中国科协精品科技期刊工程项目
CAST Promotion Project for Elite Journals

计 算 机 学 报

(JISUANJI XUEBAO)
第 35 卷 第 7 期 2012 年 7 月

目 次

综 论

移动互联网

研究论文与技术报告

CHINESE JOURNAL OF COMPUTERS

Vol.35 No.7 July 2012

CONTENTS

Surveys

Mobile Internet

Papers and Reports

计算机系统与计算机网络中的动态优化：模型、求解与应用

林 闯[1] 万剑雄[2] 向旭东[2] 孟 坤[2] 王元卓[3]

[1](清华大学计算机科学与技术系 北京 100084)

[2](北京科技大学计算机与通信工程学院 北京 100083)

[3](中国科学院计算技术研究所 北京 100190)

摘 要 动态优化是计算机系统与计算机网络中进行资源分配与任务调度等方面研究所采用的主要理论工具之一.目前,国内外已开展大量研究,致力于深化动态优化的理论研究与工程应用.文中从模型、求解与应用3个角度,对马尔可夫决策过程动态优化理论模型进行了综述,并重点介绍了将动态优化理论与随机Petri网理论相结合的马尔可夫决策Petri网和随机博弈网模型,详细讨论了这些模型的建模方法、求解算法与一些应用实例.最后,对全文进行了总结,并对未来可能的研究方向进行了展望.

关键词 动态优化;马尔可夫决策过程;随机Petri网;马尔可夫决策Petri网;随机博弈网

中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2012.01339

Dynamic Optimization in Computer Systems and Computer Networks: Models, Solutions, and Applications

LIN Chuang[1] WAN Jian-Xiong[2] XIANG Xu-Dong[2] MENG Kun[2] WANG Yuan-Zhuo[3]

[1](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

[2](*School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing* 100083)

[3](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

Abstract Dynamic optimization is one of the most popular theoretical tools to study resource allocation and task scheduling problems in computer systems and computer networks. At present, a vast number of researches have been on their way to enhance the theoretical basis and extend the industrial applications of dynamic optimization theory. This paper provides an overview of Markov Decision Process (MDP) from the perspectives of models, solutions, and applications. We also survey two types of extended dynamic optimization models, i.e., Markov Decision Petri Nets (MDPN) and Stochastic Game Nets (SGN), which combine dynamic optimization theory and stochastic Petri nets theory. We focus on the model construction, solution techniques, and applications of these models. Finally, we discuss some possible research challenges in the future.

Keywords dynamic optimization; Markov decision processes; stochastic Petri nets; Markov decision Petri nets; Stochastic Game nets

收稿日期:2012-05-03;最终修改稿收到日期:2012-06-15.本课题得到国家"九七三"重点基础研究发展规划项目基金(2010CB328105,2009CB320505)、国家自然科学基金重点项目(60932003)和国家自然科学基金面上项目(61070182,60973144,60973107,61173008,61070021)资助.**林 闯**,男,1948年生,博士,教授,博士生导师,主要研究领域为计算机网络、系统性能评价、安全分析和随机Petri网.E-mail: chlin@tsinghua.edu.cn. **万剑雄**,男,1982年生,博士研究生,主要研究方向为性能评价和最优控制. **向旭东**,男,1986年生,博士研究生,主要研究方向为性能评价和最优控制. **孟 坤**,男,1980年生,博士研究生,主要研究方向为性能评价和随机模型. **王元卓**,男,1978年生,博士,副教授,主要研究方向为随机Petri网与网络安全.

1 引 言

随着计算机网络与计算机系统在国民生活各个领域应用的不断拓展,其承载的业务种类与数量也在不断增加.如何在复杂的应用环境中合理地分配系统资源并进行调度任务,以提高计算机系统与计算机网络的运行效率,降低运行成本,是一个亟待解决的问题.

优化理论是学术界研究计算机系统与计算机网络中资源分配与任务调度问题普遍采用的方法之一.从时间这个维度进行分类,优化理论可分为静态优化与动态优化两种.其中,静态优化将系统看作为一个时不变系统,即将系统的资源需求量与资源保有量视为一个与时间无关的常量.但是,实际的系统往往都是随时间变化的,而且会受到各种外部随机事件的影响.静态优化模型忽略了未来可能的系统变化,也不能反映决策者当前行为对未来的影响,无法刻画系统随时间变化的特性.因此,本文着重研究动态优化理论在计算机系统与计算机网络中的应用.在动态优化理论中,系统的目标函数是系统收益关于时间的累积量.相对于静态优化理论,动态优化理论可以较好地对系统的时变性进行刻画,更好地反映系统当前决策对时间累积目标函数的影响.

动态优化的基本理论模型是马尔可夫决策过程(Markov Decision Process,MDP).MDP 可以用来描述这样一类离散时间决策过程:系统 $t+1$ 刻状态的转移,只依赖于 t 时刻的系统状态与决策者的行为,而与$[0,t-1]$时间段内的系统状态与决策者行为无关.MDP 可以从执行时间、决策者观测能力、状态转移的确定关系、时间的连续性、状态转移/收益的确定性、是否具有附加限制条件以及决策目标数量等角度进行分类.通常情况下,对于计算机系统与计算机网络中的资源管理问题,由于资源的种类繁多,数量庞大,因而所建立的 MDP 模型通常会遇到"状态空间爆炸"问题,即 MDP 模型的状态空间随着问题规模指数级增长,这使得传统精确求解算法如值迭代与策略迭代等无法应用.因此,本文详细讨论了 MDP 模型的近似求解算法,将这些算法归为 3 类:贪心算法、基于状态聚合的算法以及基于近似动态规划(Approximate Dynamic Programming,ADP)的算法.

马尔可夫决策过程在实际应用中体现出一些不足,主要表现在:(1)模型不够直观.一方面,MDP 中的各个模型要素都使用了严格的形式化定义,虽然具有较强的逻辑性与严密性,但是模型的直观性与可理解性却相对较低.另一方面,模型建立需要较强的数学背景,例如在推导系统状态转移概率时,往往需要建模者具有一定的随机数学基础,增加了模型建立的难度.(2)在一些复杂应用环境中,单纯的 MDP 模型难以精确刻画系统的特点.例如,在网络安全问题中,MDP 难以描述网络拓扑与各个组件之间的逻辑关系.这些不足激励学者们进行了进一步的模型拓展研究,其中较有代表性的是马尔可夫决策 Petri 网(Markov Decision Petri Nets,MDPN)与随机博弈网(Stochastic Game Nets,SGN).这些模型方法将动态优化理论与随机 Petri 网理论相结合,在一定程度上克服了上述缺点.随机 Petri 网模型语义明确,使用图形化表示方式,直观易懂.系统中各个组件之间的关系可以灵活地使用组件之间的连接弧与变迁可实施函数等方式表现.建模者可以将精力更多地放在研究目标系统与精确描述系统与决策者行为方面,而状态转移概率等其它较为复杂的模型元素则可以利用 Petri 网工具中集成的功能实现自动化推导.

MDPN 模型将 MDP 理论与随机 Petri 网理论相结合,可以体现出系统与决策者宏观层面上的行为交替.利用 MDPN 模型,可以方便地借助 Petri 网图形工具对系统进行建模,并对模型的可达图进行规约得到 MDP 模型.SGN 模型是动态优化模型的进一步扩展,它将动态随机博弈与 Petri 网理论相结合,允许系统中存在多个决策者.每个决策者一般都有各自的目标函数,他们之间既可以是合作关系,也可以是竞争关系.在建立 SGN 模型时,可以先单独从各个决策者的角度出发,建立 SGN 子模型,再利用模型组合与化简技术,得到完整的 SGN 模型.求解 SGN 是一个寻求每个决策者均衡策略的问题,可归结为一个静态非线性规划问题.

动态优化模型是当前计算机系统与计算机网络的资源分配与任务调度等问题中的研究热点,对降低系统维护成本、提高系统运行效率具有重要的意义.本文从建模、求解与应用等角度,论述了马尔可夫决策过程、马尔可夫决策 Petri 网以及随机博弈网等动态优化模型在计算机系统与计算机网络中的应用.

2　基于马尔可夫决策过程的动态优化模型

2.1　马尔可夫决策过程

一个基本的马尔可夫决策过程包括以下要素：

(1) 状态集合 S，描述系统的状态.

(2) 行为集合 A，描述决策者在状态空间中可能的行为. 通常行为集合会依赖于当前状态，即可将其记为 $A(s)$.

(3) 收益函数 $R(s,s',a)$，$s,s'\in S$，$a\in A$，描述系统在决策者行为的影响下运行所产生的收益.

(4) 状态转移关系 S^M，描述系统状态在决策行为影响下的转移过程.

马尔可夫决策过程的一个显著特征是无后效性，即系统在下一时刻的状态仅依赖于当前所处的状态与决策行为，而与系统的历史无关.

根据 S^M 的性质不同，MDP 可以分为确定 MDP 与随机 MDP 两大类. 对于确定 MDP，在某个状态下的某个行为会导致唯一确定的状态转移，即 S^M：$S\times A\to S$，此时状态转移方程可记为 $s'=S^M(s,a)$；对于随机 MDP，未来系统状态不仅取决于当前系统状态下决策者的行为，还受到外部随机变量 W 的影响，即 S^M：$S\times A\times W\to S$，此时状态转移方程可记为 $s'=S^M(s,a,W(\omega))$，其中 $W(\omega)$ 为外部随机变量的一个实现样本. 随机 MDP 的未来状态一般服从某种分布，该分布可记为 $P(s'|s,a)$. 本文主要研究随机 MDP，下文中提到的马尔可夫决策过程，一般均指随机马尔可夫决策过程. 定义 $R(s,a)=\sum_{s'\in S}P(s'|s,a)R(s,s',a)$ 为状态 s 下采用行为 a 所产生的收益.

在马尔可夫决策过程中，策略 π 定义为从状态集合 S 到行为集合 A 的一个映射. 决策者根据策略 π 来得到当前所需的决策行为. 一个典型的马尔可夫决策过程的执行流程如下：

1. 决策者观察当前所处的状态 s.
2. 根据当前状态确定决策行为 $\pi(s)$.
3. 执行行为 $\pi(s)$，系统状态发生转换.
4. 重复 1.

MDP 在系统演进过程中，会产生一个收益序列. 为比较 MDP 中决策的优劣程度，引入了目标函数 J. 它将一个收益序列映射为一个单一的实数值. 对于无限时间 MDP 来说，其设置一般有 3 种方法：

(1) 在无限时间 MDP 中截取一个足够长的有限时间 MDP，则无限时间 MDP 的目标函数可近似地看作有限时间 MDP 的收益的和.

(2) 依照时间的推移对未来所得收益进行逐步折扣，保证对时间累加的总收益总是收敛的. 这种方式更看重当前所得的收益.

(3) 平均收益在时间趋于无穷处的极限值.

通过目标函数 J，可以定义策略之间的偏序关系，这样就可以对策略的优劣进行比较了.

MDP 中另一个重要概念是值函数 $V^\pi(s)$. $V^\pi(s)$ 是从 $\pi\times S$ 到实数集 $\mathbb{R}$ 的映射，其含义为在采用策略 π 的前提下，在状态 $s\in S$ 下所得到的目标函数 J 的期望. 无限时间 MDP 的值函数满足 Bellman 递推方程，即式(1)：

$$V^\pi(s_t)=R(s_t,\pi(s_t))+\alpha\sum_{s_{t+1}\in S}P(s_{t+1}|s,\pi(s))V^\pi(s_{t+1}) \tag{1}$$

其中 α 为折扣因子. 式(1)说明，给定策略 π，则在状态 s 的值函数等于当前一步决策所得收益与下一时刻折扣后值函数期望的和.

式(1)也可以写成如式(2)所示的向量形式，即

$$\boldsymbol{V}^\pi=\boldsymbol{R}^\pi+\alpha\cdot\boldsymbol{P}^\pi\boldsymbol{V}^\pi \tag{2}$$

2.2　马尔可夫决策过程建模与分析

在利用马尔可夫决策过程对系统进行建模分析时，可使用如下步骤：

(1) 明确系统运行目标.

该步骤中需要确定 MDP 的收益函数 R 与目标函数 J. 一方面，不同系统的运行目标可能不同. 另一方面，即使是对于同一系统，研究角度的差异也会导致不同的收益函数与目标函数. 以计算机网络为例，较为常用的目标函数有

① 节点吞吐量[1-3]；

② 能量消耗[4-6]；

③ 信道利用率[7-8]；

④ 延迟[9-10]；

⑤ 分组丢失率[11].

对于随机 MDP，通常使用带有期望形式(E)的目标函数. 一般期望目标函数具有如下形式：

有限马尔可夫决策过程：

$$J=\mathrm{E}\left\{\sum_{t=1}^{T}R(s_t,a_t)\right\} \tag{3}$$

无限马尔可夫决策过程：

$$J=\mathrm{E}\left\{\sum_{t=1}^{\infty}\alpha^tR(s_t,a_t)\right\} \tag{4}$$

$$J=\lim_{T\to\infty}\frac{1}{T}\mathrm{E}\left\{\sum_{t=1}^{T}R(s_t,a_t)\right\} \tag{5}$$

其中，s_t与a_t分别为t阶段系统所处状态与决策者采取的行为.式(4)与式(5)分别为无穷时间折扣情形与无穷时间平均情形下的目标函数.系统的运行目标通常是最大化或最小化上述目标函数J.

(2)确定系统运行状态空间与决策者的行为空间.

系统的状态空间与决策者的行为空间可能是离散可列的.例如，在认知无线电系统中，信道可以用两个离散的状态刻画，即{空闲，占用}，用户的行为也可能是离散的，如{发送数据，监听信道}.状态空间与行为空间也可能是连续的.例如，在上例中，若用户的行为变为“以概率p发送数据”，则用户行为空间是连续的，且其取值范围为[0,1].

(3)根据系统状态之间的动态转移关系建立Bellman递推方程.

该步骤中要找到状态之间的转移关系.对于随机MDP来说，转移关系包括状态转移方程$s'=S^M(s,a,W(\omega))$与转移概率$P(s'|s,a)$.有时状态转移概率无法精确得知，此时可以使用强化学习[12]的方法来求解马尔可夫决策过程.Bellman方程描述的是值函数V的递推关系，该方程在求解最优策略时发挥了重要作用.

(4)根据所建立的Bellman递推方程，对模型进行求解，得到最优策略π^*.

以最大化目标函数为例，求解过程中的关键步骤包括

$$\pi^*(s_t)=\arg\max_{a_t\in A}\left\{R(s_t,a_t)+\alpha\sum_{s_{t+1}\in S}P(s_{t+1}|s_t,a_t)V^*(s_{t+1})\right\} \tag{6}$$

$$V^*(s_t)=R(s_t,\pi^*(s_t))+\alpha\sum_{s'\in S}P(s_{t+1}|s_t,\pi^*(s_t))V^*(s_{t+1}) \tag{7}$$

式(6)按照最大化策略，根据当前所得的值函数V^*求得在状态s下应该采取的策略π^*，而式(7)则根据π^*计算其所对应的值函数.式(6)、(7)实际上是一个迭代的过程.不同的求解算法，如值迭代、策略迭代等，均需要使用以上两个步骤，只是顺序不同.

下面以一个接纳控制为例(图1)，举例说明MDP的建模方法.外部任务到达后，首先缓存在接纳控制器的等待队列中.在每个时间槽的开始，接纳控制器将其等待队列中的任务按照某种策略或将任务丢弃，或将任务分配给服务器$1\sim n$中的一个.在这个系统中，决策者为接纳控制器，其t时刻的行为是向量$\boldsymbol{x}_t=\{x_t^i\}$，其中每个分量$x_t^i$表示向服务器$i$分配的任务数.系统中的外部随机变量$W$包括两部分：①$[t-1,t]$内到达接纳控制器的任务数$\lambda_t$；②$[t,t+1]$内服务器$i$完成的任务数$\mu_t^i$.

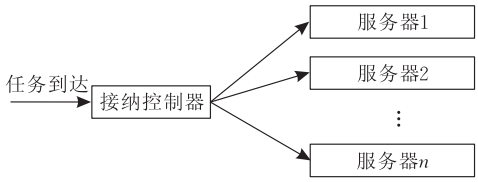


图1 MDP建模举例：一个接纳控制问题

系统在t时刻的状态可用$\{\lambda_t,q_t^1,\cdots,q_t^n\}$表示.其中$q_t^i$表示服务器$i$中的队列长度.假设若等待队列中的任务没有得到及时服务，则下一时刻这些任务会丢失.系统的状态转移方程可写为

$$q_{t+1}^i=q_t^i+x_t^i-\mu_t^i,\quad \forall i\in[1,\cdots,n] \tag{8}$$

此时决策行为受到如下流守恒条件约束：

$$\sum_{i=1}^{n}x_t^i\leq\lambda_t.$$

若假设系统每个时间槽内的收益与完成的任务数成正比，与服务器中驻留的任务数成反比，则系统在$[t,t+1]$时间段内的收益函数可定义如下：

$$r\sum_{i=1}^{n}\mu_i-\sum_{i=1}^{n}c_i(q_t^i+x_t^i),$$

其中r为每完成一个任务所得的收益，c_i为在服务器i上每个时间槽内服务一个任务所需要的成本.该系统的无穷时间折扣MDP的目标函数为

$$J=\mathrm{E}\left\{\sum_{t=1}^{\infty}\sum_{i=1}^{n}\alpha^t(r\mu_i-c_i(q_t^i+x_t^i))\right\}.$$

MDP的求解方法将在2.4小节进行详细讨论.

2.3 马尔可夫决策过程的分类

根据不同的划分依据，可将马尔可夫决策过程进行分类，如表1所示.

表1 马尔可夫决策过程的分类

划分依据	种 类
执行的时间	有限时间MDP，无限时间MDP
决策者的观测能力	完全可观测MDP，部分可观测MDP(POMDP)
转移关系的确定性	确定MDP，随机MDP
时间的连续性	连续时间MDP，离散时间MDP
转移概率/收益的确定性	普通MDP，带有强化学习的MDP
是否有附加限制条件	不受限MDP，受限MDP(CMDP)
目标的数量	单目标MDP，多目标MDP

(1)按照系统的执行时间分类.

现实中系统的运行时间都是有限的.对于有限时间马尔可夫决策过程，其目标函数可以简单地写成在系统运行期间内收益的和，如文献[3].当系统

运行时间很大时，也可以近似地认为系统的运行时间是无限的. 此时针对该系统建立的马尔可夫决策过程就是无限时间马尔可夫决策过程. 无限时间马尔可夫决策过程通常采用折扣累积收益或平均收益作为其目标函数，即式(4)和(5). Haas 等人[13]分别针对这两种目标函数研究了无线多媒体网络环境中的资源分配问题. 折扣累积收益目标函数已经被广泛研究，理论成果较为完善.

(2) 按照决策者的观测能力分类.

一般情形下，决策者可以完全观测到系统的状态，并根据所观测到的状态进行决策. 但是，有些时候决策者不能完全观测到系统状态. 这时，需要利用部分可观测马尔可夫决策过程(Partially Observable Markov Decision Process，POMDP)进行建模. Zhao 等人[2-3]研究了认知无线电系统中次用户的信道监听与接入的问题. 在该问题中，由于次用户监听可能发生错误，因此系统是一个部分可观测马尔可夫决策过程. POMDP 求解相较于 MDP 来说较为复杂. 因为决策者没有系统状态的精确信息，所以需要维护一个信任向量，用来描述系统当前位于各个状态的概率. 信任向量随着系统的演进而不断更新.

(3) 按照转移关系的确定性分类.

决策者在某个状态下所做的行为，有时会导致一个确定的结果，即以概率 1 转移到下一个状态，这称为确定马尔可夫决策过程. 有时，决策者的行为会导致不确定的结果，这称为随机马尔可夫决策过程.

(4) 按照时间的连续性分类.

现实中的一些问题是离散时间的，例如在库存管理问题中，仓库管理员一般每隔一个固定的时间间隔采购商品，更新库存. 还有一些问题是连续时间的，典型问题如队列管理问题[14]与设备维护问题[15-16]. 在这些问题中，系统的状态转换间隔时间(如顾客到达的间隔时间与设备正常运转的时间等)服从指数分布，每次系统状态发生转换时都需要决策者进行决策.

(5) 按照转移概率/收益的确定性分类.

在一些复杂系统中，系统的状态转移概率 $P(s'|s,a)$难以精确测量，收益函数 $R(s,a)$也无法推导出显式的解析式. 这时，就需要建立基于强化学习的 MDP 模型，采用跟踪实际系统运行过程或 Monte Carlo 模拟等方法，不断学习系统的未知特性. 值得注意的是，基于强化学习的 MDP 也是一种 MDP 的近似求解方法，简化了 Bellman 方程中值函数期望的计算.

(6) 根据是否有附加限制条件分类.

有时决策者行为会受到一些客观条件的影响，这时可将该问题归结为一个受限马尔可夫决策过程(Constrained Markov Decision Process，CMDP)问题. 以折扣情形为例，一个 CMDP 模型可表达为

$$\max_{a_t} \mathrm{E}\left\{\sum_{t=1}^{\infty}\alpha^t R(s_t,a_t)\right\}$$

$$\text{s.t.}: \mathrm{E}\left\{\sum_{t=1}^{\infty}\beta^t c(s_t,a_t)\right\}\leq C \tag{9}$$

约束(9)中 $c(s_t,a_t)$可视为阶段 t 所产生的资源消耗，C 为客观资源总量限制. 这类问题在计算机系统内有大量的应用，如 Djonin 等人[17]研究了在 MIMO 系统中，在数据延迟受限的情况下，最小化平均发射功率的问题. 求解 CMDP 可以使用线性规划法[18]与拉格朗日法[19]等.

(7) 按照目标数量分类.

很多动态优化问题只考虑一个目标函数，也就是常见的单目标优化问题. 如果目标函数有多个，就需要用多目标优化建模. 一般处理多目标问题的方式有 3 种：① 将一部分目标函数转化为约束，进而转化为 CMDP 模型[20]；② 将各个目标加权平均，组合成一个整体目标[21]；③ 求解帕雷托前沿(Pareto Frontier)[22-23].

2.4　马尔可夫决策过程的求解

由于篇幅所限，本文主要讨论无穷折扣马尔可夫决策过程的求解. 求解算法分类如图 2 所示.

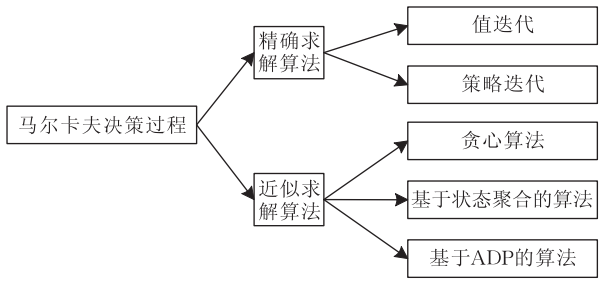


图 2　MDP 求解算法分类

2.4.1　精确求解算法

折扣情形下的最优解满足

$$\boldsymbol{V}^* = T(\boldsymbol{V}^*) \tag{10}$$

运算符 T 定义如下：

$$[T(\boldsymbol{V})]_s = \max_{a\in A(s)}\left\{R(s,a)+\alpha\sum_{s'\in S}P(s'|s,a)[\boldsymbol{V}]_{s'}\right\} \tag{11}$$

其中，$[\cdot]_s$为向量的第 s 个分量. 满足式(10)的值函数即是最优值函数 $\boldsymbol{V}^*$. 可以看到，式(11)实际上是采取最大化策略的式(1)的变形.

(1)值迭代算法.值迭代算法实际上是近似算法,随着迭代过程的进行,该算法会不断逼近最优解.值迭代算法如算法1所示.

算法1. 值迭代算法.

1. $n=0$,给定初值 $\boldsymbol{V}_0=\boldsymbol{v}$.
2. 根据迭代式 $\boldsymbol{V}_n=T(\boldsymbol{V}_{n-1})$,计算第 n 次迭代的值函数与策略.
3. 重复步2.

可以证明,算法1在 $n\to\infty$ 时收敛于最优值函数 $\boldsymbol{V}^*$.此外,还可以在每一次迭代时估计出最优解的区间,即

$$\boldsymbol{V}_n+\frac{\alpha}{1-\alpha}\cdot\beta_n\cdot\boldsymbol{e}\leq\boldsymbol{V}^*\leq\boldsymbol{V}_n+\frac{\alpha}{1-\alpha}\cdot\alpha_n\cdot\boldsymbol{e} \tag{12}$$

其中,$\boldsymbol{e}$ 为全1向量,α_n 与 β_n 定义如下:

$$\alpha_n=\max_{s\in S}\{[\boldsymbol{V}_n]_s-[\boldsymbol{V}_{n-1}]_s\} \tag{13}$$

$$\beta_n=\min_{s\in S}\{[\boldsymbol{V}_n]_s-[\boldsymbol{V}_{n-1}]_s\} \tag{14}$$

式(12)也可以作为值迭代算法运行结束的判定方法.例如,可事先指定一精度 ε,使得

$$\frac{\alpha}{1-\alpha}\cdot(\alpha_n-\beta_n)\cdot\boldsymbol{e}\leq\varepsilon$$

成立时算法终止.

(2)策略迭代算法.可以证明,当状态集合与行为集合有限时,策略迭代算法可以在有限迭代次数内获得最优解,且迭代次数上界为策略数,即 $\prod_{s\in S}|A(s)|$,其中 $|\cdot|$ 为集合内的元素个数.策略迭代算法如算法2所示.算法2首先确定一个初始策略 π_0,并直接根据式(2)求解得到该策略所对应的值函数.最后,再根据所得的值函数对策略进行更新.若更新前后的策略相同,则说明已经找到了最优策略,算法结束.

算法2. 策略迭代算法.

1. $n=0$,给定初始策略 π_0.
2. 通过求解 $(\boldsymbol{I}-\alpha\boldsymbol{P}^{\pi_n})\boldsymbol{V}_n=\boldsymbol{R}^{\pi_n}$ 确定 $\boldsymbol{V}_n$.
3. 确定 π_{n+1} 使其满足
$$\pi_{n+1}=\arg\max_{\pi_{n+1}}\{\boldsymbol{R}^{\pi_{n+1}}+\alpha\boldsymbol{P}^{\pi_{n+1}}\boldsymbol{V}_n\}.$$
4. if $\pi_{n+1}=\pi_n$,算法终止,设定最优策略 $\pi^*=\pi_n$.
else $n=n+1$,转到步2.

此外,学者们还基于以上两种基本算法设计了一些变形算法,如修正的策略迭代(Modified Policy Iteration)等,此处不再赘述.

2.4.2 近似求解算法

在一个实际系统中,资源种类与资源数量都极其庞大,导致所建立的MDP模型无法利用精确算法进行求解,原因在于:① 需要为每个状态存储其值函数.在状态数较多时,现有的技术无法提供足够的存储空间;② 在迭代过程中,计算值函数要遍历所有状态,会导致迭代一次所需时间较长,算法收敛速度太慢.基于这些考虑,人们开始寻找MDP的近似求解算法,使得在有限的时空复杂度范围内,得到可接受的次优解.

(1)贪心算法

贪心算法又称为近视策略(myopic policy),它可表示为:在时刻 t,求解如下优化问题

$$\max_{a_t\in A(s_t)}R(s_t,a_t) \tag{15}$$

例如,在图1的接纳控制问题中,贪心算法为

$$\max_{x_t^i}\sum_{i=1}^{n}\{r\mu_t^i-c_i(q_t^i+x_t^i)\}$$

$$\text{s.t.}:\sum_{i=1}^{n}x_t^i\leq\lambda_t.$$

贪心算法是最简单的一类近似算法.它只关注系统当前的收益,而忽略当前决策对未来收益的影响.这种方法虽然未必是最优的,但是至少提供了一种动态优化问题的简单求解方案.贪心算法的最大优点在于,其求解过程没有算法1与算法2中的迭代过程,因而时间复杂度较低.此外,也不需要提供存储值函数的空间.

在一些特殊的动态优化模型中,贪心策略就是最优策略.Karush与Dear[24]将一个学习过程利用POMDP建模,并证明了贪心策略在该类问题中的最优性.Krishnamurthy等人[25]研究了目标跟踪中的动态传感器调度问题,并给出了贪心策略是最优策略的一些充分条件.文献[26-28]分别从不同角度研究了机会频谱接入问题,并证明了贪心策略的最优性.

然而,通常情况下贪心策略并非最优策略.如Ahmad等人[29]指出,在负相关系统转移的机会频谱接入问题中,若信道都是独立同分布的Gilbert-Elliot信道,当信道数量大于3时,贪心策略并不是最优策略.虽然如此,在很多应用中,贪心算法表现出较好的适应性[30-31].

(2)基于状态聚合的算法

精确求解算法应用的最大障碍是状态空间爆炸问题.因此,一种很直观的近似求解策略是将问题空间进行聚合化简,使得问题规模减少,便于精确算法求解.

以图1中问题为例,假设接纳控制器的等待队列与所有服务器的服务队列最大容量均为100个任

务，则该问题 MDP 模型的状态空间共有 100^{n+1} 个状态. 此时，可设定如下状态聚合策略：为每个队列设定一个阈值，当队列长度低于该阈值时，则认为处于宏状态“低负载”，反之，则处于宏状态“高负载”. 这样，每个队列的状态可以化简为 2，整个系统的状态也缩小为 2^{n+1}.

一种常用的 MDP 状态聚合方法来源于马尔可夫过程的近似求解理论，见算法 3. 假设有状态转换如图 2 所示的马尔可夫决策过程. 如果存在一种对状态空间的划分，在每个划分内，任选一个状态，使得：① 以该状态作为起始状态，则转移到该状态所属划分内状态的概率很大；② 以该状态作为起始状态，则转移到不属于该状态所属划分内状态的概率很小，则这个 MDP 可以进行状态聚合化简. 例如在图 3 中，实线转移概率比虚线转移概率大很多，则该模型可以利用图中所示方式进行聚合. Liu 等人[32]利用该方法近似求解了分布式 Web 服务系统中的服务器选择问题.

算法 3.　MDP 状态空间化简算法.

1. 将状态空间 S 进行划分：$\{S_1, S_2, \cdots, S_n\}$.
2. for $i=1$ to n
3.　将所有转到 S_i 以外状态的概率都设置为 0.
4.　将 S_i 内的状态转移概率进行归一化处理.
5.　计算 S_i 内状态的稳态概率分布，利用 $\pi=\pi\boldsymbol{P}$，其中 $\boldsymbol{P}$ 为归一化的 S_i 内部转状态移概率矩阵.
6.　计算 S_i 到 S_j 的转移概率 $P_{ij}=\sum_{k\in S_i}\pi_k p_{kj}$.
7. end for

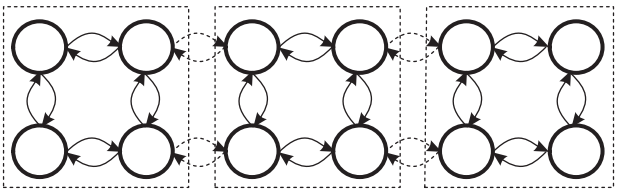


图 3　一个 MDP 状态空间的聚合

值得注意的是，只有在上面两个假设条件都满足的时候，算法 3 才能得到精度较高的近似解. 否则，误差会比较大. 此外，这种近似方法还有一个缺点，即不能得到近似解与精确解之间的关系. 为了克服这些问题，学者们又提出了其它解决方案，有界参数 MDP（Bounded-Parameters MDP，BMDP）就是这些方案中较有影响力的方法之一.

BMDP 由 Givan 等人[33-34]提出，它是非精确状态转移概率 MDP（Markov Decision Processes with Imprecisely Known Transition Probabilities，MDPIPs）模型的一种特殊情况. BMDP 是一个 4 元组 $\{S, A, R_\updownarrow, P_\updownarrow\}$. 与传统 MDP 不同，在 BMDP 中每个状态的收益函数 $R_\updownarrow$ 与状态转移概率 $P_\updownarrow$ 是一个区间，而不是一个点值. 若一个 MDP M，其状态、行为集与 BMDP $M_\updownarrow$ 的状态、行为集完全相同，且 M 的收益函数 R 与转移概率 P 都在 $M_\updownarrow$ 所规定的区间内，则称 $M\in M_\updownarrow$.

在一个 BMDP $M_\updownarrow$ 中，给定一个决策策略 π，则该策略所产生的值函数也是一个区间，称为区间值函数

$$V_\updownarrow^\pi(s)=\left[\min_{M\in M_\updownarrow} V_M^\pi(s), \max_{M\in M_\updownarrow} V_M^\pi(s)\right] \tag{16}$$

其中 $V_M^\pi(s)=R_M(s,a)+\alpha\sum_{s'\in S}P_M(s'|s,a)V_M^\pi(s')$ 是 $M_\updownarrow$ 中的一个 MDP M 的值函数. 可根据实际工程应用背景，定义区间值函数的比较方法. 例如对于策略 a 与 b，可定义：

① 乐观最优

$$V_\updownarrow^a \gg V_\updownarrow^b \Leftrightarrow V_\uparrow^a \geq V_\uparrow^b \vee (V_\uparrow^a = V_\uparrow^b \wedge V_\downarrow^a \geq V_\downarrow^b).$$

② 悲观最优

$$V_\updownarrow^a \gg V_\updownarrow^b \Leftrightarrow V_\downarrow^a \geq V_\downarrow^b \vee (V_\downarrow^a = V_\downarrow^b \wedge V_\uparrow^a \geq V_\uparrow^b).$$

可以证明，存在 $M\in M_\updownarrow$，使得所有状态的值函数能同时达到最大或最小，并称这两个 MDP 分别为关于策略 π 的最大 MDP 与最小 MDP. 寻找最大或最小 MDP 的过程，相当于寻找关于值函数上界降序排列状态空间序列与值函数下界升序排列状态空间序列的序列最大 MDP（Order Maximizing MDP）. 具体来讲，一个状态空间序列 $O=\{s_1, s_2, \cdots, s_n\}$ 为状态空间中所有状态的一个排列顺序，则状态空间序列 O 的序列最大下标 r 与序列最大 MDP M_O 可定义如下.

定义 1（序列最大下标与序列最大 MDP）[34]. 对于某个状态 s 与决策行为 a，其关于序列 O 的序列最大下标 r 为

$$\arg\max_{1\leq r\leq n}\sum_{i=1}^{r-1}P_\uparrow(s_i|s,a)+\sum_{i=r}^{n}P_\downarrow(s_i|s,a) \tag{17}$$

相应的序列最大 MDP 是一个满足式(18)的 MDP $M_O\in M$

$$P_{M_O}(s_i|s,a)=\begin{cases}P_\uparrow(s_i|s,a), & i<r\\ P_\downarrow(s_i|s,a), & i>r\end{cases} \tag{18}$$

$$P_{M_O}(s_r|s,a)=1-\sum_{i=1,i\neq r}^{n}P_{M_O}(s_i|s,a) \tag{19}$$

利用 BMDP 可以对问题空间进行状态聚合. 一个精确 MDP 经过聚合后，一般都可以归结为一个 BMDP 问题，可以利用区间迭代求解算法进行求解，即

$$IVI_{\updownarrow\text{opt}}(V_{\updownarrow})(s)=\max_{a\in A(s)}\left[\min_{M\in M_{\updownarrow}}VI^{a}_{M}(V_{\downarrow})(s),\max_{M\in M_{\updownarrow}}VI^{a}_{M}(V_{\uparrow})(s)\right] \quad (20)$$

计算式(20)实际上可以看作为具有2个决策者的2步博弈过程.以乐观最优为例,在第1步中,决策者1与决策者2为合作配合关系,决策者1利用所定义的乐观最优比较运算符≥求得最大化区间值函数上界的策略$\pi_{\uparrow\text{opt}}$.在第2步中,决策者1与决策者2为对立竞争关系,决策者2求得策略$\pi_{\uparrow\text{opt}}$的最小MDP,并计算区间值函数的下界.该过程可用算法4描述.其中,*Sort_Dec_Order*与*Sort_Inc_Order*为排序函数,*Order_Max_Ind*利用式(17)求得对应的序列最大下标.这样,就可以在缩小的问题空间中,求得原问题具有边界的解.

算法4. 区间迭代算法.

1. $O_{\text{up}}=Sort\_Dec\_Order(V_{\uparrow})$,
   $O_{\text{down}}=Sort\_Inc\_Order(V_{\downarrow})$.
2. for all $s\in S$ do
3. for all $s\in S$ do
4. $r_{\text{up}}=Order\_Max\_Ind(M_{\updownarrow},O_{\text{up}},s,a)$.
5. $r_{\text{down}}=Order\_Max\_Ind(M_{\updownarrow},O_{\text{down}},s,a)$.
6. for $i=1$ to n do
7. 根据式(18)、(19)计算$P_{\text{up}}(s_{O_{\text{down}}(i)}|s,a)$与$P_{\text{down}}(s_{O_{\text{up}}(i)}|s,a)$.
8. end for
9. end for
10. $V_{\uparrow}=\max_{\boldsymbol{a}\subseteq A(s)}R_{\uparrow}(s,a)+\alpha\sum_{s'\in S}P_{\text{up}}(s'|s,a)V_{\uparrow}(s')$.
11. if $|a|=1$ and $a=\{a\}$ then
12. $V_{\downarrow}=R_{\downarrow}(s,a)+\alpha\sum_{s'\in S}P_{\text{down}}(s'|s,a)V_{\downarrow}(s')$.
13. $\pi(s)=a$.
14. else
    $V_{\downarrow}=\max_{a\in\boldsymbol{a}}R_{\downarrow}(s,a)+\alpha\sum_{s'\in S}P_{\text{down}}(s'|s,a)V_{\downarrow}(s')$.
15. $\pi(s)=a$.
16. end if
17. end for

(3)基于近似动态规划的算法

近似动态规划(Approximate Dynamic Programming,ADP)是一种解决大规模动态优化问题的现代近似求解方法.目前,关于近似动态规划的代表性专著,主要有3本[12,35-36],分别从人工智能、控制论以及运筹学的角度对近似动态规划进行了详细的论述.近似动态规划能有效解决马尔可夫决策过程中的状态空间爆炸问题.

在ADP中,式(11)通常改写为

$$V(s_t)=\max_{a_t\in A(s_t)}R(s_t,a_t)+\alpha\cdot\mathrm{E}\{V(s_{t+1})\} \quad (21)$$

在式(21)中,状态空间爆炸问题表现为:(1)问题状态空间S太大,现有的技术无法提供足够的存储空间;(2)外部随机变量有时无法精确测量其分布,或即使分布已知,也会由于随机变量状态太多而导致其期望难于计算.在近似动态规划中,主要使用基于值函数近似(Value Function Approximation)与后决策状态(Post-Decision State Variable)的前向动态规划方法来克服以上问题.

令系统的状态转换方程为

$$s_{t+1}=S^{M}(s_t,a_t,W(\omega_t)) \quad (22)$$

其中,$W(\omega_t)$是t时刻外部随机变量的一个样本,则基本的近似动态规划算法可表述为算法5.

算法5. 基本近似动态规划算法.

1. 初始化:
   对每个状态s,初始化$\bar{V}(s)$,
   选择初始状态s_0.
2. for $t=0$ to T do
3. 求解
   $\hat{v}_t=\max_{a_t\in A(s_t)}\{R(s_t,a_t)+\alpha\cdot\mathrm{E}\{\bar{V}(s_{t+1})|s_t\}\}$,
   并令a_t为以上最大化问题的解.
4. 利用下式对$\bar{V}(s_t)$进行更新
   $\bar{V}(s_t)\leftarrow(1-\eta_t)\bar{V}(s_t)+\eta_t\hat{v}_t$.
5. 选定一个采样路径ω_t.
6. 计算下一个状态
   $s_{t+1}=S^{M}(s_t,a_t,W(\omega_t))$.
7. end for

算法5首先初始化所有状态的值函数,并指定一个初始状态.然后,利用Monte Carlo方法对随机外部信息进行一次采样.算法的核心是步2~7,首先求解一步优化问题(步3),并利用所得出的$\hat{v}_t$对值函数进行更新.其中η_t是步长.

该算法与用于求解一般马尔可夫决策过程迭代算法的最根本区别,在于时间是顺序演进的,而不是倒序演进的.算法运行的过程,实际上是一个系统仿真的过程.以图1中接纳控制问题为例,其ADP求解算法可描述如下:

1. 设定每个状态值函数的初始值,选取起始状态,并令$t=0$.

2. 采集t时刻系统状态,根据当前的值函数,计算当前决策行为x_t,并得到当前状态值函数的一个样本$\hat{v}_t$(算法5步3).其中,$\mathrm{E}\{\bar{V}(s_{t+1})|s_t\}$可用Monte Carlo模拟的方法求得.

3. 根据值函数样本,更新当前状态的值函数(算法5步4).

4. 使用 Monte Carlo 方法得到外部随机变量的样本，即任务到达数 λ_t 与任务完成数 μ_t^i.

5. $t \leftarrow t+1$，并根据式(8)得到 $t+1$ 时刻的系统状态，重复步 2.

该算法其优点显而易见，在迭代过程中不需要枚举系统的所有状态来计算值函数，一定程度上规避了状态空间爆炸问题.

但是，算法 5 中仍然存在不足. 例如，该算法为每个状态均设立一个变量 $\bar{V}(s)$ 用以存储其值函数. 当问题状态空间较大时，难以提供足够的存储空间. 同时，该算法只更新所遍历到的状态的值函数，而未遍历到的状态的值函数却得不到更新. 下面我们就算法 5 中的各个步骤展开论述，详细介绍近似动态规划算法克服状态空间爆炸问题的主要手段.

① 后决策状态

后决策状态是决策者做完决策后、且外部随机信息到达前系统的状态. 这样，式(22)就分为了两步：

$$s_t^x = S^{M,x}(s_t, a_t) \tag{23}$$

$$s_{t+1} = S^{M,\omega}(s_t^x, W(\omega_t)) \tag{24}$$

其中，s_t^x 称为 t 时刻的后决策状态，s_{t+1} 称为 $t+1$ 时刻的前决策状态. 后决策状态可以看作为前决策状态与决策行为的确定函数.

以图 1 中接纳控制问题为例，$\{\lambda_t, q_t^1, \cdots, q_t^n\}$ 为系统的前决策状态，而其后决策状态为 $\{q_t^{x,1}, \cdots, q_t^{x,n}\}$，它们之间的状态转移如下

$$q_t^{x,i} = q_t^i + x_t^i,$$

$$q_{t+1}^i = q_t^{x,i} - \mu_t^i.$$

后决策状态的值函数定义如下：

$$V(s_t^x) = \mathrm{E}\{V(s_{t+1}) \mid s_t^x\} \tag{25}$$

即它是下一时刻前决策状态值函数的期望. 此时，步 3 可以改写为

$$\hat{v}_t = \max_{a_t \in A(s_t)} R(s_t, a_t) + \alpha \cdot \bar{V}(S^{M,x}(s_t, a_t)) \tag{26}$$

注意到式(26)中，等式右边已经没有期望运算.

② 值函数近似

在算法 1 与算法 2 中，值函数表现为一种“查表”形式(Table Lookup Form)，即算法需要维护一个值函数表，表项为每个状态 s 所对应的值函数 $V(s)$. 这种方式使得值函数的存储与计算都较为困难. 在 ADP 中，可以利用函数近似的方法，利用一些简单的函数形式拟合后决策状态的值函数. 线性值函数近似是普遍使用的一种值函数近似方法. 令 $\mathcal{F}$ 为动态优化问题中的特征集，该特征集与问题结构本身有较大相关性. 如在分布式库存管理问题中，特征集可以包括各地库存量、各地仓库在单位时间内到达的货物量、库存变化的方差以及这些特征的平方等[37].

定义基函数(Basis Function) $\phi_f(s_t^x)$，$f \in \mathcal{F}$ 为关于后决策状态 s_t^x 中某一特征 f 数量关系的函数，即 $\phi_f(s_t^x)$ 为从后决策状态集合到实数集合的映射，则后决策状态的值函数可以利用如下方式进行近似：

$$V(s_t^x) \approx \bar{V}(s_t^x \mid \boldsymbol{\theta}) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(s_t^x) \tag{27}$$

此时算法 5 中步 3 可以进一步改写为

$$\hat{v}_t = \max_{a_t \in A(s_t)} \{R(s_t, a_t) + \alpha \cdot \sum_{f \in \mathcal{F}} \theta_f \phi_f(s_t^x)\} \tag{28}$$

这样，估计值函数的过程，就转化为估计 θ_f 的过程，即 θ_f 随时间演进而不断更新，因此也可记作 $\theta_{f,t}$. 一般情况下，特征集的空间远小于问题的状态空间. 因此，值函数近似可以较好地解决状态空间爆炸的问题.

③ 值函数样本的取得

2.1 小节提到，状态 s_t 的值函数 $V(s_t)$ 为从状态 s_t 开始到时间趋于无穷时收益函数的累加. 在策略 π 作用下，$V(s_t)$ 的一个无偏估计样本可以直观地写为

$$\hat{v}(s_t) = \sum_{\tau=t}^{\infty} \alpha^{\tau-t} R(s_\tau, a_\tau^\pi) \tag{29}$$

式(29)可以用一个有限时间累计收益进行近似，即取一个足够大的 T，使得 $\alpha^{T-t} \to 0$，则

$$\hat{v}(s_t) \approx \sum_{\tau=t}^{T} \alpha^{\tau-t} R(s_\tau, a_\tau^\pi) \tag{30}$$

式(29)还可改写为

$$\hat{v}(s_t) = \sum_{\tau=t}^{\infty} \alpha^{\tau-t} R(s_\tau, a_\tau^\pi) - \sum_{\tau=t}^{\infty} \alpha^{\tau-t} (V(s_\tau) - \alpha V(s_{\tau+1})) + V(s_t) - \alpha^\infty V(s_\infty) \tag{31}$$

由于 $\alpha \in (0,1)$ 且 $V(s)$ 有界，因而 $\alpha^\infty V(s_\infty) \to 0$，式(31)可近似地变换为

$$\hat{v}(s_t) = V(s_t) + \sum_{\tau=t}^{\infty} \alpha^{\tau-t} (R(s_\tau, a_\tau^\pi) - V(s_\tau) + \alpha V(s_{\tau+1})) \tag{32}$$

其中 $R(s_\tau, a_\tau^\pi) - V(s_\tau) + \alpha V(s_{\tau+1})$ 称为即时差分(Temporal Difference，TD)或 Bellman 误差(Bellman Error)，表示当前值函数估计值与上次值函数估计值之间的差. 在一些文献中，折扣因子 α 有时用 λ 表示，因此这种取得值函数样本的方法又叫做 $TD(\lambda)$. 当折扣因子 $\alpha = 0$ 时，又可以得到一种特殊的表示方式：

$$\hat{v}(s_t) = R(s_t, a_\tau^\pi) + \alpha V(s_{t+1}) \tag{33}$$

式(33)称为 $TD(0)$. 注意式(33)与带有后决策状

态变量的 Bellman 方程(26)极为相似. 当 π 为式(26)中的最大化策略时,$V(s_{t+1})$ 为式(26)中 $\bar{V}(S^{M,x}(s_t,a_t))$的无偏样本.

当利用形如式(27)所示的值函数近似方法时,ADP 算法并不关注值函数本身,而着重考察值函数的导数 θ_f. 例如,在资源管理问题中,$\phi_f(s_t^x)$可以代表具有某一特性 f 的资源的数量,这时,θ_f的物理含义是该类资源的边际收益[38]. θ_f的样本可以通过以下两种方法得到:

(i) 优化问题的对偶变量. 一般资源管理问题都存在资源数量的约束,该约束所对应的对偶变量 $\hat{\theta}_f$就是资源的影子价格,即 θ_f的样本.

(ii) 数值微分. 在状态 s_t,根据式(33)可得 $\hat{v}(s_t)$. 此时,可将状态 s_t的 f 类资源的数量减 1 得到状态 $\tilde{s}_t$,重新进行优化,得到 $\hat{v}(\tilde{s}_t)$,则数值微分可表示为

$$\hat{\theta}_f=\hat{v}(s_t)-\hat{v}(\tilde{s}_t).$$

④ 值函数更新方法

随机梯度法是一种常用的值函数更新方法,可以通过逐步学习值函数样本 $\hat{v}$,使 $\bar{V}$ 不断逼近真实值函数. 随机梯度法的目标是

$$\min_{\bar{V}(s)} \mathrm{E}\left\{\frac{1}{2}(\bar{V}(s)-\hat{v}(s))^2\right\} \tag{34}$$

即寻找最符合样本 $\hat{v}$ 的值函数 $\bar{V}$. 由于 $\hat{v}(s)$是一个随机变量,因此该问题为一个随机优化问题,其求解算法与静态优化问题中的梯度法类似,称为随机梯度法. 若在 t 时刻,系统位于状态 s,对应的步长为 η_t,则

$$\bar{V}(s)\leftarrow\bar{V}(s)-\eta_t(\bar{V}(s)-\hat{v}(s))=(1-\eta_t)\bar{V}(s)+\eta_t\hat{v}(s).$$

注意到该式就是算法 5 中步 4.

若使用后决策状态的值函数,则优化目标(34)变为

$$\min_{\bar{V}(s_t^x)}\mathrm{E}\left\{\frac{1}{2}(\bar{V}(s_{t-1}^x)-\hat{v}(s_t))^2\right\},$$

此时更新方法为

$$\begin{aligned}\bar{V}(s_{t-1}^x)&\leftarrow\bar{V}(s_{t-1}^x)-\eta_t(\bar{V}(s_{t-1}^x)-\hat{v}(s_t))\\&=(1-\eta_t)\bar{V}(s_{t-1}^x)+\eta_t\hat{v}(s_t)\end{aligned} \tag{35}$$

若使用形如式(27)的后决策状态值函数近似策略,则随机梯度法的目标变为

$$\min_{\boldsymbol{\theta}}\mathrm{E}\left\{\frac{1}{2}(\bar{V}(s_{t-1}^x|\boldsymbol{\theta})-\hat{v}(s_t))^2\right\} \tag{36}$$

即寻找最接近实际值函数的后决策状态近似值函数 $\bar{V}(s_{t-1}^x|\boldsymbol{\theta})$. 此时只需更新 $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\eta_t(\bar{V}(s_{t-1}^x|\boldsymbol{\theta})-\hat{v}(s_t))\nabla_{\boldsymbol{\theta}}\bar{V}(s_{t-1}^x|\boldsymbol{\theta}) \tag{37}$$

其中,由式(27)得

$$\nabla_{\boldsymbol{\theta}}\bar{V}(s_{t-1}^x|\boldsymbol{\theta})=\begin{pmatrix}\dfrac{\partial\bar{V}(s_{t-1}^x|\boldsymbol{\theta})}{\partial\theta_1}\\[2ex]\dfrac{\partial\bar{V}(s_{t-1}^x|\boldsymbol{\theta})}{\partial\theta_2}\\[2ex]\vdots\\[2ex]\dfrac{\partial\bar{V}(s_{t-1}^x|\boldsymbol{\theta})}{\partial\theta_{|\mathcal{F}|}}\end{pmatrix}=\begin{pmatrix}\phi_1(s_{t-1}^x)\\\phi_2(s_{t-1}^x)\\\vdots\\\phi_{|\mathcal{F}|}(s_{t-1}^x)\end{pmatrix} \tag{38}$$

此外,还有一些基于线性回归的值函数更新算法,如最小二乘即时差分(Least Squares Temporal Differences,LSTD)与最小二乘策略估计(Least Squares Policy Evaluation,LSPE)[35]. 这两种算法的主要区别在于,LSTD 采集所有值函数样本后一次进行拟合,而 LSPE 为一种边采集值函数样本边拟合的递归算法.

⑤ 状态聚合

2.4.2 节介绍了一些基于状态聚合的近似求解算法. 事实上,ADP 中也可以使用状态聚合. 不失一般性,一个聚合状态 s_g的值函数 $V(s_g)$可定义为该聚合状态所包含状态的值函数的平均值,即

$$V(s_g)=\frac{\sum_{s\in s_g}V(s)}{|s_g|} \tag{39}$$

状态聚合解决了状态空间爆炸问题,但是随之而来的问题是如何确定合理的状态聚合策略以获得较好的近似解. George 等人[39]提出了一种多层状态聚合的思想,将状态的近似值函数定义为不同层次聚合值函数的加权平均. 令 G 为聚合层次的集合,则

$$V(s)=\sum_{g\in G}w_gV(s_g) \tag{40}$$

其中 s_g为非聚合状态 s 在第 g 层聚合中所对应的聚合状态. w_g可以通过跟踪各层聚合状态值函数的误差与方差等参数确定. 这种方法在实际样本较少的问题中,显示出较强的适应性,可以加速算法的收敛速度.

⑥ 步长

步长一般可分为两类,一类是确定步长,如 $\eta_t=1/(t+1)$或 $\eta_t=a/(t+a)$等;另一类是随机步长,这类步长与每次取得的样本 $\hat{v}$ 或 $\hat{\theta}$ 等相关,一般收敛速度较快. 本文中以确定步长为例,简要介绍 ADP 值函数更新算法中的步长.

为保证随机梯度法收敛,一般要求确定步长 η 满足如下条件:(i) $\eta_t\geq0$;(ii) $\sum_{t=0}^{\infty}\eta_t=\infty$;(iii) $\sum_{t=0}^{\infty}\eta_t^2<$

∞. 在算法 5 的步 4 中，由于值函数样本 $\hat{v}_t$ 与所估计的值函数 $\bar{V}(s_t)$ 单位相同，因而可以简单地取 $0\leq\eta_t\leq 1$，如令 $\eta_t=1/(t+1)$，$t=0,1,\cdots$. 然而，在随机梯度法式(37)中，由于等式右边 $(\bar{V}(s|\boldsymbol{\theta}_{t-1})-\hat{v})\nabla_{\boldsymbol{\theta}}\bar{V}(s|\boldsymbol{\theta}_{t-1})$ 与 $\boldsymbol{\theta}$ 的单位不一定相同，η 的取值还需仔细调整. Powell 对步长进行了较为详细的介绍[36]，有兴趣的读者可以参考.

⑦ 探索(Exploration)与利用(Exploitation)问题

算法 5 中采用前向动态规划法，且下一个状态 s_{t+1} 的选取都与当前状态 s_t 所做的决策 a_t 有关，这称为依照策略的学习方式(On-Policy Learning). 这种方法充分利用了前期估计得到的统计信息，会不断提高所遍历到的状态的值函数，而没有遍历过的状态的值函数的数值则相对较低. 这很容易导致算法收敛于局部最优解而非全局最优解.

针对这个问题，学者们又提出了不依照策略的学习方式(Off-Policy Learning). 但是，这种方式不能保证 ADP 算法收敛. 因此，又提出了一些折中的方案，如 Boltzmann 探索[40]等，在算法前期，先利用 Off-Policy Learning 遍历尽量多的状态，采集足够多的统计信息，而在算法后期，则使用 On-Policy Learning 方法，加快收敛速度.

3　基于马尔可夫决策 Petri 网的动态优化模型

Beccuti 等人[41-42]于 2007 年提出了马尔可夫决策 Petri 网(Markov Decision Petri Nets，MDPN)，将 MDP 的思想融入了 Petri 网中，其目的是为了提供一种比 MDP 更高层的建模工具，从宏观的角度反映决策者行为与系统行为的交替，并从语义的角度严格定义两种行为的转换过程.

3.1　马尔可夫决策 Petri 网

马尔可夫决策 Petri 网可分为两种子网：代表系统行为的随机子网(Probabilistic Subnet)以及代表决策者行为的非确定子网(Nondeterministic Subnet). 这两种子网通过立即变迁 $NdtoPr$ 与 $PrtoNd$ 同步. 随机子网的行为通过两类变迁 $Trun^{pr}$ 与 $Tstop^{pr}$ 来描述. $Trun^{pr}$ 代表系统运行的中间过程，而 $Tstop^{pr}$ 代表系统当前阶段运行过程的终止. 随机子网中的每个变迁都对应一个权值(weight)，用来计算某个状态下系统可实施变迁的概率. 此外，每个变迁还对应系统中一个触发该变迁的行为(act)，包括组件集合的一个子集.

在 MDPN 中，系统由多个组件构成. 这些组件有些是可控的，有些是不可控的. 非确定子网用两类变迁 T_g^{nd} 与 T_l^{nd} 来描述. T_g^{nd} 代表决策者系统级的控制行为，而 T_l^{nd} 代表组件级的控制行为. 与随机子网中的变迁类似，非确定子网中的这两类变迁又可以细分为 $Trun_g^{nd}$、$Tstop_g^{nd}$、$Trun_l^{nd}$、$Tstop_l^{nd}$. 每个非确定子网中的变迁还对应一个对象，用以说明该变迁对应行为的施加组件对象.

定义 2(马尔可夫决策 Petri 网)[42]. 一个马尔可夫决策 Petri 网是一个四元组 $MN=\{Comp^{pr},Comp^{nd},N^{pr},N^{nd}\}$，其中

$Comp^{pr}$ 是一个有限非空系统组件集合；

$Comp^{pr}\subseteq Comp^{pr}\cup\{id_s\}$ 是非空可控组件集合，其中 id_s 代表整个系统；

N^{pr} 由 3 部分构成：① 一个带有优先级的 Petri 网 $\{P,T^{pr},I^{pr},O^{pr},H^{pr},prio^{pr},m_0\}$；② 一个对应的权值 $weight$：$T^{pr}\rightarrow\mathbb{R}$；③ 一个对应的行为 act：$T^{pr}\rightarrow 2^{Comp^{pr}}$，其中 $T^{pr}=Trun^{pr}\cup Tstop^{pr}$；

N^{nd} 由两部分构成：① 一个带有优先级的 Petri 网 $\{P,T^{nd},I^{nd},O^{nd},H^{nd},prio^{nd},m_0\}$；② 一个对应的对象 obj：$T^{nd}\rightarrow Comp^{nd}$，其中 $T^{nd}=Trun^{nd}\cup Tstop^{nd}$.

此外，MDPN 还需满足以下条件：① 一个变迁不能既是非确定变迁又是随机变迁；② 每个系统组件至少可以触发一个 $Tstop^{pr}$ 类型的变迁；③ 每个可控系统组件至少是一个 $Tstop^{nd}$ 类型变迁的对象.

在 MDPN 中，收益分为两部分. 第一部分是状态收益，即系统到达某个状态后得到的收益. 第二部分是行为收益，定义为一连串决策行为所得到的收益. 行为收益与行为序列的顺序无关.

3.2　马尔可夫决策 Petri 网的建模与分析

当构建好决策者行为子模型与系统行为子模型后，需要加入一些附加的位置与变迁，将两个子模型连接起来.

一个基本的 MDPN 模型如图 4 所示. 位置 $Stop_i^{pr}$、Run_i^{pr}、$Stop_i^{nd}$、Run_i^{nd}、$Stop_0^{nd}$、Run_0^{nd} 用来在系统组件、整个系统以及决策者之间进行同步. 对于每个组件 i，都有一个 $Stop_i^{pr}$ 与 Run_i^{pr} 位置. 若决策者采取了针对整个系统的全局性行为，则需插入位置 $Stop_0^{nd}$ 与 Run_0^{nd}. 若采取的是针对某个系统组件的局部行为，则需插入位置 $Stop_i^{nd}$ 与 Run_i^{nd}.

变迁 $NdtoPr$ 与 $PrtoNd$ 描述系统行为与决策者行为的交替进行. $NdtoPr$ 只有在 $Stop_0^{nd}$ 与所有

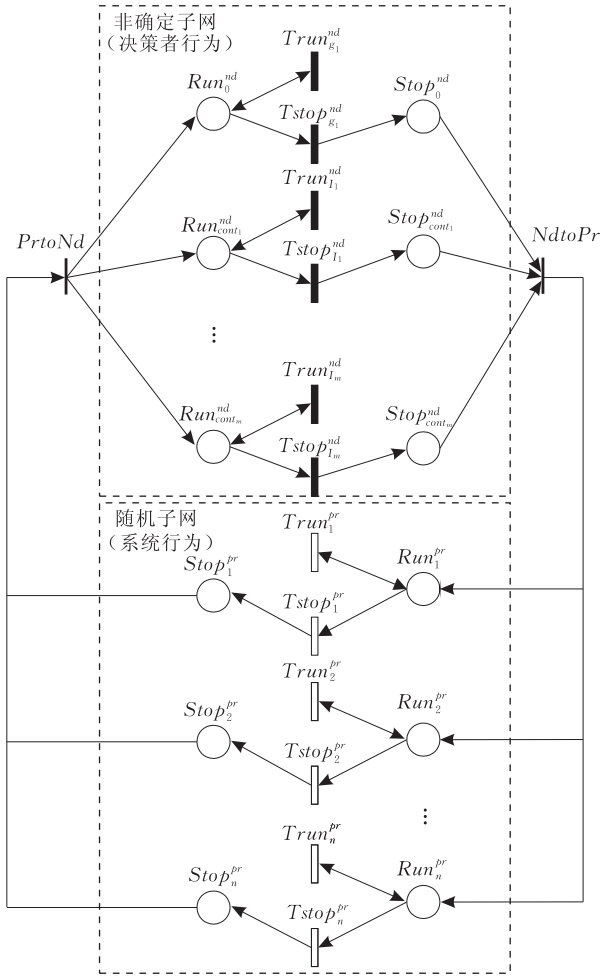


图 4 一个基本的 MDPN 模型

$Stop_i^{nd}$ 位置都有标记时才能实施,代表模型由决策态转移到系统运行态. 而 $PrtoNd$ 相反只有在 $Stop_i^{pr}$ 位置都有标记才能实施,代表由系统运行态转移到决策态.

3.3 马尔可夫决策 Petri 网的求解

MDPN 的求解过程可以分为如下 4 个步骤[41]:

(1) 由 MDWN 模型求得该模型的可达图 RG

可达状态集合(Reachability Set,RS)可分为两部分:非确定状态(RS_{nd})与随机状态(RS_{pr}). 在非确定状态中,只有 T_{nd} 类型的变迁是可实施的,而在随机状态中,只有 T_{pr} 类型的变迁是可实施的.

(2) 将可达图 RG 规约为非确定可达图 RG_{nd}

在 RG 中,定义非确定子路径与随机子路径分别为 RG 中经过同样类型状态的最大路径. 搜索所有非确定子路径,并将每个非确定子路径压缩为一个决策状态,代表所有可能的决策行为,得到非确定可达图 RG_{nd}.

(3) 将非确定可达图 RG_{nd} 规约为 MDP 可达图 RGMDP

搜索所有随机子路径,通过路径途中经过变迁的权值,计算各个路径的概率,并将每个随机子路径压缩为 RG 中的一条有向弧,代表宏观的系统状态转移,得到 MDP 可达图 RG_{MDP}.

(4) 计算对应 MDP 的转移概率

转移概率矩阵为

$$\boldsymbol{P}=\Big(\sum_{n=0}^{\infty}(\boldsymbol{P}^{(pr,pr)})^n\boldsymbol{P}^{(pr,nd)}\Big) \tag{41}$$

其中,$\boldsymbol{P}^{(pr,pr)}$ 为 RG 中从一个随机状态转移到另一个随机状态、且途中没有非确定状态的概率,$\boldsymbol{P}^{(pr,nd)}$ 为从一个随机状态转移到非确定状态的概率. 转移矩阵 $\boldsymbol{P}$ 可用式(42)进行计算:

$$\boldsymbol{P}=\begin{cases}\Big(\sum\limits_{k=0}^{n_0}(\boldsymbol{P}^{(pr,pr)})^k\Big)\boldsymbol{P}^{(pr,nd)}, \\ \qquad\text{若随机状态集合不存在回路} \\ (\boldsymbol{I}-\boldsymbol{P}^{(pr,pr)})^{-1}\boldsymbol{P}^{(pr,nd)}, \\ \qquad\text{若随机状态集合中存在回路}\end{cases} \tag{42}$$

(5) 根据 Bellman 方程计算 MDP 中的最优策略

可根据算法 1 或算法 2 求得 MDP 中的最优策略,也就是 MDWN 模型中的最优控制策略.

3.4 应用与扩展

本小节中将以一个可修复系统为例[41],对 MDWN 模型的各个要素进行说明,其模型如图 5 所示. 左半部分为随机子网,描述一个既可能正常工作(变迁 $Workfine$)、又可能失效(变迁 $FailProc$)的系统组件. 右半部分为非确定子网,描述决策者的行为,包括分配资源以维修失效组件(变迁 $AssignRes$)与

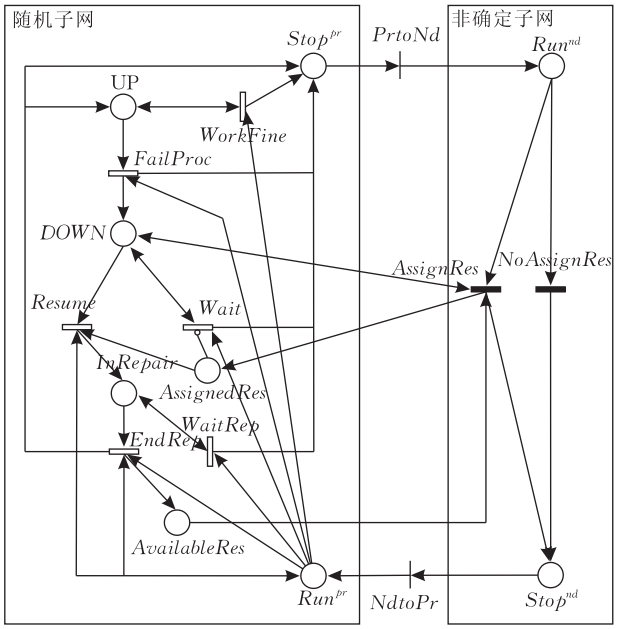


图 5 一个可修复系统的 MDWN 模型

不分配资源（变迁 *NoAssignRes*）. 在随机子网中，$Tstop^{pr}$ 类型的变迁有 *WorkFine*、*Fail*、*Wait*、*EndReq*，$Trun^{pr}$ 类型的变迁仅有一个，即 *Resume*. 非确定子网中所有的变迁均为 $Tstop^{nd}$ 类型.

在 MDPN 模型中，在位置、标记以及变迁中增加颜色的概念后，可进一步得到马尔可夫良构 Petri 网（Markov Decision Well-formed Nets，MDWN）模型. 这种模型可以较好地处理具有对称属性的系统，有效地缩小问题空间. MDPN 与 MDWN 的模型与算法都已经集成在 GreatSPN 工具中[43-45].

目前，针对 MDPN 与 MDWN 模型的应用研究已经逐步开展. 文献[46]分别利用 MDPN/MDWN 研究了高质量视频处理中的资源管理问题. 文献[47]研究了无线传感器网络中，对象移动跟踪的最优能源管理问题. 文献[48-49]研究了一类非确定可维修故障树（Non deterministic Repairable Fault Trees，NdRFT）模型与 MDWN 模型转换的方法，并将 MDWN 模型作为求解 NdRFT 模型最优策略的方法.

3.5　MDPN 与 ADP 的结合

3.3 小节中的 MDPN 求解方法，将 MDPN 规约为 MDP，然后再利用精确求解算法进行求解. 这种方式使得 MDPN 的求解仍然存在“状态空间爆炸”问题. 为此，我们将 MDPN 与 ADP 结合，利用 ADP 中 Monte Carlo 仿真的方法，解决 MDPN 的近似求解问题.

在结合 ADP 方法的 MDPN 中，不需要通过 Petri 模型得到完全的可达图 RG，也不需要通过式(42)计算状态之间的转移概率. 相反的，在模拟系统与决策者行为的同时，不断地更新可达状态集 RS. 由于 MDP 中只关注决策与系统运行的最终状态，因此只需记录位置 $Stop_i^{nd}$ 中全部都有标记的状态（称为决策终结状态），或者位置 $Stop_i^{pr}$ 中全部都有标记时的状态（称为系统终结状态）. 若该状态在 RS 中不存在，才将该状态加入 RS 中. 若新加入 RS 中的状态为决策终结状态，则为其关联一个后决策值函数并设定其初始值，其功能与式(25)类似.

进行仿真时，在每个系统终结状态可利用式(26)进行决策，并得到一个值函数样本. 注意行为 a_t 可能是一个行为序列. 此时，可利用式(35)，在值函数样本的基础上，更新其上一时刻决策终结状态的值函数. 这样，就将 ADP 中的前向动态规划算法集成到了 MDWN 中.

4　基于随机博弈网的动态优化模型

上述 MDP、MDPN 以及 MDWN 模型，都只能描述具有集中式控制设施的系统，即系统内只有一个决策者. 在现实生活中，还存在着大量具有多个决策者的系统. 上述模型在处理这类问题时，只能从各个决策者的角度分别建模，而将其他决策者视为不可控外部随机事件，无法体现出决策者之间的联系. 文献[50]于 2008 年首次提出了随机博弈网（Stochastic Game Nets，SGN），将动态随机博弈与随机 Petri 网结合，能够对具有多个决策者的系统进行建模分析.

动态随机博弈可以看作是马尔可夫决策过程的扩展，可包含多个决策者并能体现出他们之间的复杂关系，包括：(1) 竞争关系. 即每个决策者只关心最大化自己的收益；(2) 合作关系. 即所有决策者作为一个群体关心的是总收益. 将动态随机博弈与随机 Petri 网相结合，有助于系统的细粒度建模与简化求解.

4.1　随机博弈网

定义 3（随机博弈网）[51].　一个随机博弈网是一个 9 元组：$SGN=\{N,P,T,F,\pi,\lambda,R,U,M_0\}$，其中：

$N=\{1,2,\cdots,n\}$ 是决策者（博弈局中人）的集合；

P 是有限的位置集合；

$T=T^1\cup T^2\cup\cdots\cup T^n$ 是有限变迁的集合，其中 T^k 是第 $k\in N$ 个决策者的行为；

$\pi:T\to[0,1]$ 是决策者选择某个变迁的概率；

$F\subseteq I\cup O$ 是弧的集合，其中 $I\subseteq(P\times T)$，$O\subseteq(T\times P)$，且有 $P\cap T=\varnothing$，$P\cup T\neq\varnothing$. 记 x 的前集合为 ${}^{\cdot}x=\{y\mid(y,x)\in F\}$，$x$ 的后集合为 $x^{\cdot}=\{y\mid(x,y)\in F\}$；

$R:T\to(R_1,R_2,\cdots,R_N)$ 为决策者采用某个变迁所对应行为所得的收益函数，其中 $R_i\in(-\infty,+\infty)$，$i\in N$；

$\lambda=\{\lambda_1,\lambda_2,\cdots,\lambda_W\}$ 为变迁的实施速率，其中 W 是变迁的个数；

U 是决策者的总收益函数；

M_0 是起始状态，代表所有决策者的最初状态.

在该定义中，P 是博弈的状态，在某个位置 $p\in P$ 中有标记意味着所有决策者都在该状态中. 位置 p 中的标记 s 对应一个收益向量

$$\boldsymbol{h}_p(s)=(h_p^1(s),h_p^2(s),\cdots,h_p^k(s)) \tag{43}$$

其中 $h_p^k(s)$ 为决策者 k 在状态 p 中所得的收益. 当变迁 t 实施,所有决策者都会得到收益

$$\boldsymbol{R}(t)=(R_1(t),R_2(t),\cdots,R_k(t)) \tag{44}$$

其中 $R_i(t)$ 为决策者 i 所得的收益. 若标记经过变迁 t 到达位置 p,则收益都会累加在标记的收益向量 $\boldsymbol{h}_p(s)$ 中.

在 SGN 中,当系统运行至状态 p 时,决策者 k 的策略可定义为

$$\pi_k(p)=\{\pi(t_j^k)\}_{(t_j^k\in p^\cdot)\wedge(t_j^k\in T^k)} \tag{45}$$

其中,$\pi(t_j^k)$ 是决策者 k 采取变迁(即行为)t_j 的概率. 显然,对于所有状态 p,都有

$$\sum_{(t_j^k\in p^\cdot)\wedge(t_j^k\in T^k)}\pi(t_j^k)=1 \tag{46}$$

进一步,参照博弈论中纳什均衡的概念,可以定义 SGN 中的均衡策略 $\boldsymbol{\pi}^*=(\pi_1^*,\pi_2^*,\cdots,\pi_n^*)$ 满足

$$U_k(\pi_1^*,\cdots,\pi_{k-1}^*,\pi_k^*,\pi_{k+1}^*,\cdots,\pi_n^*)\geq U_k(\pi_1^*,\cdots,\pi_{k-1}^*,\pi_k,\pi_{k+1}^*,\cdots,\pi_n^*),\ \forall k\in[1,2,\cdots,n] \tag{47}$$

其中 π_k 是除 π_k^* 外所有其它可能的策略. 均衡策略的含义在于,某个决策者在其他决策者都不偏离均衡策略的情况下,采用非均衡策略不会取得比采用均衡策略更高的收益. 换句话说,该决策者没有偏离均衡决策的动机.

值得注意的是,在 MDP 中一般都使用确定行为(称为纯策略)作为最优解(一些例外的情况如探索/利用问题中会采取一些不确定行为来主动学习值函数). 而在 SGN 中,一般采用在行为空间的概率分布(混合策略)作为均衡解,因为在多人决策问题中,在纯策略意义下一般不存在均衡解,而在混合策略意义下一定存在均衡解.

4.2 随机博弈网的建模与分析

构建一个 SGN 模型一般分为 4 个步骤[52]:

(1) 建立每个决策者的子 SGN 模型.

在实际系统中,识别出 SGN 对应的要素,包括

① 变迁. 变迁代表决策者的行为. 注意行为集合中也可能包括空行为 ϕ,即决策者不采取任何行为.

② 收益. 对于每个变迁 t,赋予其一个收益函数 R,其每个分量 R_i 代表决策者 i 在该行为结束后所得的收益.

③ 位置集合 P. 每个位置 p 代表系统的一个状态.

(2) 描述纳什均衡条件.

对于竞争博弈,每个决策者的目标是最大化自己的收益;对于合作博弈,每个决策者的目标是最大化所有决策者的收益的总和. 对于有限时间 SGN,可仿照 MDP 中式(3)定义决策者 i 的总收益:

$$U_i^\pi=\mathrm{E}\Big\{\sum_{n=0}^{N}\alpha^n R_n^\pi\Big\} \tag{48}$$

其中 N 是时间的长度,R_n^π 是阶段 n 使用策略 π 时所得的收益. 注意 U_i^π 与所有决策者的策略都相关,而并非只与 i 自己的策略相关. 对于只有两个决策者的系统,均衡策略 $\pi^*=\{\pi^{1*},\pi^{2*}\}$ 满足 $U_1^{\pi^{1*},\pi^{2*}}\geq U_1^{\pi^1,\pi^{2*}}$ 且 $U_2^{\pi^{1*},\pi^{2*}}\geq U_2^{\pi^{1*},\pi^2}$.

(3) 求解纳什均衡策略.

一般情形下求解均衡策略难度较大. 本文仅对只有两个决策者的特殊情况进行讨论,此时,系统求解问题可以化归为一个静态非线性规划问题,详细请参见 4.3 小节.

(4) 合并子模型,建立全局 SGN 模型.

将子模型中含义相同的位置合并,可将所有子模型进行组合,得到全局 SGN 模型.

4.3 随机博弈网的求解

文献[52-54]给出了二人动态博弈的纳什均衡求解方法,该方法基于文献[55],将二人动态博弈问题化归为一个静态非线性规划(Non Linear Programming,NLP)问题:

$$\min_{U_1,U_2,\pi_1,\pi_2}\ \mathbf{1}^{\mathrm{T}}[U_k-R_k(\pi_1,\pi_2)-\alpha\boldsymbol{P}(\pi_1,\pi_2)U_k]$$

s.t.:

$$R_1(p_i)\pi_2(p_i)+\alpha\boldsymbol{T}(p_i,U_1)\pi_2(p_i)\leq\mathbf{1}^{\mathrm{T}}U_1(p_i),$$

$$(\pi_1(p_i))^{\mathrm{T}}R_2(p_i)+\alpha(\pi_1(p_i))^{\mathrm{T}}\boldsymbol{T}(p_i,U_2)\leq\mathbf{1}^{\mathrm{T}}U_2(p_i),$$

其中,值向量为

$$\boldsymbol{T}(p,U)=\{[\boldsymbol{P}(p_1|p,t^1,t^2),\cdots,\boldsymbol{P}(p_{|P|}|p,t^1,t^2)]^{\mathrm{T}}U_k\}$$

且有 $k\in\{1,2\}$,$i\in\{1,\cdots,|P|\}$,$p_i\in\boldsymbol{P}$,$t^1\in\boldsymbol{T}^1$,$t^2\in\boldsymbol{T}^2$. 该非线性规划的最小全局解,就是 SGN 中的纳什均衡解.

4.4 随机博弈网的模型化简与合并

当利用 SGN 对实际问题建模时,通常会遇到的一个问题是决策者的行为复杂,导致所建立的 SGN 模型难于求解分析. 文献[56]针对这一问题,给出了一些 SGN 模型化简的方法,例如在图 6 左半部分所示的模型,可以等价地化简为右半部分的简单模型.

4.2 小节中提到,在构建 SGN 全局模型时,需要进行子模型合并. 文献[57]讨论了在利用 SGN 对网络攻防进行建模时,子模型合并的方法,将决策者之间的关系分为两类:禁止类型与结束类型,相应的组合方法如图 7 所示.

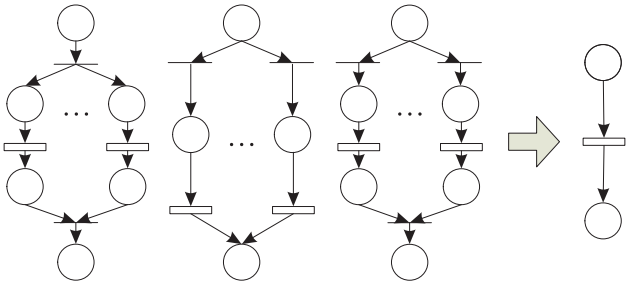
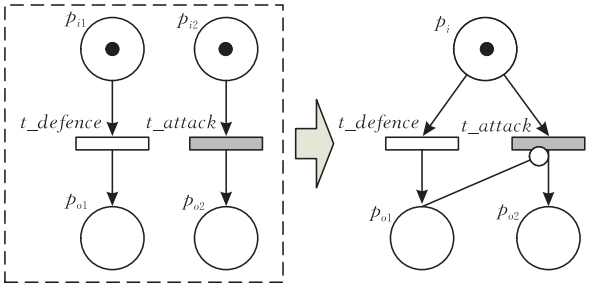
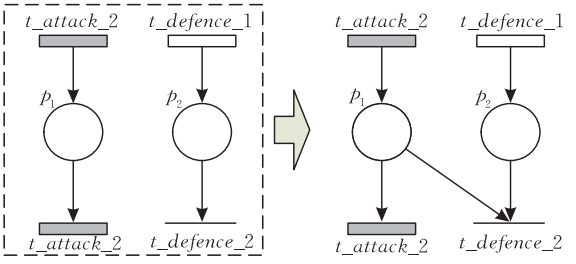


图 6　SGN 模型的化简



(a) 禁止类型



(b) 结束类型

图 7　模型的合并

一方面，当防御与攻击行为都可以实施时，若防御先实施，则可禁止攻击行为的实施(图 7(a))．另一方面，防御行为的实施，也可以使得整个攻击过程结束(图 7(b))．

4.5　应用与扩展

文献[57]在 SGN 的基础上进行延伸，进一步提出了针对网络安全攻防的攻击-防御随机博弈网(Attack-Defense Stochastic Game Nets，ADSGN)，准确地刻画了网络攻击者与防御者之间的零和竞争博弈关系．本小节中以企业网中的安全攻防问题为例，说明 SGN 的建模方法．

在一个典型企业网中，从攻击者与网络管理员的观点来看，网拓扑结构可抽象为图 8．攻击者可进行一些攻击行为，如扫描网络脆弱性、攻击数据库、破译服务器密码等．网络管理员可以进行一些相应的防御措施，例如利用入侵检测系统进行扫描、阻止攻击者 IP 进入系统、移除嗅探器等．攻击者 SGN 子模型与防御者 SGN 子模型分别如图 9、图 10 所示．这两个子模型从不同决策者的角度刻画了决策者在每个决策时间可能采取的攻防行为．应用 4.4 小节中的模型化简与合并技术，可将图 9、图 10 中的 SGN 子模型合并为图 11 所示的 SGN 完整模型．

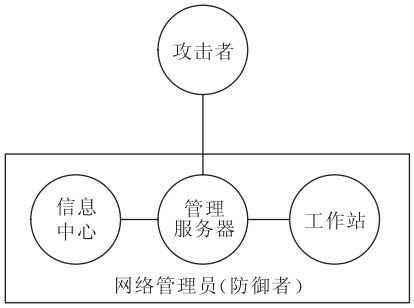


图 8　一个企业网网络拓扑结构

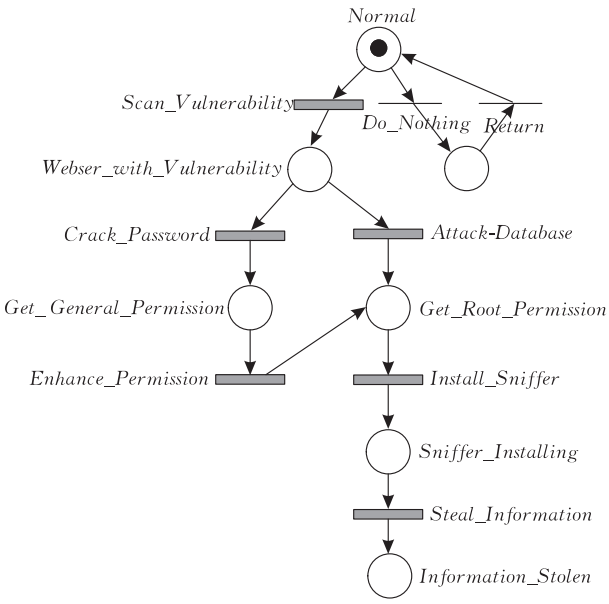


图 9　SGN 攻击者子模型

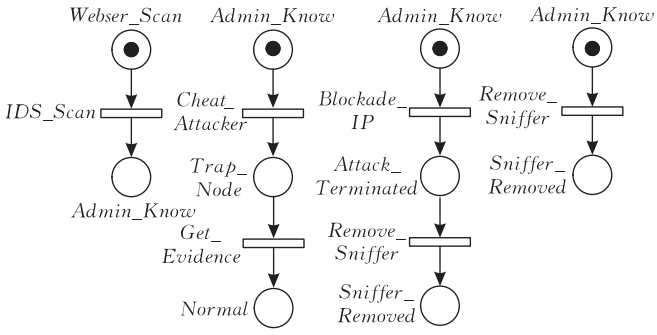


图 10　SGN 防御者子模型

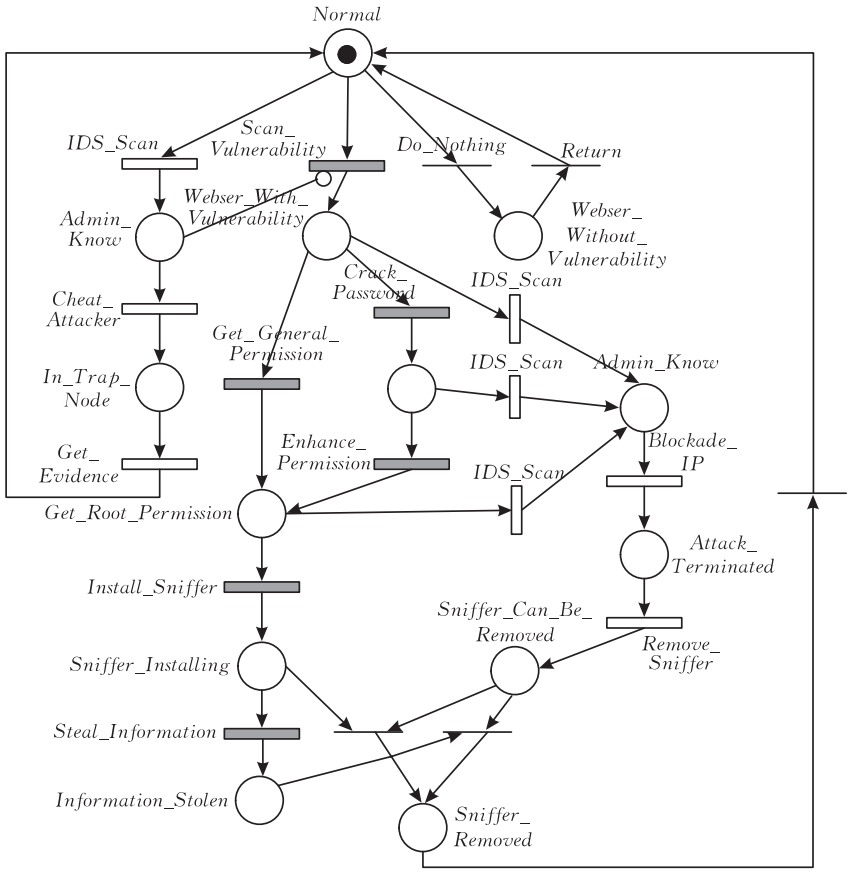


图 11 SGN 完整模型

目前,关于 SGN 的应用研究绝大多数都集中在网络安全方面,如文献[58]研究了利用网络连接关系与脆弱性信息等输入数据生成 SGN 模型的方法,文献[52-53,57]研究了企业网中的安全问题,文献[56,59]研究了电子商务中的若干安全问题,文献[54]研究了电子邮件蠕虫病毒的传播问题.另外,在无线网络领域,也有一些初步的研究成果,如文献[60]研究了无线网络中共享信道竞争的性能评价问题.总之,SGN 是一个正在发展与完善中的研究领域,在理论与应用方面均具有较为广阔的前景.

5 结论与展望

本文对动态优化在计算机系统与计算机网络中的建模、求解与应用进行了综述.相较于静态优化,动态优化可以精确地刻画系统的时变性.本文主要讨论了 3 种理论模型,即马尔可夫决策过程模型、马尔可夫决策 Petri 网模型以及随机博弈网模型,对这些模型的建模方法、求解算法、与应用实例进行了较为深入的研究.

计算机系统与计算机网络中的资源种类复杂,数量众多.面对这种复杂的应用环境,如何合理地运用动态优化理论对系统进行建模,并采取适当的求解算法进行(近似)求解具有极大挑战性.在本文最后以以下几点为例,列举一些未来可能的研究方向:

(1)马尔可夫决策过程的近似求解算法.众所周知,目前还不存在适用于所有 MDP 近似求解的统一"万能药"算法.很多看似合适的算法得出的近似解往往质量较差,在某些环境下甚至会出现算法不收敛的情况.近似解的质量在很大程度上还取决于算法设计者对领域专业知识的理解程度与算法设计经验,Powell 甚至将 ADP 近似值函数中的特征函数选取称为一种"艺术(art)"[36].对于近似求解算法的应用范围、解的质量以及收敛性等一系列问题,还需要进一步深入研究.

(2)马尔可夫决策 Petri 网与随机博弈网等模型的近似求解问题.一方面,在 MDPN/MDWN 与 SGN 模型中,虽然存在一些对模型进行化简的方法(如 4.4 节),但是这些方法往往局限于对某些特定模型结构的化简,还无法处理更为复杂的模型.另一方面,这些模型均采用精确求解算法,这使得利用这两种模型对大规模系统进行建模分析时求解较为困

难，大大限制了其应用范围. 在 3.5 节中我们对 MDWN 中结合 ADP 算法的方式进行了一些初步的探索，但还不够深入. 后续工作还应对这些模型的近似求解算法进行研究.

(3) 随机博弈网的应用研究拓展. 目前随机博弈网模型方法主要应用于网络安全分析中，而就随机博弈网的模型特点来说，它可以适用于模型分析具有多个独立决策者参与的计算机系统应用，如无线网络、对等网络(P2P)以及社交网络等. 进一步的研究工作将针对这些应用的特点，研究有针对性随机博弈网的建模与分析方法，拓展随机博弈网的应用领域.

参　考　文　献

[1] Murugesan S, Schniter P, Shroff N B. Multiuser scheduling in a Markov-modeled downlink using randomly delayed ARQ feedback. IEEE Transactions on Information Theory, 2012, 58(2): 1025-1042

[2] Zhao Qing, Swaml Anathram. A decision-theoretic framework for opportunistic spectrum access. IEEE Wireless Communications, 2007, 14(4): 14-20

[3] Zhao Q et al. Decentralized cognitive MAC for opportunistic spectrum access in Ad Hoc networks: A POMDP framework. IEEE Journal on Selected Areas in Communications, 2007, 25(3): 589-600

[4] Simunic Tajana, Benini Luca, Glynnt Peter, De Micheli Giovanni. Dynamic power management for portable systems// Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. New York, USA, 2000: 11-19

[5] Srivastava Rahul, Koksal Can Emre. Energy optimal transmission scheduling in wireless sensor networks. IEEE Transactions on Wireless Communications, 2010, 9(5): 1550-1560

[6] Chen Huan, Huang Cheng-Wei. Power management modeling and optimal policy for IEEE 802.11 WLAN System// Proceedings of the IEEE 60th Vehicular Technology Conference. Los Angeles, USA, 2004: 4416-4421

[7] Choi Kae Won. Adaptive sensing technique to maximize spectrum utilization in cognitive radio. IEEE Transactions on Vehicular Technology, 2010, 59(2): 992-998

[8] Ouyang Wenzhuo, Murugesan Sugumar, Eryilmaz Atilla, Shroff Ness B. Exploiting channel memory for joint estimation and scheduling in downlink networks//Proceedings of the IEEE INFOCOM. Shanghai, China, 2011: 3056-3064

[9] Su Chi-Jiun, Tassiulas Leandros, Tsotras Vassilis J. Broadcast scheduling for information distribution//Proceedings of the IEEE INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Kobe, 1997: 109-117

[10] Wang Rui, Lau V K N, Huang Huang. Delay optimal power control and relay selection for two-hop cooperative OFDM systems via distributive stochastic learning//Proceedings of the 2010 IEEE International Symposium on Information Theory (ISIT). Austin, Texas, 2010: 1843-1847

[11] Karmokar Ashok K, Djonin Dejan V, Bhargava Vijay K. Optimal and suboptimal packet scheduling over correlated time varying flat fading channels. IEEE Transactions on Wireless Communications, 2006, 5(2): 446-456

[12] Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge, MA: The MIT Press, 1998

[13] Haas Zygmunt, Halpern Joseph Y, Li Erran L, Wicker Stephen B. A decision-theoretic approach to resource allocation in wireless multimedia networks//Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. New York, USA, 2000: 86-95

[14] Rykov V V, Efrosinin D. Optimal control of queueing system with heterogeneous servers. Queueing Systems, 2004, 46(3-4): 389-407

[15] Maillart L M, Cassady C R, Rainwater C, Schneider K. Selective maintenance decision-making over extended planning horizons. IEEE Transactions on Reliability, 2009, 58(3): 462-469

[16] Chana G K, Asgarpoor S. Optimum maintenance policy with Markov processes. Electric Power Systems Research, 2006, 76(6-7): 452-456

[17] Djonin Dejan V, Krishnamurthy Vikram. MIMO transmission control in fading channels—A constrained Markov decision process formulation with monotone randomized policies. IEEE Transactions on Signal Processing, 2007, 55(10): 5069-5083

[18] Derman C, Klein M. Some remarks on finite horizon Markovian decision models. Operation Research, 1965, 13(2): 272-278

[19] Beutler F J, Ross K W. Optimal policies for controlled Markov chains with a constraint. Journal of Mathematical Analysis and Applications, 1985, 112(1): 236-252

[20] Thomas L C. Constrained Markov decision processes as multi-objective problems. Translated by White D J, French S, Hartley R. London: Academic Press, 1983

[21] Mihaylova L, Lefebvre T, Bruyninckx H, Gadeyne K, Schutter J D. A comparison of decision making criteria and optimization methods for active robotic sensing numerical methods and applications//Proceedings of the 1st European Symposium on Ambient Intelligence. Veldhoven, 2003: 316-324

[22] Chatterjee K. Markov decision processes with multiple long-run average objectives//Proceedings of the FSTTCS 2007. Lecture Notes in Computer Science 4855. Springer, 2007: 473-484

[23] Chatterjee K, Majumdar R, Henzinger T A. Markov decision processes with multiple objectives//Proceedings of the STACS 2006. Lecture Notes in Computer Science 3884. Springer, 2006: 325-336

[24] Karush W, Dear R E. Optimal strategy for item presentation in a learning process. Management Science, 1967, 13(11): 773-785

[25] Krishnamurthy V, Djonin D V. Structured threshold policies for dynamic sensor scheduling—A partially observed Markov decision process approach. IEEE Transactions on Signal Processing, 2007, 55(10): 4938-4957

[26] Chen Yunxia, Zhao Qing, Swami A. Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors. IEEE Transactions on Information Theory, 2008, 54(5): 2053-2071

[27] Zhao Qing, Krishnamachari B, Liu Keqin. On myopic sensing for multi-channel opportunistic access: Structure, optimality, and performance. IEEE Transactions on Wireless Communications, 2008, 7(12): 5431-5440

[28] Liu Keqin, Zhao Qing, Krishnamachari B. Dynamic multichannel access with imperfect channel state detection. IEEE Transactions on Signal Processing, 2010, 58(5): 2795-2808

[29] Ahmad Sahand Haji Ali, Liu Mingyan, Javidi Tara, Zhao Qing, Krishnamachari Bhaskar. Optimality of myopic sensing in multichannel opportunistic access. IEEE Transactions on Information Theory, 2009, 55(9): 4040-4050

[30] Ji Shihao, Parr R, Carin L. Non-myopic multi-aspect sensing with partially observed Markov decision processes. IEEE Transactions on Signal Processing, 2005, 55(6): 2720-2730

[31] Kreucher C, Hero A, Kastella K. A comparison of task driven and information driven sensor management for target tracking//Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference (CDC-ECC'05). Spain, 2005: 4004-4009

[32] Liu Liming, Lu Yumao. Dynamic traffic controls for web-server networks. Computer Networks, 2004, 45(4): 523-536

[33] Givan R, Leach S, Dean T. Bounded parameter Markov decision processes. Artificial Intelligence, 2000, 122(1-2): 71-109

[34] Givan R, Leach S, Dean T. Bounded parameter Markov decision processes//Recent Advances in AI Planning. Lecture Notes in Computer Science 1348, 1997: 234-246

[35] Bertsekas D, Tsitsiklis J. Neuro-Dynamic Programming. Belmont, MA: Athena Scientific, 1996

[36] Powell W B. Approximate Dynamic Programming: Solving the Curses of Dimensionality. New York: John Wiley and Sons, 2007

[37] Roy B V, Bertsekas D P, Lee Y, Tsitsiklis J N. A neuro-dynamic programming approach to retailer inventory management//Proceedings of the 36th Conference on Decision and Control. San Diego, CA, 1997: 4052-4057

[38] Powell W B, George A, Ayari B B, Simao H P. Approximate dynamic programming for high dimensional resource allocation problems//Proceedings of the 2005 IEEE International Joint Conference on Neural Networks (IJCNN'05). Montreal, Canada, 2005, 5: 2989-2994

[39] George A, Powell W B, Kulkarni S R. Value function approximation using multiple aggregation for multiattribute resource management. Journal of Machine Learning Research, 2008, 9: 2079-2111

[40] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 1996, 4: 237-285

[41] Beccuti M, Franceschinis G, Haddad S. Markov decision Petri net and Markov decision well-formed net formalisms. Technical Report, TR-INF-2007-02-01-UNIPMN, 2007, available via WWW at URL http: //www. di. unipmn. it/

[42] Beccuti M, Franceschinis G, Haddad S. Markov decision Petri net and Markov decision well-formed net formalisms//Proceedings of the 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency. 2007: 43-62

[43] Beccuti M, Raiteri D C, Franceschinis G, Haddad S. A framework to design and solve Markov decision well-formed net models//Proceedings of the 4th International Conference on Quantitative Evaluation of Systems. Washington DC, USA, 2007: 165-166

[44] Baarir S, Beccuti M, Cerotti D, Pierro M D, Donatelli S, Franceschinis G. The GreatSPN tool: Recent enhancements. ACM Performance Evaluation Review, 2009, 36(4): 4-9

[45] Beccuti M, Franceschinis G, Haddad S. MDWNsolver: A framework to design and solve Markov decision Petri nets. International Journal of Performability Engineering, 2011, 7(5): 417-428

[46] Beccuti M. Modeling and analysis of probabilistic systems: Formalisms and efficient algorithms [Ph. D. dissertation]. Dipartimento di Informatica, Universita degli Studi di Torino, 2009

[47] Beccuti M, Raiteri D C, Franceschinis G. Multiple abstraction levels in performance analysis of WSN monitoring systems//Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools. Brussels, Belgium, 2009: 1-10

[48] Beccuti M, Franceschinis G, Raiteri D C, Haddad S. Parametric NdRFT for the derivation of optimal repair strategies//Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks. Lisbon, Portugal, 2009: 399-408

[49] Beccuti M, Raiteri D C, Franceschinis G, Haddad S. Non deterministic repairable fault trees for computing optimal repair strategy//Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools. Brussels, Belgium, 2008: 1-10

[50] Lin C, Wang Y Z, Wang Y. A stochastic game nets based approach for network security analysis//Proceedings of the 29th International Conference on Application and Theory of Petri Nets and Other Models of Concurrency, Concurrency Methods: Issues and Applications 2008 Workshop (Invited paper). Xi'an, China, 2008: 21-33

[51] Lin Chuang, Wang Yuan-Zhuo, Wang Yang. Stochastic Game Nets Based Network Security Evaluation and Analysis. Beijing: Tsinghua University Press, 2011(in Chinese)
(林闯，王元卓，汪洋. 基于随机博弈模型的网络安全评价与分析. 北京：清华大学出版社，2011)

[52] Wang Y Z, Lin C, Wang Y, Meng K. Security analysis of enterprise network based on stochastic game nets model//Proceedings of the 2009 IEEE International Conference on Communications (ICC'09). Dresden, Germany, 2009: 1-5

[53] Wang Y Z, Yu M, Li J Y, Meng K, Lin C, Cheng X Q. Stochastic game net and applications in security analysis for enterprise network. International Journal of Information Security, 2012, 11(1): 41-52

[54] Yu M, Wang Y Z, Liu Li, Cheng X Q. Modeling and analysis of email worm propagation based on stochastic game nets//Proceedings of the 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'11). Gwanju, Korea, 2011: 381-386

[55] Filar J, Vrieze K. Competitive Markov Decision Processes. New York: Springer-Verlag, 1996

[56] Wang Y Z, Lin C, Meng K. Security analysis for online banking system using hierarchical stochastic game nets model//Proceedings of the GLOBECOM'09. Hilton Hawaiian Village, Honolulu, Hawaii, USA, 2009: 1-6

[57] Wang Y Z, Li J Y, Meng K, Lin C, Cheng X Q. Modeling and security analysis of enterprise network using attack-defense stochastic game net. Security Communication Networks, 2012

[58] Wang Yuan-Zhuo, Lin Chuang, Cheng Xue-Qi, Fang Bin-Xing. Analysis for network attach-defense based on stochastic game model. Chinese Journal of Computers, 2010, 33(9): 1748-1762(in Chinese)
(王元卓，林闯，程学旗，方滨兴. 基于随机博弈模型的网络攻防量化分析方法. 计算机学报，2010，33(9)：1748-1762)

[59] Wang Y Z, Lin C, Meng K, Lv J J. Analysis of attack actions for E-commerce based on stochastic game nets model. Journal of Computers, 2009, 4(6): 461-467

[60] Wan J X, Lin C, Chen X, Meng K, Wang Y Z. Performance analysis of channel contention in wireless Ad Hoc networks: A stochastic game nets approach//Proceedings of the GLOBECOM 2010. Miami, USA, 2010: 1-5

LIN Chuang, born in 1948, Ph. D., professor, Ph. D. supervisor. His research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications.

WAN Jian-Xiong, born in 1982, Ph. D. candidate. His research interests include performance evaluation and optimal control.

XIANG Xu-Dong, born in 1986, Ph. D. candidate. His research interests include performance evaluation and optimal control.

MENG Kun, born in 1980, Ph. D. candidate. His research interests include performance evaluation and stochastic models.

WANG Yuan-Zhuo, born in 1978, Ph. D, associate professor. His research interests include Petri net theory, and network security.

Background

Dynamic optimization theory is a powerful theoretical tool for modeling and solving sequential decision problems. It receives much attention from many academic areas such as operation research community, artificial intelligence community, and control theory community. This paper presents a brief overview of the models, solution techniques, as well as application of dynamic optimization theory in the field of computer systems and computer networks. Specifically, we discuss two kinds of extended dynamic optimization model, i. e., Markov Decision Petri Nets and Stochastic Game Nets.

This work is partly supported by the National Basic Research Program (973 Program) of China (Nos. 2010CB328105, 2009CB320505), National Natural Science Foundation of China (Nos. 60932003, 61070182, 60973144, 60973107, 61173008, 61070021). These projects aim to provide design principles for computer systems and computer networks from theoretical perspectives to improve the system performance and reduce the maintenance cost. Our group has been working on the performance evaluation and the optimization of the computer networks and computer systems for years. Many good papers have been published in respectable international conferences and transactions, such as INFOCOM, IEEE Journal on Selected Areas in Communication, IEEE Transactions on Information Theory, and IEEE Transactions on Signal Processing, etc. This paper summarizes these results and purposes some future research challenges.

多射频无线 Mesh 网络组播端到端时延建模与优化

王 维 杨 明 罗军舟 刘 波
(东南大学计算机科学与工程学院 南京 211189)

摘 要 针对 802.11 多射频无线 Mesh 网络(MR-WMN)不能有效支持端到端低时延组播的问题,首先围绕 MAC 层传输时延和 Mesh 层排队时延建模分析,并提出低时延组播路由模型 MR-MED(Multi-Radio Multicast End-to-End Delay).其次证明全局流干扰最小化问题是一个 NP 完全问题且全局流干扰和网络密度的数学关系符合 dPlN 分布,在此基础上提出有效减小 MAC 层流内和流间干扰的 DCA 算法.最后提出流量自适应的组播多径路由方案 MMRA,有效减小 Mesh 层排队时延.仿真与常见算法的比较表明,提出的模型准确刻画了 MR-WMN 的组播时延,且联合运用 DCA 和 MMRA 有效降低了端到端时延.

关键词 组播路由;时延优化;信道分配;多径路由;多射频无线 Mesh 网络
中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2012.01358

Modeling and Optimization of Multicast End-to-End Delay in Multi-Radio Wireless Mesh Networks

WANG Wei YANG Ming LUO Jun-Zhou LIU Bo
(*School of Computer Science and Engineering, Southeast University, Nanjing* 211189)

Abstract Multi-radio wireless mesh network (MR-WMN) is one of the key access techniques in Mobile Internet. However, current 802.11-based MR-WMN is not able to provide low multicast end-to-end delay. To address this problem, we first propose a layered and analytical model by combining overlapping channel assignment with multipath routing strategies. The proposed model can be used as a guide on multicast design. It decoupled multicast delay into transmission delay in the MAC layer and queuing delay in the Mesh layer based on that it derived a new multicast routing metric that had low end-to-end delay. Second, we prove that finding the minimum global flow interference solution is a NP-Complete problem and the relationship between global flow interference and network node densities is distributed in accord with double-Pareto lognormal (dPlN) distribution. Based on these two results, the DCA algorithm is proposed in order to minimum global flow interference which can efficiently reduce the multicast transmission delay in MAC layer. Last, to avoid the best wireless link being congested, we propose the flow adaptive-based MMRA algorithm by making use of the MR-MED routing metric and the multipath routing design philosophy, which took local channel congestion into account and can efficiently reduce the queuing delay in Mesh layer. Simulation result and comparison of the common algorithm MCM both show that the proposed model accurately characterizes the multicast delay in multi-radio

收稿日期:2011-11-22;最终修改稿收到日期:2012-05-28.本得到得到国家"九七三"重点基础研究发展规划项目基金(2010CB328104)、国家自然科学基金(60903162,60903161,61070161,61003257)、国家科技支撑计划课题(2010BAI88B03,2011BAK21B02)、高等学校博士点学科专项科研基金(20110092130002)、江苏省自然科学基金项目(BK2008030)、江苏省网络与信息安全重点实验室(BM2003201)和计算机网络与信息集成教育部重点实验室(东南大学)(93K-9)资助.**王 维**,男,1983 年生,博士研究生,主要研究方向为无线网络.E-mail: wary@seu.edu.cn.**杨 明**,男,1979 年生,博士,副教授,主要研究方向为网络安全和无线局域网.**罗军舟**,男,1960 年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为下一代网络体系结构、网络安全、无线局域网、网格与云计算.**刘 波**,女,1975 年生,博士,副教授,主要研究方向为服务调度和管理、普适计算和分布式网络管理.

wireless mesh network and the combination of the DCA and MMRA algorithms efficiently reduce the multicast end-to-end delay.

Keywords　multicast routing; delay optimization; channel assignment; multipath routing; multi-radio wireless mesh network

1　引　言

作为一种面向移动互联网环境的新一代无线多跳接入网[1]，无线 Mesh 网络(Wireless Mesh Network，WMN)正得到越来越广泛的应用. WMN 骨干网由自组织、自配置的静态 Mesh 路由器(Wireless Mesh Router，WMR)通过无线多跳互连形成[2]. 为了提高网络容量、减少信道冲突，WMR 配备多个射频接口并同时工作在不同交叠信道，这种 WMN 又叫做多射频无线 Mesh 网络(Multi-Radio WMN，MR-WMN)[3].

在 MR-WMN 中，组播是一种有效节省网络带宽的通信技术，它通过单源节点同时向一组目的节点传输信息. 被传输的信息最多在每条链路上传输一次，且仅在通向目的节点的分支处被复制. 使用组播通信技术可以显著提高 MR-WMN 容量，减少无线链路带宽消耗. 为了向终端用户提供在线游戏、视频会议、文件共享和流媒体等点对多点通信服务，同样要求 MR-WMN 支持组播. 与移动 Ad Hoc 网络侧重路径维护和节能的组播优化目标不同，MR-WMN 组播侧重为用户提供最佳接入性能[4]，例如最小端到端时延. 源节点和目的节点组之间的高时延通信会显著影响上层应用程序性能，特别是对实时性要求严格的应用程序，进而带来糟糕的用户体验. 然而在 MR-WMN 中确保低时延组播面临挑战，交叠信道间干扰及稀缺的无线带宽资源显著影响了组播端到端时延. 根据文献[5]总结的 802.11 组播传输特点，组播端到端时延分为 MAC 层传输时延和 Mesh 层排队时延两部分，因此不恰当的机制在减少其中一项时延的同时会增大另一项时延，从而不能有效优化组播端到端时延，导致 MR-WMN 无法提供令人满意的组播服务. 因此，如何同时减少组播数据传输时延和排队等待时延，使 MR-WMN 提供满意的端到端低时延组播服务，具有重要的研究意义.

本文在多射频环境下分析“流内干扰、流间干扰”和“本地拥塞”对 MR-WMN 组播端到端时延的影响，研究如何建立合理度量组播端到端时延的模型并根据模型分析以优化组播端到端时延. 论文主要研究成果包括以下 3 点：

(1) 提出分层模型 MR-MED 对组播端到端时延建模分析. 在综合权衡网络参数基础上提出组播路由判据 MR-MED，为后续组播路由提供选路依据.

(2) 证明全局流干扰最小的信道分配是一个 NP 完全问题，不存在多项式时间最优解算法；证明网络密度与信道干扰关系符合 double-Pareto Log-normal (dPlN)分布. 根据结论设计更加有效的密度感知信道分配算法，解决 MAC 层流干扰.

(3) 提出基于多路径路由算法的流量自适应多径组播路由策略，解决因最佳链路拥塞造成的时延开销. 结合实验验证了所提出的时延优化方案的有效性.

本文第 2 节简要介绍组播时延优化的相关工作，阐述这些工作的特点和存在的不足；第 3 节提出一个多射频 Mesh 网络组播端到端时延跨层模型 MR-MED，在此基础上对该模型准确性进行评价并基于该模型对组播端到端时延进行分析，提出 MR-WMN 组播端到端平均时延优化方案；根据 MR-MED 优化方案，第 4 节研究全局流干扰最小的信道分配问题. 首先给出定理并证明该问题是一个 NP 完全问题，其次研究网络密度与信道干扰的关系并给出相应的数学关系式，最后在上述证明基础上设计更加有效的密度感知的全局流干扰最小信道分配算法；根据 MR-MED 优化方案，第 5 节研究缓冲区排队引起的时延增大问题. 首先提出一个缓解本地拥塞、减少组播数据排队等待时延的多径组播路由算法，然后指出使用何种选路策略可以在多径间获得最佳端到端时延；第 6 节对提出的端到端低时延组播路由方案进行仿真实验，通过分析对比指出方案的有效性；第 7 节对全文进行总结并指出下一步工作的研究方向.

2　相关工作

目前优化无线网络端到端时延的方法主要分为

两类:一类针对流内和流间干扰问题,通过修改协议、网络编码以及信道分配优化传输时延;另一类针对数据包排队问题,通过最优队列调度策略、网关负载均衡优化数据包排队时延.

针对 MR-WMN 流内干扰和流间干扰问题,通常采用以下 3 种方法获得较低组播时延:一是修改旧有协议并提出新路由判据,例如 Kyoung 等人[6]提出低时延多射频组播路由协议 MR2_ODMRP,它是基于新的路由判据 MR2_ETT 和修改的 ODMRP 组播路由协议. MR2_ETT 根据射频当前速度计算信道传输时延,同时 MR2_ODMRP 在覆盖范围和传输速率两方面进行权衡. MR2_ODMRP 的主要问题是没有考虑端到端传输中存在的流间干扰,且容易产生较高的数据包排队时延. 二是基于网络编码技术使网络支持尽可能多的并发流,例如 Eryilmaz 等人[7]和 Yeow[8-9]等人分析了点对多点文件共享服务的组播时延,分别提出基于网络编码的组播时延优化技术. 他们指出,用网络编码技术代替传统的调度技术,可以显著降低组播单跳传输时延. 然而他们各自提出的网络编码方案没有考虑组播端到端传输时延,并且网络编码需要路由器在硬件上支持,而目前部署的无线路由器大多数不支持网络编码. 三是利用信道分配技术以减少射频干扰,例如 Cheng 等人[10]提出基于禁忌搜索(Tabu Search)优化方法的组播路由算法,联合考虑路由和信道分配. 但是该算法的缺陷是优化条件仅考虑链路的传输时延,没有考虑数据包的排队时延. Lim 等人[11]提出基于组成员邀请机制的分布式组播树构建算法,在此基础上利用自底向顶的信道分配方法以优化传输时延. 它的主要问题是基于节点加入/离开的组播树构建算法使用跳数作为路由判据,未考虑流间干扰,无法有效降低端到端组播时延.

针对组播数据在缓冲区排队问题,Key 等人[12]提出 FMRC 多径路由,通过对整个 Mesh 网络建立排队网络模型,为流量选择高性能多径链路. 它的主要问题是不支持组播. Nagesh 等人[13]提出 MDOS 路由,将流量进行分割,分发至不同链路进行传输. MDOS 的主要问题是在链路没有发生拥塞时,流量没有全部流经低端到端时延的路径,因此端到端时延较大. Popa 等人[14]提出基于地理位置的路由 IPS-BGR,在负载均衡的基础上实现数据包准确送达目的节点. 它的主要问题是需要节点具有探测自身所在地理位置的装置(例如 GPS),而目前大多数 MR-WMN 网络节点不具备该硬件. He 等人[15]提出基于 LBA 的分布式负载均衡策略,缓解 WMN 终端和网关之间的拥塞. 它的主要问题是只考虑网关的负载均衡,不能有效缓解节点发生的本地拥塞. Li 等人[16]提出流量感知的组播路由判据 FLMM,考虑流内干扰以及信道多样性. 它的主要问题是没有考虑流间干扰对组播端到端时延的影响. Zeng 等人[17]提出一种集中式算法 Multi-Channel Multicast (MCM),首先最小化组播源节点和目的节点之间转发节点数及跳数,然后分配交叠信道以减小流间干扰. 但是 MCM 存在隐藏信道及干扰因子不易确定等问题,不能有效优化组播性能.

通过对以上相关工作分析可以看出,在组播时延优化方面,现有的研究工作没有充分结合跨层信息获得满意的组播端到端时延;在设计拓扑控制的信道分配算法时,大多考虑正交信道,对交叠信道的利用率不高;在设计组播路由算法时或者只考虑网关负载均衡,不考虑转发路径拥塞,或者只考虑建立最佳路径,不考虑因此造成的本地拥塞.

3 组播端到端时延跨层模型

通过分析基于 802.11 MAC 协议的多射频无线 Mesh 网络组播链路层传输机制,对 MR-WMN 组播端到端时延建立跨层模型 MR-MED,并对 MR-MED 的准确性进行仿真评价;在此基础上指出如何根据该模型对网络进行优化,减少 MR-WMN 组播端到端平均时延.

3.1 组播时延模型

基于 802.11 的多射频无线 Mesh 网络,是当前主流的无线 Mesh 网络,已经应用于无线北京(奥运)、无线伦敦和 MIT Roofnet 等众多项目中. MR-WMN 组播端到端时延定义为源节点数据包到每个目的节点时延的平均值. 根据 IEEE802.11 MAC 协议,组播帧不会被分段,无需得到 ACK 肯定确认,在传输出现差错或数据丢失时也不会被重传. 整个原子过程只牵涉到一个数据帧,根据基于竞争的访问控制规则加以传输,竞争窗口为[0,31],计数单位为时隙. 一旦发现信道不可用,节点必须重新竞争信道. 传输结束后,所有节点必须等待一段 DIFS,然后在竞争窗口倒数随机产生的时延时间. 组播数据帧交换过程如图 1 所示.

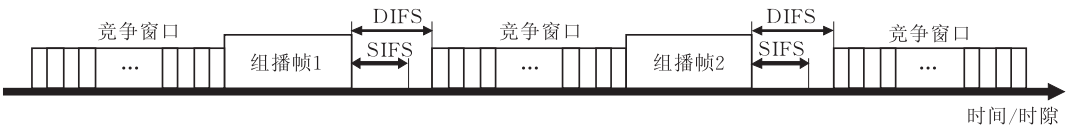


图 1　802.11MAC 组播数据帧交换过程

分析组播数据传递过程可知，多射频组播端到端时延(Multi-Radio Multicast End-to-End Delay，MR-MED)分为传输时延和排队时延. MR-MED 等于链路上所有组播单跳时延 MHD(Multicast Hop Delay)的总和，因此组播无线链路时延又可分为链路传输时延(TD)和组播数据包排队时间(QD)两部分. 进一步地，链路传输时延又可分解为数据发送时间(SD)和竞争信道的时延(BD)两部分. 从图 1 可知 802.11 MAC 组播没有 ACK 肯定确认机制，也没有回退机制，因此用如下公式表示：

$$MR\text{-}MED(u)=\sum_{i=1}^{d(u)}MHD_i(u),\ \forall u\in R \tag{1}$$

式(1)中，R 为组播目的节点集；$d(u)$代表源节点到节点 u 的路径长度，计数单位为跳数. 为求解式(1)，其余分量由下式给出：

$$\begin{aligned}MHD_i(u)&=QD_i(u)+TD_i(u)\\&=\sum_{j=1}^{Q_i(u)+1}[TD_i(u)]_j\\&=\sum_{j=1}^{Q_i(u)+1}[SD_i(u)+BD_i(u)]_j\end{aligned} \tag{2}$$

$$SD_i(u)=\frac{P_k}{\sum_{m=1}^{N(u,i)}B_m(u,i)\times p_m(u,i)} \tag{3}$$

$$BD_i(u)=\sum_{j=1}^{\infty}\sum_{m=1}^{N(u,i)}[1-p_m(u,i)]^{j-1}p_m(u,i)^j\sum_{t=1}^{j}E[S_t] \tag{4}$$

其中，$Q_i(u)$为源节点到节点 u 路径上第 i 跳节点的当前队列长度；用 j 标识节点当前队列中第 j 个数据包；$p_m(u,i)$ 代表源节点到节点 u 的路径上第 i 跳节点的第 m 个射频所在信道的成功传输率，注意当 j 超过节点最大队列长度时，这些数据包将被丢弃，即此时这些数据包的成功传输率 $p_m=0$，式(4)仍然成立；$B_m(u,i)$代表节点 u 第 m 个射频在第 i 跳的链路带宽；$N(u,i)$代表源节点到节点 u 的路径上第 i 跳节点的发送射频数量；S_t 为第 t 次争用信道时的竞争窗口所占的时隙大小. 由 $S_t\in[0,31]$，$t\in N$ 可以计算，$E[S_t]=15.5$ 时隙.

为求解式(1)，流间干扰和流内干扰对传输时延的影响用链路带宽损耗度量. 这里使用 Capone 等人的物理干扰模型[18]和物理层信噪比(SNR)计算流间干扰链路层传输等效带宽为

$$B_m^O(u,i)=\left(1-\frac{\sum_{k\in i'}P_i(k)}{P_{\max}}\right)\times B_{\text{base}} \tag{5}$$

其中，$P_i(k)$代表节点 i 收到来自节点 k 发射的功率；i'代表 i 的干扰域内节点；$P_{\max}$为最大允许的干扰功率；B_{base}为网络理想带宽. 在计算流内干扰时，一条路径上的链路并非同时可以使用. 例如干扰域为 R_I，则在任意时刻，以路径上节点 u 为参照，u 的前 R_I 跳链路和后 R_I 跳链路最多仅允许一条非正交信道传输数据，也即该路径在同一时刻仅允许传输一个包. 针对该问题，设以节点 u 为参照的一段信道组成的路径的等效端到端带宽为 $B^I(u)$，则式(6)成立

$$\begin{aligned}MHD(u)=\frac{P_k}{B^I(u)}&=\frac{P_k}{B_{A-R_I}(u)}+\cdots+\\&\frac{P_k}{B_A(u)}+\cdots+\frac{P_k}{B_{A+R_I}(u)}\end{aligned} \tag{6}$$

经过数学变换得到

$$B^I(u)=\frac{1}{\sum_{t=A-R_I}^{A+R_I}\frac{1}{B_t(u)}} \tag{7}$$

又因为一条信道的等效带宽取决于该信道流内等效带宽和该信道流间等效带宽两者中较小者，因此

$$B_m(u,i)=\min[B_m^O(u,i),B^I(u)] \tag{8}$$

至此式(1)得到解答. 下面对该模型的有效性进行验证.

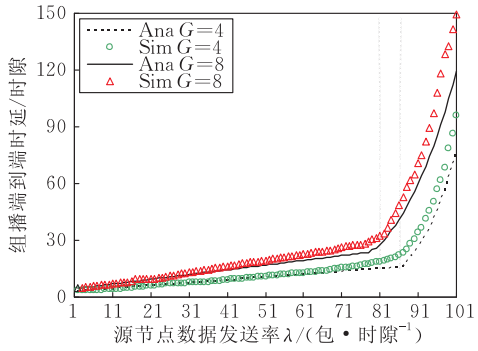
3.2　模型评价

在使用 MR-MED 跨层模型之前，需要验证该模型的准确性，这通过对比模型的分析结果和实验仿真数据来确定. 仿真使用 NS-3(版本 NS-3.10). 根据 IEEE802.11 MAC 对组播的要求，使用基于 CSMA/CA 的无 RTS/CTS、无 ACK 肯定确认机制的分布式点协调控制策略(DCF). 信道速率采用 802.11 规定的基本速率集速率，即 1.0 Mbps. 组播源节点发送数据包的时间间隔符合指数分布，速率范围为 0.01(数据包/时隙)～1.00(数据包/时隙)递增. 数据头部 32 Bytes，数据大小 512 Bytes，末尾校验位 4 Bytes. 因此一个时隙的长度为 4.384 ms.

仿真使用 MIT Roofnet 网络的真实拓扑作为 Mesh 骨干，包括 24 个关键节点，如图 2(a)所示. 根据 802.11 规定，总共有$|C|=11$ 条交叠信道可用，$C=\{1,2,\cdots,11\}$. 每个 WMR v 配备 $N(v)=2$ 个全向射频接口，所有射频接口统一使用 Wi-Fi 默认最大功率 100 mW. 假定驻波比 1∶1，则该功率对应传输半径 $R_T=250$ m，干扰半径 $R_I=450$ m. WMR 队列长度为 50. 仿真在组播组规模不同的两个场景下进行，组播组规模分别是 $G=4$ 和 $G=8$. 每个场景各进行 100 次仿真，并对结果计算平均值.

图 2(b)分别比较了采用模型分析和仿真得到的组播端到端平均时延. 观察到模型分析结果与仿真结果非常接近，这验证了所提出的跨层模型在刻画组播端到端平均时延方面的准确性. 此外，注意到当源节点发送数据包速率超过 0.80 包/时隙($G=8$)、0.85 包/时隙($G=4$)时，组播端到端时延曲线呈现指数增长趋势，仿真曲线逐渐偏离分析曲线. 这主要由于此时网络负载已经非常重，开始出现缓冲区排队. 此时若要改善组播端到端平均时延，需要在路由上做出改进，使路由能够缓解本地拥塞.

(a) 拓扑环境



(b) 模型评价结果

图 2 仿真拓扑环境和模型评价结果

根据跨层模型有效性可知，*MR-MED* 值可以作为基于先验式表驱动的组播树生成算法的路由判据，减小组播端到端平均时延. 此外分析跨层模型可知，当源节点数据发送速率、组播组规模、节点配备射频接口数及网络可用信道数确定时，数据包的传输速度和排队速度之间存在权衡. 高速链路具有最小链路传输时延，但可能因耗尽带宽而导致本地拥塞，增大排队时延. 根据以上分析，本文提出组播端到端平均时延优化方案：通过设计信道分配算法以减小全局流干扰；通过设计多径路由策略以减少数据排队时间，最终优化端到端平均时延，该优化方案因此可称为 MR-MED，它所包含的上述两部分优化内容分别在第 4 节和第 5 节进行讨论.

4 全局流干扰最小的密度感知信道分配

根据 MR-MED 模型知，优化组播端到端时延需要最小化全局流干扰. 首先对全局流干扰最小信道分配问题进行理论分析，通过顶点着色证明其属于 NP 完全问题. 其次研究节点密度与信道干扰之间的关系，证明全局流干扰与网络密度的关系满足双帕累托对数正态(double-Pareto Lognormal, dPlN)分布. 最后给出密度感知的信道分配算法，使全局流干扰最小.

4.1 理论分析

全局流干扰分为流内干扰和流间干扰，其本质是交叠信道的传输干扰. 差值≥5 的相邻交叠信道之间不会相互干扰，因此与相邻信道差值≥5 的信道将具有最大信道容量(可达到其理论上限)；反之，与相邻信道差值越小，信道容量越小. 当两条相邻信道差值等于 0 时，两者的各自信道容量为 0，此时它们严重相互干扰，无法传输数据. 将 MR-WMN 抽象为无向、全连通图 $G(V,E)$，其中顶点集合 V 代表所有 WMR，无向边集合 E 代表所有在物理上有直接连通关系的一跳传输. IEEE802.11 MAC 协议将无线频率划分为多个交叠信道，令 $C=\{1,2,\cdots\}$ 代表网络中所有可用信道构成的集合，总数为 $|C|$. 根据调度策略，每一个 WMR $v\in V$ 配备 $N(v)$ 个工作在特定无线信道 $c\in C$ 的射频接口. WMR v 的射频接口 I 表示为 $v(I)$，$v(I)\in N(v)$，工作在信道 c 的射频接口 $v(I)$ 表示为 $c[v(I)]$. 假定所有射频接口都是全向的，并且具有同样的传输距离 R_T 和干扰距离 R_I，这里 $R_I=2R_T$. 令 $d[u(I),v(J)]$ 代表 $u(I)$ 和 $v(J)$ 之间的距离，从而这两个射频可以相互通信，当且仅当满足 $d[u(I),v(J)]\leq R_T\wedge c_1[u(I)]\cap$

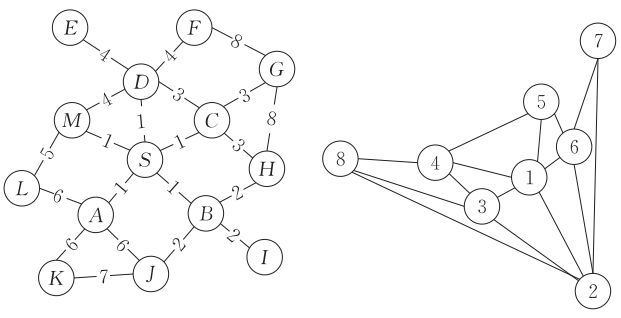
$c_2[v(J)]\neq\varnothing$. 进一步假定射频接口 $u(I)$ 和 $v(J)$ 具有相同的、常数的传输成功率 $p_s[u(I),v(J)]$. 如果满足条件 $R_T\leq d[u(I),v(J)]\leq R_I \wedge c_1[u(I)]\cap c_2[v(J)]\neq\varnothing$，$u(I)$ 和 $v(J)$ 不仅不能相互通信，并且在各自传输过程中还会干扰对方. 使用不同正交信道的射频之间不会相互干扰，即使 $d[u(I),v(J)]\leq R_I$. 节点 u 的干扰域内的所有节点构成节点集合 $I(u)$. 为了度量全局流干扰，给出了如下定义.

定义 1.　组播流内干扰强度：组播路径上一条分支路径内相互干扰信道之间的差值之和. 组播流间干扰强度：组播路径上各条分支路径之间相互干扰信道的差值之和. 组播全局流干扰强度：组播流内干扰强度和流间干扰强度的总和.

因此，寻找组播全局流干扰强度最小的信道分配，就是寻找一种信道分配策略，满足组播源节点和所有目的节点间存在连通，并且各干扰域内信道之间差值最大. 这个问题由于计算复杂，相关方案给出的都是近似解[3,19]. 首先给出定理 1，证明该问题是一个 NP 完全问题.

定理 1.　组播全局流干扰最小的交叠信道分配问题是一个 NP 完全问题.

证明.　用干扰图重新定义该问题. 即有干扰图 $G'(V',E')$，对于 $G(V,E)$ 中 $\forall e\in E$，$\exists v'\in V'$. 给定 $\forall u',v'\in V'$，它们在 $G(V,E)$ 中代表的边有相互干扰关系，当且仅当它们之间 $\exists e'\in E'$. 因此如图 3(a) 所示具有 14 个节点的网络可以转换为图 3(b)所示的干扰图. 在转换时需要注意，由于通信采用组播方式，网络中各链路并非平等关系，因此从靠近源节点 S 的链路开始编号；并且由于无线采用广播方式，一次广播均可达的链路被赋予相同编号. 在图 3(b) 中，问题等价于：将 $C=\{1,2,\cdots,11\}$ 分配给各顶点，使得所有位于线段两端的顶点对的信道差值(的绝对值)之和最大，即



(a) 原始网络拓扑图　　(b) 转换后的干扰图

图 3　网络拓扑图转换为信道干扰图

$$\text{maximize}\sum_{\substack{\forall(u',v')\in E'\\ c_1(u'),c_2(v')\in C}}|c_1(u')-c_2(v')| \tag{9}$$

其中，$c_1(u')$ 代表 u' 分配的信道. 这可以视为一个顶点着色问题. 又由于 $|c_1(u')-c_2(v')|\geq 5$ 的各正交信道的顶点着色问题是一个 NP 完全问题，因此网络使用交叠信道时的顶点着色问题也是一个 NP 完全问题.　　证毕.

4.3 节给出使用贪婪方法的近似算法求解全局流干扰最小信道分配问题. 在给出具体算法以前，4.2 节首先讨论网络密度对信道分配算法的潜在影响，该影响决定了信道分配算法的有效性.

4.2　网络密度与信道干扰的定量分析

传统信道分配算法大多忽略网络密度. 虽然有些算法承认网络的稀疏、稠密程度与信道干扰有关，在信道分配算法时却不予考虑，部分原因是这一关系很难准确度量. 本节通过研究网络密度和信道干扰之间的关系，给出数学关系式，用于指导信道分配算法的设计和优化.

网络的稀疏、稠密表现在节点和连接各节点的边的密度上，最终表现为链路的密度. 直观认为：链路密度越大，干扰边数越多，越难以进行信道分配. 可是由于无线网络的广播特性，网络越稠密，节点一次传输覆盖的相邻节点也越多，传输次数可以更少，信道分配更容易进行. 本小节通过分析给出网络密度与信道干扰的量化表达式，用于指导信道分配算法的设计.

定义 2.　节点 A 的干扰边数定义为以节点 A 为圆心的一定区域内所有与其它链路有干扰关系的链路数.

定理 2.　各节点与链路位置在网络中任意时，对于节点 A 的一定范围内所有干扰边数的平均增长率为固定常数.

证明.　以 A 为圆心，干扰半径为 k，长度 r 为半径作圆形区域. 假定网络边的平均密度为 ρ，则圆边界处的边数为 $2\pi r\cdot\Delta r\cdot\rho$. 又干扰半径为 k，即相距 k 之外的边不会引起干扰，因此随着半径扩大，圆形区域内部靠近圆心的边不再与圆形外部区域的边存在干扰关系，故干扰边数可由 r，k 表示为

$$2\pi\rho\cdot[r\cdot\Delta r-(r-k)\cdot\Delta(r-k)] \tag{10}$$

因此干扰边数的平均增长率为

$$\eta=\lim_{\Delta r\to 0}\frac{2\pi\rho\cdot[r\cdot\Delta r-(r-k)\cdot\Delta(r-k)]}{2\pi\rho\cdot\Delta r}=k \tag{11}$$

所以当干扰半径为定值 k 时，A 的一定范围内干扰

边数的平均增长率为固定常数. 证毕.

定理 3. 干扰边数以 η 倍增长过程中,存在以 η 为中心的波动 $W_r \sim N(0, c^2 r)$.

证明. 由于节点和连接节点的各边在网络中非均匀分布,随着半径增长,一些新干扰边加入,一些边不再是干扰边,因此数量上呈现无限小波动,W_r 是独立增量且关于 r 连续,又在有限 Δr 范围内围绕 $\eta\Delta r$ 波动并趋向于 $\eta\Delta r$,因此 W_r 服从正态分布 $N(0, c^2 r)$,c 为任意常数. 证毕.

定理 4. 干扰边数在初始时刻非零.

证明. 对于组播通信,这是显然的. 因为在非平凡组播网络中,至少有一个组播源节点和两个组播目的节点,且源节点到两个组播目的节点分别各有一条边相连. 这两条边在源节点的干扰域内,因此干扰边数初始值非零. 证毕.

定理 5. 干扰边数的增长符合 double-Pareto Lognormal(dPlN)分布.

证明. 由定理 2～4 知,干扰边数的增长过程服从几何布朗运动(GBM),即

$$dX_r = (\eta dr + \sigma dW_r) \cdot X_r \tag{12}$$

其中:X_r 代表干扰边数;σ 代表扰动量. 再根据 Itô 公式可得

$$X_r = X_0 \cdot e^{(k-\sigma^2/2)r+\sigma W_r} \tag{13}$$

因此

$$\ln X_r = \ln X_0 + (\mu - \sigma^2/2)r + \sigma W_r \tag{14}$$

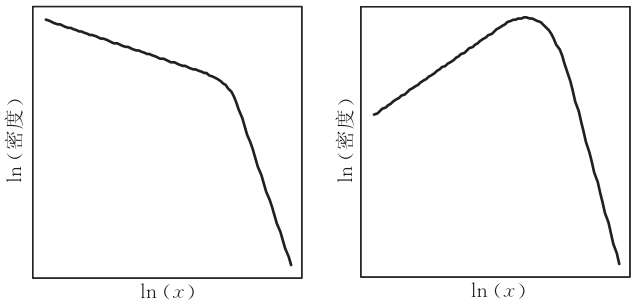
$$X \sim dPlN(\alpha, \beta, 0, c^2) \tag{15}$$

其中 α 和—$\beta(a, \beta > 0)$分别是 z 变换后方程(16)的两个根.

$$\frac{\sigma^2}{2}z^2 + \left(\mu - \frac{\sigma^2}{2}\right)z - \lambda = 0 \tag{16}$$

所以干扰边数的增长服从 dPlN 分布[20]. 证毕.

利用所求解析式对干扰边数 X 分析绘制图 4 特征曲线. 从绘制曲线中可以看出,β 值大于或者小于 1,密度曲线变化不同. 从图 4 可以看出,干扰边数与链路密度并非成固定的正比例关系,即对于 $\forall x_1, x_2, f(x_1) > f(x_2)$ 并非恒成立. 具体地说,图 4(a)代表稀疏网络的密度特点:大多数边具有大致相同的干扰边数,因此优化干扰时宜整体考虑;相反,图 4(b)代表稠密网络的密度特点:拥有非常少干扰边数或者非常多干扰边数的边占绝大部分,因此重点优化这些边以减少干扰. 为减少流内、流间干扰对组播传输时延的影响,需要根据网络稀疏、稠密特点有效选择信道分配方式,减少无线链路并行传输干扰,4.3 节对此进行讨论.



(a) $\beta<1$ (b) $\beta>1$

图 4 dPlN 特征曲线

4.3 密度感知信道分配算法

网络的稀疏、稠密由 dPlN 中 β 值确定,可在信道分配前事先由网络信息获得. 该假定是合理可行的,因为 MR-WMN 的拓扑具有最小变化性且通常在一定时期内(例如数日甚至数个星期)较稳定. 因此网络各节点可提前通过周期广播 Probe Message 与邻节点交互信息,统计链路数并计算 β 值. 该值在相当长一段时间内将保持不变,可供信道分配参考. 信道分配算法开始时,首先以组播源节点为中心,使用广度优先遍历(BFS)对网络全部节点划分层次.

定义 3. 如果位于第 i 层的节点 A 和位于第 $i+1$ 层的节点 B 满足 $0 \le d[A,B] \le R_T \wedge c_1[A] \cap c_2[B] \neq \varnothing$,则 A 是 B 的父节点,同理 B 是 A 的子节点.

定义 4. 优化函数定义如式(17)所示. 其中 RR 代表用于接收数据的射频接口,SR 代表用于发送数据的射频接口.

$$F(u,c) = \prod_{\substack{\forall v \in I(u) \\ c_i, c_j \in C}} \{|c - c_i[v(RR)]| + |c - c_j[v(SR)]|\} \tag{17}$$

式(17)的意义在于通过贪心法对节点 u 的射频分配信道 c,使 c 与干扰节点集 $I(u)$ 中所有已分配信道的差值最大,从而最小化干扰,最大化信道容量. 算法按照层次结构自顶向下分配信道,在分配过程中始终以最大化优化函数 $F(u,c)$ 为目标. 如果存在多于一个 $F(u,c)$ 具有相同取值,算法随机选择一个. 当 u 分配信道 c 后,所有 u 的子节点统一调整其 RR 到 c 信道以便与 u 通信. 密度感知信道分配算法(Density-aware Channel Assignment, DCA)如算法 1 所示.

算法 1. 密度感知信道分配算法 DCA.

```
1. procedure DCA(C, G, β)
                    //C: channel set, G: network graph
2.     s's SR uses channel 1, its children use channel 1
```

```
        for their RRs
3.    s uses BFS to divide WMRs into different layers
4.    for l←2, layerNo do
5.      if β>1 then A←Sorting all node u at l by their
              degrees
6.      else A←all node u at l without sorting
7.      end if
8.      for each a in A do
9.        select channel i∈C to minimize F(a,v)
10.       a adjusts SR to i, its children adjust RRs to i
11.     end for
12.   end for
13. end procedure
```

DCA 算法拥有两个主要特点：易于实现和全局流干扰近似最小化. 此外，通过在相邻 WMR 之间使用 Probe Message，可以较方便地构建算法的分布式形式.

5　流量自适应多径组播路由策略

组播端到端时延受数据包排队影响，因此不合理的组播传输路径会增大端到端平均时延. 在全局流干扰优化基础上，根据 MR-MED 模型可知，尽管路由算法可以使用 MR-MED 作为路由判据以优化组播端到端时延，但由于无线链路带宽有限，密集的数据包将导致网络拥塞，增大排队时延. 该假定是合理的，因为所有涉及链路质量为判据的路由算法均会使尽可能多的数据通过质量最高的链路，从而导致该链路发生本地拥塞. 针对该问题，提出流量自适应的多径组播路由策略，使用多径组播路由算法(MMRA)建立多径传输路径，对密集网络流量进行分流，从而有效缓解本地拥塞. MMRA 见算法 2.

算法 2.　多径组播路由算法 MMRA.

```
1.  procedure MMRA(s,R)
                        //s: source, R: all destinations
2.    s uses BFS to compute MR-MED(v) towards each
        WMR v and divide them into different layers
3.    V(T)←R∪{s}, E(T)←∅
4.    for ∀WMR v∈R do
5.      p←v
6.      P←all p's parents //P!=∅
7.      while ∀p∈P∧p∉V(T) do
8.        select u∈P that MR-MED(u)+MHD_l(p) is
              minimum
9.        select w∈(P−u} that MR-MED(w)+MHD_l(p)
              is minimum
10.       V(T)←V(T)∪{u}
11.       E(T)←E(T)∪{(u,p)}
12.       if w!=∅ then
13.         w.tag←"backup"
14.         V(T)←V(T)∪{w}
15.         E(T)←E(T)∪{(w,p)}
16.       end if
17.       p←u
18.     end while
19.   E(T)←E(T)∪{(u,p)}
20.   end for
21.   return T
22. end procedure
```

流量自适应体现在策略根据网络流量计算 *MR-MED* 值并动态选择最佳传输路径，在传输时延和排队时延之间寻找平衡，也就是在自适应策略下，数据并非全部流经质量最高的链路，而是流经较差链路以换取较小排队时延. 当网络流量较大时，自适应策略利用分流路径进行分流，尽可能缓解本地拥塞；当网络流量较小时，多径仅做备份，数据全部流经链路质量最高的路径. 策略根据动态计算的 *MR-MED* 值决定何时将分流数据路由至分流路径，一旦原路径拥塞缓解后再将分流数据路由回原路径. 考虑到在某些特定网络场景下，可能存在因分流而使得数据在原路径和分流路径之间频繁切换，自适应策略使用式(18)计算 $E[MR\text{-}MED]$，当原路径 $E[MR\text{-}MED]$ 值大于分流路径 *MR-MED* 值时再进行切换.

$$E[MR\text{-}MED]=\frac{1}{CLength}\sum_{t=now-CLength}^{now} MR\text{-}MED_t,\quad now\geq CLength \tag{18}$$

其中，$MR\text{-}MED_{now}$ 表示最新一次 *MR-MED* 值. 式中 *CLength* 称为防抖动参数，在默认情况下其取值为足够大的正整数. 特别地，该值反映了选路和切换之间的权衡：*CLength* 取值为 1 则数据切换最频繁，此时每个数据包都经过最佳路径，但切换开销较大；取值越大则切换越不频繁，但数据的实时响应越差. 本文下一步工作将研究选路和切换之间的权衡问题.

6　仿真分析

本节通过仿真实验，将提出的组播端到端时延优化方案 MR-MED 与 MCM 在不同场景、不同网络密度下进行对比分析. 首先给出仿真实验参数及环境设置，然后进行实验并对结果进行分析比较. 实验结果证明了 MR-MED 在优化组播端到端时延方面的有效性.

6.1 实验参数及环境设置

仿真实验基于 NS-3 网络模拟器，使用带分布式点协调功能(DCF)的 IEEE802.11 作为 MAC 层协议.根据 802.11 标准对组播传输的要求，不使用 RTS/CTS 和 ACK 机制.各节点物理位置以及源节点均随机生成在 1000 m×1000 m 大小的网络区域内，但要求网络必须连通，否则重新生成网络拓扑.各射频传输半径为 250 m，干扰半径为 450 m.组播源节点流量模型使用固定比特速率(CBR)，数据大小为 512 Bytes，数据发送时间为 500 s.针对每个场景各进行 100 次仿真并求期望值作为最终实验结果.用 MR-MED-S 和 MCM-S 表示两种方案在稀疏网络的性能表现，用 MR-MED-D 和 MCM-D 表示两种方案在稠密网络的性能表现.为清晰表述内容，将实验参数的记法和取值说明列于表 1，将实验场景和说明列于表 2.

表 1 实验参数和取值说明

实验参数	取值说明
网络覆盖区域	1000 m×1000 m
节点数量	稀疏(β<1):30；稠密(β>1):60
节点队列及长度	FIFO，2.0 MBytes
信道类型及数量	2.4～2.485 GHz，非正交信道:11
链路质量	0.9 (constant)
链路基本速率	1.0 Mbps
射频接口类型	Omni-Simplex-Two Ray
射频接口数量	4 (2×2 MIMO)
发射功率	100 mW (SWR=1∶1)
数据包大小	512 Bytes

表 2 实验场景和说明

场景	说明
A	稀疏和稠密网络.组播组规模:5～30 可变，增幅为 1；可用交叠信道数:11；组播源节点速率:120 pkts/s
B	稀疏和稠密网络.可用交叠信道数:1～11 可变，增幅为 1；组播组规模:5；组播源节点速率:120 pkts/s
C	稀疏和稠密网络.组播速率:50～150 pkts/s 可变，增幅为 2；组播组规模:5；交叠信道数:11

实验仿真单源节点向一组目的节点组播的平均吞吐量和端到端平均时延，组播性能评估参数定义如下：

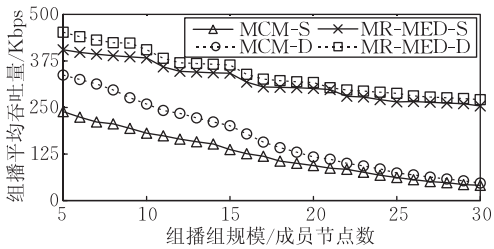
(1) 组播平均吞吐量(Multicast Average Throughput，MAT)：组播目的节点吞吐量定义为它收到的组播字节数与它收到最后 1 个字节和第 1 个字节所花费时差的比值.组播平均吞吐量定义为所有组播目的节点吞吐量的平均值.该性能评估参数表明网络能够承载的组播最大速率.

(2) 组播端到端平均时延(Multicast End-to-End Average Delay，MEAD)：组播端到端时延定义为组播目的节点收到源节点发送的数据包所花费时间.组播端到端平均时延定义为所有组播目的节点收到源节点发送的数据包所花费的平均时间.该性能评估参数表明网络传递组播数据包所耗费的时间.

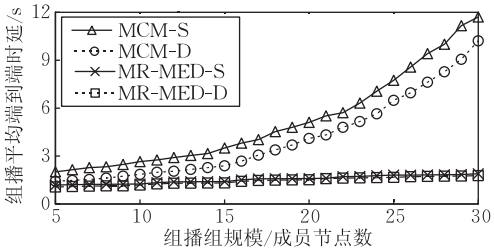
6.2 性能分析

6.2.1 组播组规模对性能的影响

场景 A 用于分析比较 MR-MED 方案和 MCM 方案中组播组规模对组播吞吐量和端到端时延的影响，其中组播组规模从 5 增至 30，每次增幅为 1.实验结果如图 5 所示.综合图 5(a)和 5(b)看出：MR-MED 方案组播性能优于 MCM 方案.具体地说，从图 5(a)可以看出，实验过程中，MR-MED 方案的组播吞吐量高于 MCM 方案；并且随着组播组规模增长，两种方案在稀疏网络和稠密网络中吞吐量均逐渐减小，但 MR-MED 方案的组播吞吐量下降幅度小于 MCM；此外 MR-MED 方案在稀疏网络和稠密网络的吞吐量差异小于 MCM，表明方案对网络密度具有较好适应性.从图 5(b)可以看出，实验过程中，MR-MED 方案的组播端到端时延低于 MCM 方案；并且随着组播组规模的增长，MCM 方案的时延显著增加，特别是 G≥13 后，而 MR-MED 方案时延增加较不明显.分析原因可知，一方面 MR-MED 方案的 DCA 算法有效减少了流内、流间干扰，另一方面当组播组规模较大使网络流量较多时，MR-MED 方案的 MMRA 算法有效缓解了本地拥塞，减小了数据包排队时延.



(a) 组播组规模对吞吐量的影响

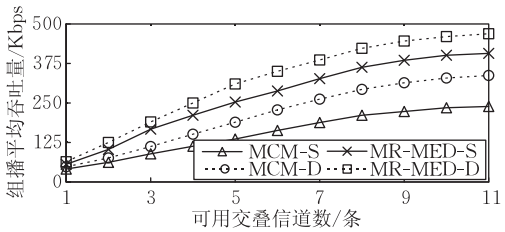


(b)组播组规模对时延的影响

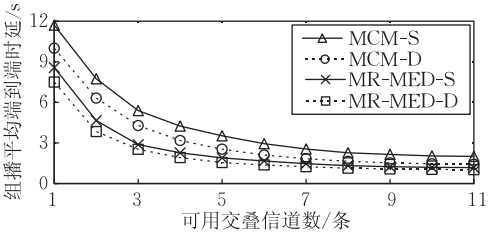
图 5 场景 A：组播源节点速率可变

6.2.2　交叠信道数对性能的影响

场景 B 用于分析比较 MR-MED 方案和 MCM 方案中交叠信道数对组播吞吐量和端到端时延的影响，其中交叠信道数从 1 增至 11，每次增幅为 1. 实验结果如图 6 所示. 综合图 6(a)和图 6(b)看出：MR-MED 方案组播性能优于 MCM 方案. 具体地说，从图 6(a)可以看出，随着可用信道数增多，两种方案在稀疏网络和稠密网络中吞吐量均显著增大，表明多信道对网络性能改善明显. 但 MR-MED 方案吞吐量增幅大于 MCM 方案. 这是因为 MR-MED 方案使用的 DCA 算法能有效分配信道，减少流内、流间干扰对传输路径的影响. 此外注意到在稀疏网络和稠密网络中，MR-MED 方案的组播吞吐量差异小于 MCM 方案，这是因为 DCA 算法在信道分配时考虑了网络密度这一重要因素. 从图 6(b)可以看出，在没有多余信道可供分配的情况下，MR-MED 方案在稀疏网络和稠密网络中时延均小于 MCM 方案，这是因为 MR-MED 方案使用的 MMRA 算法能够有效减小数据包排队时延. 当可用信道数逐渐增加时，两种方案的组播端到端时延均逐渐减小并逐渐趋于一致，但 MR-MED 方案的时延仍然略小于 MCM 方案，这是因为网络中可用信道数“充足”时，各种信道分配策略都可以实现零干扰信道分配，差异趋于消失.



(a) 交叠信道数对组播吞吐量的影响



(b) 交叠信道数对组播时延的影响

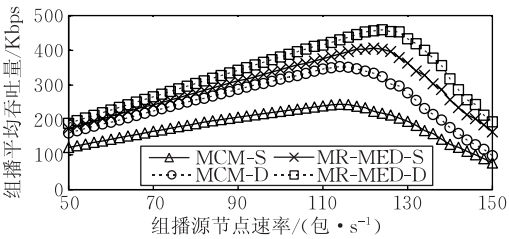
图 6　场景 B：组播源节点速率可变

6.2.3　组播速率对性能的影响

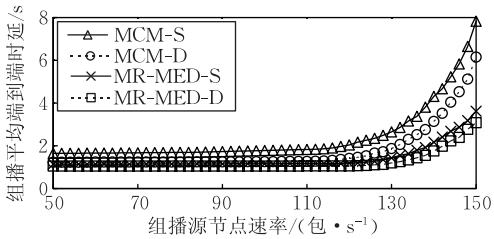
场景 C 用于分析比较 MR-MED 方案和 MCM 方案中组播源节点速率对组播吞吐量和端到端时延的影响，其中源节点速率从 50 packets/s 增至 150 packets/s，每次增幅 2 packets/s. 实验结果如图 7 所示. 综合图 7(a)和 7(b)看出：MR-MED 方案组播性能优于 MCM 方案. 具体地说，从图 7(a)可以看出：随着源节点速率增大，两种方案在稀疏网络和稠密网络中吞吐量先逐渐增大再逐渐减小. 这是因为当网络轻载时，随着组播源节点速率增大，组播平均吞吐量逐渐增大；当网络重载时，随着组播源节点速率增大，网络带宽逐渐耗尽，各链路发生本地拥塞越来越频繁，导致队列丢包，组播平均吞吐量逐渐降低. 但对比图中曲线可以发现，MR-MED 方案吞吐量大于 MCM 方案，这是因为 MR-MED 方案中 MMRA 算法能够有效缓解本地拥塞、减小数据包排队时延. 从图 7(b)可以看出，网络轻载时，两种方案在稀疏网络和稠密网络中时延增加均不明显；当网络负载重时，MR-MED 方案的曲线增幅小于 MCM 方案，表明 MR-MED 组播端到端时延优于 MCM 方案. 这一方面是因为 DCA 算法有效减小流内、流间干扰，另一方面是因为 MMRA 算法有效缓解了本地拥塞. 此外还可以看出，MR-MED 时延对于稀疏和稠密网络差异小于 MCM，表明 MR-MED 方案对网络密度具有较好适应性.

7　结束语

组播端到端平均时延是评价多射频无线 Mesh 网络点到多点通信性能的关键指标，该指标受到 MAC 层传输和 Mesh 层排队的双重影响. 本文提出的跨层模型 MR-MED 较好地刻画了组播端到端时



(a) 源节点速率对组播吞吐量的影响



(b) 源节点速率对组播时延的影响

图 7　场景 C：组播源节点速率可变

延,同时该模型的 *MR-MED* 值可以作为组播路由判据,优化选路.对 MR-MED 模型分析可知:优化组播端到端时延需要在链路传输速率和排队时延之间权衡,本文据此提出优化方案,解决链路传输速率受到流内干扰和流间干扰影响的问题及最佳路径发生本地拥塞造成的排队问题.由于优化方案要求的全局流干扰最小的信道分配是 NP 完全问题,论文先求出干扰与网络密度的关系,最后提出密度感知的信道分配算法,使优化方案在稀疏网络和稠密网络均可有效优化全局流干扰.优化方案使用的流量自适应多径组播路由策略对本地拥塞具有较好缓解作用,可以在最优链路传输时间和缩短数据包排队时间之间寻找平衡点.实验表明,本文提出的 MR-MED 优化方案在稀疏和稠密网络中均可有效降低组播端到端平均时延.

下一步工作将对组播多径路由算法的防抖动机制进行研究,权衡数据选路和路径切换.

参 考 文 献

[1] Luo Jun-Zhou, Wu Wen-Jia, Yang Ming. Mobile internet: terminal devices, networks and services. Chinese Journal of Computers, 2011, 34(11): 2029-2051(in Chinese)
(罗军舟, 吴文甲, 杨明. 移动互联网:终端、网络与服务. 计算机学报, 2011, 34(11): 2029-2051)

[2] Akyildiz I F, Wang X D, Wang W L. Wireless mesh networks: A survey. Computer Networks, 2005, 47(4): 445-487

[3] Si W S, Selvakennedy S, Zomaya A Y. An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks. Journal of Parallel and Distributed Computing, 2010, 70(5): 505-524

[4] Roy S, Koutsonikolas D, Das S et al. High-throughput multicast routing metrics in wireless mesh networks. Ad Hoc Networks, 2008, 6(6): 878-899

[5] IEEE draft standard for information technology-telecommunications and information exchange between systems-Local and metropolitan area networks-specific requirements-part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications-amendment 10: mesh networking [Z]. 2011: 1-361

[6] Kyoung J O, Lee C Y. Multicast routing protocol with low transmission delay in multi-rate, multi-radio wireless mesh networks//Proceedings of the IEEE International Conference on Communications (ICC). Cape Town, 2010: 1-6

[7] Eryilmaz A, Ozdaglar A, Medard M. On delay performance gains from network coding//Proceedings of the 40th Annual Conference on Information Sciences and Systems. Princeton, 2006: 864-870

[8] Yeow W L, Hoang A T, Tham C K. On average packet delay bounds and loss rates of network-coded multicasts over wireless downlinks//Proceedings of the IEEE International Conference on Communications (ICC). Dresden, 2009: 1-6

[9] Yeow W L, Hoang A T, Tham C K. Minimizing delay for multicast-streaming in wireless networks with network coding//Proceedings of the 32nd IEEE International Conference on Computer Communcations (INFOCOM). Rio de Janeiro, 2009: 190-198

[10] Cheng H, Yang S X. Joint multicast routing and channel assignment in multiradio multichannel wireless mesh networks using tabu search//Proceedings of the 5th International Conference on Natural Computation (ICNC). Tianjin, 2009: 325-330

[11] Lim S H, Kim C, Ko Y B et al. Efficient multicasting for multi-channel multi-interface wireless mesh networks//Proceedings of the Military Communications Conference (MILCOM). Boston, 2009: 1-7

[12] Key P, Massoulie L, Towsley D. Combining multipath routing and congestion control for robustness//Proceedings of the 40th Annual Conference on Information Sciences and Systems. Princeton, 2006: 345-350

[13] Nagesh S N, Deepti S N, Dharma P A. Multipath routing in wireless mesh networks//Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS). Vancouver, 2006: 741-746

[14] Popa L, Raiciu C, Stoica I et al. Reducing congestion effects in wireless networks by multipath routing//Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP). Santa Barbara, 2006: 96-105

[15] He B, Dongmei S, Agrawal D P. Diffusion based distributed internet gateway load balancing in a wireless mesh network//Proceedings of the Global Telecommunications Conference (GLOBECOM). Honolulu, 2009: 1-6

[16] Li F, Fang Y, Hu F et al. Load-aware multicast routing metrics in multi-radio multi-channel wireless mesh networks. Computer Networks, 2011, 55(9): 2150-2167

[17] Zeng G K, Wang B, Ding Y et al. Efficient multicast algorithms for multichannel wireless mesh networks. IEEE Transactions on Parallel and Distributed Systems, 2010, 21(1): 86-99

[18] Capone A, Carello G, Filippini I et al. Routing, scheduling and channel assignment in wireless mesh networks: Optimization models and algorithms. Ad Hoc Networks, 2010, 8(6): 545-563

[19] Naveed A, Kanhere S S, Jha S K. Topology control and channel assignment in multi-radio multi-channel wireless mesh networks//Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS). Pisa, 2007: 1-9

[20] Reed W J, Jorgensen M. The double Pareto-lognormal distribution—A new parametric model for size distributions. Communications in Statistics-Theory and Methods, 2004, 33(8): 1733-1753

WANG Wei, born in 1983, Ph. D. candidate. His main research interest is wireless networks.

YANG Ming, born in 1979, Ph. D., associate professor. His research areas are network security and wireless local area network.

LUO Jun-Zhou, born in 1960, Ph.D., professor, Ph. D. supervisor. His main research interests include next generation network architecture, network security, wireless local area network, grid and cloud computing.

LIU Bo, born in 1975, Ph. D., associate professor. Her research areas are service scheduling and management, ubiquitous computing and distributed network management.

Background

Multi-radio Wireless mesh network (MR-WMN) is regarded as one of the key technologies of the backbone of Mobile Internet. Though it has been deployed in many areas for ubiquitous accessing the Internet wirelessly, the nature of WMN, i. e. wireless broadcasting and channel interference, makes it a challenge to efficiently support multicast services with considerably low transmission delay, which will deteriorate the quality of services in upper layers and lead to unpleasant user experience. In the world, exiting multicast optimization efforts for MR-WMNs have traditionally focused on either the flow interference or the gateway load balancing. Though it improves the multicast delay, yet the result is far from expectation. Experiments and applications have all shown that multicast performance can be efficiently improved by explicitly taking into account the overlapping channels and the multipath routing strategies.

Therefore, in this paper, we propose an analytical model for 802.11 based MR-WMN by combining overlapping channel assignment with multipath routing strategies. This model derives a new routing metric, which can be used in subsequent routing stage for obtaining low end-to-end multicast delay. To reduce interflow and introflow interferences, first we prove that finding the minimum global flow interference solution is a NP-Complete problem, and then a quantitative analysis is proposed to reveal the relationship between global flow interference and network node densities in order to design more effective channel assignment algorithm, at last we propose the DCA algorithm to minimum global flow interference, which can efficiently reduce the multicast transmission delay in MAC layer. To avoid the best wireless link being congested, we propose the flow adaptive-based MMRA algorithm by making use of the *MR-MED* routing metric and the multipath routing design philosophy, which takes local channel congestion into account and can efficiently reduce the queuing delay in Mesh layer. The simulation compares our scheme with MCM and the results show that the proposed model accurately characterizes the multicast delay in multi-radio wireless mesh network; the DCA and MMRA efficiently improve the multicast throughput and reduce the end-to-end delay.

This work is supported by National Key Basic Research Program of China under Grant No. 2010CB328104, National Natural Science Foundation of China under Grant Nos. 60903162, 60903161, 61070161, and 61003257, China National Key Technology R&D Program under Grant Nos. 2010BAI88B03 and 2011BAK21B02, China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20110092130002, Jiangsu Provincial Natural Science Foundation of China under Grant No. BK2008030, Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grant No. 93K-9. All these projects aim to improve our country's academic and applied study on networking and are helpful to make advances in network communication technologies. The research team of this paper has focused on the research of multicasting optimization in wireless mesh networks for many years, and has published several papers in international conferences and journals.

Ad Hoc网络中一种基于防策略支付模型的安全激励合作算法

王 博[1)] 黄传河[2)]

[1)](国家计算机网络应急技术处理协调中心 北京 100029)

[2)](武汉大学计算机学院 武汉 430072)

摘 要 Ad Hoc网络中节点之间的正常通信都是通过节点相互合作来进行中继转发.但是,Ad Hoc网络由于受到自身能量、可用带宽和计算能力的限制,节点往往表现出自私性,因此激励节点合作转发的积极性成为当前Ad Hoc网络的研究热点.该文基于算法机制设计中的思想,对Ad Hoc-VCG模型进行具体分析,指出其存在的问题,提出了一种防策略和防共谋攻击的支付模型,设计了一种包含路由建立和数据包转发过程的安全激励合作算法ICTP.最后,通过仿真实验来验证该算法的有效性,并与Ad Hoc-VCG、COMMIT和LMOCP算法进行了性能对比.仿真结果表明:ICTP算法较其它3种算法在性能上有了显著的改善.

关键词 Ad Hoc-VCG;防策略;激励合作;支付模型;共谋

中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2012.01370

Secure Incentive Algorithm Based on Strategy-Proof Payment Model for Cooperation in Ad Hoc Networks

WANG Bo[1)] HUANG Chuan-He[2)]

[1)](*CNCERT/CC*, *Beijing* 100029)

[2)](*School of Computer*, *Wuhan University*, *Wuhan* 430072)

Abstract Cooperation among nodes is important in Ad Hoc networks since in such networks nodes depend on each other for forwarding packets. However, cooperation in such operations consumes nodes energy and recourses. Therefore, it is necessary to design incentive mechanisms to improve the enthusiasm of cooperation among nodes. In this paper, we analyze the Ad Hoc-VCG model based on algorithmic mechanism design, point out its main problems, propose a strategy-proof and collusion-resistant payment model and design an algorithm called ICTP (Incentive Cooperative sTrategy-proof Payment algorithm), which consists of two procedures: routing establishment and data packets forwarding. At last, we verify the effectiveness and correctness of this algorithm by simulation using NS2, compare its performance with other classical algorithms: Ad Hoc-VCG, COMMIT and LMOCP. Simulation results show that ICTP performs better than other algorithms.

Keywords Ad Hoc-VCG; strategy-proof; incentive cooperation; payment model; collusion

1 引 言

Ad Hoc网络不依赖于任何固定的基础设施,节点既是终端又是路由器,这种双重功能决定了节点既要保证自身的正常通信需求,又要为网络的其它节点提供转发数据和寻找路由的服务,即需要所有节点的相互协作来实现.这种合作性也是现有诸

收稿日期:2012-01-04;最终修改稿收到日期:2012-05-15.本课题得到国家自然科学基金项目(60633020、60970117)资助.**王 博**,男,1982年生,博士,主要研究方向为计算机网络与通信、信息安全.E-mail: wbxyz@163.com.**黄传河**,男,1963年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为计算机网络、分布并行处理和网络安全.

多路由协议设计的一个基本假设前提.随着当前研究的发展,Ad Hoc 网络逐步由单一控制的军用领域,延伸到民用领域.那么由于节点受不同组织中实体的操作和控制,并且其自身能量和带宽有限,节点往往表现出不合作的自私行为[1].此外,仿真分析表明[2]:当具有自私行为的节点数目占全网总节点数的 10%～40%时,网络的整体性能会严重地下降 16%～32%.如何有效地加强节点间的合作,从而保障 Ad Hoc 网络的可用性及其整体性能,是 Ad Hoc 网络当前研究的热点之一.

当前针对自私节点的一种新的解决思路是为节点转发数据提供补偿,并且使得补偿大于转发数据的代价,以激励节点参与数据转发过程.但是每个节点的私有信息转发价格(主要与传输功率和电池剩余电量有关)可能不同,自私节点为了获得超额收益(效用),可能会虚报自己的转发价格,这就要求我们必须采用某种机制迫使节点真实地报告自己的转发价格.于是,一些研究者开始探索利用博弈论中的机制设计方法[3]来进行解决.本文以算法机制设计思想作为研究出发点,对 Ad Hoc-VCG 支付模型进行分析,指出其存在的主要问题,提出了防策略和防共谋的增强支付模型,并采用密码学的方式将此模型运用到在网络层节点如何执行数据转发和路由的问题中,设计了一种安全可靠的激励合作算法:ICTP.该算法考虑了路由建立和和数据包转发两个阶段,来提高节点的合作积极性.最后,通过仿真实验来对其进行验证和分析,并与其它的经典算法进行了性能对比.仿真结果表明:ICTP 算法在性能上较其他算法有显著的提高.

本文第 2 节介绍相关工作;第 3 节给出机制设计原理和 VCG 机制的思想;第 4 节分析基于 Ad Hoc-VCG 支付模型中存在的问题,并提出一种新的支付模型;第 5 节详细介绍安全激励合作算法 ICTP,并给出理论分析;第 6 节是仿真实验;最后第 7 节给出全文总结和未来的工作.

2　相关工作

当前对 Ad Hoc 网络中激励合作问题的研究主要集中在基于博弈论思想来提高节点之间的合作性,减少自私节点产生的攻击.近年的相关研究进展如下:

文献[4-17]考虑将存在自私行为的 Ad Hoc 网络归类于非合作博弈的研究领域,采用相应的博弈论知识来增强节点合作转发的积极性.Felegyhazi 等人[4]提出了一种基于节点拓扑依赖关系的博弈分析模型,Altman 等人[5]则在前者基础上进一步提出了一个调整转发概率的模型.此外,Srinivasan 等人[6]还提出了 GTFT(Generous Tit-For-Tat)模型,尝试以针锋相对的转发策略来平衡节点间的相互贡献;进一步 Levin[7]提出了一种以拥塞信道手段来强制协作的思路.其中,Felegyhazi 和 Altman 的模型均引入了对拓扑的依赖性,其均衡状态的结论唯有在满足特定条件时方可应用,而这要求节点必须了解全局拓扑结构.与前者不同,GTFT 从节点与网络间而不是多个节点间博弈的角度分析,从而避免了上述依赖性,但它仅考虑了历史收益对节点决策的影响,而没有考虑其将来获利的期望.在 Levin 的工作中,尽管作者证明了纳什均衡的存在性,但却没有提出具体的合作增强机制.文献[8]对节点间转发数据包的过程进行建模,也证明了纳什均衡存在的条件,提出了对自私行为约束的惩罚措施.文献[9]认为到实际网络中物理层的信道干扰、冲突和碰撞,导致数据包的丢失,此种情况的发生也不能归结为由节点的"自私性"所致,导致该节点遭受无辜的惩罚.同理,文献[10]提出了一种 DARWIN 模型,避免可能出现善意节点被误认为"自私节点"所经受的惩罚,帮助该节点快速恢复到合作转发数据包的状态.该模型具有较强的鲁棒性,能够抵抗共谋攻击的影响.文献[11]提出一种在路由发现阶段建立的转发困境博弈模型,并通过邻居发现协议来发现参与博弈的节点数,网络中的节点通过混合策略中的概率转发策略,实现了网络的混合策略纳什均衡.文献[12]通过建立基于信任的演化博弈模型,激励自私节点动态地选择合适的转发策略,进一步分别通过决定式模型和随机模型,来预测和了解邻居节点的转发行为,同时,结合遗传算法来促使节点优先选择转发策略,并且在 AODV 协议上进行了实现和仿真.文献[13]针对自组网络节点的预期收益及其协作交互过程建立了一个重复博弈模型,提出了一个激励一致性条件,在此条件下,节点会迫于惩戒机制威慑而自愿采取合作策略;并分析了节点对将来利益的重视程度、机制参数和作弊检测效率对协作效果的影响.文献[14]总结了无线网络中因自私节点的存在而带来的一些关键问题,特别对含有自私节点的无线环境中基于非合作博弈理论的路由机制进行了分析和研究.文献[15]针对基于邻居节点中继和生成的路由请求包过程,提出了一种适用于按需

路由协议寻路阶段的自私行为检测和惩罚机制，对算法激励合作的有效性进行分析. 文献[16]提出一种建立在路由发现阶段的转发困境博弈模型，并通过邻居发现协议来发现参与博弈的节点数，网络中的节点通过混合策略中的概率转发策略，实现了网络的混合策略纳什均衡. 文献[17]首次将博弈论和信任模型综合考虑，来实现节点的合作性. 通过博弈策略的演化和遗传算法的结合，达到最优合作策略的快速收敛. 最后，该文在非合作博弈的背景下提出了一种分布式演化算法来快速选择最优的合作策略. 文献[18]提出了一种满足激励兼容条件的有限时间信誉系统 FITS 来增强节点的合作性. 该系统主要采用博弈论思想中的有限重复博弈和信誉估计模型来提高节点数据包转发概率，并通过详细的理论和实验进行论证. 文献[19]针对 Ad Hoc 网络中相邻节点的合作转发过程建立无限重复合作转发博弈模型，结合该模型提出了激励节点之间合作的积极性条件，并给出了一种通用惩罚机制，对节点的不合作行为给予比较严厉的惩罚，使得节点从不合作行为中得到的收益由于被惩罚而抵消，确保节点恢复到合作状态，然后给出了该惩罚机制的实现框架.

文献[20]提出了基于 VCG 支付模型的合作机制 Ad Hoc-VCG. 这种基于 VCG 的机制是从经济学上的经典拍卖模型 VCG 改进而来的，可以有效地解决自私节点为路径选择和数据转发带来的问题. Ad Hoc-VCG 机制在转发数据包的过程中优先选择一条能耗最小的路径，激励该路径中的节点积极转发数据包，根据节点的真实报价信息，给予较高的补偿来弥补节点的合作转发代价，从而使得节点获得的效用最大(可以将此过程理解为二级密封价格拍卖活动)，最后基于该模型实现了一种路由协议，通过证明，该协议具有防策略性，能够找到一条成本有效的路径. 此外，文献[21]也基于 VCG 支付模型对网络中的恶意节点和自私节点进行分析，提出一种安全诚实路由协议，而文献[22]实现了一种 LOTTO 协议，通过该协议不仅也能选择一条成本最优的路径，并且能够大大降低路由建立过程中的开销至 $O(n^2)$. 其它相关研究如 CORSAC[23]、TEAM[24]、COMMIT[25]和 LMOCP[26]等，其大多基于 VCG 机制[3]，而忽略了数据包在转发过程中的重复性. 由中继者报价、发送者给予选定路由中中继者一定数量的超额红利来激励合作是其核心思想. 为鼓励诚实报价，发送者经拍卖支付的总费用一般远高于实际所需的代价，这使得收支平衡问题进一步恶化.

还有一类是利用密码学方式来增强激励合作的安全机制：文献[27]提出一种比较实用的激励合作系统 PIS. 该系统主要是在 Sprite[28]模型的基础上进行改进，对每次会话产生的数据包收据大小进行压缩，减少整个系统的存储空间. 此外采用签名技术来确保源节点和目的节点之间通信的完整性、真实性和不可抵赖性. 文献[29]提出了一种安全高效的激励合作协议 ESIP. 该协议主要采用基于公钥密码(Hash 链和签名等)和身份鉴别技术来实现，从而确保整个通信过程数据包转发的完整性、可用性和不可抵赖性. 文献[30]也是在 Sprite 模型的基础上，通过采用轻量级的统计方法来辨别为了提高自身利益的欺骗节点和共谋节点的攻击行为，实现了一种基于有效的激励合作框架的欺骗监测系统. 文献[31]主要围绕网络环境下数据包丢失攻击行为的发生可能性开展讨论，从节点自身理性情况下发生的自私攻击行为和由于非理性节点故意丢弃数据包的行为两个角度来考虑，提出了一种新的激励合作机制 TRIPO. 该机制对理性的自私节点采取基于微支付的方式来进行激励，而对非理性的恶意节点则采取信誉系统的思想来对节点的故意丢包行为进行监视，并通过实验验证了该机制的有效性.

本文的研究主要是基于 Ad Hoc-VCG 机制，对该机制中的支付模型进行详细分析，指出其存在的主要问题，对其支付模型进行改进，提出一种防策略、防共谋的增强支付模型，对路由建立和数据包转发过程统一考虑，建立有效激励合作算法. 此外，借鉴密码学的一些思想，进一步提高激励合作算法运行的安全性，加强整个通信过程中数据包转发的完整性、可用性和不可抵赖性.

3 算法机制设计预备知识

本节仅给出机制设计中的基本知识以及机制设计中的典型应用：VCG 机制，详细内容可参考文献[3].

3.1 算法机制设计原理

定义 1(机制). 一个机制 $M=(O,P)$ 包含了两个部分：结果规则函数 O 和支付函数 P. 当网络中需要通信的节点对 (S,D) 建立了一条 h 跳的路径，结果规则函数 O 映射了该路径上的中间节点向机制报告其整个类型向量 $\boldsymbol{a}=(a_1,a_2,\cdots,a_h)$ 时的输出结果；而支付函数 P 则规定了每个中间节点 i 在

该类型 a_i 下的支付 $Pay(S,i,\boldsymbol{a})$.

此外，每个节点 i 都有一个真实的类型 t_i，由于受自身实际情况的约束，可能表现出 $t_i \neq a_i$，即节点 i 不以真实的类型来报告给机制 M，同时节点 i 也在该类型 a_i 下有相应的估价函数 $V_i(O,\boldsymbol{t})$. 其中，$\boldsymbol{t}=(t_1,t_2,\cdots,t_h)$ 为路径上 h 个中间节点的真实类型组成的向量. 那么可以得出节点 i 以类型 a_i 报告时的效用函数：

$$U_i(O(\boldsymbol{a}),\boldsymbol{t}) = V_i(O(\boldsymbol{a}),\boldsymbol{t}) + Pay(S,i,\boldsymbol{a}) \quad (1)$$

定义 2(机制实现). 对于机制 $M=(O,P)$，如果对于 $\forall(a_1,a_2,\cdots,a_h)\in M$，则存在 $P(a_1,a_2,\cdots,a_h)=F(\boldsymbol{a})$，可以说机制 M 实现了社会选择函数 $F(\boldsymbol{a})$，其中 $a_1,a_2,\cdots,a_h$ 是由机制 M 导出的博弈均衡解.

定义 3(占优解). 由于各个节点都是理性的，在不管其它 $h-1$ 个节点选择的策略 $\boldsymbol{a}_{-i}$ 情况下，其中 $\boldsymbol{a}_{-i}=(a_1,a_2,\cdots,a_{i-1},a_{i+1},\cdots,a_h)$，节点 i 在选择 a_i 时其效用函数能够达到最大，那么节点 i 选择的策略 a_i 是占优策略，也即存在占优解.

定义 4(防策略性). 如果每个节点的类型都是策略空间的一部分，每个节点通过报告其真实类型来最大化它们的效用函数，且不受其它节点的类型影响，则该机制实现了防策略性. 如果机制 $M=(O,P)$ 满足以下 3 个具体条件，可以说机制 M 具有防策略性：

(1) 激励兼容性

对于一个机制 M 具有防策略性，支付函数要满足激励兼容性，即每个节点 i 需要满足：

$$V_i(O(\boldsymbol{a}\,|^i t_i),\boldsymbol{t}) + Pay(S,i,\boldsymbol{a}\,|^i t_i) \geq V_i(O(\boldsymbol{a}),\boldsymbol{t}) + Pay(S,i,\boldsymbol{a}) \quad (2)$$

其中，$\boldsymbol{a}\,|^i t_i=(a_1,a_2,\cdots,a_{i-1},t_i,a_{i+1},\cdots,a_h)$.

(2) 节点理性

防策略性机制保证各个节点都自愿参加. 每个节点都可以得到一个非负的效用函数值. 即对每个节点 i 满足：

$$V_i(O(\boldsymbol{a}\,|^i t_i),\boldsymbol{t}) + Pay(S,i,\boldsymbol{a}\,|^i t_i) \geq 0 \quad (3)$$

(3) 多项式时间可计算性

防策略性机制需要保证所有的结果规则函数 O 和支付函数 P 的计算都满足多项式时间.

3.2 VCG 机制思想

算法机制设计中典型的应用就是 VCG 机制. 该机制由 Vickrey[32]、Clarke[33] 和 Groves[34] 提出. VCG 机制主要解决的问题是结合各个节点给出的估价函数 $V_i(O,\boldsymbol{t})$，来最大化社会选择函数 $g(O,\boldsymbol{t})$：$g(O,\boldsymbol{t})=\sum_i V_i(O,\boldsymbol{t})$. 这样的最大化机制设计问题在经济学领域中称为该机制具有功利性.

定义 5(VCG 机制). 一个机制 $M=(O(\boldsymbol{t}),P(\boldsymbol{t}))$ 被认为是 VCG 机制，需要满足的条件如下：

(1) 当各个节点(如节点 i)都以其真实的类型 t_i 向机制 M 报告时，整个机制得到的输出结果 $O=O(\boldsymbol{t})$ 可以最大化 $g(O,\boldsymbol{t})$. 其中，$g(O,\boldsymbol{t})=\sum_i V_i(O,\boldsymbol{t})$；

(2) 每个节点 i 可以获得的支付 $Pay(S,i,\boldsymbol{t})$ 如下：

$$Pay(S,i,\boldsymbol{t}) = \sum_{j\neq i} V_j(O,\boldsymbol{t}) + H^i(t_{-i}) \quad (4)$$

其中，$t_{-i}=(t_1,t_2,\cdots,t_{i-1},t_{i+1},\cdots,t_h)$，$H^i$ 为 t_{-i} 的任意函数.

Groves 在文献[34]中已经证明 VCG 机制具有防策略性，Green 和 Laffont 在文献[38]中指出，当在给出适当的假设条件下，VCG 机制在很多功利性的问题中表现出防策略性.

本文的思想就是基于 VCG 机制和文献[20]中提出的 Ad Hoc-VCG 机制，为了激励网络的节点积极主动地参与网络中的各项工作，要求各个节点将其真实的私有类型信息(例如：节点成本或无线链路成本)报告给需要发生数据包的源节点，源节点根据网络的实际情况，确定各个节点的支付费用，从而确保它们自身效用的最大化，实现整个支付模式的防策略性，达到激励自私节点进行合作的目的.

4 模型建立

4.1 模型假设

为了便于形式化分析和说明，对本文的模型条件进行假设：

(1) 网络是一个有向图(或双向连通图)，用加权图 $G=(V,E)$ 来表示 Ad Hoc 网络，其中 V 为网络的非空节点集合，E 为连接节点对的通信链路集合(相邻节点都在彼此的通信范围之内)，$|V|$ 和 $|E|$ 分别表示该网络中节点和链路的数目.

(2) 网络中各个节点都是理性的，都很自愿地参与到 VCG 机制中来，但都以追求其自身的效用 U_i 为最大化的目的.

(3) 由源节点 S 来为网络中的节点支付转发费用，并且在建立路由之前，预先设置一个最大的支付费用阈值 m_s.

(4) 在需要建立通信的 (S,D) 节点对中，一般情

况下,目的节点 D 都是确定的、可信的、不存在攻击行为,并且其具有较强的存储能力和计算能力.

(5)网络中的节点 i 在转发数据包的过程中,可能由于消耗自身能量、CPU 计算能力、带宽资源,产生一定的转发成本 c_i,并且,只有该节点自己获知其私有类型成本信息.由于受到自私利益的驱动,节点 i 可能以成本信息 d_i 向源节点报告,一般情况下,$d_i \neq c_i$.

(6)网络中的链路状态比较良好,不存在受物理信道干扰、冲突的影响情况,在转发过程中不考虑数据包丢失重传的情况,避免节点过多地增加转发成本.

(7)在下次网络拓扑结构变化之前,整个网络的状态都比较稳定,基本上避免了各个节点的转发成本受节点移动和拓扑结构变化的影响.

(8)本模型的建立是以单个数据包的转发来考虑,此模型也可以延伸到多数据包转发的情况.

4.2 Ad Hoc-VCG 支付模型

假如节点对(S,D)需要建立通信,通信过程中需要经历若干 h 个中间节点.为了激励这 h 个中间节点的合作转发积极性,源节点 S 需要支付一定的费用来补偿中间节点转发数据包所消耗的成本,而各个节点需要获得的具体支付情况,则通过各个节点 i 向源节点 S 报告其对应的私有类型成本信息 d_i 来计算.这 h 个中间节点所报告的类型信息,可以组合成一个类型向量 $\boldsymbol{d}$:$\boldsymbol{d}=(d_1,d_2,\cdots,d_h)$.通过此支付思想,源节点 S 可以选择一条成本最小的路径 $LCP(S,D,\boldsymbol{d})$来转发数据包.具体的支付模型如下:

$$Pay(S,i,\boldsymbol{d}) = \begin{cases} 0, & i \notin LCP(S,D,\boldsymbol{d}) \\ \|LCP_{-i}(S,D,\boldsymbol{d})\| - \|LCP(S,D,\boldsymbol{d})\| + d_i, & i \in LCP(S,D,\boldsymbol{d}) \end{cases} \tag{5}$$

其中,$LCP_{-i}(S,D,\boldsymbol{d})$表示从 $S\sim D$ 的所有路径中不包含中间节点 i 的最小成本路径,而$\|LCP_{-i}(S,D,\boldsymbol{d})\|$为对应的路径成本值.当节点 i 在 $LCP(S,D,\boldsymbol{d})$中,其获得源节点 S 给予的支付费用为 $Pay(S,i,\boldsymbol{d})=\|LCP_{-i}(S,D,\boldsymbol{d})\|-\|LCP(S,D,\boldsymbol{d})\|+d_i$;而其它不在 $LCP(S,D,\boldsymbol{d})$中的节点获得的支付费用为 0.

该 Ad Hoc-VCG 支付模型是 VCG 机制在 Ad Hoc 网络中的典型应用.当节点 i 向源节点 S 报告其真实类型信息 c_i 时,即 $d_i=c_i$,则节点 i 获得的效用 $U_i(\boldsymbol{d})$最大:

$$U_i(\boldsymbol{d}) = Pay(S,i,D) - d_i \tag{6}$$

并且保证节点 i 的效用值是非负的:$U_i(\boldsymbol{d})\geq 0$.

源节点 S 在实际的支付过程中,所需要支付的总费用为 $\sum_{i\in LCP, i\notin (S,D)} Pay(S,i,D)$,而源节点 S 获得的效用为 $U_s=m_s-\sum_{i\in LCP, i\notin (S,D)} Pay(S,i,D)$.此外,源节点 S 能够建立路由的条件 $U_s\geq 0$.

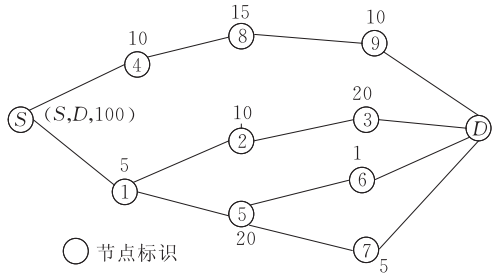
4.3 问题描述

进一步对上节的 Ad Hoc-VCG 支付模型进行分析,发现 Ad Hoc-VCG 支付模型存在以下问题:

(1) Ad Hoc-VCG 模型不具有防策略性.

文献[25]中已经指出 Ad Hoc-VCG 模型不具有防策略性,现举例分析和说明.

如图 1 所示,假设源节点 S 可以提供的最大支付为 100,S 到 D 的最小成本路径为 LCP,对应的 $LCP=\{1,5,6\}$,通过 Ad Hoc-VCG 支付模型计算,得出 LCP 中各个节点的所获支付情况(利用式(5)):节点 1 的所获支付情况 $Pay(S,1,\boldsymbol{d})$:$Pay(S,1,\boldsymbol{d})=35-26+5=14$;节点 5 的所获支付情况 $Pay(S,5,\boldsymbol{d})$:$Pay(S,5,\boldsymbol{d})=35-26+20=29$;节点 6 的所获支付情况 $Pay(S,6,\boldsymbol{d})$:$Pay(S,6,\boldsymbol{d})=35-26+1=10$.



节点旁边的数字表示节点的成本信息

图 1 网络拓扑结构图

同时,这 3 个节点对应的效用情况(利用式(6)):节点 1 的效用 $U_1(\boldsymbol{d})$为 $U_1(\boldsymbol{d})=14-5=9$;节点 5 的效用 $U_5(\boldsymbol{d})$为 $U_5(\boldsymbol{d})=29-20=9$;节点 6 的效用 $U_6(\boldsymbol{d})$为 $U_6(\boldsymbol{d})=10-1=9$.

而当节点 5 虚报其成本类型信息为 22 时,LCP 中各个节点(1,5,6)的支付情况需要重新计算:$Pay(S,1,\boldsymbol{d})=35-28+5=12$;$Pay(S,5,\boldsymbol{d})=35-28+22=29$;$Pay(S,6,\boldsymbol{d})=35-28+1=8$.

对应的效用也相应更新值为$U_1(\boldsymbol{d})=12-5=9$;$U_5(\boldsymbol{d})=29-20=9$;$U_6(\boldsymbol{d})=8-1=7$.

通过对比发现:当节点 5 虚报信息后,其自身所获的支付费用仍然保持为 29,而节点 1 和 6 的支付分别更新为 12 和 8,由于节点 5 的虚报行为,即使节点 1 和 6 仍以真实类型信息报告,则其分别对应

的效用值为 7(7)也达不到最大值 9(9). 因此，Ad Hoc-VCG 模型不具有防策略性.

(2) Ad Hoc-VCG 模型存在过度支付问题.

结合图 1 的示例所示，当节点 5 虚假报告其成本类型信息为 10 时，则 LCP 中所有节点(1,5,6)所获支付情况可以更新如下：

$$Pay(S,1,\boldsymbol{d}) = 35-16+5 = 24;$$
$$Pay(S,5,\boldsymbol{d}) = 35-16+10 = 29;$$
$$Pay(S,6,\boldsymbol{d}) = 35-16+1 = 20.$$

对应的效用情况为

$$U_1(\boldsymbol{d}) = 24-5 = 19;$$
$$U_5(\boldsymbol{d}) = 29-20 = 9;$$
$$U_6(\boldsymbol{d}) = 20-5 = 15.$$

LCP 中的总支付为：24+29+20=73. 而节点 5 真实报告成本信息为 20 时，LCP 中的总支付为 53. 很明显，源节点 S 需要多支付金额为 20 的费用. 因此，网络中的源节点 S 存在过度支付问题.

(3) Ad Hoc-VCG 模型中消息负载的开销过高，大致为 $O(n^3)$.

在 Ad Hoc-VCG 模型中，一次路由发现过程中总共需要转发 $O(n^3)$ 数量级的路由请求数据包. 当网络中的节点数量较多时，消息负载开销会更大，将严重影响网络的性能.

(4) Ad Hoc-VCG 模型中存在共谋攻击的可能性.

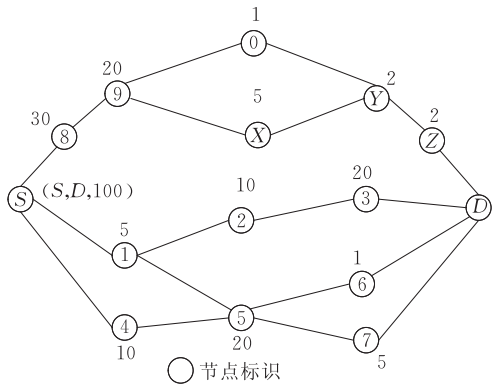
如图 2 所示，当节点 5 和 6 形成共谋小团体时，它们将联合获取较高的支付，其表现形式可能为：节点 5 故意降低自己的成本信息为 10，而节点 6 则故意抬高其成本信息为 10. 在此情况下，LCP 中的中间节点所获源节点 S 的支付为

$$Pay(S,1,\boldsymbol{d}) = 35-25+5 = 15;$$
$$Pay(S,5,\boldsymbol{d}) = 35-25+10 = 20;$$
$$Pay(S,6,\boldsymbol{d}) = 35-25+10 = 20.$$

对应的效用情况为

$$U_1(\boldsymbol{d}) = 15-5 = 10;$$
$$U_5(\boldsymbol{d}) = 20-20 = 0;$$
$$U_6(\boldsymbol{d}) = 20-5 = 15.$$

由于网络中存在共谋攻击的影响，节点 6 的效用值相对于其报告真实类型信息时，增加了 6，但是，为了满足共谋小团体的效益，假如节点 6 转移其 50% 的效用值来补偿节点 5 为了整个小团体的利益而做出的牺牲，使其获取了效用值为 3 的收益，但此时节点 5 的效用并没有达到最大.



节点旁边的数字表示节点的成本信息

图 2　网络拓扑结构图

4.4　模型建立

4.4.1　防策略支付模型

针对 Ad Hoc-VCG 支付模型中存在的问题，本文首先考虑解决防策略性方面的问题，对其支付模型进行改进. 从 Ad Hoc-VCG 支付模型的分析中可以看出：$LCP(S,D,\boldsymbol{d})$ 中的节点为了获取最大化的效用，可能会表现出虚报其真实类型信息的情况，不过此虚报没有任何意义，即对该节点自身的效用没有任何影响，而对 $LCP(S,D,\boldsymbol{d})$ 中其它节点的效用产生影响，使得路径中的节点以真实的类型信息向源节点 S 报告时，得到的相应效用值不能达到最大. 因此，本节给出一个新的定义，具体如下.

定义 6(全局替代路径).　全局替代路径是指一条从节点 S 到 D 去掉 $LCP(S,D,\boldsymbol{d})$ 中所有中间节点之外的最小成本路径.

根据 4.1 节的模型假设条件：网络是双向连通图，能够确保在该网络拓扑结构下，源节点 S 能够通过拓扑发现协议，找到一条最优的全局替代路径. 因此，改进后的防策略支付模型和 4.2 类似：

$$Pay(S,i,\boldsymbol{d}) = \begin{cases} 0, & i \notin LCP(S,D,\boldsymbol{d}) \\ \|LCP_{-i}(S,D,\boldsymbol{d})\| - \|LCP(S,D,\boldsymbol{d})\| + d_i, & i \in LCP(S,D,\boldsymbol{d}) \end{cases} \tag{7}$$

其中，当节点 i 在 $LCP(S,D,\boldsymbol{d})$ 路径上时，其获得源节点 S 给予的支付费用为 $Pay'(S,i,\boldsymbol{d}) = \|LCP_{-i}(S,D,\boldsymbol{d})\| - \|LCP(S,D,\boldsymbol{d})\| + d_i$，其它不在 $LCP(S,D,\boldsymbol{d})$ 中的节点所获支付仍为 0. $LCP_{-i}(S,D,\boldsymbol{d})$ 仍然表示从 S 到 D 的所有路径中不包含中间节点 i 的最小成本路径. 而源节点 S 能够建立路由的条件如下：$U_s' = m_s - \|\overline{LCP(S,D,\boldsymbol{d})}\| \geq 0$. 而 $\overline{LCP(S,D,\boldsymbol{d})}$

表示 $LCP(S,D,\boldsymbol{d})$的全局替代路径.本节改进的防策略支付模型主要是通过$\|\overline{LCP(S,D,\boldsymbol{d})}\|$来约束源节点 S 支付的总费用,使源节点 S 也具有防策略性.

可以证明该支付模型具有防策略性,详细过程见 5.3 节中的定理 3.

现以一个示例说明该支付模型的思想.具体见图 2 所示.

在图 2 中,假设源节点 S 仍以最大支付 100 来建立路由.首先 S 仍然通过拓扑发现协议,计算出从节点 S 到节点 D 的最小成本路径为 $LCP(S,D,\boldsymbol{d})=\{1,5,6\}$,其对应的路径成本仍为$\|LCP(S,D,\boldsymbol{d})\|=26$.同时,全局替代路径也可以通过全局的网络拓扑结构计算得出$\overline{LCP(S,D,\boldsymbol{d})}=\{8,9,0,Y,Z\}$,对应的路径成本$\|\overline{LCP(S,D,\boldsymbol{d})}\|=55$.

由于$\|\overline{LCP(S,D,\boldsymbol{d})}\|<m_s=100$,源节点 S 可以发起路由发现和数据包转发过程.利用本节的支付模型分别计算出节点 1,5,6 的支付费用如下:

$$Pay'(S,1,D)=31-26+5=10;$$

$$Pay'(S,5,D)=35-26+20=29;$$

$$Pay'(S,6,D)=35-26+1=10;$$

$$\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)=10+29+10=49<\|\overline{LCP(S,D,\boldsymbol{d})}\|=55.$$

很明显,存在以下问题:$\|\overline{LCP(S,D,\boldsymbol{d})}\|-\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)=6$,源节点$S$ 支付的总费用为 55,而 LCP 中节点需要支付的总费用为 49,多余的支付没有得到有效地利用,即经济学上所谓的“盈余”现象.围绕此示例仔细分析,当$\overline{LCP(S,D,\boldsymbol{d})}$中的节点 8 更改其成本信息为 20,则$\|\overline{LCP(S,D,\boldsymbol{d})}\|=45$,不过此时,$\|\overline{LCP(S,D,\boldsymbol{d})}\|=45<m_s=100$(路由发现和数据包转发过程仍然可以进行),但是$\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)=49>\|\overline{LCP(S,D,\boldsymbol{d})}\|=45$,出现了实际支付的费用不足,超出了预计的最大支付,即经济学上所谓“亏空赤字”现象.因此,以上问题总结为:支出和预算不平衡问题,即

$$\|\overline{LCP(S,D,\boldsymbol{d})}\|\neq\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D).$$

为了解决源节点 S 预先定义的支付费用和实际的支付不平衡问题,本节通过可信的目的节点 D 来进行调控,实现源节点 S 费用的支出和预算之间的平衡.实际上在现实社会中,这种预算不平衡的现象出现很正常.为了便于理解,假定一种场景:源节点 S 为无线接入网络的用户,而目的节点 D 则为提供无线接入网的服务提供商.该服务提供商是诚实可信的,用户 S 为了使用无线接入提供的服务,必须向无线接入网的服务提供商支付一定的费用$\|\overline{LCP(S,D,\boldsymbol{d})}\|$,费用的分摊可以补偿给为用户 S 提供接入建立路由的中间节点,具体的费用通过 $LCP(S,D,\boldsymbol{d})$来计算得到$\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)$,分两种情况讨论:

(1) 当$\|\overline{LCP(S,D,\boldsymbol{d})}\|>\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)$时,服务提供商 D 将用户 S 支付的剩余费用要么直接退付给用户 S,要么通知用户 S 在下次享用服务时,可以作为下次支付费用的一部分来解决;

(2) 当$\|\overline{LCP(S,D,\boldsymbol{d})}\|\leq\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)$时,为了吸引用户 S 以后更多的使用该服务提供商提供的服务,服务提供商 D 可能采取以下措施:① 由 D 来负责代替支付用户 S 支付其不足的费用;② D 将登记支付费用不足的用户 S 的详细情况,以便下次及时提醒其补交;③ D 考虑到实际成本的情况,下次再向其他用户提供接入服务时,调整用户的支付价格以及预算费用情况.因此,当用户 S 预算不足时,由 D 来调控处理的思想也是可行的.

4.4.2 防共谋支付模型

从 4.3 节看到,Ad Hoc-VCG 支付模型中也存在共谋攻击问题.而在 4.4.1 节提出的防策略支付模型仅仅解决了 Ad Hoc-VCG 支付模型的防策略问题,该支付模型也存在共谋攻击的可能性.如何在 4.4.1 节支付模型的基础上进行改进,建立有效的防共谋支付模型是本节的研究重点.

为便于挖掘网络中共谋攻击发生的表现形式和条件,本节对支付模型中存在的共谋攻击情况进行了分析和总结,特以 2 个节点形成的共谋小团体为例(多于 2 个节点形成的共谋场景就是此简单场景的延伸).团体中的 2 个节点通过密谋协商虚报其类型信息,来获取较高的支付费用,并根据具体的支付费用情况,弥补共谋节点为了整个团体的收益而牺牲的所获得的收益.本节首先分析几种典型的共谋情况,以便对以上提出的支付模型进行改进.结合网络的实际情况,可能发生攻击的条件如下:

(1) 一个节点在 LCP 中,另外一个节点不在 LCP 中.此种情况已在文献[35]中进行了详细的分析,提出的支付模型可以实现防策略性,并且可以抵制共谋攻击产生的影响.

(2) 两个节点都不在 LCP 中，从 4.3 节中可以看出，不在 LCP 中的节点可以通过降低自己的类型信息来获取较高的支付. 当这两个节点降低的类型信息足以确保这两个节点加入到 LCP 中时，那么该两个节点获得的效用将减少；当这两个节点降低的类型信息不足以加入到 LCP 中，则其获得的支付为 0，对 LCP 路径不产生影响. 因此，此情况发生共谋的可能性很小.

(3) 两个节点都在 LCP 中.

① 两个节点都抬高自己的真实成本类型信息.

此种情况出现共谋的可能性很小，原因是由于两个节点提高自己的成本信息过高，那么这两个节点出现在 LCP 中的可能性减少，极有可能被网络中其它路径所替代，即可能获得的支付变为 0.

② 两个节点都降低自己的真实成本类型信息.

通过降低这两个节点的成本类型信息，这两个节点仍然在 LCP 中，但是这两个节点获得的支付很可能增加很多，但是网络的总支付最终受限于 $\|\overline{LCP(S,D,\boldsymbol{d})}\|$. 如果在总支付在小于 $\|\overline{LCP(S,D,\boldsymbol{d})}\|$ 的情况下，源节点 S 发起路由建立的可能性较大，则在此路由建立过程中很可能出现共谋.

③ 一个节点抬高其类型信息，一个节点则降低其对应的信息.

此种情况结合节点的虚报成本信息的程度，进一步分两种子情况来分析：

子情况 1.　一个节点成本提高的程度高于另一个节点成本降低的程度.

此子情况出现共谋的情况也较少. 虽然两个节点通过此方式增加了效用，但是 LCP 中的其它节点的效用受这两个节点的影响，所获得的效用比发生共谋之前明显降低. 此外，这两个节点虚报成本信息的折中，导致其 LCP 的路径成本增加，很有可能存在其它路径来替换当前的 LCP.

子情况 2.　一个节点成本提高的程度不高于另一个节点的成本降低的程度.

从表面上来看，其中一个虚报成本降低的节点效用降低，另外一个节点虚报成本增加的节点效用增加，但是由于效用降低的程度不足以用增加的程度去弥补，因此，此子情况发生共谋的可能性很小.

通过对上述分析结果进行总结，具体可能发生共谋攻击的条件和场景见如表 1 所示.

两个节点不互为邻居节点的分析思路就是以上讨论过程的延伸，其对应的具体情况也对应有 3 种，

表 1　两个节点都在 *LCP* 中发生共谋的情况总结

有一个节点在 LCP 的情况	两个都不在 LCP 的情况	两个节点都在 LCP 的情况			
		(高，高)	(低，低)	(高，低)	
				(高＞低)	(高≤低)
共谋存在	共谋不存在	共谋不存在	共谋存在	共谋不存在	共谋不存在

分析思路与上述过程类似，本文略去相应的分析过程. 共谋小团体无非就是通过设计支付费用交换协议[23]来对通过共谋所获的额外支付进行转移，本文通过 5.3 节实现的算法来避免转移情况的发生.

基于以上分析思路，对 4.4.1 节提出的支付模型进行改进如下：

$$Pay''(S,i,\boldsymbol{d})=\begin{cases}0, & i\notin LCP(S,D,\boldsymbol{d})\\ \|LCP_{-Q(i)}(S,D,\boldsymbol{d})\|-\|LCP(S,D,\boldsymbol{d})\|+d_i, & i\in LCP(S,D,\boldsymbol{d})\end{cases} \tag{8}$$

其中，$Q(i)$ 表示为包含节点 i 以及与 i 组成共谋小团体的其它所有节点的集合，而 $LCP_{-Q(i)}(S,D,\boldsymbol{d})$ 表示为从 S 到 D 的所有路径中不包含 $Q(i)$ 中所有节点的最小成本路径，$\|LCP_{-Q(i)}(S,D,\boldsymbol{d})\|$ 则为对应的最小路径成本值. 其它表示的意义与 4.4.1 节相同. 此时，源节点 S 能够建立路由的条件：$U'_s=m_s-\|\overline{LCP(S,D,\boldsymbol{d})}\|\geq 0$. 为了实现整个网络的支付和预算平衡，仍然采用 4.4.1 节中由目的节点 D 进行调控的思想.

定理 1(防共谋性＋防策略性).　该防共谋支付模型具有防策略性.

证明.　假如网络中存在有两个节点 x 和 y，则此两个节点可以互为邻居节点，也可以不互为邻居节点. 为了简化证明的过程，利用 VCG 机制中的节点效用计算思想(参见式(4))，对这两个节点的效用计算根据本节的支付模型进行变换，得到如下计算公式：

$$U'_x(S,x,\boldsymbol{d})=\sum_{i=1}^{h}V_x(O(\boldsymbol{d}),d_i)+H^{-x}(\boldsymbol{d}^{-Q(x)})$$

$$U'_y(S,y,\boldsymbol{d})=\sum_{i=1}^{h}V_y(O(\boldsymbol{d}),d_i)+H^{-y}(\boldsymbol{d}^{-Q(y)}) \tag{9}$$

将 x 和 y 两个节点的效用值进行相加，得到如下计算公式：

$$U'_x(S,x,\boldsymbol{d})+U'_y(S,y,\boldsymbol{d})=2\times\sum_{i=1}^{h}V_x(O(\boldsymbol{d}),d_i)+H^{-x}(\boldsymbol{d}^{-Q(x)})+H^{-y}(\boldsymbol{d}^{-Q(y)})$$

其中，$H^{-x}(\boldsymbol{d}^{-Q(x)})+H^{-y}(\boldsymbol{d}^{-Q(y)})$不依赖于 x 和 y 具体的自身成本类型信息 d_x 和 d_y，此外，当 LCP 中所有节点都将真实的成本类型信息报告给源节点，以此来计算各自的支付时，$\sum_{i=1}^{h}V_x(O(\boldsymbol{d}),d_i)$ 和 $\sum_{i=1}^{h}V_y(O(\boldsymbol{d}),d_i)$都能够达到最大，即 x 和 y 都将以真实的成本类型信息来最大化其对应的效用值. 此证明过程可以延伸到网络中共谋小团体中至少包含 2 个节点的情况. 因此，该支付模型能够抵抗共谋的攻击，具有防策略性. 证毕.

5 安全激励合作算法

5.1 算法体系

本节围绕上节所建立的具有防策略性和防共谋的支付模型，提出安全可靠的激励合作算法 ICTP. 当前已有的相关文献[21-26]，重点在路由发现阶段去分析激励合作问题，而很少考虑在路由建立之后的数据包转发阶段，本文提出的激励合作算法则将这两个阶段统一设计，避免 LCP 中的节点在路由发现阶段表现出很好的合作性，而在数据包转发阶段则又表现出自私性发生的可能性. 最后，考虑到在 4.4.2 节不互为邻居节点的共谋小团体在产生共谋之后，可能会出现所获支付的内部转移等情况发生，以及避免本身网络中出现的篡改、窃听和重放等攻击，而采用密码学方式的加密、认证方式来实现算法的保密性、完整性、不可否认性和可用性. 具体的体系结构图如图 3 所示.

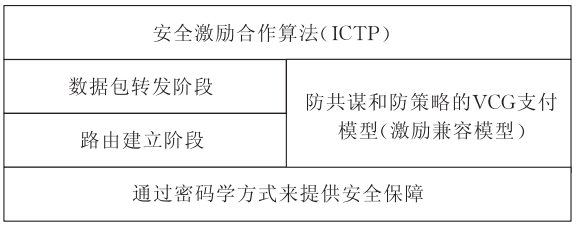


图 3 安全激励合作算法体系

5.2 安全激励合作算法

5.2.1 初始化阶段

本文采用 Diffie-Hellman 密钥交换协议，该协议的有效性依赖于计算离散对数的难度，其目的在于使节点之间安全地交换一个秘密密钥以便用于后续阶段的数据包加密. 本文通过该协议预先给网络中的各个节点分配私钥和公钥，以及在需要通信的节点对之间建立对称密钥. 节点 i 通过该协议选择一个作为私钥的随机数 k_i，并计算其公钥 $K_i=g^{k_i}$. 类似地，目的节点 j 也得到对应的私钥、公钥，分别为 k_j 和 K_j. i 与 j 都可以计算出其对称密钥 $k_{i,j}$：

$$k_{i,j}=g^{k_ik_j}=(K_j)^{k_i}=(K_i)^{k_j},$$

其中，g 为离散对数问题中素数的原根. 具体的协议思想可以参考文献[36].

5.2.2 路由建立阶段

本节的路由建立过程是对基于 Ad Hoc 网络中典型按需路由算法(AODV,DSR)的思想改进实现. 路由建立阶段包括两个子过程：路由请求过程和路由回复过程. 假设 S 和 D 需要建立一条最小成本的路由，具体过程如下：

(1) 路由请求过程

① 源节点 S 向其所有邻居节点(假设其中一个节点为 i)广播路由请求信息：

$$S\rightarrow brdcast:[RDP,K_{S,D}(S,D,m_S,N_S,TTL)],$$

其中，RDP 是路由请求包，$\{S,D,m_S,N_S,TTL\}$信息通过 S 和 D 的对称密钥 $K_{S,D}$加密来确保信息的保密性. N_S是由 S 产生的随机数，目的是用来唯一地确定 RDP 来自哪一个节点，每一次源节点 S 发起路由发现请求时，产生一个单调递增的 N_S源节点序列号，在网络的整个生命周期中 N_S是递增的. TTL 是数据包在网络中的生存时间，增加该参数的目的是防止数据包在网络中逗留时间过长，减少广播风暴对整个网络通信性能的影响.

② 邻居节点 i 收到路由请求信息之后，向源节点 S 发送确认信息：

$$i\rightarrow S:[ACK,K_{S,D}(S,D,m_S,N_S,TTL)]K_{i,S},$$

其中，ACK 是确认包，一般情况下，该确认包的长度比较小，对网络的整个通信开销不造成影响，基本上可以忽略. 节点 i 向源节点 S 发送此确认包的目的是为了通知源节点 S 确认其哪一跳节点收到了路由请求包，进一步提高网络的可靠性. 确认成功之后，进入执行过程③. 否则，源节点 S 需等待其它邻居节点发送的确认包.

③ 节点 i 继续广播路由请求信息至其所有邻居节点 j：

$$i\rightarrow brdcast:[RDP,K_{S,D}(S,D,m_S,N_S,TTL),K_{i,D}(i,D,d_i,N_S,TTL)].$$

节点 i 将其私有类型信息 d_i以及其它相关信息通过 i 和 D 的对称密钥 $K_{i,D}$加密，确保信息转发过程中的保密性. 利用类似过程②的方法，向上一跳节点发送确认信息，确认成功之后，则执行过程④. 否则，节点 i 需等待其它邻居节点发送的确认包.

④ 节点 j 继续广播路由请求信息直至到达目的

节点 D 的前一跳节点 k：

$j \rightarrow brdcast:[RDP, K_{S,D}(S, D, m_S, N_S, TTL), K_{i,D}(i, D, d_i, N_S, TTL), \cdots, K_{k,D}(k, D, d_k, N_S, TTL)]$.

重复过程③直至到达目的节点 D. 利用类似过程③的方法，向上一跳节点发送确认信息，确认成功之后，则执行过程⑤. 否则，节点 k 需等待其它邻居节点发送的确认包.

⑤ 目的节点 D 处理接收到的路由请求信息.

由于 Ad Hoc 网络本身具有广播特性，目的节点 D 最终会收到多个来自源节点 S 发送的不同路由请求包信息，节点 D 通过利用其上一跳邻居节点对应的对称密钥进行解密，因为只有目的节点 D 可以通过它们的对称密钥进行解密，也从侧面上保证了对上一跳邻居节点签名和认证，进而获取到网络中各个节点的私有类型信息. 通过节点 D 获取到的信息，可以构建一个全局拓扑结构图以及通过 Dijkstra 算法在该拓扑结构中形成最小成本路径 $LCP(S, D, \boldsymbol{d})$，并且计算出全局替代路径 $\overline{LCP(S, D, \boldsymbol{d})}$. 本文基于 4.4.1 节和 4.4.2 两节的支付模型，结合源节点 S 的最大支付阈值 m_S 以及 $\overline{LCP(S, D, \boldsymbol{d})}$ 的计算值，目的节点 D 来确定 $LCP(S, D, \boldsymbol{d})$ 中的中间节点支付转发费用. 存在如下两种情况需考虑：

情况 1.　当 $m_S - \|\overline{LCP(S, D, \boldsymbol{d})}\| \geq 0$，则继续执行后续路由回复过程.

情况 2.　当 $m_S - \|\overline{LCP(S, D, \boldsymbol{d})}\| < 0$，源节点的效用为负值，违背了支付模型的防策略性的要求，则不再继续执行后续的路由回复过程.

(2) 路由回复过程

目的节点 D 确定各个中间节点的支付费用之后，则向源节点 S 发送路由回复信息，通知其需要支付的总金额为 $\|\overline{LCP(S, D, \boldsymbol{d})}\|$，并向 LCP 路径中的各个节点通知其给予支付的具体金额以及 $\overline{LCP(S, D, \boldsymbol{d})}$ 和 $LCP(S, D, \boldsymbol{d})$ 等重要信息. LCP 路径中需要支付的节点和支付金额，则通过各自与目的节点 D 的对称密钥进行加密，在路由回复过程中，相应地解密获取对应的支付信息. 具体的过程如下：

① 目的节点 D 向 LCP 路径中节点 D 的上一跳节点 l 发送路由回复信息：

$D \rightarrow l:[REP, K_{S,D}(D, S, LCP, \overline{LCP}, \|\overline{LCP}\|, N_S, TTL), K_{l,D}(D, l, Pay(S, l, \boldsymbol{d}), N_S, TTL), K_{w,D}(D, w, Pay(S, w, \boldsymbol{d}), N_S, TTL), \cdots, K_{x,D}(D, x, Pay(S, x, \boldsymbol{d}), N_S, TTL), K_{y,D}(D, x, Pay(S, x, \boldsymbol{d}), N_S, TTL)]K_l$，

其中，REP 是路由回复包. 节点 D 将重要信息以及 LCP 路径上节点的支付金额，发送给 LCP 路径中距离目的节点最近的上一跳节点 1，REP 路径中包含了 LCP 所要经历的所有节点，以便加快路由回复的过程. 节点 l 通过其私钥解密，如果成功，则执行过程②. 否则，等待重新发送的路由回复包.

② 节点 l 向目的节点 D 发送确认信息：

$l \rightarrow D:[ACK, K_{l,D}(D, l, Pay(S, l, \boldsymbol{d}), N_S, TTL)]K_D$.

同时也考虑到满足可靠性的要求，节点 l 向目的节点 D 确认收到的路由回复信息. 如果目的节点 D 验证成功，则执行过程③，如果失败，则目的节点 D 重新发送路由回复信息.

③ 节点 l 继续沿着反向 LCP 路径发送路由回复包给节点 l 的下一跳节点 w：

$l \rightarrow w:[REP, K_{S,D}(D, S, LCP, \overline{LCP}, \|\overline{LCP}\|, N_S, TTL), K_{w,D}(D, w, Pay(S, w, \boldsymbol{d}), N_S, TTL), \cdots, K_{x,D}(D, x, Pay(S, x, \boldsymbol{d}), N_S, TTL), K_{y,D}(D, x, Pay(S, x, \boldsymbol{d}), N_S, TTL)]K_w$.

节点 w 收到回复包之后，进行节点 l 的签名认证，如果认证成功，则向节点 l 发送类似过程②确认信息，确认成功之后，节点 w 通过解密获取到目的节点 D 发送的支付信息，同时，执行过程④；否则，l 重新发送路由回复信息.

④ 节点 w 继续沿着反向 LCP 路径发送路由回复包直至源节点 S 的下一跳节点 y：

$w \rightarrow y:[REP, K_{S,D}(D, S, LCP, \overline{LCP}, \|\overline{LCP}\|, N_S, TTL), \cdots, K_{x,D}(D, x, Pay(S, x, \boldsymbol{d}), N_S, TTL), K_{y,D}(D, x, Pay(S, x, \boldsymbol{d}), N_S, TTL)]K_y$.

此过程类似于过程③，主要实现通过节点 y 的公钥来进行加密和向其上一跳节点 x 的确认，同时节点 y 通过解密获取到其对应的支付金额.

⑤ 源节点 S 处理接收到的节点 y 发送的路由回复包：

源节点 S 收到节点 y 发送的路由回复包，也要首先进行回复包来源的认证和确认，确保源节点 S 接收到的路由回复包准确无误，通过对称密钥进行解密出预先发送的重要信息，并计算其对应的效用值，进一步加强源节点 S 的防策略性. 此外，通过周期性地发送探测包来验证 $\overline{LCP(S, D, \boldsymbol{d})}$ 路径是否存在.

5.2.3　数据包转发阶段

5.2.2 节的路由建立阶段为本节的数据包转发阶段提供了前提条件. 本文将此两个阶段综合考虑

的出发点是为了保证数据包转发阶段也能提高激励合作的效率,防止部分节点在路由建立阶段积极主动地参与到该部分的过程中,而在数据包转发阶段又表现出其自私性.因此,结合前面支付模型的基础是以单个数据包的合作转发来进行展开,本节进一步延伸,根据源节点 S 实际需要发送的数据包个数来为中间节点支付额外的费用,即源节点 S 需要转发 N 个数据包,则 LCP 中的节点 i 需要源节点 S 提供的支付费用为 $N\times Pay'(S,i,\boldsymbol{d})$,不在 LCP 中的节点给予的支付费用仍然为 0.同时,源节点 S 的效用值就相应地改变为 $U''_S=m_s-N\times\sum_{i\in LCP,i\notin(S,D)}Pay'(S,i,D)$,也要保证 $U''_S\geq 0$.本节的数据包转发算法仍然结合密码学方式来进行,假定通过路由建立过程获得的最优 LCP 路径为 $\{S,1,2,\cdots,n,D\}$,LCP 路径中各个节点的本地缓存中都保存了到达目的节点 D 的最优 LCP 路径下一跳节点的路由表.具体的过程如下:

① 源节点 S 向下一跳节点 1 发送 $Data$ 数据包:

$$S\rightarrow 1:[K_{S,D}(S,D,Data),N'_S,TTL,LCP]K_{1,D}.$$

源节点 S 为了统计发送数据包的个数,通过 N'_S 来统计,如果目的节点 D 成功接收到数据包,则 N'_S 自动递增.添加 LCP 信息是验证接收到数据包的节点是否在 LCP 中.

② 节点 1 接收到数据包后向源节点 S 发送确认:

$$1\rightarrow S:[ACK,LCP]K_{1,S}.$$

为了提高数据包传输的可靠性,节点 1 通过签名认证和 LCP 判断:如果验证成功,则向源节点 S 发送确认信息,同时在本地缓存中记录源节点 S 转发数据包的序列号,继续执行过程③;如果失败,则等待源节点 S 重新发送数据包.

③ 节点 1 向下一跳节点 2 发送 $Data$ 数据包:

$$1\rightarrow 2:[K_{S,D}(S,D,Data),N'_S,TTL,LCP]K_{1,2}.$$

此过程类似过程①,并且验证成功之后,节点 2 也要向节点 1 发送确认信息.

④ 重复以上过程,直至节点 n 向目的节点 D 发送 $Data$ 数据包:

$$n\rightarrow D:[K_{S,D}(S,D,Data),N'_S,TTL,LCP]K_{n,D}.$$

目的节点 D 验证节点 n 签名的有效性以及其是否在 LCP 中,如果验证成功,则执行过程⑤;如果验证不成功,则等待数据包的重新发送.

⑤ 目的节点 D 向源节点 S 发送数据包收到的确认:

$$D\rightarrow S:[ACK,N'_S,TTL,LCP]K_{S,D}.$$

此过程沿反向 LCP 路径,发送该确认信息,中间节点收到确认信息后通过本地缓存的路由表信息,来转发此确认信息给下一跳节点,直至到达源节点 S.当源节点 S 收到此信息后,如果还有数据包要发送,则开始下一个时刻的 N'_S+1 数据包的发送;如果没有,则源节点 S 要向 LCP 路径中的节点支付费用,执行过程⑥.

⑥ 源节点 S 支付费用过程:

每个节点都很积极主动地获取相应的支付费用,通过设计一个简单的控制协议来实现整个支付的过程,避免恶意节点的窃听、篡改等攻击的可能发生,仍然采用密码学方式来加强这方面的考虑.此部分为了避免 LCP 路径中的部分节点优先获取到了支付费用,不愿意为距离源节点 S 较远的节点转发支付费用的情况发生,那么本文采用优先为距离源节点 S 较远的节点支付转发费用,最后支付给距离较近的节点的方式,增强了支付费用过程的有效性.

5.3 算法正确性和有效性分析

本节将从理论上来证明 ICTP 算法的有效性和正确性.具体从以下几个部分进行展开:

定理 2(无环性). 路由建立阶段一定不会存在重复计算环.

证明. 需要证明:设从源节点 S 到目的节点 D 存在一条包含环的路径 $Path$:

$$Path=\{S,v_1,\cdots,v_i,v_j,\cdots,v_{j+k},v_i,v_{i+1},\cdots,v_n,D\}.$$

当 RDP 沿着路径 LCP 第 2 次经过节点 v_i 时,节点 v_i 将其丢弃并不影响目的节点 D 对 LCP 和全局替代路径的计算,从而对本算法的正确性不造成任何影响.具体证明如下:

(1) 因为 $\{S,v_1,\cdots,v_i,v_{i+1},\cdots,v_n,D\}$ 组成一条不含环的从 S 到 D 的路径,沿着这条路径的 RDP 不会被丢弃,因此目的节点 D 可以获得 v_i 的私有类型信息;

(2) 对于环上 $\{v_j,\cdots,v_{j+k}\}$ 的任意节点 $v_l(0\leq l\leq k)$,如果存在一条从 S 到 D 的经过 v_l 的简单路径(不含环),则目的节点 D 也能够获得 v_l 的私有类型信息;

(3) 对于环上 $\{v_j,\cdots,v_{j+k}\}$ 的任意节点 $v_l(0\leq l\leq k)$,如果存在一条从 S 到 D 的路径中都包含环,则节点 v_l 的信息对于计算 LCP 和全局替代路径没有任何意义(因为 LCP 和全局替代路径中都不包含环),因此,LCP 和全局替代路径中都不会包含有 v_l,丢弃该节点不会影响本算法的正确性. 证毕.

定理 3(防策略性). ICTP 算法一定具有防策略性.

证明. 本节结合 4.4.1 节和 4.4.2 节的支付模型,分别从源节点 S、中间节点 i 和目的节点 D 的防策略性来分析和证明该算法的防策略性:

(1) 源节点的防策略性

源节点 S 在寻找最小成本路径 LCP 之前,已经给出预支付的最大值 m_S,即源节点 S 实际支付给 LCP 中节点的总费用一定不会超过 m_S,因此,确保其效用值为非负 $U_S \geq 0$. 下面分两种情况来说明源节点 S 真实报告其私有类型是 VCG 机制的占优策略,其中 m_f 为 S 虚报的类型信息,$\|\overline{LCP(S,D,\boldsymbol{d})}\|$ 为全局替代路径的成本:

当 $m_f \leq m_S$ 时,存在以下 3 种子情况:

子情况 1:$\|\overline{LCP(S,D,\boldsymbol{d})}\| < m_f \leq m_S$.

在该情况下,源节点 S 实际支付的费用为 $\|\overline{LCP(S,D,\boldsymbol{d})}\|$,其效用 U_S 仍为 $m_S - \|\overline{LCP(S,D,\boldsymbol{d})}\|$,则 U_S 不依赖于其它报告的类型信息.

子情况 2:$m_f < \|\overline{LCP(S,D,\boldsymbol{d})}\| < m_S$.

在该情况下,源节点 S 以虚报信息 m_f 报告时,由于 $m_f < \|\overline{LCP(S,D,\boldsymbol{d})}\|$,产生过度支付问题,则整个算法将不执行,即 U_S 将为 0,则源节点 S 报告 m_S 为占优策略.

子情况 3:$m_f \leq m_S < \|\overline{LCP(S,D,\boldsymbol{d})}\|$.

很明显,源节点 S 虚报信息 m_f 时,由于支付的实际费用超过了预算,整个算法仍将不执行,即 U_S 将为 0. 很显然,源节点 S 报告 m_S 为占优策略.

当 $m_f > m_S$ 时:

证明过程与 $m_f \leq m_S$ 类似,同理也可以证明:源节点 S 仍以 m_S 报告时为占优策略,因此,源节点具有防策略性.

(2) 中间节点的防策略性

这里假定中间节点 i 的真实信息为 c_i,而虚报的信息为 d_i,则可能存在如下两种情况:

当 $c_i < d_i$(虚报的成本类型信息过高)时:

子情况 1:中间节点 i 不在 LCP 中,则节点 i 以报告 d_i 时,其效用 U_i 仍为 0.

子情况 2:中间节点 i 在 LCP 中时,如果节点 i 虚报的信息 d_i 过高,那么节点 i 很可能不能加入 LCP 中,则其效用 $U_i = 0$;如果节点 i 以不太高的 d_i 信息报告时,那么该节点仍在 LCP 中,但报告 d_i 对其效用不产生任何影响.

当 $c_i \geq d_i$(虚报的成本类型信息过低)时:

子情况 1:中间节点 i 在 LCP 中,通过报告信息 d_i,该节点仍在 LCP 中,对节点 i 的效用不产生任何影响,即不影响其支付费用.

子情况 2:中间节点 i 不在 LCP 中,通过报告 d_i,则此时网络从源节点 S 到目的节点 D 形成的新 LCP 包含该节点,其对应的效用在信息虚报前后变化过程如下:

节点 i 在虚报 d_i 信息之前(以 c_i 报告)时,该节点的效用情况为

对应效用的为:$U_i = \|LCP_{-i}(S,D,\boldsymbol{d})\| - \|LCP(S,D,\boldsymbol{d})\|$,很显然,$U_i \geq 0$.

而节点 i 以信息 d_i 报告时,节点 i 可能形成新的 LCP,则其支付更新为

$$Pay'_d(S,i,D) = \|LCP'_{-i}(S,D,\boldsymbol{d})\| - \|LCP'(S,D,\boldsymbol{d})\| + d_i.$$

很显然,原来节点 i 以 c_i 信息报告时的 $LCP(S,D,\boldsymbol{d})$,成为了当前网络拓扑下从 S 到 D 的所有路径中不包含中间节点 i 的最小成本路径 $LCP'_{-i}(S,D,\boldsymbol{d})$,而 $\|LCP'(S,D,\boldsymbol{d})\|$ 更新为 $\|LCP'(S,D,\boldsymbol{d})\| = \|LCP_{-i}(S,D,\boldsymbol{d})\| - c_i + d_i$.

通过化简,计算其支付为

$$Pay'_d(S,1,D) = \|LCP(S,D,\boldsymbol{d})\| - \|LCP_{-i}(S,D,\boldsymbol{d})\| + c_i.$$

对应实际获取的效用为 $U'_i = \|LCP(S,D,\boldsymbol{d})\| - \|LCP_{-i}(S,D,\boldsymbol{d})\|$. 由于 $U_i \geq 0$,所以 $U'_i \leq 0$,即通过虚报信息 d_i,节点 i 的效用比以 c_i 报告时的明显减少,而且变为负值.

综合以上情况分析,中间节点报告真实类型 c_i 是占优策略,即该节点具有防策略性.

(3) 目的节点的防策略性

本文中预先假定了目的节点 D 是可信的,能够真实地报告其类型信息,一般情况下,目的节点 D 不会与源节点 S 形成共谋来减少源节点 S 实际的支付,并且节点 D 也不会与中间节点形成共谋来为中间节点提供较高的支付. 因此,目的节点 D 也具有防策略性. 证毕.

定理 4(算法开销). ICTP 算法的开销降为 $O(|M|^2 d)$,其中 M 为除 S 和 D 之外所有中间节点的子集,d 为保证网络连通性时的最大连通度.

证明. 本文算法的执行条件要求从源节点 S 到目的节点 D 存在一条 LCP 路径,其成本要小于源节点 S 的最大支付 m_S. 很明显,$|M| \leq |V| - 2$,而算法的开销主要集中在集合 M 中所有节点的数据包转发. 由于数据包转发的开销至多为 $O(|M|d)$,因此该算法的整个通信开销大致为

$O(|M|^2 d)$. 相对于以 Ad-Hoc VCG 模型为基础的算法通信开销(见 4.3 节问题描述)来说,本文的 ICTP 算法开销大大降低.

定理 5(安全性分析). ICTP 算法能够抵抗网络中常见攻击的影响.

证明. 算法 ICTP 在基于对称密钥加密的基础上结合公钥加密的方式来实现安全保证. 针对当前网络中出现的常见攻击方式,本节对该算法的鲁棒性进行评估,对其安全性进行分析.

(1)节点运行算法的安全性

在网络初始化阶段,节点都需要通过 Diffie-Hellman 密钥交换协议与网络中的其它节点以及目的节点 D 分发各自的公钥和对称密钥信息,各个节点在本地缓存中保存对应节点的公钥信息以及合法的安全邻居列表,确保网络中的节点都是安全的、可信的,为下一步的路由建立过程提供认证,签名提供强有力的安全保证.

(2)发送伪装的路由请求数据包信息来攻击

由于每个节点都有一张邻居节点列表,节点在广播路由请求数据包 *RDP* 时,不能随意伪装成网络中的其它节点,只有下一跳节点的邻居节点存在伪装攻击的可能;否则,通过确认和监听方式很容易为下一跳节点发起伪装攻击提供可能. 此外,本文采用逐跳的方式进行上游节点的确认和签名,接收节点也进行安全确认回复,即使节点刻意对数据包进行伪装攻击,那么每个节点的私钥也是无法获得的,因此该攻击出现的可能性较小.

(3)更改路由信息攻击

在本算法的路由建立阶段,恶意节点有可能更改路由信息,但是其更改也只能修改先于该节点之前的路由信息,而不能更改该节点之后的路由信息,因为如果其在该节点之后任意加入节点信息,则下一跳节点可以对其进行监测和发现. 而当其更改前面的路由信息,当 *REP* 转发至被其更改的节点时,被更改节点和其邻居节点都很容易被监测,并将举报其篡改行为.

(4)黑洞和灰洞攻击

本文重点通过支付模型,让节点真实地报告其私有类型信息,来获取最大的支付,同时确保自身的效用最大化,即网络中的节点都较理性地去考虑自己的收益情况,此外本文的支付模型具有防策略性(见定理 3),节点不可能表现出自私性,不会出现故意丢弃需要转发的数据包的行为,因此避免网络中存在黑洞和灰洞攻击的可能性.

(5)重放攻击

由于 *RDP* 和 *REP* 数据包中都包含数据包的序列号和时间戳等重要信息,因而,整个的算法过程可以阻止重放攻击的发生.

(6)共谋攻击

本文的出发点是通过改进 Ad Hoc-VCG 支付模型来避免共谋攻击产生的可能,4.4.2 节给出了解决的方法,另外,本文在设计算法时也充分考虑了共谋小团体之间收益转移的可能性. 因此,首先通过对邻居节点的流量信息实行监控,然后利用加密、认证和鉴别的方式进一步避免共谋攻击的发生.

6 仿真与分析

6.1 仿真环境建立

本文为了进一步验证以上思路的有效性和正确性,采用 NS2[①] 仿真平台来实现上述算法 ICTP,并与其它经典的 3 种算法 Ad Hoc-VCG、COMMIT、LMOCP 进行对比.

本文的仿真环境基于 MAC 层使用 IEEE 802.11b 的 DCF,移动模型采用 Random Waypoint Model 模型以及将 CBR(Constant Bit Rate)作为传输流量模型,发送数据包的速率为 4 packet/s,每个数据包随机地从一个源节点发送到另一个随机目的节点,移动速度在 0 m/s~20 m/s 之间任意选择. 到达目的节点后,经过一个暂停时间(0 s~200 s)再开始新的转发过程. 仿真过程中建立的通信连接数为 30,一共生成 25 种随机拓扑,每种情况对应 5 种,最后的数据为 5 种拓扑所产生数据的平均值. 为了合理地计算出各个节点的私有类型转发成本信息,各个节点结合物理层的实际转发数据包的能量消耗,来计算其成本类型信息. 其它参数的设置见表 2 所示.

表 2 仿真参数设置

参数	意义	取值
$Area$	拓扑区域	1000 m×1000 m
N	网络节点数	50
r	节点传输半径	250 m
S	移动节点最大移动速度	20 m/s
P	分组长度大小	512 Bytes/packet
T	整个网络的仿真时间	200 s
M	网络中的自私节点数	1~30
m_s	源节点 S 的最大支付阈值	100
E	节点的初始化能量	100 J

① http://www.isi.edu/nsnam/ns/

本文的拓扑结构采用改进的 Waxman[37] 方法思想，它可以产生实际网络的拓扑结构. 具体 Waxman 方法的思想：网络中随机产生 N 个节点，并将这些节点均匀分布在一定拓扑区域中，任意两个节点 i 和 j 之间按一定的概率来建立连接，节点间连接概率的取值具体由节点间的欧氏距离来决定，从而将节点间链路的存在与否与节点间的距离关联起来. 而链路的产生概率可以通过如下公式来计算：

$$p(i,j)=\beta\exp\frac{-Dis(i,j)}{\alpha\times L_{\max}},$$

其中，$Dis(i,j)$ 表示节点 i 和 j 之间的欧氏距离；$L_{\max}$ 表示整个网络中任意节点间的最大距离；α 和 β 为对应的动态调整参数($0<\beta<1$，$0<\alpha<1$). 当 β 逐渐增大时，网络中边的密度也在增大，而 α 减少时，网路中节点间距离较小的边相对于距离较大的边的密度也在增大. 通过对参数 α 和 β 进行调整，使得距离较小的边存在的概率大于较长的边存在的概率，并且确保图中节点平均密度为 4～6. 基于此，在假定各个节点的传输半径(具体见表 2)是相同的前提条件下，根据节点间的距离与传输半径的对比情况，优化 Waxman 拓扑思想，再结合参数 α 和 β 的调整情况，使得彼此节点在各自传输半径范围内，确保距离较小的边对应的链路连通性较高，以及整个网络的密度适中，最终实现网络拓扑图的连通性. 通过实验得到的网络拓扑结构见图 4 所示. 图 4 中存在有 40 个正常的善意节点(Benevolent node)和 10 个自私节点(Selfish node). 对这 10 个自私节点攻击行为的模拟是通过计算其自身能量的消耗成本作为其自身的私有成本类型信息，而故意增加或减少其真实消耗成本值，从而向源节点报告其虚假私有信息.

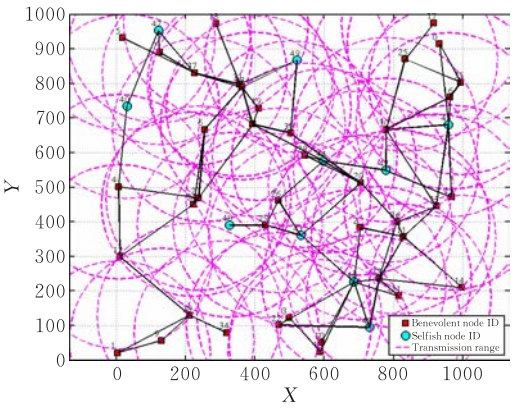


图 4　仿真过程中采用的网络拓扑图

6.2　性能对比参数

为了验证 ICTP 算法的性能，并与 Ad Hoc-VCG、COMMIT、LMOCP 算法进行实验对比，本文将围绕以下几个参数进行分析：

(1) 分组投递率，指目的节点收到的数据包数目与源节点发送的数据包数目的比值.

(2) 端到端的平均时延，包括路由查找时延、数据包在接口队列中的等待时延，传输时延及 MAC 层的重传时延，反映了路由有效性. 端到端的平均时延＝$\sum$(接收到数据包的时间-发送数据包的时间)/发送数据包的个数).

(3) 总体效用值，指 LCP 路径中所有中间节点的效用值之和. 此参数反映了节点效用值是否在采用本文的支付模型下取得最大.

(4) 归一化路由开销，指单位时间内路由控制包的传输量，它是网络拓扑结构变化率的函数. 路由开销＝转发的路由包个数和发送的路由包个数之和/目的端接收到的路由个数.

(5) 归一化支付费用，指全局替代路径中所有节点的总成本与 LCP 路径中的所有节点需要支付的总费用之差与 LCP 中所有节点需要支付的总费用的比例值.

(6) 总体成本开销，指 LCP 路径中所有中间节点的成本之和.

(7) 转发的数据包总数，指通过采取激励合作模型之后，网络中的中间节点所转发的数据包数目.

(8) 源节点的效用，指部分源节点的效用值，主要衡量源节点是否存在过度支付情况.

6.3　仿真结果分析

本文重点分析 ICTP 算法较其它 3 种算法在 6.2 节参数下的性能对比情况，特别在网络中自私节点所占比重，归一化支付费用，总体成本开销，总体效用值和源节点的效用等方面来衡量本文提出支付模型的有效性. 具体对比情况见图 5～13 所示.

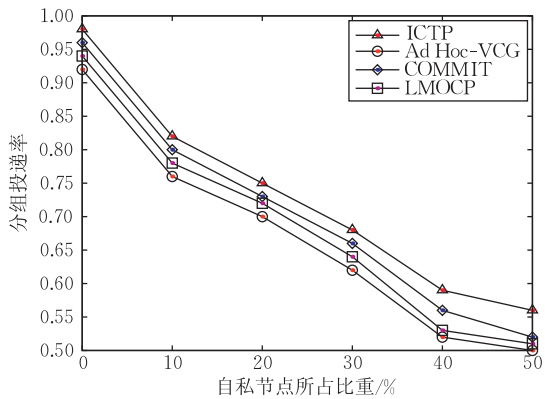


图 5　分组投递率和自私节点所占比重的对比

图 5 给出了 4 种算法的分组投递率在不同自私节点所占比重下的表现情况. 很明显，随着自私节点

逐渐增加,4 种算法的分组投递率都在逐步下降. 整体上来说,ICTP 的分组投递率较高,Ad Hoc-VCG 的最低,而 COMMIT 和 LMOCP 介于前面两者之间. 这其中的主要原因:随着自私节点的增加,网络中的分组投递率肯定是减少的,如何采取有效的激励措施是解决问题的关键,而这 4 种算法都是通过给予中间的转发节点一定的成本补偿来调动其合作的积极性,源节点支付的总费用是否足以维持这些自私节点,这 4 种算法则各有所长,而 ICTP 算法较其它 3 个算法的优势体现在:源节点的防策略性,避免过度支付策略以及路由建立过程和数据包转发阶段的双重激励合作. 而 COMMIT 和 LMOCP 都没有考虑源节点的防策略性,而 Ad Hoc-VCG 算法虽然考虑了源节点的防策略性,但是没有考虑过度支付的情况,因而,导致出现图 5 的所示结果.

图 6 给出了 4 种算法的端到端的平均时延在不同自私节点所占比重下的对比情况. 整体上来说,这 4 种算法的平均时延都随着自私节点数的增加而显著增加. Ad Hoc-VCG 的时延从 13 ms 增加到 20 ms,LMOCP 的时延从 12 ms 增加到 18 ms,COMMIT 则从 11 ms 增加到 16 ms,而 ICTP 算法则从 11 ms 增加到 14 ms. 直观上来看,Ad Hoc-VCG 的时延开销最高,LMOCP 次之,而 ICTP 和 COMMIT 的开销基本上差不多,达到了最小. 导致存在这种差异的原因在于:源节点为了激励数目逐渐增加的自私节点的转发积极性,不得不向其支付一定的费用,并且保证支付的费用都是自私节点通过真实报告其类型来获得的,整个的支付和报告类型过程,都是通过路由发现过程来实现,目的节点通过广播的请求包,收集网络的全局拓扑结构图,选择一条最小成本的 LCP,并向源节点进行回复确认,通知其需要支付的节点集合和相关费用. 因此,整个的预处理过程较长,数据包转发的时延开销较高,但是 Ad Hoc-VCG 和 LMOCP 相比来说,由于这两种算法的存在共谋和源节点不具有防策略性问题,导致其中间节点和源节点可能存在不真实的报告其类型的行为,影响其数据包的转发效率,从而导致这两种算法的时延开销最高,而 LMOCP 是对 Ad Hoc-VCG 控制开销的减少的改进,使其开销略低;ICTP 和 COMMIT 解决了源节点的防策略性问题,使 COMMIT 的时延开销较低,而 ICTP 也解决了共谋问题,使其真正实现了防策略性,进一步减少了网络通信的时延开销.

图 7 显示了在不同自私节点数所占比重下 4 种

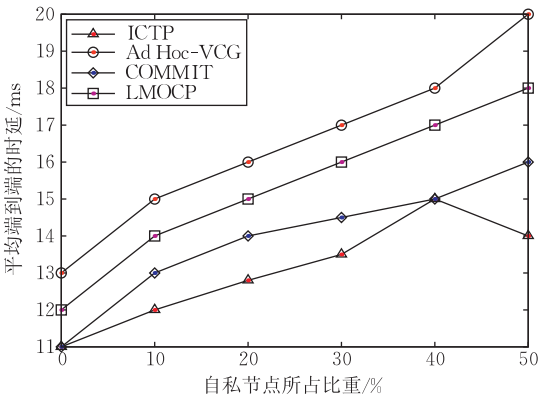


图 6 时延和自私节点所占比重的对比

算法的路由开销对比情况. 该图中也反映了这 4 种算法的通信开销情况. 从防策略性的特点中可以看出,网络的通信开销要在多项式时间内实现可计算性,也即整个通信过程包括防策略性的实施和支付费用都在保证通信开销维持的多项式时间内,网络的性能能够实现最优. 和图 6 的分析类似,时延开销的情况也从侧面反映了通信开销的情况. 这 4 种算法的路由开销随着自私节点数的增加而增加,COMMIT,LMOCP 和 ICTP 这 3 种算法基本上开销变化很小,而 Ad Hoc-VCG 的开销要高些. 存在这种对比差异的原因:Ad Hoc-VCG 明确指出其通信开销维持在 $O(n^3)$,其它 3 种算法的开销维持在都维持在 n^2 的数量级,而这 3 种算法的主要区别主要体现在 n^2 数量级的常数关系上. LMOCP 在路由发现过程中,对发送路由请求包进行改进,针对重复发送的情况进行抑制以及网路拓扑变化时,动态局部更新路由表的过程,而 COMMIT 和 ICTP 算法则包含了改进思想,但 ICTP 可能由于使用了对称加密和公钥认证的方式以及长度较小的确认数据包,来增强其确认的安全机制,略微增加了网络的通信开销.

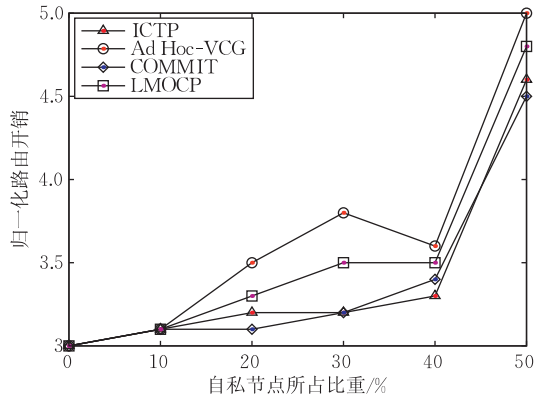


图 7 路由开销和自私节点所占比重的对比

接下来图 8～13 主要从源节点的支付情况、中间节点的总效用、总成本和转发数据包个数等情况以及共谋攻击的影响方面来进一步对性能进行分析. 图 8 给出了在不同自私节点所占比重下的 *LCP* 中的节点所获支付费用情况来进行对比. 实际上，图 8反映的是源节点预支付和实际支付的差额比值关系，进一步说明源节点的支付能力. 当网络中不存在自私节点时，*LCP* 中的节点都很自愿地转发数据包，而本文假定中间节点不存在支付的情况. 随着自私节点数的增加，源节点需要支付给自私节点的费用增加，因而导致 4 种算法的支付情况都在逐渐下降. 但是，ICTP 算法的归一化支付较高，说明中间节点所需支付的总费用相比其它 3 种算法不会出现中间节点虚报信息而支付费用过高的可能性，进一步验证了 ICTP 算法的防策略性. 不具有防策略性的 Ad Hoc-VCG 算法则其归一化支付费用过低，可能存在支付较高的可能性. 而 COMMIT 和 LMOCP 算法是基于 Ad Hoc-VCG 支付模型的改进，支付的有效性可能有了改进.

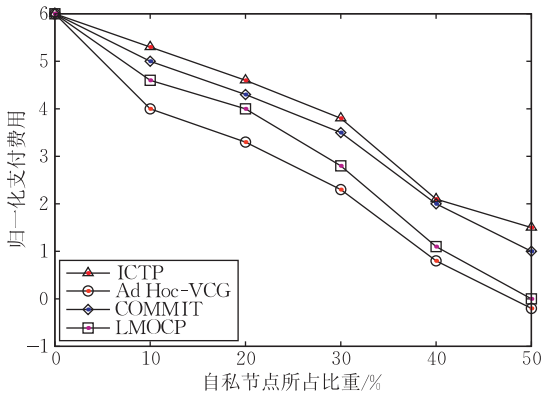


图 8　支付费用和自私节点所占比重的对比

图 9 则对应给出了 *LCP* 中间节点的总体效用值的变化情况. 该图反映了中间节点是否以真实的类型来报告其信息，如以真实的类型信息来报告，结合节点自身理性的特点，那么该节点获取的效用一定是最高的. 也从侧面验证 4 种算法的防策略性. 与图 8 的分析情况相同，很明显，ICTP 算法的总体效用值最高，Ad Hoc-VCG 算法的最低，COMMIT 和 LMOCP 算法的总体效用值则介于以上两种算法的效用值之间.

图 10 进一步验证了 4 种算法在存在自私节点情况下的总成本开销. 由于不在 *LCP* 中的节点其所获支付费用为 0，这些节点的成本开销主要体现在路由发现过程中的接收和转发路由请求包，实际的总体成本开销则指在路由发现过程和数据包转发过

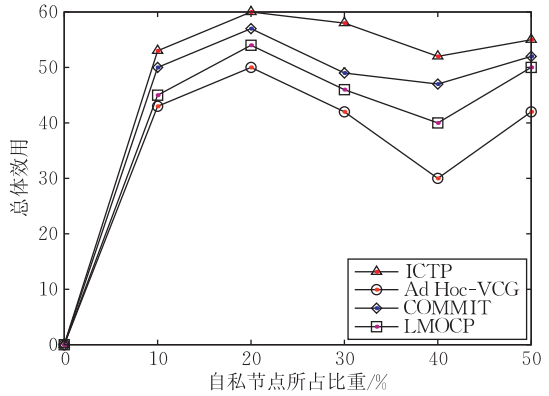


图 9　总体效用值和自私节点所占比重的对比

程的两个阶段下 *LCP* 中所有中间节点的成本之和. 很显然，其它 3 种算法都仅仅考虑路由发现过程中 *LCP* 中的节点转发成本，而忽略了数据包转发阶段的成本开销，也即该阶段下的自私激励过程，从而导致在路由发现过程中 *LCP* 中的节点具有合作转发的积极性，而在数据包转发阶段又表现出自私性. 较其它 3 种算法，ICTP 的成本开销较高，但是其在数据包转发阶段也对应获取到源节点给予积极合作的支付费用，起到一定的鼓励和补偿的目的.

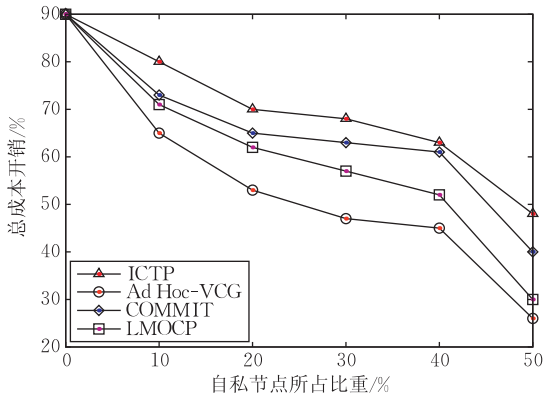


图 10　总体成本开销和自私节点所占比重的对比

另外一个衡量 *LCP* 中节点的合作积极性的度量指标为转发数据包的总个数. 图 11 给出了具体的对比情况. 整体上来说，随着自私节点的增加，4 种算法的数据包转发总数都在减少. 相对而言，Ad Hoc-VCG 算法的转发数据包总数递减的幅度最快，ICTP 算法递减的幅度较慢，COMMIT 和 LMOCP 算法的递减程度则介于上述两种算法之间. 存在这种情况的原因仍然是由于 Ad Hoc-VCG、COMMIT 和 LMOCP 的支付模型缺少防策略性以及没有考虑数据包转发过程的激励措施所导致. ICTP 算法针对以上 3 种算法的不足提出了改进方案，能够在自私节点较多的情况下，保持良好的性能.

为了验证本文提出的防共谋攻击的支付模型性

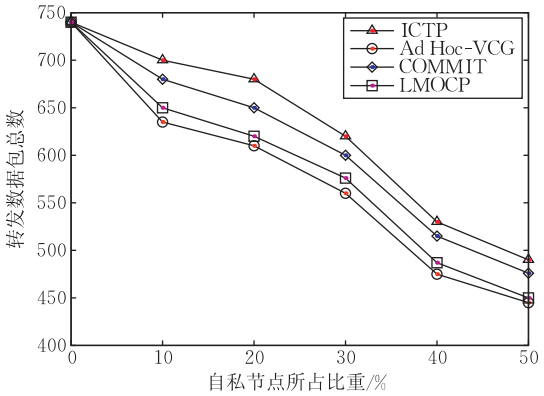


图 11 转发数据包总数和自私节点所占比重的对比

能情况，通过在一固定通信对的 *LCP* 中，随机选择至少两个中间节点来组成共谋小团体进行仿真测试. 图 12 给出了测试结果. 总效用值的计算思想和图 10 类似. 当共谋节点数逐渐从 4%增加到 8%时，4 种算法的总效用在逐渐增加；当从 8%增加到 16%，总效用都在逐渐减少；而从 16%增加到 20%时，则总效用又出现了逐渐递增的情况. 相对图 9 所示来说，图 12 中 ICTP 算法的总体效用不再像图 9 表现出最大的优势，而是略低于 COMMIT 算法，仍高于其他两种算法. 总体效用表现的最大，则并不意味着节点之间都以真实的类型信息来报告，很有可能是由于共谋小团体为了获取高额的支付，通过协商故意让小团体中的节点虚报节点的信息(具体见 4.4.2 节)，因此一部分节点获取了较高的支付，而部分节点为了小团体的利益，减少了其自身的支付，但是通过支付转移来平衡节点的支付. 本文的支付模型能够抵抗共谋攻击的影响，一方面通过加密和签名的方式防止支付转移的可能性发生，另一方面通过剔除共谋小团体来计算其各自的支付两种措施来进行保护，确保各个中间节点以真实类型信息进行报告. 基于此，可能导致 ICTP 算法的总体效用相对于 COMMIT 算法不能达到最大，略高于 LMOCP 和 Ad Hoc-VCG 算法的总体效用.

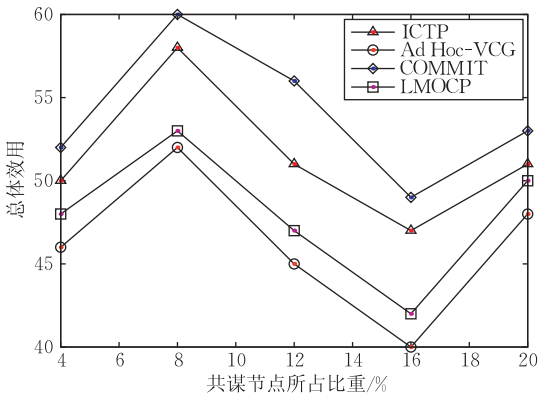


图 12 总体效用值和共谋节点所占比重的对比

本文通过对 30 个通信对的源节点效用值在不同仿真时间下进行观察分析. 图 13 给出了当网络中存在 16%的自私节点数时，在 30 个源节点中随机选择出 7 个节点的效用情况. 从图中可以看出：在 ICTP 算法的背景下，7 个节点的效用值都大于 0；而在使用其它 3 种算法的情况下，都存在或多或少的效用值小于 0 的情况. 此现象发生的主要原因是由于 ICTP 算法解决了源节点的防策略的问题，利用全局替代路径来取代其它 3 种算法中的除去对应源节点的最小成本路径，减小源节点的过度支付，最大化其自身的效用值. 因此，ICTP 算法的性能较其它 3 种算法略好一些.

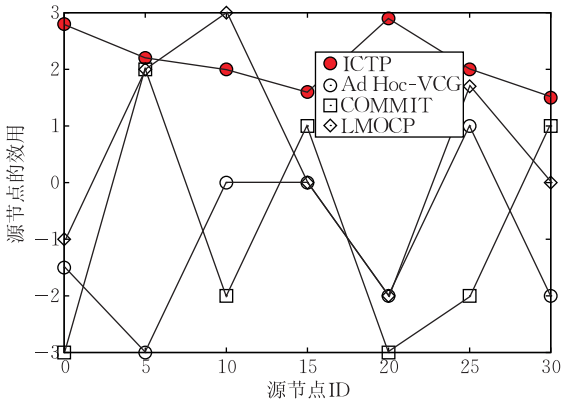


图 13 源节点的效用值在停留时间为 100 s 的情况

6.4 安全性能保证开销分析

为了进一步验证 ICTP 算法的安全保证开销情况，本文对签名、对称和公钥密钥加密方式进行了效率的对比与分析. 本文采用 Crypto++库[①]中的 RSA 加密算法和 DES 加解密算法，并在不同密钥安全等级下进行仿真实验，从而评估该算法的安全性能和计算开销. 各种算法均假定数据包长度为 128 Bytes 下进行测试，仿真的计算机硬件配置环境为 Intel Core 2.2 GHz. RSA 签名算法采用了 1024 bit 和 512 bit 两种等级的密钥，分别对应运算 10 000 次，DES 加密及解密算法采用 128 bit 和 256 bit 两种等级的密钥分别进行 1 000 000 次运算，以上密钥等级均具有当前较高的安全等级密钥强度. 通过对上次运算重复 5 次实验，得出：RSA 加解密算法在密钥为 1024 bit 时，加解密的时间开销基本上都在 355 s，而在密钥为 512 bit 时，加解密的时间开销维持在96 s～98 s 之间；而采用 DES 算法分别在密钥为 128 bit 和 256 bit 时，其对应的时间开销基本上维持在190 s～191 s 以及 198 s. 加解密算法的效率明显要远远高

① http://www.cryptopp.com/

于同安全等级的签名和验证算法，且平均每次运算相差的时间为两个数量级. 另外，一般情况下，对称加解密的运算速度比非对称加解密的运算要快，即该算法开销主要集中在 RSA 的算法上. 此外，在路由建立过程中，由于确认包的长度较小，所以路由建立阶段的大部分时间开销都花费在 RSA 算法中的加解密操作中. 当网络中广播的路由请求包 RDP 过多，以及需要经过的较多中间节点进行转发时，路由建立的开销可能更高. 因此，在未来的工作中，选择有效的签名方法来对本文的算法进行改进和扩展.

7　总结与展望

本文基于当前的研究热点——算法机制设计思想以及将此思想运用于 Ad Hoc 网络中的支付模型——Ad Hoc-VCG，提出了一种防策略和防共谋的支付模型，使自私节点报告其真实的私有类型成本信息来实现效用最大化，通过获取源节点提供的支付费用作为转发补偿，实现激励合作的目的，并进一步对路由发现过程和数据包转发过程一起考虑，加强数据包转发阶段节点转发的积极性，提出了对应的激励合作算法 ICTP. 最后，通过仿真实验验证其有效性，并与其它的经典算法（Ad Hoc-VCG，COMMIT 和 LMOCP）进行了性能对比. 仿真结果表明：ICTP 算法在性能上有了显著的提高. 在未来的工作中，将继续测试该协议的性能，以及将其延伸到多播路由协议和 QoS 保证中来进一步提高网络的性能.

参　考　文　献

[1] Zhang C, Zhou M C, Yu M. Ad hoc network security: A review. International Journal of Communication Systems, 2007, 20(8): 909-925

[2] Marti S, Giuli T J, Lai K, Baker M. Mitigating routing misbehavior in mobile ad hoc networks//Proceedings of the MOBICOM 2000. Boston, USA, 2000: 255-265

[3] Nisan N, Ronen A. Algorithmic mechanism design. Games and Economic Behavior, 2001, 35: 166-196

[4] Felegyhazi M, Hubaux J P, BuRyan L. Nash equilibria of packet forwarding strategies in wireless ad hoc net works. IEEE Transactions on Mobile Computing, 2006, 5(5): 463-476

[5] Altman E, Kherani A. Non-Cooperative forwarding in ad-hoc networks//Proceedings of the IFIP Networking 2005. Waterloo, Canada, 2005: 486-498

[6] Srinivasan V, Nuggehalli P. Cooperation in wireless ad hoc networks//Proceedings of the IEEE INFOCOM 2003. Washington, USA, 2003: 808-817

[7] Levin D. Punishment in selfish wireless networks: A game theoretic analysis//Proceedings of the ACM Workshop on the Economics of Networked Systems (NetEcon2006). Ann Arbor, Michigan, USA, 2006: 9-14

[8] Lu Yan, Stephen Hailes, Licia Capra. Analysis of packet relaying models and incentive strategies in wireless ad hoc networks with game theory//Proceedings of the International Conference on Advanced Information Networking and Applications. Gino-wan, Okinawa, Japan, 2008: 1062-1069

[9] Ng S K, Seah W K G. Game-theoretic approach for improving cooperation in wireless multihop networks. IEEE Transactions on System Man and Cybernetics Part B—Cybernetics, 2010, 40(3): 559-574

[10] Jaramillo J J, Srikant R. A game theory based reputation mechanism to incentivize cooperation in wireless ad hoc networks. Ad Hoc Networks, 2010, 8(4): 416-429

[11] Naserian M, Tepe K. Game theoretic approach in routing protocol for wireless ad hoc networks. Ad Hoc Networks, 2009, 7(3): 569-578

[12] Komathy K, Narayanasamy P. Trust-based evolutionary game model assisting AODV routing against selfishness. Journal of Network and Computer Applications, 2008, 31(4): 446-471

[13] Li Li, Dong Shu-Song, Wen Xiang-Ming. The analyses of cooperation mechanism based on game theory in wireless Ad hoc network. Journal of Electronics & Information Technology, 2007, 29(6): 1299-1303(in Chinese)
(李莉，董树松，温向明. 基于博弈理论建立无线自组网中激励合作机制的研究. 电子与信息学报，2007，29(6)：1299-1303)

[14] Wang Yang, Lin Chuang, Li Quan-Lin et al. Non-cooperative game based research on routing schemes for wireless networks. Chinese Journal of Computers, 2009, 32(1): 54-68 (in Chinese)
(汪洋，林闯，李泉林等. 基于非合作博弈的无线网络路由机制研究. 计算机学报，2009，32(1)：54-68)

[15] Huang Lei, Liu Li-Xiang. Study on cooperation stimulation mechanism in route discovery of Ad hoc networks. Chinese Journal of Computers, 2008, 31(2): 262-269(in Chinese)
(黄蕾，刘立祥. Ad hoc 网络寻路阶段的合作激励机制研究. 计算机学报，2008，31(2)：262-269)

[16] Lu Yin, Shi Jin, Xie Li. Repeated-Game modeling of cooperation enforcement in wireless ad hoc network. Journal of Software, 2008, 19(3): 755-768(in Chinese)
(陆音，石进，谢立. 基于重复博弈的无线自组网络协作增强模型. 软件学报，2008，19(3)：755-768)

[17] Mejia Marcela, Peña Nstor, Muñoz, Jose L et al. A game theoretic trust model for on-line distributed evolution of cooperation in MANETs. Journal of Network and Computer Applications, 2011, 34(1): 39-51

[18] Chen Ting-Ting, Wu Fan, Zhong Sheng. FITS: A finite-time reputation system for cooperation in wireless Ad hoc networks. IEEE Transactions on Computers, 2011, 60(7): 1045-1056

[19] Wang Bo, Huang Chuan-He et al. Cooperative forwarding model based on punishment mechanism in wireless Ad hoc networks. Computer Research and Development, 2011, 48(3): 398-406(in Chinese)
(王博, 黄传河等. Ad hoc 网络中基于惩罚机制的激励合作转发模型. 计算机研究与发展, 2011, 48(3): 398-406)

[20] Anderegg L, Eidenbenz S. Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents//Proceedings of the MobiCom03. San Diego, CA, USA, 2003: 245-259

[21] Mehdi Kargar, Mohammad Ghodsi. Truthful and secure routing in Ad hoc networks with malicious and selfish nodes. International Journal of Security and its Applications, 2009, 3(1): 117-128

[22] Wang Y, Singhal M. On improving the efficiency of truthful routing in manets with selfish nodes. Pervasive and Mobile Computing, 2007, 3(5): 537-559

[23] Zhong S, Li L, Liu Y, Yang Y. On designing incentive-compatible routing and forwarding protocols in wireless ad hoc networks — An integrated approach using game theoretical and cryptographic techniques//Proceedings of the MobiCom05. Cologne, Germany, 2005: 117-131

[24] Cai J, Pooch U. Play alone or together-truthful and efficient routing in wireless ad hoc networks with selfish nodes//Proceedings of the MASS04. Fort Lauderdale, United States, 2004: 457-465

[25] Eidenbenz S, Resta G, Santi P. The COMMIT protocol for truthful and cost-efficient routing in Ad hoc networks with selfish nodes. IEEE Transactions on Mobile Computing, 2008, 7(1): 19-33

[26] Guo Jian-Li, Wu Zhi-Bo, Dong Jian et al. A cooperation protocol for Ad hoc networks with selfish nodes based on mechanism design. Chinese Journal of Computers, 2009, 32(3): 483-492(in Chinese)
(郭建立, 吴智博, 董剑等. 基于机制设计理论的自组网节点合作协议. 计算机学报, 2009, 32(3): 483-492)

[27] Mahmoud M E, Shen X. PIS: A practical incentive system for multi-hop wireless networks. IEEE Transactions on Vehicular Technology, 2010, 59(8): 4012-4025

[28] Zhong S, Chen J, Yang Y R. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks//Proceedings of the IEEE Infocom'03. San Francisco, CA, USA, 2003: 1987-1997

[29] Mahmoud M, Shen X. ESIP: Secure incentive protocol with limited use of public-key cryptography for multi-hop wireless networks. IEEE Transactions on Mobile Computing, 2011, 10(7): 997-1010

[30] Mahmoud M, Shen X. Stimulating cooperation in multi-hop wireless networks using cheating detection system//Proceedings of the IEEE INFOCOM. San Diego, CA, 2010: 776-784

[31] Mahmoud M, Shen X. An integrated stimulation and punishment mechanism for thwarting packet dropping attack in multihop wireless networks. IEEE Transactions on Vehicular Technology, 2011, 60(8): 3947-3962

[32] Vickrey W. Counterspeculation, auctions and competitive sealed tenders. Journal of Finance, 1961, 16(1): 8-37

[33] Clarke E H. Multipart pricing of public goods. Public Choice, 1971, 11(1): 17-33

[34] Groves T. Incentives in teams. Econometrica, 1973, 41(4): 617-631

[35] Wang Wei-Zhao, Li Xiang-Yang. Low-cost routing in selfish and rational wireless ad hoc networks. IEEE Transactions on Mobile Computing, 2006, 5(5): 596-607

[36] Stinson D R. Cryptography: Theory and Practice. 3rd Edition. London: CRC Press, 1995

[37] Waxman Bernard M. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 1988, 6(9): 1617-1622

[38] Green J, Laffont J J. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. Econometrica, 1977, 45(2): 427-438

WANG Bo, born in 1982, Ph. D.. His main research interests include computer network and communication, network security.

HUANG Chuan-He, born in 1963, Ph. D., professor, Ph. D. supervisor. His main research interests include computer network, distributed parallel processing and network security.

Background

Mobile Ad hoc networks are the collection of wireless mobile hosts forming a temporary, self-organized network without the help of any centralized administration or standard support services. In such an environment, it may be necessary for one mobile host to enlist the aid of others in forwarding a packet to its destination, due to the limited propagation range of each mobile host's wireless transmission. It is not limited by the time and space to build quickly and convenient-

ly, so that a MANET is attractive for applications such as disaster relief, emergency operations, military service, maritime communications, vehicle networks, casual meetings, campus networks, robot networks, and so on. Similar to conventional fixed networks, security of the Ad hoc networks is considered from the attributes such as availability, confidentiality, integrity, authentication, non-repudiation, access control and usage control. But security approaches used for the fixed networks are not feasible due to the salient characteristics of Ad hoc networks. New security mechanisms are needed to adapt the special characteristics of Ad hoc networks.

Traditional MANET routing protocols assume that all nodes in the network work in a benevolent and cooperative manner between communication partners. However, selfish nodes will not spend their resources such as battery energy, CPU cycles, and bandwidth to relay others' packets without sufficient incentive. Therefore, it is necessary to design incentive mechanisms to improve the enthusiasm of cooperation among nodes. In this paper, we analyze the Ad Hoc-VCG model based on algorithmic mechanism design, point out its existing problems, propose a strategy-proof and collusion-resistant payment model and design an efficient secure incentive algorithm, which consists of two procedures: routing establishment and data packets forwarding.

The work was supported in part by National Natural Science Foundation of China (Project Nos. 60633020 and 60970117). The former foundation focuses on the research of theory and application in trusted mobile Internet, the latter foundation researches on the low delay data distribution and aggregation in the multi-radio multi-channel wireless sensor network. Our group has been working on the research of security for wireless Ad hoc network for many years, especially in routing security and trusted routing, many related research papers have been published in journals and conferences.

一种提高系统吞吐量的协助下载补偿模型

刘建航[1),2)] 毕经平[2)] 徐 鹏[2)] 边永超[2)] 李忠诚[2)]

[1)](中国石油大学(华东)计算机与通信工程学院 山东 青岛 266555)

[2)](中国科学院计算技术研究所 北京 100190)

摘 要 随着人们对互联网依赖性的日益提高,人们希望随时随地能够接入Internet,即使在行驶中的汽车里.相比于3G或4G网络,通过路边接入点(AP)接入互联网具有高带宽低延迟的特点.利用经过的车辆携带用户所需要的数据可以有效地减少由于AP分布稀疏所引起的间歇性连接的影响.然而高速行驶的节点,快速的拓扑变化以及传输碰撞域的重叠将导致协助车所携带的数据不能完全地传递给用户,从而降低了系统的吞吐量.文中在先前工作的基础上提出了一种提高系统吞吐量的协助下载补偿模型.在高速公路场景下,协助车在与同向和对向的其它车辆相遇时根据速度和位置信息预测其与目标车相遇的概率,并选取满足条件的车辆备份携带的数据以弥补因传输碰撞域重叠等因素所引起的部分数据包传送失败的损失,从而达到有效地利用盲区(Dark Area)延伸用户下载区域的目的.实验结果表明该方法显著地提高了下载的吞吐量,减少间歇性连接带来的影响.

关键词 协助下载;车联网;动态时槽;存储转发;DTN

中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2012.01390

A Compensation Model of Cooperative Downloading Improving System Throughput

LIU Jian-Hang[1),2)] BI Jing-Ping[2)] XU Peng[2)] BIAN Yong-Chao[2)] LI Zhong-Cheng[2)]

[1)](*College of Computer and Communication Engineering, China University of Petroleum, Qingdao, Shandong* 266555)

[2)](*Institute of Computing Technology, Chinese Academic of Sciences, Beijing* 100190)

Abstract Internet is playing a more and more important role in shaping our lives, which raises the demand for being connected anytime and anywhere, even while on driving cars. Compared with 3G or 4G networks, roadside APs have the advantages of low cost and high bandwidth. Making use of passing vehicles to carry the requested data for the client can reduce the influence of intermittent connectivity due to APs' sparse distribution. High node mobility, fast topology changes and collision domain overlay those cause the data carried by cooperative vehicles cannot be received totally. Based on previous work, this paper proposes a compensation model of cooperative downloading improving system throughput. In highway scenarios, according to the information of speed and position, a cooperative vehicle estimates the probability of other vehicles encountering users while meeting, and chooses adapted vehicles to backup partial data aiming at compensation for possible collisions. Simulation results indicate the benefits of the proposed scheme in terms of increasing throughput and reducing delay.

Keywords cooperative downloading; VANET; dynamic slot; carry and forword; delay tolerant networks

---

收稿日期:2011-12-26;最终修改稿收到日期:2012-05-09.本课题得到国家自然科学基金(60803138)、国家"九七三"重点基础研究发展规划项目前期研究专项(2011CB302505)资助.**刘建航**,男,1978年生,博士研究生,讲师,主要研究方向为下一代互联网. E-mail: liujianhang@ict.ac.cn. **毕经平**,女,1974年生,研究员,博士生导师,主要研究领域为下一代互联网和网络性能测试. **李忠诚**,男,1962年生,研究员,博士生导师,主要研究领域为计算机网络. **边永超**,男,1986年生,硕士研究生,主要研究方向为下一代互联网. **徐 鹏**,男,1990年生,硕士研究生,主要研究方向为下一代互联网.

1 引　言

随着对互联网依赖性的日益提高，人们希望能随时随地接入 Internet，即使在行驶的汽车里. 利用 3G 通信网络可以达到此目的，但是昂贵的费用和较差的服务质量成为制约这种接入方式的障碍. 近些年 WIFI 接入点(AP)的大规模部署为车载用户接入互联网提供了一条新的途径. 然而，相比于其它类型的移动互联网，车联网本身具有节点移动速度快、拓扑变化频繁的特点，其运行路线具有一定的可知性. 另一方面 AP 接入点覆盖范围有限，车辆运行在 AP 之间的 DA 盲区(Dark Area)时将与网络失去联系，从而导致间歇性连接. 若车载用户在一个 AP 区不能完成下载，只能等到下一次接入 AP 再进行下载任务，这种延迟在 AP 节点部署比较稀疏的高速公路场景中尤为严重. 其 AP 点通常设置在加油站或服务区内，距离分布较远，一般是在 8km～16km 以上. 使用 IEEE802.11a/b/g/p 标准，AP 的通信范围在 300m～1300m，其中 production phase 在 1km[1]，汽车使用 WLAN 接入 Internet，将有 7km～15km 与互联网失去联系，如果汽车行驶速度为 100km/h，则在 DA 中的运行时间为 5min～10min，这种延迟是很难让人接受的. 因此通过其它车辆携带用户所需数据来延伸 AP 下载区域的方法成为了近些年的研究热点.

我们先前的研究中，提出了 DSRelay[2]，即针对 AP 通信范围有限的问题提出了一套适应于高速公路场景下，通过 AP 接入点以及其它车辆进行协助下载的方法. 当移动用户在一个 AP 区内不能完成它的下载任务时，中心服务器计算用户和注册车辆在 AP 间的 DA 区中相遇时间和通信时长，在下一个 AP 通信区内选择一组协助车辆来携带用户所需数据. 每辆协助车在 DA 中不同时间段内将数据传给用户，从而达到利用 DA 延伸移动用户的下载区域，提高用户下载吞吐量的目的. 仿真结果显示，与 Greedy 选车算法比较，用户获取的数量提高了 40% 以上. 协助车在预定的时间将所携带的数据传递给目标车，该转发过程发生的区域我们称为传输碰撞域. 文献[2]中提出了根据协助车与目标车车速的加权平均值和注册时间划分传输碰撞域的方法. 然而研究中发现由车速变化所产生的碰撞域叠加仍然是一个影响系统的吞吐量的主要因素. 尽管高速公路场景下车速变化率不大，但仍然存在着影响车速的诸多因素，比如驾驶员的驾驶习惯，多变的路况条件等. DSRelay 规划了协助车在不同的时间和地点与目标车相遇，如果协助车都在预定的时间和地点与目标车相遇，系统将达到下载量的最大值. 然而由于车速变化等方面因素的影响，目标车和协助车相遇的时间和地点可能与预计的时间产生偏差. 因此协同车所携带的数据不可能百分之百地传递给用户. 根据仿真实验结果车速变化率在20%～30%，用户只能获取 75%～80%协助车所携带的数据. 更重要的是虽然 DSRelay 能够使目标车下载吞吐量提高 40%以上，但是如果目标车在下载数据文件时，即使 1%的传输损失也将使用户无法获取整个文件，用户不得不等到下个 AP 接入点再获取数据. 显然这种情况下吞吐量的提高并没有带来延迟的减少. 解决这个问题的一种方式是通过 AP 下载冗余的数据给协助车，让不同的协助车携带部分相同的数据包以弥补因为车速变化产生冲突的传输损失. 根据本文 3.1 节提出的式(1)可以推断出这种方式无疑会降低 AP 的有效传输率，在达到减少延迟的目的同时却降低了系统总的吞吐量.

根据高速公路的特点我们假设这样的场景：AP 的覆盖范围 $L=800$m，AP 间的 DA 区在 $D=8$km～16km，车辆之间的通信半径为 $r=200$m. 理论上 DA 中的碰撞域将有 $D/2r$ 个即(20 个～40 个)，如果 AP 的下载带宽 W_{AP} 是车间下载带宽 W_v 的 2 倍[1]，那么即使在 AP 区内直接下载率(直接把数据下载给请求用户所占用的时间比率)为 30%，那么 DSRelay 方法中 DA 的利用率为 7%(DA 区域为 8km，即 20 个碰撞域). 这意味着 DA 中大部分空间是空闲的. 如果使用 DA 区来完成数据的备份以解决上文提到的“用不同的协助车携带部分相同的数据包以弥补因为车速变化产生冲突的传输损失”，这样可以在不占用 AP 下载时间的前提下有效地降低用户获取完整信息的延迟.

通过以上的分析，本文在 DSRelay 方法的基础上提出了一个高速公路场景下的协助下载补偿模型. 该模型通过运行在 DA 中的与目标车同向和对向车辆，根据其运行速度和在 DA 中的位置计算与目标车相遇的概率，将协助车中携带的数据备份到能在 DA 区域内进入用户通信区的其它车辆上以弥补因为车速变化产生冲突的传输损失. 仿真结果表明该模型有效地利用 Dark Area 盲区，DA 中的下载量达到 AP 有效下载量的 99%以上，从根本上提高了系统吞吐量，降低了用户获取数据的延迟.

本文第 2 节介绍目前相关的研究工作;第 3 节详细阐述该模型的设计方案;第 4 节讨论模拟实验的结果,验证此模型的有效性;最后一节是结论和未来的工作.

2 相关工作

自从 Nandan 等人[3]提出 SPAWN 协议,第一次将协助下载(Cooperative downloading)的概念引入到车联网,越来越多的研究者投入到协助下载方面的研究中. SPAWN 协议与本文所做的工作不同,其针对 P2P 的应用场景,假定目标车辆和协助车辆都下载同一资源,再通过其提出的方法寻找资源进行下载;而本文要解决的问题是用户下载不同的数据,利用 AP 间的 DA 由 AP 选择车辆,提供协助下载服务. 相似的研究[4-5]重点集中在:P2P 汽车网络中在其它汽车里寻找所需要的资源进行协助下载.

Fiore 等人[6]提出的协助下载方法主要考虑城市下载场景,通过以往经过 AP 的车辆的速度和相关数据来预测车辆的行驶路线与目标车辆相遇的时刻,选定协助车辆下载协助数据. 其工作主要集中在 AP 的部署、不同协助车辆选择算法的评估以及数据分块方案. Trullols-Cruces 等人[7]提出了高速公路场景下的协助下载模式,主要讨论通过同向车辆协助传输解决无线下载过程中产生的丢包问题,以及通过对向车辆传输增加总的吞吐量. 但是文中没有对对向车辆协助下载进行具体的规划,也没有提到相邻 AP 点如何根据车辆的速度选择不同的车辆协助下载数据. MobTorrent[8]提出了使用 WIFI 和 WWAN 相互补充、利用同向以及对向车辆协助下载的方案,但是前提是需要所有车辆安装 GPS 导航,以及 WIFI 接口和 WWAN 接口,当目标车辆提出下载请求时,通过 WWAN 提前通知 AP 站点,AP 站点根据 GPS 的行驶路线信息将数据下载到对向或同向的车辆上. 文中没有明确地给出 AP 的选车算法和协助车辆携带的数据量,没有考虑由于车速变化产生的协助车重叠问题.

其它相关工作还有 Cabernet[9]通过优化建立连接的过程和传输层协议来提高传输性能,Balasubramanian 等人[10]提出了利用 AP 点分布的差异来减少冲突的一种支持车内用户使用交互式应用程序的方法. Zhang 等人[11]提出了汽车从路边基础设施上传和下载的方案. Mahajan 等人[12]提出了一种测量 WIFI 连通性的方法. DTcoop[13]解决的是单向的信息广播分发问题,通过协助下载的方式减少高速情况下丢包率较高的问题. 在文献[14]中作者提出了一种通过车间中继延伸 AP 通信范围的方法以便于车辆有更大的机会通过 AP 接入 Internet.

3 多任务协助下载模型的设计与实现

为了清楚地阐述本文提出的模型,本文假定 AP 通过有线网络可以直接与中心服务器通信. 每辆车都装有可用于车与 AP 通信和车车通信的 WIFI 接口. 中心服务器维护两个列表,一个是车辆列表,记录车辆进入 AP 通信区向其注册的 ID(首次进入系统的车辆由 AP 服务器分配 ID)、速度和注册时间等信息. 另一个是任务列表,记录着有下载请求的用户进入 AP 区后通过 AP 向中心服务器提出下载请求信息. 车辆运行在 DA 盲区中时每隔 1 s 要广播一次自己的 ID、速度、运行方向以及携带的数据信息. 根据该信息每辆车维护一个一跳内的邻居列表. 带有协助下载任务的协助车辆的广播信息中还要额外包括下载发生的时间和位置. 图 1 展示了该模型的基本结构.

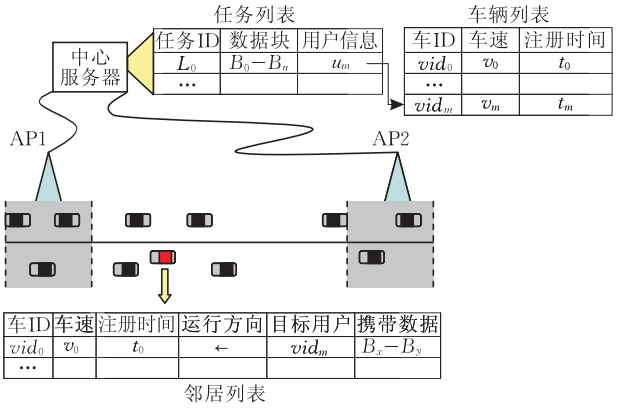


图 1 协助下载补偿模型

多用户情况下中心服务器需要维护一个任务列表而不是在文献[2]中讨论的单个任务项,选择协助车辆的时候要根据任务列表中任务的优先级确定什么时刻为哪辆车提供协助下载的服务. 任务优先级定义不是本文讨论的重点. 模型中要用到的参数见表 1.

表 1 参数表

参数	标签	参数	标签
AP 通信范围	L	DA 的长度	D
车辆通信半径	d	选择第 n 辆的时刻	T_n
AP 区内的下载带宽	W_{AP}	车间下载带宽	W_v
第 x 辆选中车辆的注册时间	t_x	第 x 辆选中车辆车速	v_x
第 n 个请求用户注册时间	t_s^n	第 n 个请求用户车速	v_s^n

3.1　模型运行机制

实际上用户在 DA 盲区中获取的所有数据都来自 AP，因此系统的吞吐量是由 AP 的有效下载量决定的，设包长为 pl，第 i 个包被目标车利用次数为 k_i，则 AP 有效带宽为

$$w_e=\sum_{i=0}^{n} pl\times k_i/s \tag{1}$$

其中 $n=W_{AP}/pl$，如果协助车辆携带的数据能够全部传递给目标车辆，那么系统理想吞吐量为

$$Ts=w_e\times \text{AP 工作时间} \tag{2}$$

因此影响系统总的吞吐量的因素除了包被利用的次数（多个用户下载相同的数据）以外，还有一个重要的因素就是协助车辆携带的数据是否能够百分百地传递给目标车辆. 引言中提到的通过 AP 下载给协助车冗余数据来弥补 DA 转发过程中的 20% 左右碰撞损失的方法，其代价是使部分数据包的 $k_i=0$，根据式(2)，将导致系统的吞吐量减少，因此利用 AP 下载冗余数据的方法不是一个理想的解决方案. 在保持 $k_i\geq 1$ 的前提下，我们将用冗余数据提高系统吞吐量的工作放在 DA 中来完成，对于 AP 通信区内任意协助车辆 c_x：(v_x,t_x)和中心服务器列表中当前区域的第 n 个下载请求的用户 u_n：(v_s^n,t_s^n)，c_x 和 u_n 相遇的开始时间 B_x^n 和地点 PB_x^n，以及通信结束的时间 E_x^n 和地点 PE_x^n 可以通过式(3)、(4)计算出.

$$B_x^n=\frac{D+2L-d+v_s^n t_s^n+v_x t_x}{v_s^n+v_x},\ PB_x^n=(B_x^n-t_s^n)\cdot v_s^n \tag{3}$$

$$E_x^n=\frac{D+2L+d+v_s^n t_s^n+v_x t_x}{v_s^n+v_x},\ PE_x^n=(E_x^n-t_s^n)\cdot v_s^n \tag{4}$$

根据高速公路环境的特点，一般情况下，相邻两个 AP 区中对向行驶的车辆一定会在两个 AP 通信范围之外的盲区相遇（两条高速线路相交时会出现车辆驶向第 3 个 AP 通信区，这种情况不在本文讨论范围之内）. 而同向行驶的车辆，存在距离和车速差，快车有超越慢车的可能. 根据这两个特点，该补偿模型是基于车辆相遇，超越和被超越的前提下，我们将其具体化以下 4 种情况.

(1) 因为运行在 DA 中的车辆每秒钟都要广播自己的 ID、车速和当前位置，所以每辆车都可以知道所在通信范围内其它车辆的速度和位置信息. 携带用户数据的协助车 A 在与目标车 D 相遇之前如果检测到与 D 同向的车辆 S 能够在 D 到达下个 AP 之前被 D 赶超（图 2(a)），那么 A 就将所携带的 D 所需的部分数据包传递给 S. 由于车辆间每秒交互一次信息，因此每辆车都维护一个邻居列表（见图 1）. 图 2(a)中，在 A 的邻居列表中车速 v_y 满足式(5)的协助车即为所选车辆 S_y.

$$\frac{L'+d}{v_y}-\frac{(B_x^n-T)(v_s^n+v_x)+L'+d}{v_s^n}\geq\frac{data}{w_v} \tag{5}$$

其中，T 为 A 检测到的 S_y 时间；L' 为 A 离开 AP 的距离；$data$ 是 A 所携带的数据长度. 这里规定 A 给 S 传递数据包的顺序是按逆序传递，A 给 D 传输数据包按顺序传递，这样保证 D 接收数据的最大化. 如果 A 在与 D 相遇时传递数据给 D 的过程中，由于速度变化的原因与另一组下载过程 A'-D' 产生碰撞（图 2(b)），A 所携带的数据就不能全部传递给 D. 那么当 D 在赶超 S 的时候（图 2(c)），由 S 将剩余部分数据传递给 D 以弥补碰撞带来的损失. 该协助下载过程简称为 USP(User Surpass Process).

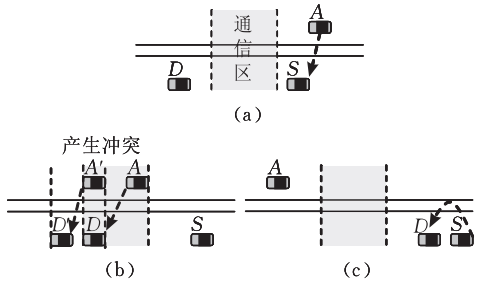


图 2　USP 协助下载过程

(2) 如果 A 在与 D 相遇时传递数据给 D 的过程中，由于速度变化的原因与另一组下载过程 A'-D' 产生碰撞（图 3(a)），那么 A 所携带的数据就不能全部传递给 D. 当 A 离开 D 的通信区后，检测到与 D 同向行驶的车辆 F，根据 F 提供的速度和位置信息，如果 F 能够在到达下个 AP 之前赶超 D，A 将携带的未能传递给 D 的数据包传递给 F（图 3(b)）. 当 F 赶超 D 时（图 3(c)），将其携带的数据包传递给 D 以弥补碰撞带来的损失. 图 3(c)中在 A 的邻居列表中车速 v_y 满足式(6)的协助车即为所选车辆 F_y. 该过程简称为 SUP(Surpass User Process).

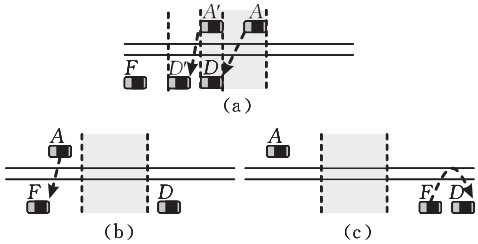


图 3　SUP 协助下载过程

$$\frac{(B_x^n-T)(v_s^n+v_x)+L'+d}{v_s^n}-\frac{L'+d}{v_y}\geq\frac{data}{w_v} \quad (6)$$

(3)上文提到带有协助下载任务的协助车辆的广播信息中包括下载发生的时间和位置,下载的时间和位置是根据协助车和用户车速和注册时间等信息确定的[2].因此当 A 在与 D 相遇之前,如果检测到 A' 发生下载的区域与自己的相叠加,这表明发生碰撞的概率非常大,这时 A 将随机找到一个与目标车同向的车辆 R,将可能由于冲突而未能传输的数据包下载到 R 上(图 4(a)).其发生的条件满足式(7).

$$(B_{A'}^n\leq B_A^m\leq E_{A'}^n)\bigcup(B_{A'}^n\leq E_A^m\leq E_{A'}^n)\bigcup$$
$$(PB_{A'}^n\leq PB_A^m\leq PE_{A'}^n)\bigcup(PB_{A'}^n\leq PE_A^m\leq PE_{A'}^n) \quad (7)$$

需要注意的是 R 是一辆与 D 同向的随机车辆,该车辆不一定会在当前 DA 区内被 D 赶超.如果其不能被 D 赶超,R 在离开 A 的传输区域后遇到与 A 同向的车辆 C,将所携带的 D 所需数据传至 C 上(图 4(b)).当 D 在与 C 相遇时下载未能获取的相应的数据(图 4(c)).该过程简称为 CAP(Collision Avoided Process).如果 R 可以在该 DA 区内被 D 赶超就不必将数据传至 C,而是执行 USP 过程.

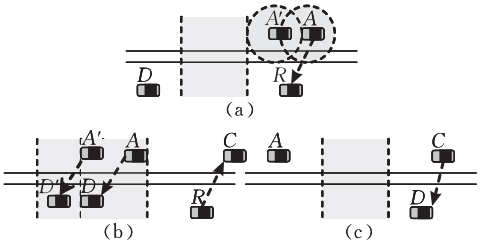


图 4 CAP 协助下载过程

(4)以上 3 种情况讨论的是当协助车 A 为单个任务服务,也就是说 A 的数据是被一辆目标车下载.然而很多情况许多用户会请求相同的数据,比如下载热点视频.在这种情况下,如果 AP 将多个任务所需要的相同数据包下载到一辆用户车,那么意味着式(1)中的 $k_i>1$,这将提高系统的吞吐量.事实上这种方案是可行的,因为 AP 任务列表中的任务所对应的用户车一定是正在向当前 AP 行驶的车辆,这样对向行驶的协助车 A 一定会在 DA 区域中与这些用户车$\{D_n\}$相遇.然而 AP 不能预测 A 能够在指定的时间和地点将所携带的数据传递给$\{D_n\}$中的所有车辆.根据文献[2]中提出的算法,AP 可以至少预测 A 与其中一辆用户车 D_1 在无冲突的情况下完成下载,如(图 5(a)).在第一辆用户车 D_1 收到数据后,因为不确定 A 是否能将数据包完整的传递给 D_2,D_1 在离开 A 的通信区域后将数据转发给与 A 同向的车辆 C(图 5(b)).如果在 D_2 与 A 相遇过程中与其它下载过程产生冲突,那么在与 C 相遇时将可以下载剩余的数据包(图 5(c)).该过程被简称为 MTBP(Multi-Task Backup Process).

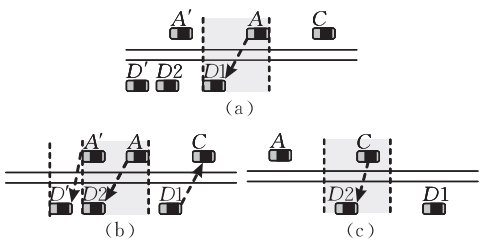


图 5 MTBP 协助下载过程

3.2 数据流图

根据上述 4 种协助下载过程中数据在协助车中的流动方向,我们把其抽象成数据流图的形式如图 6 所示.图中的椭圆中的字母表示不同协助车所处的角色,单个字母表示单个车辆,花括号表示符合条件的车辆的集合.字母上方的箭头表示车辆运行的方向.线是数据流动的方向.实线表示只要符合条件就一定发生的事件,虚线则表示如果协助车已经将所携带的数据全部转发给目标车了,则即使遇到满足条件的车辆也不会发生的事件.下面从几个方面介绍该模型.

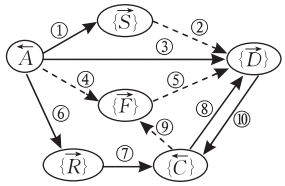


图 6 协助下载数据流图

(1)角色

A 表示从 AP 获取原始数据的协助车.

$\{D\}$表示请求 A 所携带数据的用户的集合,由于 A 所携带的数据可能是多个用户共同请求的数据,所以用户可能多于一个,这里用集合$\{D\}$表示.

$\{S\}$表示在当前 DA 中,A 在与$\{D\}$相遇前所遇到的可以被$\{D\}$赶超的车辆的集合.

$\{F\}$表示在当前 DA 中,A 在与$\{D\}$相遇后所遇到的可以赶超$\{D\}$的车辆的集合.

$\{R\}$是 A 在与$\{D\}$相遇前所遇到的与$\{D\}$同向行驶的任意车辆的集合.很明显$\{S\}$是$\{R\}$的一个子集.

$\{C\}$表示在当前 DA 中,行驶在 A 后面的任意同向车辆集合.

显然 A 和 C 是同向的，因此 A 和 C 上方的箭头方向相同. 它们与 S，D，F，R 是对向行驶的.

(2) 数据流向

图中线上的数字表示数据流向过程.

过程①表示当 A 遇到可被用户{D}赶超的车辆 S_n，将数据在不影响其它目标车下载数据的前提下，按所携带数据包倒序传送给 S_n.

过程②表示当在{D}中的车辆赶超集合{S}中的车辆的过程中，下载 S 中所携带的并且未能在 A 中获取的数据.

过程③表示 A 与{D}相遇时，将所携带的{D}需要的数据转发给的{D}过程.

过程④表示如果 A 在与{D}相遇过程中未能将数据传递给用户，在离开{D}通信区后找到可以在当前 DA 中赶超{D}的车辆在{F}中，将未转发的数据传递给{F}.

过程⑤表示当集合{F}中的车辆赶超集合{D}中的车辆时，将从 A 下载的{D}所需数据转发给{D}.

过程⑥表示当 A 检测到以当前的速度行驶会在与{D}相遇过程中与其它下载过程发生冲突，提前将部分数据传递给最先接触的对行车辆{R}.

过程⑦表示如果集合{R}中的车辆携带的数据所对应的{D}不能在当前 DA 中赶超自己，该车就将从 A 下载到的数据传递给与 A 同向的车辆{C}，因为过程⑦发生在过程⑥之后，所以{C}中的车辆一定晚于 A 与{D}相遇.

过程⑧表示当集合{C}中车辆与{D}相遇时将从{R}下载到的{D}所需数据传递给{D}.

过程⑨表示如果集合{C}中车辆所携带的数据未能全部传递给{D}，但在离开{D}的通信区后遇到能够在当前 DA 中赶超{D}的车辆{F}，{C}将未完成数据传递给{F}.

过程⑩表示因为由于预测到与当前用户车 D_n 请求相同数据的后续用户车 D_{n+k} 可能不能完全下载数据，D_n 将数据传递给对行的车辆{C}，以便{C}中车辆在与 D_{n+k} 相遇时完成数据转发.

(3) 传输优先级

传输过程中，当不同的下载请求传输中出现碰撞域叠加，根据各过程的优先级确定传输顺序.

②③⑤⑧过程中，都是协助车向用户转发数据，因此优先级最高.

④⑨发生在已经确认目标车未下载所有数据，处于第二优先级.

⑥⑦发生在协助车已经预测到会发生碰撞，需要进行备份措施. 相似的，⑩也是因为由于预测到与 D_n 请求相同数据的 D_{n+k} 可能不能完全下载数据的一种数据备份方式. 因此都处于第三优先级.

①的优先级最低，因为其发生在并不确定是否发生碰撞的一种预防措施.

4　性能评价

本文通过仿真实验对该模型的性能进行评估. 实验场景：高速公路的 AP 点通信范围设为 800 m[15]，两个 AP 间相距 8 km，符合高速公路上 AP 点设置在加油站或服务区的实际情况. 车辆的通信半径设为 250 m. 协助下载过程中取 1 s 作为用户与协助车辆建立连接所用的时间(信道争用与接入方法作为未来的研究工作). AP 区下载速率设为 150 KB/s，同向车辆协助下载速率设为 200 KB/s，对向车辆协助下载速率设为 50 KB/s[15]. 车速在 90 km/h 到 150 km/h 之间随机产生，在车速变化上，假定车速变化的概率为 p，变化的范围在 90 km/h～150 km/h，并且符合对数正态分布[16]，假定用户车速为 100 km/h.

为了验证上文提出的 4 种协助下载过程的可行性，我们先测试在车速变化率在 30%、车流密度 $\lambda=10$ 的情况下，在距离 AP 3 km，4 km，5 km，具有不同车速的用户能够在 DA 区中超越前车和被前车超越的概率. 图 7 中 3 条实线分别表示用户车距离 AP 5 km，4 km，3 km 时超越前车的可能性. 即 USP 过程，3 条虚线表示用户的 SUP 过程概率. 从图 7 的测试结果可以看出，在 USP 过程中，当用户车速在 90 km/h 的情况下，用户车在距离 AP 的 3 个测试点的超越率几乎为 0. 距离 AP 5 km 时，当车速上升到 100 km/h，超越率为 80%，当车速超过 110 km/h 时，用户能够超越前车的概率就达到 100%. 距离 AP 3 km 时，用户车速达到 120 km/h 才会一定超越前车. SUP 过程正相反，用户车速在低于 120 km/h 时，用户一定会被同向行驶的车辆超越. 在距离 AP 3 km，用户车速在 140 km/h 时，基本上不会有车超越用户车. 测试结果表明，无论用户车速为多少，USP 过程和 SUP 过程中至少有一个过程会发生的概率是 100%. 这就证明了本文提出模型的可行性. 图 8 中比较了无协助下载时，使用贪心选车法(每次中心服务器选择 AP 通信区域内最先与用户相遇的车辆作为协助车辆)，随机选车法和我们之前提出的 DSRelay 协助下载方法以及本文在 DSRelay 提出的协助下载补偿模型，这 5 种下载方法在两个 AP 区以及之间的盲区的下载数据量.

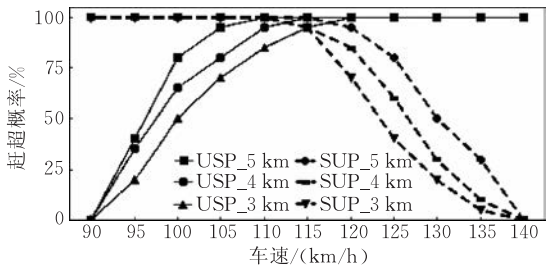


图 7 不同车速下的 USP 和 SUP 概率

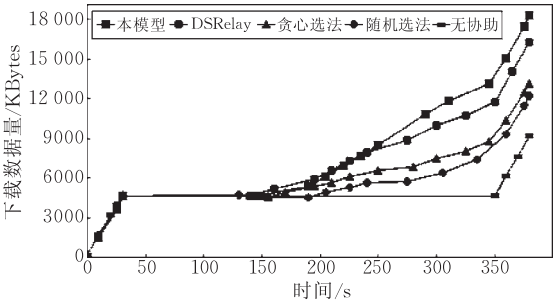


图 8 5 种下载方法比较

如图所示,开始的 30 s 左右的时间,用户车辆进入 AP 区,提出下载请求后直接在 AP 区下载数据,下载速率设为 150 KB/s,用户能获得约 4.5 MB 的数据后即离开 AP 通信区.用户提出下载请求后下中心服务器根据选车算法在下一个 AP 区内选择协助车辆携带用户不能下载完成的数据.在 30 s～140 s 的时间,用户车还没有遇到从下个 AP 对向行驶的并携带其所需数据的车辆,所以这段时间下载总量一直保持在 4.5 MB.大约 140 s 后用户开始遇到对向协助车辆并下载数据.由于随机选车法和贪心选车法没有规划协助车与用户的相遇时间,所以会频繁出现车辆的碰撞域重叠,因此在这个区间(两个 AP 区和一个 DA 区)中用户只能下载到约 12 MB 的数据.相比之下使用 DSRelay 规划了 DA 中的下载区间,减少了碰撞重叠的时间和区域,下载量提高到 16 MB.然而即便是进行了规划,由于车速的变换,碰撞域叠加是不可避免的.因此仅使用 DSRelay 方法利用 DA 区完成协助下载并未达到下载的最大值.使用本文提出的补偿模型解决了由于车速变化而产生的碰撞域叠加的问题,实验结果显示下载总量可以达到 18 MB 以上,基本接近使用对向车辆协助下载方法中的理想值.图 8 中 250 s 之前使用补偿模型下载量会稍有下降,这是由于上文提出的 4 种过程对信道的争用频繁所产生一些影响.250 s 后下载量明显提高,这是由于 USP 和 SUP 以及 CAP 过程都是发生在 DA 的后半段.

图 9 显示了使用补偿模型在 DA 区中各个过程的下载量.该实验验证了不同车速下直接下载,USP、SUP 和 SUP 过程用户获取的数据比例.车速在 100 km/h 时,用户赶超概率比较低,所以 USP 过程平均仅能获取 0.3 MB 的数据,而被赶超的概率较高,SUP 能获取 1.4 MB 的数据,CAP 获取的数据为 0.9 MB.相比之下,车速在 115 km/h 时各个过程获取的数据量相差不多.车速高达 130 km/h,能够被赶超的概率很低,SUP 过程只能获取 0.2 MB 数据.用户获取的数据总量随车速提高而减少,原因是车速越快在 DA 内停留的时间越短.从测试结果中可以算出,3 个补偿下载过程所获取的数据量占用户获取数据总量的 20%左右,因此可以得出这种补偿模型可以将数据下载总量提高 20%以上.

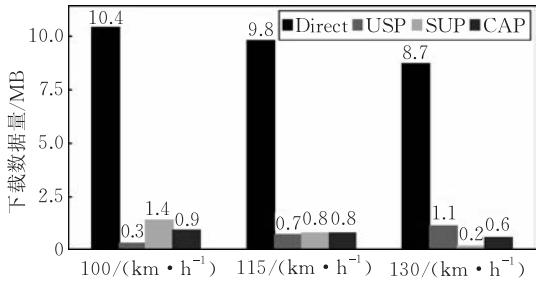


图 9 不同车速下用户在各过程中获取的数据量

为了比较本文提出的补偿模型在下载实际数据文件时的优势,我们使用了在文献[2]研究中使用的 3 个不同大小的测试文件.表 2 统计了测试结果.可以明显地看出,相比较 DSRelay,使用补偿模型,平均带宽得到进一步提高.尤其在下载比较大的文件时,比如一个时长为 120 min,大小为 379.8 MB 的 rmvb 电影文件,用户只需要约 1 小时 45 分钟即可获取所有数据,平均带宽达到 60 KB/s,下载时间要低于视频时长.因此该模型在高速公路场景下的视频观看方面具有一定的应用前景.

表 2 下载不同大小文件的延迟(*D*)和平均带宽比较(*AB*)

文件长度/MB	未协助下载的结果		DSRelay 的结果		补偿模型的结果	
	D/s	*AB*/(KB·s^{-1})	*D*/s	AB/(KB·s^{-1})	*D*/s	AB/(KB·s^{-1})
27.0	1773	15.2	683	39.5	569	47.4
55.3	3880	14.2	1284	42.8	1030	53.7
379.8	27812	13.7	7908	48.0	6314	60.1

5 结束语

针对高速公路场景下车速变化产生碰撞域叠加的问题，本文在前期工作 DSRelay 方法的基础上提出了利用行驶在 DA 中无任务的车辆去备份协助车携带数据的一种数据补偿模型. 通过该模型，协助车可以预测可能发生或是已经发生的碰撞域重叠. 将在与用户相遇时未能传递给用户的数据提前或者延后备份到可以在 DA 区内的其它时间和地点与用户相遇的其它车辆上. 从而减少了由于碰撞域重叠产生的冲突对 DSRelay 协助下载方法的影响，有效地利用了 DA 区，从根本上提高了系统的整体吞吐量.

下一步的研究工作包括研究符合高速公路特点的接入控制方法，主要工作是信道的分配和接入方法. 另一方面是激励机制和信息安全研究，通过完善的经济模型鼓励更多的车辆参与到该模型中.

参 考 文 献

[1] Ott J, Kutscher D. A Disconnection-tolerant transport for drive-thru Internet environments//Proceedings of the IEEE INFOCOM. Miami, USA, 2005, 3: 1849-1862

[2] Liu Jian-Hang, Sun Jiang-Ming, Bi Jing-Ping et al. VANET cooperative downloading approach study based on dynamic slot. Chinese Journal of Computers, 2011, 34(8): 1378-1386(in Chinese)
(刘建航，孙江明，毕经平等. 基于动态时槽的车联网协助下载方法研究. 计算机学报，2011，34(8)：1378-1386)

[3] Nandan A, Das S, Pau G, Sanadidi M, Gerla M. Cooperative downloading in vehicular adhoc wireless networks//Proceedings of the International Conference on Wireless on demand Network Systems and Services. St. Moritz, Switzerland, 2005: 32-41

[4] Liu Che-Liang, Wang Chih-Yu, Wei Hung-Yu. Mobile chord: Enhancing P2P application performance over vehicular Ad Hoc network//Proceedings of the IEEE GLOBECOM. New Orleans, USA, 2008: 241-247

[5] Lin Sung-Han, Hu Junn-Yen, Chou Cheng-Fu, Chang Ing-Chau, Hung Chien-Chun. A novel social cluster-based P2P framework for integrating VANETs with the Internet//Proceedings of the IEEE WCNC. Budapest Hangury, 2009: 1-6

[6] Fiore M, Barcelo-Ordinas J M. Cooperative download in urban vehicular networks//Proceedings of the IEEE MASS. Lyon, France, 2009: 20-29

[7] Trullols-Cruces O, Morillo-Pozo J, Barcelo Jose M, Garcia-Vidal J. A cooperative vehicular network framework//Proceedings of the IEEE ICC. Dresden, Germany, 2009

[8] Chen Bin Bin, Chan Mun Choon. MobTorrent: A framework for mobile Internet access from vehicles//Proceedings of the IEEE INFOCOM. Barcelona, Spain, 2009

[9] Eriksson J, Balakrishnan H, Madden S. Cabernet: Vehicular content delivery using WiFi//Proceedings of the MobiCom. San Francisco, CA, USA, 2008: 199-210

[10] Balasubramanian A, Mahajan R, Venkataramani A, Levine B, Zahorjan J. Interactive WiFi connectivity for moving vehicles//Proceedings of the SIGCOMM. Seattle, WA, USA, 2008: 427-438

[11] Zhang Y, Zhao J, Cao G. On scheduling vehicle-roadside data access//Proceedings of the VANET. Montreal, QC, Canada, 2007: 10-19

[12] Mahajan R, Zahorjan J, Zill B. Understanding WiFi-based connectivity from moving vehicles//Proceedings of the IMC. San Diego, CA, USA, 2007

[13] Liang Hao, Zhuang Weihua. DTcoop: Delay tolerant cooperative communications in DTN/WLAN integrated networks//Proceedings of the Vehicular Technology Conference Fall (VTC 2010-Fall). Taipei, Tanwan, China, 2010

[14] Zhao Jing, Arnold Todd, Zhang Yang, Cao Guohong. Extending drive-thru data access by vehicle-to-vehicle relay//Proceedings of the VANET'08. San Francisco, California, USA, 2008: 66-75

[15] Ott Jorg, Kutscher Dirk. Drive-thru Internet: IEEE 802.11b for "Automobile" users//Proceedings of the IEEE INFOCOM. Hong Kong, China, 2004: 362-373

[16] Gerlough D L. Poisson and traffic: Use of Poisson distribution in highway traffic. Eno Foundation for Highway Traffic Control, 1955

LIU Jian-Hang, born in 1978, Ph. D. candidate, lecturer. His research interest is next generation Internet.

BI Jing-Ping, born in 1974, Ph. D., professor, Ph. D. supervisor. Her research interest is next generation Internet and network test.

LI Zhong-Cheng, born in 1962, Ph. D., professor. His research interest is computer networks.

BIAN Yong-Chao, born in 1986, M. S. candidate. His research interest is next generation Internet.

XU Peng, born in 1990, M. S. candidate. Her research interest is next generation Internet.

Background

This paper focuses on the research of compensation model of cooperative downloading on highway scenarios against intermittent connectivity. Cooperative downloading in vehicular networks is first introduced by Nandan et al. as a part of the protocol SPAWN for cooperative content retrieval and sharing among users aboard vehicles in 2005. From then on, researchers propose successively some schemes of cooperative downloading to help the users access the Internet via APs, which include MobTorrent, Cabernet, VADD and so on. However, these schemes do not give a definite solution when cooperative vehicles cannot forward all data to users. Making use of passing vehicles to carry the requested data for the client can reduce the influence of intermittent connectivity due to APs' sparse distribution. High node mobility, fast topology changes and collision domain overlay those cause the data carried by cooperative vehicles cannot be received totally. Based on previous work, this paper proposes a compensation model of cooperative downloading which cooperative vehicles estimates the probability of other vehicles encountering users while meeting, and chooses adapted vehicles to backup partial data aiming at compensation for possible collisions.

This research work is supported by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302505 and the National Natural Science Foundation of China under Grant No. 60803138.

基于联合信道特征的中继物理层安全传输机制

李翔宇 金 梁 黄开枝 吉 江
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要 针对中继节点不可信的问题,提出了一种基于联合信道特征的中继物理层安全传输机制.首先将中继前后的两个信道等效合并为一个,得到联合信道特征.然后在联合信道特征的零空间中,增加人工噪声,使参与转发的中继节点无法获得有效信息量.仿真结果表明:当增加的人工噪声处于合法接收者信道特征的零空间时,可以提高协作中继系统的保密容量,合法接收者的误码率要远低于不可信中继;保密容量随中继数量增加的变化趋势与窃听者的分布相关.

关键词 物理层安全;协作中继系统;保密容量;人工噪声
中图法分类号 TN918 **DOI号**: 10.3724/SP.J.1016.2012.01399

A Physical Layer Security Transmission Mechanism of Relay System Based on Joint Channel Characteristics

LI Xiang-Yu JIN Liang HUANG Kai-Zhi JI Jiang
(*National Digital Switching System Engineering & Technological Research Center, Zhengzhou* 450002)

Abstract Motivated by the problem of untrusted relays, this paper proposes a physical layer security transmission mechanism of relay system based on joint channel characteristics. First of all, we combine the two channels before and after the relay into one, and get the joint channel characteristics. Then in the null space of the channel characteristics, the artificial noise is added to make the internal wire-tappers get none of the information. Simulation results show that when the artificial noise lies in the null space of the legitimate receiver's channel, secrecy capacity of the cooperative relay system can be improved and the bit-error ratio of the legitimate receiver is much smaller than untrusted relays. The trend of the secrecy capacity, changed by increasing relay number, relates to the distribution of eavesdroppers.

Keywords physical-layer security; cooperative relay system; secrecy capacity; artificial noise

1 引 言

近年来,协作中继系统已成为无线网络通信技术的研究热点之一.在4G通信LTE-Advanced系统中[1],通过基站(eNodeB)和中继站(Relay Station,RS)之间的协作,可以提高小区边缘用户的通话质量,增加小区的覆盖范围和系统吞吐率,减少传输所消耗的功率,并降低相邻小区之间的干扰.但是,由于其中间节点多,网络结构复杂,开放性强,通信信息极易被第三方用户获取,因此面临较大的安全威胁.尤其是在中继节点不可信的情况下,通信双

收稿日期:2011-12-26;最终修改稿收到日期:2012-05-16.本课题得到国家自然科学基金(61171108)、国家"八六三"高技术研究发展计划项目基金(2011AA010604)资助.**李翔宇**,男,1987年生,硕士研究生,助理工程师,主要研究方向为移动通信网络、物理层安全等.E-mail: luckyxiangyu@gmail.com.**金 梁**,男,1969年生,博士,教授,博士生导师,主要研究领域为移动互联网、第三代移动通信、物理层安全、超宽带通信等.**黄开枝**,女,1973年生,博士,副教授,主要研究方向为无线通信、第三代移动通信等.**吉 江**,男,1983年生,博士研究生,主要研究方向为无线通信、物理层安全等.

方的信息安全问题变得尤为突出.

传统无线网络安全的研究主要集中于密钥加密体制.随着无线通信网络日益复杂,需要构建更严密的密钥生成和管理协议.并且,由于无线通信的开放性,一但高层的加密机制被破解,通信的信息将完全暴露.所以,在无线通信中,防止信息在传输过程中被截获往往比防止信息被解密更有积极意义.1975 年,Wyner[2] 提出了点对点通信的物理层安全模型和保密容量(Secrecy Capacity)的概念,为从物理层提高无线中继系统的安全性提供了可行思路.文献[3-4]对中继节点的可信性问题进行了研究,计算了中继节点不可信时的保密容量,但主要是借助编码方式对各种模型下的理论极限进行计算.文献[5-6]分别采用接收者发送干扰信号和外部节点协助干扰的方式,对非可信中继节点进行适当的干扰,为解决中继节点不可信的问题提供了可行思路,但该文献只研究了单一中继窃听者的情况.文献[7]在此基础上提出了一种更明确、更优化的干扰权值计算方法.Ekrem 等人[8] 提出了一个更普遍的模型,中继节点本身也是发射节点,也有信息要传输,建立了协作中继广播信道模型,但要求发送者和接收者之间有直达径.文献[9]提出了一种中继窃听者广播虚假信道信息,欺骗发送者的情况,依托于编码方式计算了保密容量域.Yuksel 等人[10] 还研究了中继节点协助窃听者干扰正常接收者的情况.在这些方式中,大多数信道模型要求发送端和接收端之间有直达径,并且只研究了单一不可信节点参与协作转发的模型,因此应用场合有一定的局限性.

基于上述不足,本文提出了一种基于联合信道特征的中继物理层安全传输机制,首先建立了多中继协作信道模型,将中继前后的两个信道等效合并为一个,得到联合信道特征矩阵.然后以联合信道特征为参数,增加人工噪声[11-12],解决了中继节点不可信情况下的安全传输问题,计算了系统的保密容量和信号最佳发射功率比例.通过仿真证明,当增加的人工噪声处于合法接收者信道特征的零空间时,可以提高协作中继系统的保密容量,且合法接收者的误码率要远低于不可信中继.随着中继数量的增加,保密容量的变化趋势与窃听者的分布相关.本文方法能够解决中继节点不可信时的安全传输问题,其系统模型能够在移动蜂窝网等场景中获得较为广泛的应用.

2 系统模型

协作中继信道的系统模型如图 1 所示,Alice 是拥有 $M(M>1)$根天线的基站,利用 $N(N>1)$个中继节点 Relay$\{R_1,R_2,\cdots,R_N\}$,向 Bob 发送信息.Alice 与 Bob 之间没有直达径,Relay 和 Bob 均只有 1 根天线.系统中的某些 Relay 同时作为内部窃听者 Eve 存在,即不可信的中继节点,在执行正常转发功能同时进行窃听.

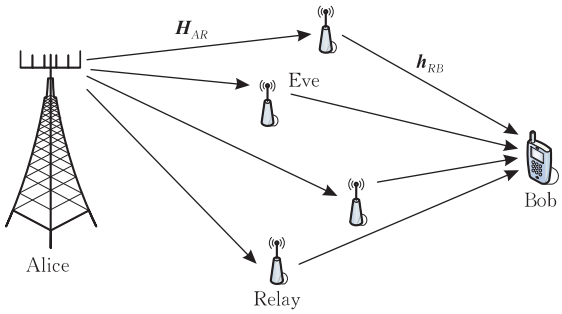


图 1 协作中继信道系统模型

Alice 发出的源信息 W 在信息集合$\{1,2,\cdots,2^{nR}\}$中服从均匀分布,并通过编码器映射到码字 ν^n,编码器使用的码本服从高斯分布.将系统整个传输过程分为一个个相等的时间单位,每个时间单位传输一个已编码的信源符号,各信源符号在发射前会经过预处理向量 $\boldsymbol{g}_1$,信号总功率为 $E\{|\boldsymbol{g}_1\nu|^2\}=P_s$,天线发射功率最大值为 P_t.系统内各节点均受到加性高斯白噪声的影响.

假设信道为平坦衰落信道.Alice 的 M 根天线和 N 个 Relay 之间的信道可以表示为

$$\boldsymbol{H}_{AR}=\begin{bmatrix} h_{AR(1,1)} & h_{AR(1,2)} & \cdots & h_{AR(1,N)} \\ h_{AR(2,1)} & h_{AR(2,2)} & \cdots & h_{AR(2,N)} \\ & & \ddots & \\ h_{AR(M,1)} & h_{AR(M,2)} & \cdots & h_{AR(M,N)} \end{bmatrix} \tag{1}$$

其中,第 n 列表示 Alice 与第 n 个 Relay 之间的信道特征,用 $\boldsymbol{h}_{AR(,n)}$ 表示.

Relay 均处于半双工模式,采用放大转发(Amplify-and-Forward,AF)[13] 的方式进行工作,Relay 和 Bob 之间的信道用 $\boldsymbol{h}_{RB}$ 表示,将中继转发所增加的放大系数 ω_n 和延时系数 τ_n 均计入 $\boldsymbol{h}_{RB}$ 中

$$\begin{aligned}\boldsymbol{h}_{RB}&=[h_{R_1B}\ \ h_{R_2B}\ \ \cdots\ \ h_{R_NB}]^{\mathrm{T}}\\&=[h_{R_1,B}\omega_1 e^{-j\omega\tau_1}\ \ h_{R_2,B}\omega_2 e^{-j\omega\tau_2}\ \ \cdots\ \ h_{R_N,B}\omega_N e^{-j\omega\tau_N}]^{\mathrm{T}}\end{aligned} \tag{2}$$

3　一种基于联合信道特征的中继物理层安全传输机制

Alice、Bob 和 Eve 是已知整个系统的信道状态信息(Channel State Information，CSI)，Alice 基于 CSI 采用增加人工噪声的方式对传输数据进行加密.

假设，每个 Relay 都是非可信节点，并且各个 Relay 之间不会联合处理信息，要保证 Relay 对转发的信息一无所知，必须保证 Relay 获得的信息量 $I(X;Z_R)=0$. 因此提出联合信道特征的概念，将 Alice 到 Relay 再到 Bob 之间的信道看成一个整体，忽略内部的传输和转发细节，只考虑其对信号幅度与相位等参数的影响，等效成 Alice 直接传输到 Bob 的一个信道. 即将 Alice 与 Relay 之间的信道特征 $\boldsymbol{H}_{AR}$ 和 Relay 与 Bob 之间的信道特征 $\boldsymbol{h}_{RB}$ 合并，组成联合信道特征 $\boldsymbol{h}_{ARB}$.

$$\begin{aligned}\boldsymbol{h}_{ARB}&=\boldsymbol{H}_{AR}\boldsymbol{h}_{RB}\\&=\begin{bmatrix}h_{AR(1,1)} & h_{AR(1,2)} & \cdots & h_{AR(1,N)}\\ h_{AR(2,1)} & h_{AR(2,2)} & \cdots & h_{AR(2,N)}\\ & & \ddots & \\ h_{AR(M,1)} & h_{AR(M,2)} & \cdots & h_{AR(M,N)}\end{bmatrix}\begin{bmatrix}h_{R_1B}\\ h_{R_2B}\\ \vdots\\ h_{R_NB}\end{bmatrix}\\&=\begin{bmatrix}h_{A_1B} & h_{A_2B} & \cdots & h_{A_MB}\end{bmatrix}^{\mathrm{T}}\end{aligned}\tag{3}$$

使用该联合信道特征构造人工噪声，对传输数据进行加密[11]. 定义 k 时刻的发射信号为 v_k，服从高斯分布. 基于上述信道模型，将整个传输过程分为两个阶段.

第 1 个阶段，Alice 向所有 Relay 发送信号 $\boldsymbol{x}_k$，$\boldsymbol{x}_k$ 包括信息承载信号 $\boldsymbol{s}_k$ 和构造的人工噪声信号 $\boldsymbol{n}_k$ 两个部分：

$$\boldsymbol{x}_k=\boldsymbol{s}_k+\boldsymbol{n}_k=\boldsymbol{g}_1v_k+\boldsymbol{g}_2u_k\tag{4}$$

其中，$\boldsymbol{g}_1$、$\boldsymbol{g}_2$ 均为经过归一化处理的 $M\times1$ 维向量. $\boldsymbol{g}_1$ 用来对 v_k 进行适当的波束成型，与 $\boldsymbol{h}_{ARB}$ 同向. u_k 是人工噪声信号的码字，由 $\boldsymbol{g}_2$ 对其进行映射，确保人工噪声信号位于 $\boldsymbol{h}_{ARB}$ 的零空间，即 $\boldsymbol{h}_{ARB}^H\boldsymbol{g}_2=0$.

则第 n 个中继节点实际收到的信号为

$$r_{n,k}=\boldsymbol{h}_{AR(,n)}^H\boldsymbol{g}_1v_k+\boldsymbol{h}_{AR(,n)}^H\boldsymbol{g}_2u_k+e_{k1}\tag{5}$$

e_{k1} 为高斯白噪声，$\boldsymbol{h}_{AR(,n)}$ 是 $\boldsymbol{H}_{AR}$ 的第 n 列.

第 2 个阶段，所有 Relay 将收到的信号 $r_{n,k}$ 进行放大转发. 由于 $\boldsymbol{n}_k$ 与 $\boldsymbol{h}_{ARB}$ 正交，则 Bob 实际接收到的信号为

$$\begin{aligned}y_k&=\boldsymbol{h}_{ARB}^H\boldsymbol{g}_1v_k+\boldsymbol{h}_{ARB}^H\boldsymbol{g}_2u_k+e_k\\&=\boldsymbol{h}_{ARB}^H\boldsymbol{g}_1v_k+e_k\end{aligned}\tag{6}$$

显然 Bob 没有受到人工噪声的影响，接收信号的信噪比较高.

在 $\boldsymbol{H}_{ARB}$ 的零空间中，可以选择合适的 $\boldsymbol{n}_k$，使得 $\boldsymbol{g}_2$ 与窃听者信道特征 $\boldsymbol{h}_{AR(,n)}$ 的夹角最小，$|\boldsymbol{h}_{AR(,n)}^H\boldsymbol{g}_2|$ 最大，降低 Relay 收到信号的信噪比.

4　保密容量及性能分析

4.1　系统保密容量

假设系统中多个内部窃听者之间不联合处理信息. 信道模型可以简化为如图 2 所示的情况.

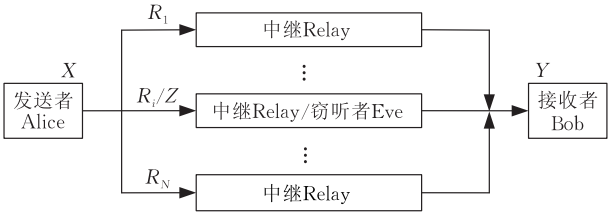


图 2　单一内部窃听者情况下的信道模型

由于 Alice 在发送时添加了人工噪声，使得 Eve 端信噪比降低，而人工噪声到达 Bob 处时正好消失，所以 Bob 收到信号的信噪比要高过 Eve，因而系统存在非 0 的保密容量.

对第 n 个中继节点来说，系统的保密容量应为[11,14]

$$\begin{aligned}C_{\mathrm{sec}\,R_n}&\geq\max_{P_XP_{YZ|X}}[I(X;Y)-I(X;Z)]\\&=\frac{W}{2}\log\left(1+\frac{|\boldsymbol{h}_{ARB}^H\boldsymbol{g}_1|^2\delta_{s_k}^2}{\delta_{e_k}^2}\right)-\\&\quad\frac{W}{2}\log\left(1+\frac{|\boldsymbol{h}_{AR(,n)}^H\boldsymbol{g}_1|^2\delta_{s_k}^2}{|\boldsymbol{h}_{AR(,n)}^H\boldsymbol{g}_2|^2\delta_{n_k}^2+\delta_{e_{k1}}^2}\right)\end{aligned}\tag{7}$$

因为系统只有一半的时间用来有效传输信息，所以系统各个信道的信道容量要乘以 1/2. 当人工噪声服从高斯分布时，等号成立.

从上述结果可以看出，当存在中继窃听者时，系统的保密容量受到信道特征矩阵以及信号和噪声功率的影响. 在实际中，每一时刻 $\boldsymbol{h}_{ARB}$ 和 $\boldsymbol{H}_{AR}$ 已知，高斯白噪声功率 $\delta_{e_k}^2$ 和 $\delta_{e_{k1}}^2$ 已经确定，在 P_t 保持不变的情况下，可以通过调整 $\boldsymbol{s}_k$ 和 $\boldsymbol{n}_k$ 的功率比，来对保密容量进行调节.

计算系统整体的保密容量，必须保证所有中继节点转发信息的安全性，对同时存在多个内部窃听者的情况进行分析. 当存在多个窃听者时，系统的保密容量应为[15]

$$C_s=\max_{j=1,2,\cdots,J}\min[I(X;Y)-I(X;Z_j)] \tag{8}$$

其中，$I(X;Z_j)$表示第 j 个窃听者获得的信息量. 因此，为了保证对所有 Relay 的安全性，系统整体的保密容量只能达到上述所有保密容量的最小值

$$C_{\sec R}=\min(C_{\sec R_1},C_{\sec R_2},\cdots,C_{\sec R_n}) \tag{9}$$

Alice 最大可以此速率发送数据，当任意一个 Relay 为窃听者时，都能保证系统的安全性.

4.2 最佳信号发射功率比例

要通过调整 $\boldsymbol{s}_k$ 和 $\boldsymbol{n}_k$ 的功率比，来对保密容量进行调节，就需要求出最佳的信号发射功率，研究信号功率 P_s 占总发射功率 P_t 的比例 ϕ 对保密容量的影响，即 $\phi=P_s/P_t$.

对第 n 个中继节点进行分析，当人工噪声服从高斯分布时，保密容量表达式等号成立

$$\begin{aligned}C_{\sec R_n}&=\max_{P_X P_{YZ|X}}[I(X;Y)-I(X;Z)]\\&=\frac{W}{2}\log\left(1+\frac{\|\boldsymbol{h}_{ARB}\|^2\|\boldsymbol{g}_1\|^2\delta_{s_k}^2}{\delta_{e_k}^2}\right)-\\&\quad\frac{W}{2}\log\left(1+\frac{\cos^2\alpha\|\boldsymbol{h}_{AR(,n)}\|^2\|\boldsymbol{g}_1\|^2\delta_{s_k}^2}{\cos^2\beta\|\boldsymbol{h}_{AR(,n)}\|^2\|\boldsymbol{g}_2\|^2\delta_{n_k}^2+\delta_{e_{k1}}^2}\right)\end{aligned} \tag{10}$$

其中，信道特征 $\boldsymbol{h}_{ARB}$ 与预处理向量 $\boldsymbol{g}_1$ 平行，$\boldsymbol{h}_{AR(,n)}$ 与向量 $\boldsymbol{g}_1$、$\boldsymbol{g}_2$ 的夹角分别为 α 和 β.

信号发射功率 $\delta_{s_k}^2=P_t\phi$，人工噪声发射功率 $\delta_{n_k}^2=P_t(1-\phi)$，则保密容量表达式可表示为

$$\begin{aligned}C_{\sec R_n}=\frac{W}{2}\Bigg[&\log\left(1+\frac{\|\boldsymbol{h}_{ARB}\|^2\|\boldsymbol{g}_1\|^2P_t\phi}{\delta_{e_k}^2}\right)-\\&\log\left(1+\frac{\cos^2\alpha\|\boldsymbol{h}_{AR(,n)}\|^2\|\boldsymbol{g}_1\|^2P_t\phi}{\cos^2\beta\|\boldsymbol{h}_{AR(,n)}\|^2\|\boldsymbol{g}_2\|^2P_t(1-\phi)+\delta_{e_{k1}}^2}\right)\Bigg]\end{aligned} \tag{11}$$

令式(11)对 ϕ 取导，可求得保密容量极大时 ϕ 的值，但所求结果比较复杂，所以可在某些情况下做相应的简化处理. 由于 $\boldsymbol{h}_{ARB}$、$\boldsymbol{h}_{AR(,n)}$、$\boldsymbol{g}_1$ 和 $\boldsymbol{g}_2$ 均已经过归一化处理，则保密容量表达式可简化为

$$\begin{aligned}C_{\sec R_n}=&\\\frac{W}{2}\log\left(1+\frac{P_t\phi}{\delta_{e_k}^2}\right)-&\frac{W}{2}\log\left(1+\frac{P_t\phi\cos^2\alpha}{P_t(1-\phi)\cos^2\beta+\delta_{e_{k1}}^2}\right)\end{aligned} \tag{12}$$

研究信号功率占总发射功率的比例 ϕ，本质上就是研究人工噪声对系统保密容量提高的影响，所以可先弱化白噪声的影响，假设系统的信道状况较好，白噪声功率较低 $\delta_{e_{k1}}^2\ll P_t$，令 $\gamma=\dfrac{\cos^2\alpha}{\cos^2\beta}$，则

$$C_{\sec R_n}\approx\frac{W}{2}\log\left(1+\frac{P_t\phi}{\delta_{e_k}^2}\right)-\frac{W}{2}\log\left(1+\gamma\frac{\phi}{1-\phi}\right) \tag{13}$$

令其对 ϕ 求导，可求得系统保密容量极大时 ϕ 的值，即

$$P_t(1-\gamma)\phi^2-2P_t\phi+P_t-\delta_{e_k}^2\gamma\approx0 \tag{14}$$

根据假设，白噪声功率较低，$\delta_{e_k}^2\ll P_t$，可再次简化

$$P_t(1-\gamma)\phi^2-2P_t\phi+P_t\approx0 \tag{15}$$

解方程(15)，可得系统保密容量最大时，信号功率占发射总功率 P_t 的比例 ϕ：

$$\phi\approx\frac{1}{1+\sqrt{\gamma}} \tag{16}$$

式(16)的结论即为整体发射功率 P_t 保持不变的情况下，信道状况较好时，信号的最佳发射功率，此时，系统的保密容量达到最大值.

5 模型仿真

5.1 信号功率占总发射功率比例对保密容量的影响

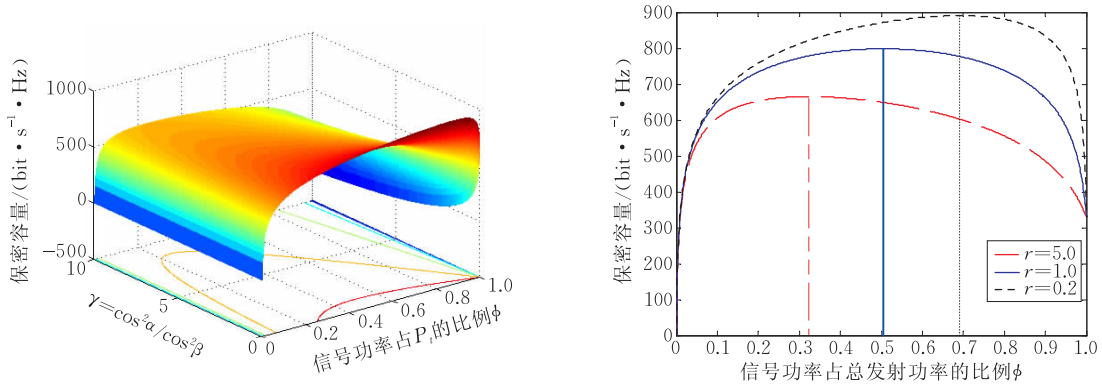
首先研究参数 ϕ 对保密容量的影响，分析 γ 为不同值时，保密容量随 ϕ 的变化趋势. 针对式(12)，设定如表 1 的仿真数据.

表 1 第 1 组仿真参数设置

参数名	取值
P_t	100 mW
$\delta_{e_k}^2$	0.1 mW
$\delta_{e_{k1}}^2$	0.1 mW
$\cos^2\alpha$	0.1
$\cos^2\beta$	0.5、0.1、0.02
W	200 Hz

仿真结果如图 3 所示，从图中可以看出，系统保密容量首先随着 ϕ 的增大而增大，当增大到一定程度之后又开始减小. 主要是由于，当 ϕ 比较小时，用于发射信号的功率很少，Bob 接收到的信噪比较低，信息量较少，保密容量自然受到影响，当 ϕ 比较大时，Bob 接收到的信息量较多，但由于人工噪声功率较小，Eve 也能接收到比较多的信息，所以保密容量也较小. 当 $\phi=100\%$ 时，人工噪声功率为 0，但由于发射天线设置了指向 Bob 的波束成形，Eve 获得的信息量仍然不如 Bob，所以有非 0 的保密容量. 因此，增加人工噪声可以显著地提高系统的保密容量.

同时，保密容量也受窃听者信道与两预处理向量 $\boldsymbol{g}_1$、$\boldsymbol{g}_2$ 夹角比值 γ 的影响，γ 的物理意义表示人工噪声对中继窃听者接收信噪比的影响. γ 越大，说明窃听者信道与人工噪声正交性越好，受人工噪声影响越小. 因此，当 γ 较大时，需要将更多的功率分配到人工噪声上，用来对窃听者进行足够的干扰，系统



(a) 保密容量与ϕ和γ的立体关系图　　(b) 保密容量与ϕ的关系

图 3　第 1 组参数仿真图

的保密容量也相对较小，且在 ϕ 较小时就已经达到极大值. 反之，若 γ 较小，则将较少的功率分配到人工噪声上就可以对窃听者产生足够的干扰，发射信号的功率就比较大，系统的保密容量也相应较大. 图 3 中还用竖线标明了当保密容量达到最大值时 ϕ 的值，验证了式(16)的相关结论.

5.2　中继节点数量对保密容量的影响

第 2 组仿真讨论中继节点数量 N 对系统保密容量的影响. 针对不同数量的中继节点，依照高斯分布随机产生信道特征矩阵 $\boldsymbol{H}_{AR}$ 和 $\boldsymbol{h}_{RB}$，然后计算出联合信道特征 $\boldsymbol{h}_{ARB}$ 以及预处理向量 $\boldsymbol{g}_1$ 和 $\boldsymbol{g}_2$，并将相应矩阵进行归一化.

当系统内只存在固定数量 $L(L\leq N)$ 的内部窃听者时，因为窃听节点是随机选取的，为不失一般性，可以在研究系统保密容量的变化趋势时，采用多个中继节点产生系统保密容量的均值作为参数，针对不同的中继数量，进行多次仿真，计算系统保密容量的均值.

$$\overline{C}_{\mathrm{sec}\,R}=\sum_{n=1}^{N}\left[\frac{W}{2}\log\left(1+\frac{|\boldsymbol{h}_{ARB}^{H}\boldsymbol{g}_1|^2\delta_{s_k}^2}{\delta_{e_k}^2}\right)-\frac{W}{2}\log\left(1+\frac{|\boldsymbol{h}_{AR(,n)}^{H}\boldsymbol{g}_1|^2\delta_{s_k}^2}{|\boldsymbol{h}_{AR(,n)}^{H}\boldsymbol{g}_2|^2\delta_{n_k}^2+\delta_{e_{k1}}^2}\right)\right]\Big/N \tag{17}$$

设定如表 2 仿真数据.

表 2　第 2 组仿真参数设置

参数名	取值
P_t	100 mW
$\delta_{e_k}^2$	0.1 mW
$\delta_{e_{k1}}^2$	0.1 mW
M	8
N	4、8、16、32
W	200 Hz

仿真结果如图 4 所示，从图中可以看出，中继节点的数量越多，系统的保密容量会越大，但增加中继节点数量对保密容量的改善也越来越小. 这主要是由于 Alice 到 Bob 之间的信道特征 $\boldsymbol{h}_{ARB}=\boldsymbol{H}_{AR}\boldsymbol{h}_{RB}$，是由每个中继节点的信道特征 $\boldsymbol{h}_{AR(,n)}$ 加权组合而成. 因此，$\boldsymbol{h}_{ARB}$ 与每个 $\boldsymbol{h}_{AR(,n)}$ 都有一定的相关性，而人工噪声信号与 $\boldsymbol{h}_{ARB}$ 完全正交. 当 N 较大时，信道 $\boldsymbol{H}_{ARB}$ 的冗余度就较高，与 $\boldsymbol{h}_{AR(,n)}$ 的相关性就较弱，则相关中继节点受人工噪声影响就比较大，收到的信号也较弱，系统会有比较大的保密容量. 反之，当 N 较小时，信道 $\boldsymbol{h}_{ARB}$ 与 $\boldsymbol{h}_{AR(,n)}$ 相关性较大，中继信道与人工噪声夹角 β 变小，受人工噪声影响也较小，所以系统的保密容量会变小.

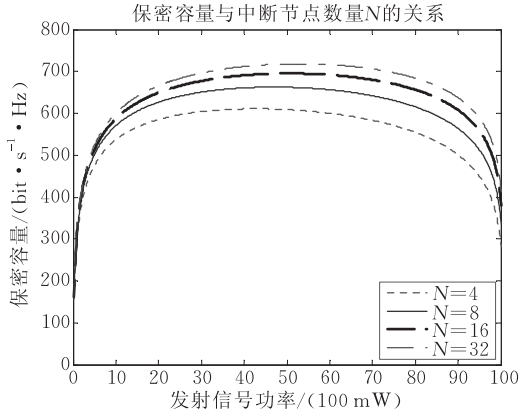


图 4　系统存在固定数量内部窃听者

当系统内所有中继节点均为窃听者时，研究增加中继数量 N 对系统保密容量的影响，采用式(9)的结论，参照表 2 的仿真参数进行多次仿真，然后取均值. 结果如图 5 所示，从图中可以看出，当所有中继节点均为内部窃听者时，系统保密容量随着中继节点数量 N 的增加而减小. 这主要是因为，中继节点越多，窃听者就越多，同时保证所有节点获得信息

量 $I(X;Z_R)\to 0$ 的难度也越大，系统的安全性自然是降低的. 分析保密容量随发射功率增长的变化趋势，与仿真 1 中 γ 比较大的情况类似，原因是 γ 比较大时，该中继窃听者受人工噪声影响较小，获得的信息量较多，导致系统保密容量较小，根据式(9)，系统保密容量随 γ 最大的节点进行变化.

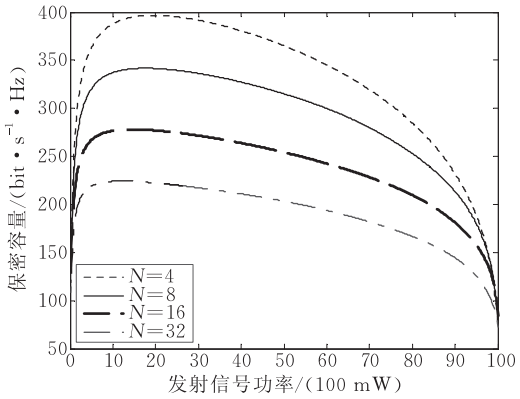


图 5 所有中继节点均为内部窃听者时保密容量与中断节点数量 N 的关系

在上述两组仿真情况中，系统保密容量随 N 的变化呈现出截然不同的变化趋势，而且在实际中，参与转发的中继节点越多，所消耗的能量也就越多，因此需要根据实际情况进行均衡，选择合适的中继数量.

5.3 节点误码率性能

最后针对引入人工噪声对系统性能的影响以及实际的安全传输效果进行探讨，建立一个简单的系统模型，对上述结果进行仿真验证. 在实际应用中，信道状态信息主要包括两个部分，一是和距离有关的信道衰落，二是随机的相位变化，例如 $\boldsymbol{h}_{AR}=d_{AR}^{-c/2}\,\mathrm{e}^{\mathrm{j}\theta}$，$d_{AR}$ 表示 Alice 与 Relay 之间的距离，距离衰落指数 c 取 3.5，θ 在 $[0,2\pi)$ 之间均匀分布. 可根据此生成 $\boldsymbol{H}_{AR}$ 和 $\boldsymbol{h}_{RB}$.

增加人工噪声的主要作用在于，当系统信道特征缓变或不变时，对窃听者 Eve 收到的信号添加随机干扰，防止其通过盲辨识或反卷积来估计信道，增加系统的安全性. 因此，仿真发送 10^6 个 BPSK 码元，系统信道特征在整个传输阶段保持稳定，增加的人工噪声在每个码元之间随机变化，但均处于 $\boldsymbol{h}_{ARB}$ 的零空间. 为了计算方便，设置第 n 个中继节点的放大系数为 $(d_{R_nB}^{-c/2})^{-1}$，用于抵消路径传输损耗，d_{R_nB} 为第 n 个中继与 Bob 之间的距离. Relay 和 Bob 节点接收到的白噪声功率均为 2.5×10^{-9} W，由于 Relay 转发的信号里也含有白噪声，所以 Bob 受白噪声的影响更大. 具体仿真参数如表 3 所示.

表 3 第 3 组仿真参数设置

参数名	取值
P_t	0～10 W
ϕ	100%，50%
d_{AR}	[100,120]m
d_{RB}	[40,60]m
M	6
N	5
测试码元数	10^6

表 3 中，$\phi=100\%$ 表示所有功率用来发送有用信号，不发送人工噪声. $\phi=50\%$ 表示有 50% 的功率用来发送人工噪声. 对 Relay 和 Bob 两种情况下的误码率进行仿真对比，随机选取 N 个 Relay 节点中的一个为例，仿真结果如图 6 所示.

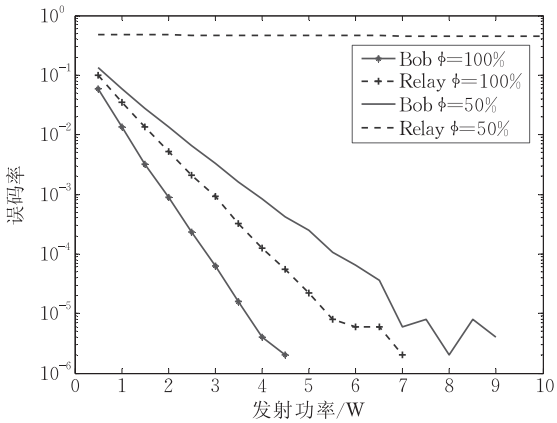


图 6 Relay 和 Bob 的误码率对比

从图 6 中可以看出，当 $\phi=100\%$ 时，随着发射功率的增加，Bob 和 Relay 的误码率比之 $\phi=50\%$ 时有显著的降低，这主要是因为所有功率用来发送有用信号，能够更有效地克服白噪声的干扰. 显然，引入人工噪声使得功率利用率有所降低，在一定程度上导致了信道效率的下降，系统受白噪声影响更大，误码率变高，传输速率降低. 但是，人工噪声对系统安全性的改善也是显而易见的. 当 $\phi=100\%$ 时，虽然发射信号设置了指向 Bob 的波束成型，但由于 Relay 节点接收到的白噪声功率比 Bob 要小，所以 Relay 通过简单的矩阵变换完成相位调整后，其误码率性能虽比合法接收者 Bob 略差，但相差不大. 当增加人工噪声后，$\phi=50\%$ 时，不可信 Relay 节点的接收信号趋于混乱，误码率变为接近 50%，无法正确还原信息内容，系统安全性得到了保证. 因此，基于联合信道特征与人工噪声的安全机制，虽然损失了部分信道传输效率，但系统的安全性得到了很大的提高，其意义在于能够在物理层屏蔽不可信中继节点的窃听，使中继节点在转发信息的同时对信

息内容一无所知.

6　结束语

文章针对中继节点的不可信问题，结合点对点通信物理层安全中 MISO 系统的现有结论，提出了一种基于联合信道特征的中继物理层安全传输机制. 将 Alice 到 Relay 再到 Bob 之间的信道看成一个整体，忽略内部的传输和转发细节，以增加人工噪声的方式，对窃听者进行干扰. 建立了相应的信道模型，计算了系统的保密容量.

通过特定情况的仿真得出，当增加的人工噪声处于联合信道特征的零空间中时，能够提高系统的保密容量，且保密容量随信号功率占发射总功率比例 ϕ 的增大先增大后减小. 窃听者分布情况不同时，系统保密容量随中继节点数量 N 的变化趋势不同.

在以后的研究中，一方面可以对其它应用场景进行讨论，如无线自组织网和传感器网等，另一方面还可以将其它物理层安全技术应用到协作中继系统中来，满足不同应用条件的需求.

参　考　文　献

[1] Parkvall Stefan, Astely David. The evolution of LTE towards IMT-Advanced. Journal of Communications, 2009, 4(3): 146-154

[2] Wyner A D. The wire-tap channel. Bell Systems Technical Journal, 1975, 54(8): 1355-1387

[3] Oohama Y. Capacity theorems for relay channels with confidential messages//Proceedings of the IEEE International Symposium on Information Theory. Nice, France, 2007: 926-930

[4] He X, Yener A. Cooperation with an untrusted relay: A secrecy perspective. IEEE Transactions on Information Theory, 2010, 56(8): 3807-3827

[5] Zhang R, Song L, Han Z et al. Physical layer security for two way relay communications with friendly jammers//Proceedings of the IEEE Global Telecommunications Conference. Miami, USA, 2010: 1-6

[6] He X, Yener A. Two-hop secure communication using an untrusted relay: A case for cooperative jamming//Proceedings of the IEEE Global Telecommunications Conference. Los Angeles, USA, 2008: 1-5

[7] Zheng G, Choo G, Wong K. Optimal cooperative jamming to enhance physical layer security using relays. IEEE Transactions on Signal Processing, 2011, 59(3): 1317-1322

[8] Ekrem E, Ulukus E. Secrecy in cooperative relay broadcast channels. IEEE Transactions on Information Theory, 2011, 57(1): 137-155

[9] Milosavljevic N, Gastpar M, Ramchandran K. Secure communication using an untrusted relay via sources and channels//Proceedings of the IEEE International Symposium on Information Theory. Seoul, Korea, 2009: 2457-2461

[10] Yuksel M, Liu X, Erkip E. A secure communication game with a relay helping the eavesdropper. IEEE Transactions on Information Forensics and Security, 2011, 6(3): 818-830

[11] Goel S, Negi R. Guaranteeing secrecy using artificial noise. IEEE Transactions on Wireless Communication, 2008, 7(6): 2180-2189

[12] Luo Wenyu, Jin Liang, Huang Kaizhi et al. User selection and resource allocation for secure multiuser MISO-OFDMA systems. IET Electronics Letters, 2011, 47(15): 884-886

[13] Dong L, Han Z, Petropulu A P et al. Improving wireless physical layer security via cooperating relays. IEEE Transactions on Signal Processing, 2010, 58(3): 1875-1888

[14] Liang Y, Poor H V, Shamai S. Information Theoretic Security. Delft: Now Publishers, 2009

[15] Liang Y, Kramer G, Poor H V et al. Compound wire-tap channels//Proceedings of the 45th Annual Allerton Conference on Communication Control and Computing. UIUC, USA, 2007: 136-143

LI Xiang-Yu, born in 1987, M. S. candidate, assistant engineer. His research interests are mobile communication network and physical-layer security.

JIN Liang, born in 1969, Ph. D., professor, Ph. D. supervisor. His research interests are mobile Internet, the 3rd generation communication, physical-layer security, and ultra-wideband communication.

HUANG Kai-Zhi, born in 1973, Ph. D., associate professor. Her research interests are wireless communication and the 3rd generation communication.

JI Jiang, born in 1983, Ph. D. candidate. His research interests include wireless communication and physical-layer security.

Background

Nowadays, wireless communication technology has been widely used in our life. Due to the broadcast nature of wireless channels, the traditional methods to ensure security have focused on cryptographic technique in the high-level of protocol. However, if this encryption mechanism is cracked, information will be completely exposed.

Recent studies on the physical-layer security without key encryption offered another way to prevent eavesdropping. In this field, secrecy capacity, which is defined as the maximum reliable information rate sent from the source to the legitimate destination in the presence of eavesdroppers, is used to quantify the system security.

The basic idea of physical-layer security is to process the transmission signal by using the physical characteristics of wireless channels. The most widely used method is based on multiple antennas in MIMO system. When most nodes in the wireless network have only one antenna, we should find some other method to construct abundant channel characteristics, which is used to guarantee system security in physical-layer. The cooperation technique via relaying, where the source to destination transmission is helped by a bank of relays, makes the method more flexible. Meanwhile, with more nodes and more complex network structure in relay system, it is difficult for the deployment of high-level security mechanisms. As a result, using physical-layer security technology is the feasible approach to solve this problem.

Some scholars have considered the physical-layer security in cooperative relay system. However, there are few studies focused on multi-relay nodes. Most of the theses involved only one relay and needed a direct path between the sender and receiver. This paper presents a physical-layer security mechanism in cooperative relay system. We try to take full advantage of abundant channel characteristics constructed by multi-relay nodes, get the joint channel characteristics by combining the two channels before and after the relay into one and add artificial noise in its null space to make untrusted relays get none of the information.

Our group has been working on physical-layer security for several years. We have many works been published in international journals and conferences. The work in this paper is supported by the National Natural Science Foundation of China under Grant No. 61171108 and the National High Technology Research and Development Program (863 Program) of China under Grant No. 2011AA010604.

基于量子逻辑的图灵机及其通用性

李永明[1),2)] 李 平[2)]

[1)](陕西师范大学计算机科学学院 西安 710062)

[2)](陕西师范大学数学与信息科学学院 西安 710062)

摘 要 基于量子逻辑的自动机理论是量子计算模型的一个重要研究方向.该文研究了基于量子逻辑的图灵机(简称量子图灵机)及其一些变形,给出了包括非确定型量子图灵机l-VTM,确定型量子图灵机l-VDTM以及相应类型的多带量子图灵机,并引入量子图灵机基于深度优先与宽度优先识别语言的两种不同定义方式,证明了这两种定义方式在量子逻辑意义下是不等价的.进一步证明了l-VTM、l-VDTM与相应类型的多带量子图灵机之间的等价性.其次,给出了量子递归可枚举语言及量子递归语言的定义,并给出了二者的层次刻画,证明了l-VTM与l-VDTM不等价,但两者作为量子递归语言的识别器是等价的.最后,文中讨论了基于量子逻辑的通用图灵机的存在性问题,给出了一套合理编码系统,证明了基于量子逻辑的通用图灵机在其所取值的正交模格无限时不存在,而在其所取值的正交模格有限时是存在的.

关键词 量子逻辑;量子计算;量子图灵机;量子递归可枚举语言;量子递归语言

中图法分类号 TP301 **DOI号**: 10.3724/SP.J.1016.2012.01407

Turing Machines Based on Quantum Logic and Their Universality

LI Yong-Ming[1),2)] LI Ping[2)]

[1)](*College of Computer Science, Shaanxi Normal University, Xi'an* 710062)

[2)](*College of Mathematics and Information Science, Shaanxi Normal University, Xi'an* 710062)

Abstract Automata theory based on quantum logic is an important aspect of the quantum computing models. First we study Turing machines based on quantum logic (quantum Turing machines, for short) and their variants, the notions including nondeterministic quantum Turing machines l-VTM, deterministic quantum Turing machines l-VDTM and multi-tape quantum Turing machines are introduced. Two methods to recognize quantum languages by quantum Turings machines are given, which are based on depth-first technique and on width-first technique, respectively, it is shown that these two methods are not equivalent in recognizing quantum languages. Second, we introduce the notions of quantum recursively enumerable languages and quantum recursively languages, and the stratified characterization of them are also given. Moreover, we show that l-VTMs and l-VDTMs are not equivalent. However, they are equivalent when they accept quantum recursively languages. Which is an important differentia between quantum Turing machines and classical Turing machines. Finally, we study the existence of a universal quantum Turing machine and give a coding system. It is shown that the universal quantum Turing machine does not exist when the orthomodular lattice is infinite, and the universal quantum Turing machine exists when the orthomodular lattice is finite.

Keywords quantum logic; quantum computing; quantum Turing machines; quantum recursively enumerable languages; quantum recursively languages

收稿日期:2011-01-21;最终修改稿收到日期:2012-01-03.本课题得到国家自然科学基金(60873119)、教育部高等学校博士点基金(200807180005)资助.**李永明**,男,1966年生,博士,教授,研究领域为非经典计算理论、计算智能、量子计算与量子信息、格上拓扑学.E-mail: liyongm@snnu.edu.cn.**李 平**,女,1979年生,博士,讲师,研究方向为非经典计算理论、计算智能.

1 引 言

量子计算源于物理与计算之间的联系[1]，由 Shor 于 1994 年发现了在量子计算机上进行大数分解的多项式时间算法以及 Grover 发现了平方根时间的量子搜索算法之后，量子计算日益受到人们的关注与重视，量子计算模型便是其中一个很重要的研究课题.由应明生等建立的基于量子逻辑的自动机(也称量子自动机)理论[2-10]是量子计算模型方面的一个重要研究方向，目前已得到了与经典逻辑意义下不同的结果，并尝试揭示量子计算的逻辑基础.

文献[2-6,8-10]主要讨论了基于量子逻辑的有穷自动机的一些性质，文献[7]讨论了基于量子逻辑的下推自动机的一些性质，都得到了一些不同于经典自动机的结论，而对于基于量子逻辑的图灵机目前还没有研究，本文的主要工作就是进一步讨论基于量子逻辑的图灵机(也称量子图灵机)的一些性质.

经典图灵机及其许多变形识别语言的能力都是相同的，正因为如此，图灵机可以作为计算的一般模型.另外，通用图灵机(可编程图灵机)是存在的，通用图灵机可以模拟任意一个图灵机，这也是将图灵机作为现代计算机的形式模型的根本原因.我们知道，基于量子逻辑的有穷自动机与经典自动机的性质有所区别，主要是因为其真值格不满足分配律.但同时，许多重要性质，诸如 Kleene 定理、Büchi-Elgot 定理以及 Schützenberger 定理，在量子自动机情形时都成立，不过其相关构造都具有量子逻辑特性.比如，量子逻辑意义下，确定型有穷自动机与非确定型有穷自动机仍然是等价的，但经典的子集构造方法这时是不适用的.为此我们给出了量子逻辑意义下的子集构造方法，并籍此证明了量子逻辑意义下确定型有穷自动机与非确定型有穷自动机的等价性，该构造不依赖于分配性，但其复杂性远较经典构造困难，相关的构造与分析参见文献[10].因此，我们想进一步知道：在其真值格不满足分配律情况下，基于量子逻辑的图灵机及其各种变形之间识别语言的定义方式以及识别语言的能力如何?基于量子逻辑的通用图灵机存在吗?这两个重要问题就是本文所研究的主要内容.我们将给出量子逻辑意义下的相关构造，并得到一些与经典图灵机本质不同的结果.

第 2 节主要介绍量子图灵机的概念及其所识别的语言，给出不同类型的量子图灵机的定义，包括：非确定型量子图灵机 l-VTM、具有分明转移函数的量子图灵机 l-VTMc，确定型量子图灵机 l-VDTM、允许读写头不动的量子图灵机 l-VTM$_S$，相应类型的多带量子图灵机：k 带 l-VTM、k 带 l-VTMc、k 带 l-VDTM 及相应类型的双向无穷带量子图灵机：l-TVTM、l-TVTMc、l-TVDTM，并给出基于宽度优先识别语言与基于深度优先识别语言的两种(不等价)定义.并进一步证明 l-VTM、l-VTMc、l-VTM$_S$、k 带 l-VTM、k 带 l-VTMc、l-TVTM、l-TVTMc 在深度优先方式下的等价性；l-VDTM、k 带 l-VDTM、l-TVDTM 的等价性.

第 3 节给出量子递归可枚举语言和量子递归语言的层次刻画，进一步证明 l-VTM 与 l-VDTM 不等价，但两者作为量子递归语言的识别器是等价的，这也是基于量子逻辑的图灵机不同于经典图灵机的一个重要方面.

第 4 节讨论基于量子逻辑的通用图灵机(即通用 l-VTM、l-VDTM)的存在性问题，给出量子图灵机的一套合理编码系统，进一步证明当正交模格 l 无限时，通用 l-VTM M_u，l-VDTM M_{ud} 是不存在的，而当正交模格 l 有限时，通用 l-VTM M_u，l-VDTM M_{ud} 存在，并给出其构造.

2 量子图灵机概念及其基本性质

首先回顾一些关于量子逻辑的基本内容.量子逻辑是指真值为完备正交模格的逻辑，也称为正交模格值逻辑[11-12].本文采用文献[2-3]的有关记号.完备的正交模格是七元组 $l=\langle l,\leq,\wedge,\vee,\perp,0,1\rangle$，其中

(1) $l=(l,\leq,\wedge,\vee,\perp,0,1)$ 是完备格，1 和 0 分别是最大元和最小元，$\leq$ 是偏序，对 $X\subseteq l$，$\vee X$，$\wedge X$ 分别表示 X 的上确界与下确界.

(2) 一元运算 $\perp$ 是 l 上的正交补，满足如下条件：$\forall a,b\in l$，

(i) $a\wedge a^{\perp}=0$，$a\vee a^{\perp}=1$；

(ii) $a^{\perp\perp}=a$；

(iii) $a\leq b$ 蕴含 $b^{\perp}\leq a^{\perp}$；

很容易看出，条件(iii)与 De Morgan 对偶律是等价的：即 $\forall a,b\in l$，

(iii′) $(a\wedge b)^{\perp}=a^{\perp}\vee b^{\perp}$，$(a\vee b)^{\perp}=a^{\perp}\wedge b^{\perp}$；

一个正交模格就是满足如下正交模律的正交格：$\forall a,b\in l$，

(iv) $a\leq b$ 蕴含 $a\vee(a^{\perp}\wedge b)=b$.

在正交模格 l 上定义蕴含算子 $\rightarrow$ 满足 $\forall a,b\in l$，$a\rightarrow b=1$ 当且仅当 $a\leq b$. 双蕴含的定义为 $\forall a,b\in l$，$a\leftrightarrow b=(a\rightarrow b)\wedge(b\rightarrow a)$. 正交模格值逻辑(即量子逻辑)的语法与经典一阶逻辑类似. $\neg$，$\vee$，$\rightarrow$ 是 3 个原始连接词，$\exists$ 是原始量词，$\wedge$，$\leftrightarrow$ 是由 $\neg$，$\vee$，$\rightarrow$ 和 $\exists$ 定义. 语义方面，将 $\neg$，$\vee$，$\rightarrow$ 分别理解为正交模格上的 $\perp$，$\vee$，$\rightarrow$ 运算，$\exists$ 解释为下确界. 集合论公式 $x\in A$ 的真值为 $\lceil x\in A\rceil=A(x)$. 公式 φ 是有效的当且仅当 $\lceil\varphi\rceil=1$，并记为 $\overset{l}{\models}\varphi$.

定义 1.　一个基于量子逻辑的图灵机(简称为量子图灵机，记为 l-VTM)是一个七元组：

$$M=(Q,\Sigma,\Gamma,\delta,I,B,F),$$

其中，Q 为有限状态集合；$I,F:Q\rightarrow l$ 为 Q 的 l-值子集，代表初始状态与接受状态，$\forall q\in Q$，$I(q)$ 表示命题“q 为初始状态”的真值，$F(q)$ 表示命题“q 为终状态”的真值；Γ 为有限带符号表；$B\in\Gamma$ 称为空白符；$\Sigma\subseteq\Gamma-\{B\}$ 为有限输入字母表，除了空白符 B 之外，只有 Σ 中的符号才能在 M 启动时出现在输入带上；δ 是 $Q\times\Gamma\times Q\times\Gamma\times\{L,R\}$ 上的一个 l-值子集，即 $\delta:Q\times\Gamma\times Q\times\Gamma\times\{L,R\}\rightarrow l$，称 δ 为 l-值转移函数或量子转移函数，$\forall p,q\in Q,x,x'\in\Gamma,\delta(q,x,q',x',d)$ 表示命题“M 在状态 q 读入符号 x，状态改为 q'，并在 x 所在的带方格中印刷符号 x' 然后将读头向左(当 $d=L$)或向右(当 $d=R$)移动一格”的真值.

一些符号与说明：

(1) 描述 l-VTM M 所识别的逻辑语言的原子命题有如下 3 类：

“q 为初始状态”，记为“$q\in I$”；

“q 为终状态”，记为“$q\in F$”；

“M 在状态 q 读入符号 x，状态改为 q'，并在 x 所在的带方格中印刷符号 x' 然后将读头向左(当 $d=L$)或向右(当 $d=R$)移动一格”，记为“$(q,x,q',x',d)\in\delta$”.

上述命题的真值分别为 $I(q)$，$F(q)$，$\delta(q,x,q',x',d)$. 我们把描述 l-VTM M 所识别的逻辑语言的原子命题全体之集记作 $atom(M)$，即

$$atom(M)=\{\text{“}q\in I\text{”}:q\in Q\}\cup\{\text{“}q\in F\text{”}:q\in Q\}\cup\{\text{“}(q,x,q',x',d)\in\delta\text{”}:p,q\in Q,x,x'\in\Gamma\}.$$

(2) 设 $\Sigma^*=\bigcup_{n=0}^{\infty}\Sigma^n$ 表示 Σ 上所有长度有限的串的集合. 则 Σ^* 是由 Σ 通过连接运算生成的自由幺半群. $\forall\omega\in\Sigma^*$，用 $|\omega|$ 表示 ω 的长度，空串 ε 的长度为 0.

(3) 一个 Σ 上的 l-值语言(也称为量子语言)是 Σ^* 的一个 l-值子集. Σ 上的所有 l-值语言之集记作 $l^{\Sigma^*}=\{A\mid A:\Sigma^*\rightarrow l\}$. 设 $A,B\in l^{\Sigma^*}$ 为两个 Σ^* 上的 l-值子集，下面定义两个语言 A 和 B 的并运算 $A\cup B\in l^{\Sigma^*}$，交运算 $A\cap B\in l^{\Sigma^*}$，连接运算 $A\circ B\in l^{\Sigma^*}$ 和 $\lambda\in l$ 与语言 A 的数乘运算 λA：$\forall s\in\Sigma^*$，

$$s\in A\cup B\overset{\text{def}}{=}(s\in A\vee s\in B);$$

$$s\in A\cap B\overset{\text{def}}{=}(s\in A\wedge s\in B);$$

$$s\in A\circ B\overset{\text{def}}{=}(\exists u,v\in\Sigma^*)(s=uv\wedge u\in A\wedge v\in B);$$

$$s\in\lambda A\overset{\text{def}}{=}\lambda\wedge(s\in A).$$

其对应的真值分别为

$$(A\cup B)(s)=A(s)\vee B(s);$$

$$(A\cap B)(s)=A(s)\wedge B(s);$$

$$(A\circ B)(s)=\vee\{A(u)\wedge B(v):u,v\in\Sigma^*\text{且}s=uv\};$$

$$(\lambda A)(s)=\lambda\wedge A(s).$$

定义 2.　设 M 是一个 l-VTM，若 $I=q_0\in Q$，l-值转移函数 δ 为从 $Q\times\Gamma$ 到 $P(Q\times\Gamma\times\{L,R\})$ 的函数，即 $\delta:Q\times\Gamma\rightarrow P(Q\times\Gamma\times\{L,R\})$，其中，$P(X)$ 表示集合 X 的所有子集构成的集合，也记为 2^X，F 为 Q 的 l-值子集，则称 M 为一个具有分明转移函数的量子图灵机(简记为 l-VTMc). 进一步，若 δ 为 $Q\times\Gamma$ 到 $Q\times\Gamma\times\{L,R\}$ 的一个部分函数，则称 M 为一个确定型量子图灵机(简记为 l-VDTM). 注意到这两种类型的量子图灵机只有接受状态是 l-值子集.

和经典图灵机一样，我们定义 l-VTM M 的即时描述(ID)为 $\alpha_1q\alpha_2$，其中，$\alpha_1,\alpha_2\in\Gamma^*$，$q\in Q$. 该即时描述表示：$q$ 为 M 的当前状态，α_1 和 α_2 为 M 的输入带上的字符串，当 $\alpha_2\neq\varepsilon$ 时 M 的读头注视着 α_2 的最左字符，当 $\alpha_2=\varepsilon$ 时 M 的读头注视着空白符 B.

令 $ID(M)=\Gamma^*\times Q\times\Gamma^*$ 表示 M 的所有 ID 之集，我们定义 $ID(M)\times ID(M)$ 上的二元 l-值关系(记作 $\prec_M$)如下：$\forall D_1,D_2\in ID(M)$，$\alpha,\beta\in\Gamma^*$，$x,y,z\in\Gamma$，$p,q\in Q$，

$$\lceil(D_1,D_2)\in\prec_M\rceil=\prec_M(D_1,D_2)=\begin{cases}\delta(q,x,p,y,R), & D_1=\alpha qx\beta,D_2=\alpha yp\beta\\ \delta(q,x,p,y,L), & D_1=\alpha zqx\beta,D_2=\alpha pzy\beta\\ \delta(q,B,p,y,R), & D_1=\alpha q,D_2=\alpha yp\\ \delta(q,B,p,y,L), & D_1=\alpha zq,D_2=\alpha pzy\\ 0, & \text{否则}\end{cases}.$$

$\prec_M(D_1,D_2)$ 表示 l-VTM M 的即时描述 D_1 经过一步移动变为 D_2 的程度.

在不致混淆的情况下，我们经常用 $\prec$ 表示 $\prec_M$. 显然，ID 的每一步移动都完全依赖量子转移函数 δ.

对一个 l-VTM M,类似于文献[3]对自动机的处理,有两种方式定义 M 识别的语言,分别称为宽度优先(width-first)识别方式与深度优先(depth-first)识别方式,下面分别予以介绍.

先介绍宽度优先(width-first)识别方式.

对 l-值关系 $\prec_M$,对自然数 n,归纳地定义 $\prec_M^n$ 如下:$\prec_M^0$ 定义为 $ID(M)\times ID(M)$ 上的恒等关系,即 $\prec_M^0(D,D)=1$,而在其它情形下 $\prec_M^0$ 取值为 0;$\prec_M^{n+1}=\prec_M\circ\prec_M^n$. 令 $\prec_M^*=\bigcup_{n=0}^{\infty}\prec_M^n$,也即,$\prec_M^*$ 表示 $\prec_M$ 的自反传递闭包.

则图灵机 M 以宽度优先方式接受的语言定义如下.

定义 3. 设 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$,则 Σ^* 上的 l-值(一元)识别谓词 $rec_M^{[W]}$ 定义为 $rec_M^{[W]}\in l^{\Sigma^*}$:$\forall\omega\in\Sigma^*$,

$$rec_M^{[W]}(\omega)\overset{\text{def}}{=}(\exists q_0,q\in Q)(\exists\alpha_1,\alpha_2\in\Gamma^*)(q_0\in I\wedge q\in F\wedge(q_0\omega,\alpha_1q\alpha_2)\in\prec_M^*),$$

其真值为

$$\lceil rec_M^{[W]}(\omega)\rceil=\bigvee\{I(q_0)\wedge\prec_M^*(q_0\omega,\alpha_1q\alpha_2)\wedge F(q):q_0,q\in Q,\alpha_1,\alpha_2\in\Gamma^*\}\quad(1)$$

下面再介绍深度优先(depth-first)识别方式.

令 $ID(M)^+$ 表示所有的 ID 序列(路径)之集,即

$$ID(M)^+=\bigcup\{D_0D_1\cdots D_k:k\geq1,D_0,D_1,\cdots,D_k\in ID(M)\},$$

其中 D_1、D_2 表示命题"D_1 经过一次移动变为 D_2",常记为"$D_1\vdash D_2$",其真值为 $\prec(D_1,D_2)$. 一般地,ID 序列 $c=D_0D_1\cdots D_k\in ID(M)^+$ 表示命题"D_0 经过 $D_1,\cdots,D_k$ k 次移动变为 D_k". 常记为"$D_0\vdash D_1\vdash D_2\cdots\vdash D_k$". 设 $c=D_0D_1\cdots D_k\in ID(M)^+$,若 $\prec(D_i,D_{i+1})>0(i=0,1,\cdots,k-1)$,则称 c 为有效路径;否则,称 c 为无效路径. 以下用 $b(c)$ 表示路径 c 的开头 ID;$s(c)$ 表示路径 c 的第 2 个 ID,即 $b(c)$ 经过一次移动变为的 ID;$e(c)$ 表示路径 c 的末尾 ID.

$ID(M)^+$ 上的 l-值一元谓词 $path_M$ 定义为 $path_M\in l^{ID(M)^+}$:

$$path_M(c)\overset{\text{def}}{=}\bigwedge_{i=1}^{k}[D_{i-1}\vdash D_i],$$

其中,$c=D_0D_1\cdots D_k\in ID(M)^+$,其真值为

$$\lceil path_M(c)\rceil=\bigwedge_{i=1}^{k}[\prec(D_{i-1},D_i)].$$

注 1. 设 $c=D_0D_1\cdots D_k\in ID(M)^+$,若 c 为无效路径,则显然有 $\lceil path_M(c)\rceil=0$;反过来,若 $\lceil path_M(c)\rceil=0$,则 c 可能为无效路径,也可能为有效路径.

l-VTM M 在如下情况均停机:

(1) $\delta(q,x,p,y,d)=0$, $\forall p\in Q,y\in\Gamma,d\in\{L,R\}$. 此时,我们也说 $\delta(q,x)$ 无定义.

(2) 进入接受状态 q,即 $F(q)>0$.

则图灵机 M 以深度优先方式接受的语言定义如下.

定义 4. 设 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$,则 Σ^* 上 l-值(一元)识别谓词 $rec_M^{[D]}$ 定义为 $rec_M^{[D]}\in l^{\Sigma^*}$:$\forall\omega\in\Sigma^*$,

$$rec_M^{[D]}(\omega)\overset{\text{def}}{=}(\exists c\in ID(M)^+)(\exists q_0,q\in Q)(\exists\alpha_1,\alpha_2\in\Gamma^*)(q_0\in I\wedge q\in F\wedge path_M(c)\wedge(b(c)=q_0\omega)\wedge(e(c)=\alpha_1q\alpha_2)),$$

其真值为

$$\lceil rec_M^{[D]}(\omega)\rceil=\bigvee\{I(q_0)\wedge\lceil path_M(c)\rceil\wedge F(q):c\in ID(M)^+,b(c)=q_0\omega,\ e(c)=\alpha_1q\alpha_2,\ q_0,q\in Q,\ \alpha_1,\alpha_2\in\Gamma^*\}\quad(2)$$

特别地,若 M 为 l-VTMc,式(1)和式(2)可简化为

$$\lceil rec_M^{[W]}(\omega)\rceil=\lceil rec_M^{[D]}(\omega)\rceil=\bigvee\{F(q):c\in ID(M)^+,\ b(c)=q_0\omega,\ e(c)=\alpha_1q\alpha_2,\ q_0,q\in Q,\ \alpha_1,\alpha_2\in\Gamma^*\}.$$

更进一步,若 M 为 l-VDTM,式(1)和式(2)可简化为 $\lceil rec_M^{[W]}(\omega)\rceil=\lceil rec_M^{[D]}(\omega)\rceil=F(q)$,这里 $q\in Q$,存在唯一的 $c\in ID(M)^+$,使得 $b(c)=q_0\omega$,$e(c)=\alpha_1q\alpha_2$, $\alpha_1,\alpha_2\in\Gamma^*$.

因为 Σ^* 上的 l-值(一元)识别谓词 rec_M($rec_M^{[W]}$ 或者 $rec_M^{[D]}$)是一个 Σ^* 上的 l-值子集,因此,rec_M 可被看作 M 所识别的语言. 即 M 所识别的语言 rec_M:$\Sigma^*\to l$ 定义为 $rec_M(\omega)=\lceil rec_M(\omega)\rceil$, $\forall\omega\in\Sigma^*$.

我们说两个 l-VTM M_1,M_2 是等价的,是指它们识别相同的量子语言,即 $\forall\omega\in\Sigma^*$,总有

$$\overset{l}{\models}rec_{M_1}(\omega)\leftrightarrow rec_{M_2}(\omega).$$

下面定理表明,在量子逻辑意义下,图灵机以宽度优先和深度优先两种方式识别语言是不等价的,从而量子逻辑意义下的图灵机有自身的特色.

定理 1. 设 Σ 为输入字符集.

(1) 对任意的量子图灵机 M 以及输入串 $\omega\in\Sigma^*$,总有

$$\overset{l}{\models}rec_M^{[D]}(\omega)\to rec_M^{[W]}(\omega).$$

(2) 如下条件等价：

(2.1) l 是布尔代数；

(2.2) 对任意的 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$ 以及输入串 $\omega\in\Sigma^*$，总有 $\overset{l}{\models} rec_M^{[D]}(\omega)\leftrightarrow rec_M^{[W]}(\omega)$.

证明. (1) 由定义是显然的.

(2.1)⇒(2.2) 由分配性易证. 下证(2.2)⇒(2.1).

只需证：$\forall a,b,c\in l, a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$ 即可.

构造 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$，其中，$Q=\{q_0,q_1,q_2,q_3,q_4\}$，任取 $\sigma\in\Sigma$，$\delta: Q\times\Gamma\times Q\times\Gamma\times\{R,L\}\to l$ 定义为，$\delta(q_0,\sigma,q_1,\sigma,R)=a$，$\delta(q_1,\sigma,q_2,\sigma,R)=b$，$\delta(q_1,\sigma,q_3,\sigma,R)=c$，$\delta(q_2,\sigma,q_4,\sigma,R)=\delta(q_3,\sigma,q_4,\sigma,R)=1$，其它情况 δ 均取值 0；$I=\{q_0\}$，$F=\{q_4\}$ 均为单点集. 则 M 处理输入串 $\sigma\sigma$ 的有效路径有两条：$c_1=q_0\sigma\sigma\vdash\sigma q_1\sigma\vdash\sigma\sigma q_2\sigma\vdash\sigma\sigma\sigma q_4$，$c_2=q_0\sigma\sigma\vdash\sigma q_1\sigma\vdash\sigma\sigma q_3\sigma\vdash\sigma\sigma\sigma q_4$，从而，$rec_M^{[D]}(\sigma\sigma)=(a\wedge b\wedge 1)\vee(a\wedge c\wedge 1)=(a\wedge b)\vee(a\wedge c)$.

另外，$rec_M^{[W]}(\sigma\sigma)=\prec_M^*(q_0\sigma\sigma,\sigma\sigma q_4)=\prec_M^3(q_0\sigma\sigma,\sigma\sigma q_4)=\prec_M\circ\prec_M^2(q_0\sigma\sigma,\sigma\sigma q_4)=\prec_M(q_0\sigma\sigma,\sigma q_1\sigma)\wedge\prec_M^2(\sigma q_1\sigma,\sigma\sigma q_4)=a\wedge((\prec_M(\sigma q_1\sigma,\sigma q_2\sigma)\wedge\prec_M(\sigma\sigma q_2\sigma,\sigma\sigma q_4))\vee(\prec_M(\sigma q_1\sigma,\sigma q_3\sigma)\wedge\prec_M(\sigma\sigma q_3\sigma,\sigma\sigma q_4)))=a\wedge((b\wedge 1)\vee(c\wedge 1))=a\wedge(b\vee c)$. 由 $rec_M^{[W]}=rec_M^{[D]}$ 则得分配律 $a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$ 成立. 证毕.

下面先来说明 l-VTM 的初始状态可以取为分明单点集，此类图灵机记为 l-VTM$_0$，它们对后面的讨论有重要作用. 首先用 l-值子集作为状态的构造方法，来讨论 l-VTM 与 l-VTM$_0$ 之间的关系.

设 $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$ 为 l-VTM，由 M 构造一个 l-VTM$_0$ M' 如下. 令 $M'=(Q',\Sigma,\Gamma,\delta',q_0',B,F')$，其中，$Q'=\{I\}\cup Q_1$，这里 $Q_1=\{A\mid A: Q\to l$，存在唯一的 $q\in Q$，使得 $A(q)=1$，而在其它状态 A 取 0 值$\}$；$q_0'=I$；$F': Q'\to l$，$\forall A\in Q'$，

$$F'(A)=\begin{cases}F(q), & A\in Q_1\text{且存在唯一的 }q\in Q,\\ & \text{使得 } A(q)=1\\ 0, & A=I\end{cases},$$

$\delta': Q'\times\Gamma\times Q'\times\Gamma\times\{R,L\}\to l$，

$\forall(A,x,B,y,d)\in Q'\times\Gamma\times Q'\times\Gamma\times\{R,L\}$，

$$\delta'(A,x,B,y,d)=\begin{cases}\vee_{q,q_1\in Q}[A(q)\wedge\\ \delta(q,x,q_1,y,d)\wedge B(q_1)], & B\neq I\\ 0, & B=I\end{cases}.$$

M 与 M' 关系如下.

引理 1. (1) 对任意的 $\omega\in\Sigma^*$，总有

(1.1) $\overset{l}{\models} rec_M^{[W]}(\omega)\leftrightarrow rec_{M'}^{[W]}(\omega)$；

(1.2) $\overset{l}{\models} rec_M^{[D]}(\omega)\to rec_{M'}^{[D]}(\omega)$.

(2) 如下条件等价：

(2.1) l 是布尔代数；

(2.2) 对任意的 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$ 以及上述构造的 M'，总有 $\overset{l}{\models} rec_M^{[D]}(\omega)\leftrightarrow rec_{M'}^{[D]}(\omega)$.

证明. 由 M' 的构造很容易看到(1.1)是成立的. 下面证明(1.2). 设 $\omega\in\Sigma^*$，由 M' 的构造可知 $\lceil rec_{M'}^{[D]}(\omega)\rceil=\vee\{\lceil path(c')\rceil\wedge F'(A): c'\in ID(M')^+, A\in Q', \alpha_1,\alpha_2\in\Gamma^*, b(c')=q_0'\omega, e(c')=\alpha_1 A\alpha_2\}$，不妨设 $c'=D_0'D_1'\cdots D_{n-1}'D_n'(D_0'=q_0'\omega, D_n'=\alpha_1 A\alpha_2)$，则 $\lceil path_{M'}(c')\rceil=\bigwedge_{i=0}^k[\prec_{M'}^{[D]}(D_i',D_{i+1}')]$，这里，$\prec_{M'}^{[D]}(D_i',D_{i+1}')=\delta'(A_i,x_i,A_{i+1},x_{i+1},d_{i+1})(i=0,1,\cdots,n-1, A_0=I, A_n=A)$. 从而

$$\begin{aligned}&\lceil rec_{M'}^{[D]}(\omega)\rceil=\vee\{\lceil path(c')\rceil\wedge F'(A): c'\in ID(M')^+,\\ &\quad A\in Q',\alpha_1,\alpha_2\in\Gamma^*, b(c')=q_0'\omega, e(c')=\alpha_1 A\alpha_2\}\\ &=\vee\{[\vee_{q_0\in Q}I(q_0)\wedge\delta(q_0,x_0,q_1,x_1,d_1)]\wedge\\ &\quad\delta(q_1,x_1,q_2,x_2,d_2)\wedge\cdots\wedge\\ &\quad\delta(q_{n-1},x_{n-1},q_n,x_n,d_n)\wedge F(q_n): \text{存在唯一的}q_i\in\\ &\quad Q,\text{使得 } A_i(q_i)=1(i=1,\cdots,n)\}\\ &\geq\vee\{I(q_0)\wedge\delta(q_0,x_0,q_1,x_1,d_1)\wedge\\ &\quad\delta(q_1,x_1,q_2,x_2,d_2)\wedge\cdots\wedge\\ &\quad\delta(q_{n-1},x_{n-1},q_n,x_n,d_n)\wedge F(q_n): q_0\in Q,\\ &\quad\text{存在唯一的 } q_i\in Q,\text{使得 } A_i(q_i)=1(i=1,\cdots,n)\}\\ &=\vee\{I(q_0)\wedge\lceil path(c)\rceil\wedge F(q_n):\\ &\quad c\in ID(M)^+, q_0,q_n\in Q,\alpha_1,\alpha_2\in\Gamma^*, b(c)=q_0\omega,\\ &\quad e(c)=\alpha_1 q_n\alpha_2\}\\ &=\lceil rec_M^{[D]}(\omega)\rceil.\end{aligned}$$

即 $\overset{l}{\models} rec_M^{[D]}(\omega)\to rec_{M'}^{[D]}(\omega)$.

(2)(2.1)⇒(2.2). 若 l 满足分配律，则易证 $\lceil rec_M^{[D]}(\omega)\rceil=\lceil rec_{M'}^{[D]}(\omega)\rceil$，故

$$\overset{l}{\models} rec_M^{[D]}(\omega)\leftrightarrow rec_{M'}^{[D]}(\omega).$$

(2.2)⇒(2.1). 只需证：$\forall a,b,c\in l, a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$ 即可. 令 $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$，其中，$Q=\{q_0,q_1,q_2\}$，$\Sigma=\{\sigma_1,\sigma_2\}$，$\Gamma=\{B,\sigma_1,\sigma_2\}$；$I: Q\to l$，$I(q_0)=b$，$I(q_1)=c$，$I(q_2)=0$；$F: Q\to l$，$F(q_0)=0$，$F(q_1)=0$，$F(q_2)=a$；$\delta: Q\times\Gamma\times Q\times\Gamma\times\{R,L\}\to l$，$\delta(q_i,\sigma_1,q_2,\sigma_2,R)=1(i=0,1)$，其它情况 δ 均为 0. 则 M 处理 $\sigma_1\sigma_2$ 的有效路径为

$$c_1 = q_0\sigma_1\sigma_2 \vdash \sigma_2 q_2\sigma_2;$$
$$c_2 = q_1\sigma_1\sigma_2 \vdash \sigma_2 q_2\sigma_2.$$

从而有

$$\begin{aligned}\lceil rec_M^{[D]}(\sigma_1\sigma_2)\rceil &= (I(q_0)\wedge\lceil path(c_1)\rceil\wedge F(q_2))\vee\\ &\quad (I(q_1)\wedge\lceil path(c_2)\rceil\wedge F(q_2))\\ &= (b\wedge 1\wedge a)\vee(c\wedge 1\wedge a)\\ &= (b\wedge a)\vee(c\wedge a) = (a\wedge b)\vee(a\wedge c).\end{aligned}$$

注意到 M' 处理 $\sigma_1\sigma_2$ 的有效路径为

$$c' = I\sigma_1\sigma_2 \vdash \sigma_2 A_2\sigma_2.$$

从而有

$$\begin{aligned}&\lceil rec_{M'}^{[D]}(\sigma_1\sigma_2)\rceil = \lceil path(c')\rceil\wedge F(A_2)\\ &= \delta'(I,\sigma_1,A_2,\sigma_2,R)\wedge F(A_2)\\ &= (\vee_{q,q'\in Q}(I(q)\wedge\delta(q,\sigma_1,q',\sigma_2,R)\wedge A_2(q')))\wedge F(A_2)\\ &= ((I(q_0)\wedge\delta(q_0,\sigma_1,q_2,\sigma_2,R)\wedge A_2(q_2))\vee\\ &\quad (I(q_1)\wedge\delta(q_1,\sigma_1,q_2,\sigma_2,R)\wedge A_2(q_2)))\wedge F(A_2)\\ &= ((b\wedge 1\wedge 1)\vee(c\wedge 1\wedge 1))\wedge a\\ &= a\wedge(b\vee c).\end{aligned}$$

由 $rec_M^{[D]} = rec_{M'}^{[D]}$ 知 $a\wedge(b\vee c) = (a\wedge b)\vee(a\wedge c)$.

证毕.

上述引理表明,在利用 l-值子集作为状态子集的构造方法下,l-VTM 与 l-VTM$_0$ 以深度优先方式等价的充要条件为:l 是布尔代数.由于量子图灵机具有强大的模拟功能,我们也可以通过新的方法来研究量子图灵机各种变形之间的关系,且这种方法不依赖真值格的分配律,但用此法新构造的量子图灵机的规模(状态数)要比利用引理 1 的方法新构造的量子图灵机的规模大.在引理 1 中,M' 的规模为 $|Q|+1$,而在引理 2 中我们可以看到 M' 的规模为 $|Q|+2$.

引理 2. 对任意的 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$,一定存在一个 l-VTM$_0$ M',使得 $\forall\omega\in\Sigma^*$,总有 $\overset{l}{\models} rec_M^{[D]}(\omega)\leftrightarrow rec_{M'}^{[D]}(\omega)$.

证明. 令 $M'=(Q',\Sigma,\Gamma,\delta',p_0,B,F')$,其中,$Q'=Q\cup\{p_0,p_1\}$,$p_0,p_1\notin Q$;$F':Q'\to l$ 定义为:$F'(p_0)=0$,$F'(p_1)=0$,$F'(q)=F(q)$,$\forall q\in Q$;而 $\delta':Q'\times\Gamma\times Q'\times\Gamma\times\{L,R\}\to l$ 定义为:$\forall(q,x,q',x',d)\in Q\times\Gamma\times Q\times\Gamma\times\{L,R\}$,$\delta'(q,x,q',x',d)=\delta(q,x,q',x',d)$,$\forall x\in\Gamma$,$q\in Q$,$\delta'(p_0,x,p_1,x,R)=1$,$\delta'(p_1,x,q,x,L)=I(q)$,其它情况 δ' 的真值均为 0.从而对 $\omega\in\Sigma^*$ 有

$$\begin{aligned}&\lceil rec_{M'}^{[D]}(\omega)\rceil = \vee\{\lceil path(c')\rceil\wedge F'(q): c'\in ID(M')^+,\\ &\quad q\in Q',\alpha_1,\alpha_2\in\Gamma^*, b(c')=p_0\omega,\ e(c')=\alpha_1 q\alpha_2\}\\ &= \vee\{1\wedge I(q_0)\wedge\lceil path(c)\rceil\wedge F(q): c\in ID(M)^+,\\ &\quad q_0,q\in Q,\alpha_1,\alpha_2\in\Gamma^*, b(c)=q_0\omega,\ e(c)=\alpha_1 q\alpha_2\}\\ &= \vee\{I(q_0)\wedge\lceil path(c)\rceil\wedge F(q): c\in ID(M)^+, q_0,\\ &\quad q\in Q,\alpha_1,\alpha_2\in\Gamma^*, b(c)=q_0\omega,\ e(c)=\alpha_1 q\alpha_2\}\\ &= \lceil rec_M^{[D]}(\omega)\rceil.\end{aligned}$$

证毕.

下面以 l-VTM$_0$ 为桥梁来讨论 l-VTMc 与 l-VTM 的关系,即先讨论 l-VTMc 与 l-VTM$_0$ 的关系.为此,需要下面的引理.

引理 3[13]. 设 l 为一个格,X 为 l 的有限子集,则

(1) 由 X 生成的 l 的 $\wedge$-半格 $X_\wedge=\{x_1\wedge x_2\wedge\cdots\wedge x_k: x_1,x_2,\cdots,x_k\in X,k\geq 1\}\cup\{1\}$ 是有限的;

(2) 由 X 生成的 l 的 $\vee$-半格 $X_\vee=\{x_1\vee x_2\vee\cdots\vee x_k: x_1,x_2,\cdots,x_k\in X,k\geq 1\}\cup\{0\}$ 是有限的.

设 $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$ 为一个 l-VTM$_0$,令 $X=Im(\delta)\cup Im(F)\cup\{0,1\}$,其中,$Im(A)$ 表示 l-值子集 A 的象集.显然,X 为 l 的有限子集.取 $l_1=X_\wedge$,由引理 3 知 l_1 也是 l 的有限子集.因此,我们可以构造一个 l-VTMc 如下

$$M^C=(Q^C,\Sigma,\Gamma,\delta^C,q_0^C,B,F^C),$$

其中,$Q^C=Q\times l_1$;$q_0^C=(q_0,1)$;

$\delta^C: Q^C\times\Gamma\to 2^{Q^C\times\Gamma\times\{L,R\}}$,

$\forall(q,r)\in Q\times l_1, x\in\Gamma$,

$\delta^C((q,r),x)=\cup\{((q',r\wedge r'),y,d):\delta(q,x,q',y,d)=r'>0, q'\in Q,y\in\Gamma,d\in\{L,R\}\}$.

由 l_1 定义知,$\delta^C((q,r),x)\in 2^{Q^C\times\Gamma\times\{L,R\}}$,$\forall(q,r)\in Q^C, x\in\Gamma$;$F^C:Q^C\to l$ 定义为:$\forall(q,r)\in Q\times l_1$,$F^C((q,r))=r\wedge F(q)$.

如下定理说明 M 与 M^C 是等价的.

定理 2. 对任意的 l-VTM $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$,总有 $\forall\omega\in\Sigma^*$,$\overset{l}{\models} rec_M^{[D]}(\omega)\leftrightarrow rec_{M^C}^{[D]}(\omega)$.

证明. $\forall\omega\in\Sigma^*$,在 l-VTMc M^C 中,由 δ^C 定义知,由 $ID(q_0,1)\omega$ 出发的所有路径 c^C 完全由 δ 决定,且若 $e(c^C)=\alpha_1(q,r)\alpha_2$,则其唯一对应着 l-VTM M 中由 $IDq_0\omega$ 出发的一条路径 c,且 $e(c)=\alpha_1 q\alpha_2$,$r=\lceil path(c)\rceil$.从而 $\forall\omega\in\Sigma^*$,有

$$\begin{aligned}&\lceil rec_{M^C}^{[D]}(\omega)\rceil = \vee\{F^C((q,r)): c^C\in ID(M^C)^+,\\ &\quad (q,r)\in Q^C,\alpha_1,\alpha_2\in\Gamma^*, b(c^C)=q_0^C\omega,\\ &\quad e(c^C)=\alpha_1(q,r)\alpha_2\}\\ &= \vee\{r\wedge F(q): c\in ID(M)^+, q\in Q,\alpha_1,\alpha_2\in\Gamma^*,\\ &\quad b(c)=q_0\omega, e(c)=\alpha_1 q\alpha_2\}\\ &= \vee\{\lceil path(c)\rceil\wedge F(q): c\in ID(M), q\in Q,\alpha_1,\\ &\quad \alpha_2\in\Gamma^*, b(c)=q_0\omega, e(c)=\alpha_1 q\alpha_2\}\\ &= \lceil rec_M^{[D]}(\omega)\rceil.\end{aligned}$$

证毕.

由引理 2 及定理 2 易知有如下定理.

定理 3.　对任意的 l-VTM $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$，总存在与之等价的具有分明移动的 l-VTMc $M'=(Q',\Sigma,\Gamma,\delta',q_0',B,F)$，使得 $\forall\omega\in\Sigma^*,\overset{l}{\models}rec_M^{[D]}(\omega)\leftrightarrow rec_{M'}^{[D]}(\omega)$.

注 2.　定理 3 是一个非常强的结果，它表明：在深度优先方式识别语言意义下，量子逻辑意义下的图灵机与经典图灵机的差别恰好体现在接受状态的量子真值上，下面我们将看到此差别将本质地改变量子图灵机的性质. 另外，该定理也表明了把 l-VTMc 作为基于量子逻辑的图灵机的一种简单形式是合理的. 另外，l-VTMc 的定义形式也表明了基于量子逻辑的图灵机可以通过经典的图灵机和带有量子特性的终状态加以实现，量子逻辑意义下的图灵机与经典图灵机具有紧密联系与本质区别. l-VTMc 与 l-VTM 的等价性还说明了量子逻辑意义下的图灵机可以处理量子语言，但在处理的过程中我们可以直接使用经典图灵机的有关技术. 当然，M^C 的构造表明：虽然 l-VTM 与经典图灵机有紧密的联系，l-VTM 可以转化为 l-VTMc，但转化的复杂性不仅与原 l-VTM 的状态数目有关，而且与取值格 l 的大小有关. 注意到 l 可以很大，从而这种转化的复杂性很大. 因而，从实现的角度来看，l-VTM 更易于量子逻辑意义下的实现. 虽然有这种不便，但这种等价性确实为我们下面研究 l-VTM 的性质带来了极大方便，特别地，我们可以不依赖于分配性来讨论 l-VTMc（从而也是 l-VTM）的性质. 另外，由定理 1 我们知道，在宽度优先识别语言意义下，l-VTM 与 l-VTMc 一般不再等价，该注的说明也不再成立.

下面主要研究以深度优先方式识别语言的量子图灵机的性质，而针对宽度优先方式识别语言的量子图灵机的性质只在结论部分给以简单说明，更细致的性质另文研究.

首先研究基于量子逻辑的一些图灵机的变形之间的关系.

在定义 1 中，我们对转移函数 δ 做小的变动，允许读写头不动，得到量子图灵机的一个变形 l-VTM$_S$ $M_S=(Q,\Sigma,\Gamma,\delta,I,B,F)$ 其中，$\delta:Q\times\Gamma\times Q\times\Gamma\times\{L,S,R\}\to l$，但其识别语言的能力并未改变，我们可以构造一个与之等价的 l-VTM $M=(Q_1,\Sigma,\Gamma,\delta_1,I_1,B,F_1)$. 其中 $\forall(q,x,q',x',d)\in Q\times\Gamma\times Q\times\Gamma\times\{L,S,R\}$，若 $\delta(q,x,q',x',d)$ 有定义且 $d\neq S$，则 $\delta_1(q,x,q',x',d)=\delta(q,x,q',x',d)$，若 $\delta(q,x,q',x',d)$ 有定义且 $d=S$，引入一个相应的新状态 $p_{\delta(q,x,q',x',S)}\notin Q$，用两个转移来代替“不动”，即 $\delta_1(q,x,p_{\delta(q,x,q',x',R)},x',R)=\delta(q,x,q',x',S)$，$\forall y\in\Gamma,\delta_1(p_{\delta(q,x,q',x',S)},y,q',y,L)=1$，其它情况 δ_1 取值均为 0.

$Q_1=Q\cup\{$上述过程中引入的所有新状态$\}$；

$$I_1:Q_1\to l,\ \forall q\in Q_1,I_1(q)=\begin{cases}I(q), & q\in Q\\ 0, & \text{否则}\end{cases};$$

$$F_1:Q_1\to l,\ \forall q\in Q_1,F_1(q)=\begin{cases}F(q), & q\in Q\\ 0, & \text{否则}\end{cases}.$$

易验证，$rec_{M_S}=rec_M^{[D]}$.

基于量子逻辑的多带图灵机和 l-VTM 类似定义，只是有多个带子，每个带子有自己的读写头. 开始时，输入出现在第 1 个带子上，其它带子都是空白. 转移函数改为允许同时进行读、写和移动读头. 其形式定义如下.

定义 5.　一个基于量子逻辑的 k 带图灵机（简记为 k 带 l-VTM）$M(k\geq 2)$ 是一个七元组 $M=(Q,\Sigma,\Gamma,\delta,I,B,F)$，其中，$Q,\Sigma,\Gamma,I,B,F$ 和定义 1 中相同，转移函数 δ 定义为

$$\delta:Q\times\Gamma^k\times Q\times(\Gamma\times\{L,R\})^k\to l.$$

特别地，若 $I=q_0\in Q$，转移函数 δ 是一个 $Q\times\Gamma^k$ 到 $P(Q\times(\Gamma\times\{L,R\})^k)$ 或 $Q\times(\Gamma\times\{L,R\})^k$ 的部分函数，则称 M 为一个 k 带 l-VTMc 或 k 带 l-VDTM.

与定义 1 类似，我们可以给出描述 k 带 l-VTM 的原子命题及其真值. 下面主要是给出其即时描述 ID 形式.

一个 k 带 l-VTM 的一个 ID D 是一个 $(k+1)$-元组：$(q,u_1v_1,\cdots,u_kv_k)$，其中，q 是当前状态，第 i 个带上串为 u_iv_i，第 i 个带上的读头正注视着 v_i 的最左字符.

其余的定义、符号全部沿用 l-VTM 的. 显然，以深度优先方式所识别的语言 $rec_M^{[D]}\in l^{\Sigma^*}$ 可以定义为

$$\lceil rec_M^{[D]}(\omega)\rceil=\bigvee\{I(q_0)\wedge\lceil path(c)\rceil\wedge F(q):$$
$$c\in ID(M)^+,b(c)=(q_0,\omega,B,\cdots,B),$$
$$e(c)=(q,u_1v_1,\cdots,u_kv_k),$$
$$q_0,q\in Q,u_iv_i\in\Gamma^*,i=1,2,\cdots,k\}.$$

类似地也可给出 k 带 l-VTMc 和 k 带 l-VDTM 所识别的语言以及 k 带 l-VTM 以宽度优先方式所识别的语言. 相对于多带图灵机来说，前面的图灵机

都叫做单带图灵机，用和经典图灵机类似的证明方法[14-15]：用单带图灵机的有限步移动来模拟多带图灵机的一步移动，不同的是：在单带图灵机 l-VTM 的有限步移动中，我们规定最后一步移动的真值为其所模拟的多带图灵机 k 带 l-VTM 的一步移动的真值，其余每步移动的真值均为 1. 这样，两者处理任何一个输入串的有效路径是一一对应的且真值都是相同的. 从而有如下定理.

定理 4. k 带 l-VTM（k 带 l-VTMc 或 k 带 l-VDTM）与 l-VTM（l-VTMc 或 l-VDTM）等价.

注 3. 我们还可以像单带图灵机那样，允许多带图灵机的读头不动，则类似有 k 带 l-VTM_S、k 带 l-VTMc_S、k 带 l-VDTM_S 识别语言的能力分别和 k 带 l-VTM、k 带 l-VTMc、k 带 l-VDTM 相等.

另外，我们还可以给出基于量子逻辑的双向无穷带图灵机 l-TVTM 及带分明转移的双向无穷带图灵机 l-TVTMc、确定型双向无穷带图灵机 l-TVDTM，并类似地可以证明其识别语言的能力分别和 l-VTM、l-VTMc、l-VDTM 相等.

3 量子递归可枚举语言及量子递归语言的层次刻画

定义 6. 一个 l-值子集 $A:\Sigma^*\to l$ 称为量子递归可枚举（简记为：l-$r.e.$）语言，若存在 l-VTM M 使得 $A=rec_M^{[D]}$. 进一步，若对任意的 $\omega\in\Sigma^*$，M 总停机，则称 A 为量子递归语言. 显然，量子递归语言是量子递归可枚举语言的子类.

量子递归可枚举语言也称为量子图灵机以深度优先方式可识别的语言（即可以被一个 l-VTM 以深度优先方式所识别的语言），量子递归语言也称为量子图灵机以深度优先方式可判定的语言（即可以被一个 l-VTM 以深度优先方式所判定的语言）.

在经典图灵机中，确定型和非确定型的图灵机所识别的语言是等价的，即它们所识别的语言都是递归可枚举语言，但在量子图灵机中，确定型和非确定型的量子图灵机识别语言的能力是不相同的. 为了给出二者之间的关系，我们先给出量子递归可枚举语言、量子递归语言的一些性质及刻画. 由定理 2 及定理 3 知，给定一个量子递归可枚举语言，可以设计一个 l-VTM 来识别它，也可以设计一个 l-VTMc 来识别它，或者设计一个 k 带 l-VTM（k 带 l-VTMc）带来识别它，更多的时候，我们采用 l-VTMc 或 k 带 l-VTMc，因为它们的转移是分明的，模拟起来比较容易（实际上，在第 4 节讨论 l-VTM 的通用性的时候，也可以看到采用的模型均为 l-VTMc 或 k 带 l-VTMc）.

引理 4. 两个 l-$r.e.$语言（量子递归语言）的并还是 l-$r.e.$语言（量子递归语言）.

证明. 设 A_1,A_2 是 l-VTMc $M_1=(Q_1,\Sigma,\Gamma_1,\delta_1,q_{01},B,F_1)$ 及 $M_2=(Q_2,\Sigma,\Gamma_2,\delta_2,q_{02},B,F_2)$ 分别所识别的量子语言，不妨设 $Q_1\cap Q_2=\varnothing$. 我们构造一个新的 l-VTMc $M=(Q_1\cup Q_2,\Sigma,\Gamma_1\cup\Gamma_2,\delta,q_0,B,F)$，其中 $q_0\notin Q_1\cup Q_2$；$\delta(q_0,x)=\delta_1(q_{01},x)\cup\delta_1(q_{02},x)$，$\delta(q,x)=\delta_i(q,x)$，若 $q\in Q_i$，$i=1,2$；$F(q)=F_i(q)$，若 $q\in Q_i$，$i=1,2$，$F(q_0)=0$. 则 $\forall\omega\in\Sigma^*$，由上述定义知 ω 引导 M 由 $ID q_0\omega$ 有选择地进入 M_1 的 $ID\ D_1$（此时，$q_0\omega D_1=q_{01}\omega D_1\in ID(M_1)^+$）或 M_2 的 $ID\ D_2$（此时，$q_0\omega D_2=q_{02}\omega D_2\in ID(M_2)^+$），而以后的 ID 变动完全由 M_1 或 M_2 决定. 从而有

$$\begin{aligned}
&\lceil rec_M^{[D]}(\omega)\rceil=\bigvee\{F(q):c\in ID(M)^+,b(c)=q_0\omega,\\
&\quad e(c)=\alpha_1q\alpha_2,q\in Q,\alpha_1,\alpha_2\in\Gamma^*\}\\
&=\bigvee\{F(q):c\in ID(M)^+,b(c)=q_0\omega,s(c)\in ID(M_1)\\
&\text{或 }s(c)\in ID(M_2),e(c)=\alpha_1q\alpha_2,q\in Q,\alpha_1,\alpha_2\in\Gamma^*\}\\
&=\bigvee\{F_1(q):c\in ID(M)^+,b(c)=q_0\omega,s(c)\in\\
&\quad ID(M_1),e(c)=\alpha_1q\alpha_2,q\in Q_1,\alpha_1,\alpha_2\in\Gamma^*\}\\
&\vee\bigvee\{F_2(q):c\in ID(M)^+,b(c)=q_0\omega,s(c)\in\\
&\quad ID(M_2),e(c)=\alpha_1q\alpha_2,q\in Q_2,\alpha_1,\alpha_2\in\Gamma^*\}\\
&=\bigvee\{F_1(q):c\in ID(M_1)^+,b(c)=q_{01}\omega,\\
&\quad e(c)=\alpha_1q\alpha_2,q\in Q_1,\alpha_1,\alpha_2\in\Gamma^*\}\\
&\vee\bigvee\{F_2(q):c\in ID(M_2)^+,b(c)=q_{02}\omega,\\
&\quad e(c)=\alpha_1q\alpha_2,q\in Q_2,\alpha_1,\alpha_2\in\Gamma^*\}\\
&=A_1(\omega)\vee A_2(\omega).
\end{aligned}$$

即 $rec_M^{[D]}=A_1\cup A_2$.

显然，若 M_1,M_2 对任意的输入都停机，则 M 对任意的输入也都停机，即量子递归语言的并还是量子递归语言. 证毕.

下面给出后面将要用到的一些符号. 设 X 为一个集合，A 为 X 上的 l-值集合，即 $A:X\to l$，令 $R(A)=\{A(x):x\in X,A(x)>0\}$. 对 $r\in l$，令 $A_r=\{\omega\in\Sigma^*:A(\omega)\geq r\}$，$A_{[r]}=\{\omega\in\Sigma^*:A(\omega)=r\}$，分别称为 l-值集合 A 的 r-截集与 r-层集. 令 $supp(A)=\{x\mid x\in X,A(x)>0\}$，称为 l-值集合 A 的支撑集.

定理 5. 设 $A:\Sigma^*\to l$ 为量子语言，以下命题等价：

(1) A 为 l-$r.e.$语言；

(2) 存在 l 的有限子集$\{r_1,r_2,\cdots,r_k\}$及 Σ^* 上的有限个 $r.e.$语言$\{L_1,L_2,\cdots,L_k\}$使得 $A=\bigcup_{i=1}^{k} r_i 1_{L_i}$，其中，$1_{L_i}\in l^{\Sigma^*}$，$\forall\omega\in\Sigma^*$，$1_{L_i}(\omega)=\begin{cases}1, & \omega\in L_i\\ 0, & \text{否则}\end{cases}$；

(3) $R(A)$有限，且 $\forall r\in R(A)$，A_r为 $r.e.$.

证明.　(1)⇒(2). 不妨设 $A=rec_M^{[D]}$，$M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$为 l-VTMc，$R(F)=\{r_1,r_2,\cdots,r_k\}$为有限集，则 $M_{[r_i]}=(Q,\Sigma,\Gamma,\delta,q_0,B,F_{[r_i]})$为经典图灵机，其中，$F_{[r_i]}=\{q\in Q: F(q)=r_i\}$. 令 $L_i=L(M_{[r_i]})$($M_{[r_i]}$所识别的语言)，显然，L_i为 $r.e.$语言，$i=1,2,\cdots,k$. 从而有，$\forall\omega\in\Sigma^*$，

$$\begin{aligned}A(\omega)&=\lceil rec_M^{[D]}(\omega)\rceil=\vee\{F(q): c\in ID(M)^+,\\ &\quad b(c)=q_0\omega, e(c)=\alpha_1 q\alpha_2, q\in Q,\alpha_1,\alpha_2\in\Gamma^*\}\\ &=\vee\{r_i: c\in ID(M)^+, b(c)=q_0\omega, e(c)=\alpha_1 q\alpha_2, q\in Q\\ &\quad \text{且 } F(q)=r_i,\alpha_1,\alpha_2\in\Gamma^*\}\\ &=\vee\{r_i: c\in ID(M)^+, b(c)=q_0\omega, e(c)=\alpha_1 q\alpha_2,\\ &\quad q\in F_{[r_i]},\alpha_1,\alpha_2\in\Gamma^*\}\\ &=\vee\{r_i:\omega\in L_i\}=\vee_{i=1}^{k} r_i 1_{L_i}(\omega).\end{aligned}$$

即 $A=\bigcup_{i=1}^{k} r_i 1_{L_i}$.

(2)⇒(3). 因 $A=\bigcup_{i=1}^{k} r_i 1_{L_i}$，所以 $R(A)\subseteq\{r_1,r_2,\cdots,r_k\}_\vee$，由引理 2 知 $R(A)$有限，$\forall r\in R(A)$，

$$\begin{aligned}A_r&=\{\omega\mid A(\omega)\geq r\}\\ &=\{\omega\mid\vee_{i=1}^{k} r_i 1_{L_i}(\omega)\geq r\}\\ &=\{\omega\mid\vee\{r_i\mid\omega\in L_i\}\geq r\}\\ &=\{\omega\in\cap_{j=1}^{t} L_{i_j}\mid\vee_{j=1}^{t} r_{i_j}\geq r\}\\ &=\cup\{\cap_{j=1}^{t} L_{i_j}\mid\vee_{j=1}^{t} r_{i_j}\geq r\}.\end{aligned}$$

而 $r.e.$语言关于有限交、有限并封闭，因此 A_r为 $r.e.$语言.

(3)⇒(1). 设 A_r为 $r.e.$语言，则存在经典图灵机 $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$使得 $L(M)=A_r$. 进一步易验证，$r1_{A_r}$可以被一个 l-VTMc $M'=(Q,\Sigma,\Gamma,\delta,q_0,B,r1_F)$识别，也即 $r1_{A_r}$为 l-$r.e.$语言. 由引理 4 知，l-$r.e.$语言关于有限并封闭，从而 $A=\bigcup_{r\in R(A)} r1_{A_r}$为 l-$r.e.$语言.　　证毕.

类似于文献[16]定理 3 的证明，可给出 l-VDTM所识别语言的如下层次刻画.

定理 6.　设 $A:\Sigma^*\to l$ 为量子语言，以下命题等价：

(1) A 可以被 l-VDTM 所识别；

(2) $R(A)$有限，且 $\forall a\in R(A)$，$A_{[a]}$为 $r.e.$语言.

基于量子逻辑的图灵机 l-VDTM 与 l-VTM 不等价，即并不是所有的 l-$r.e.$语言都满足定理 6 的条件，由此给出基于量子逻辑的图灵机与经典图灵机的一个区别. 反例如下.

例 1.　设 l 为一完备正交模格，$|l|>2$，取 $a\in l$ 使得 $a\neq 0,1$. 令 $L\subseteq\Sigma^*$ 为递归可枚举语言但非递归语言，从而 Σ^*-L 非递归可枚举语言，则一定存在一个识别它的经典确定型图灵机 $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$. 下面构造一个 l-VTM $M'=(Q\cup\{p_f\},\Sigma,\Gamma,\delta',q_0,B,F\cup\{p_f\})$，其中，$p_f\notin Q$，$\delta':(Q\cup\{p_f\})\times\Gamma\times(Q\cup\{p_f\})\times\Gamma\times\{L,R\}\to l$ 定义如下：

$\forall(q,x,p,y,d)\in Q\times\Gamma\times Q\times\Gamma\times\{L,R\}$，若 $\delta(q,x)=(p,y,d)$，则 $\delta'(q,x,p,y,d)=1$，若 $\delta(q,x)$无定义，则 $\delta'=0$；$\forall(q,x)\in(Q-F)\times\Gamma$，$\delta'(q,x,p_f,x,R)=a$，其余情况 $\delta'=0$.

令 $rec_{M'}^{[D]}=A$. 则易验证：$A_{[1]}=L$，$A_{[a]}=\Sigma^*-L$，但 Σ^*-L 不是递归可枚举语言，由定理 6 知，A 不能被任何 l-VDTM 所识别.

此例也说明了，l-VDTM 所识别的语言关于并运算不封闭. 因为：令 $A_1=L$ 可以被一个经典确定型图灵机所识别，则它一定也可以被一个 l-VDTM 所识别；令 $A_2=rec_{M_2}$，这里，$M_2=(Q,\Sigma,\Gamma,\delta_2,q_0,B,F_2)$，$F_2=Q$；$\delta_2:Q\times\Gamma\to Q\times\Gamma\times\{L,R\}$，定义如下：对任意的 $(q,x)\in Q\times\Gamma$，都存在唯一的 $(p,y,d)\in Q\times\Gamma\times\{L,R\}$，使得 $\delta_2(q,x)=(p,y,d)$，显然，M_2为一个 l-VDTM，且有 $A=A_1\cup A_2$，但是 A 不能被任何 l-VDTM 所识别.

虽然 l-VTM 与 l-VDTM 不等价，但作为量子递归语言的识别器，l-VTM 与 l-VDTM 等价，即如下定理.

定理 7.　设 $A:\Sigma^*\to l$ 为一量子语言，则如下命题等价：

(1) A 是可以被一个 l-VTM 以深度优先方式所判定的语言(即 A 是一个量子递归语言)；

(2) $R(A)$有限，且 $\forall r\in R(A)$，A_r是一个递归语言；

(3) $R(A)$有限，且 $\forall r\in R(A)$，$A_{[r]}$是一个递归语言；

(4) A 是可以被一个 l-VDTM 所判定的语言.

证明.　(1)⇔(2)与定理 5(1)⇔(3)证明类似. 只要注意到量子递归语言的有限并还是量子递归语言(引理 4).

(3)⇔(4)与定理 6(1)⇔(2)证明类似. 只要注意到 l-VDTM 所判定的语言关于有限并还封闭(参见文献[14]引理 4).

(4)⇔(1). 因为任何一个 l-VDTM 都是 l-VTM 的特例，因此，对任何一个 l-VDTM 所判定的语言，一定也存在某个 l-VTM 可判定它.

下面只需证明(2)⇒(3),这是因为,$\forall r\in R(A)$,$A_{[r]}=A_r-\bigcup\{A_s \mid s\in R(A)$ 且 $s\geq r$ 但 $s\neq r\}$.

注意到递归语言关于有限并和补运算都是封闭的,从而 $A_{[r]}$ 是一个递归语言. 证毕.

下面讨论量子语言的补语言的性质.

定义 7. 设 $A:\Sigma^*\to l$ 为任一量子语言,A 的补定义为 $A^\perp:\Sigma^*\to l$,$\forall\omega\in\Sigma^*$,$A^\perp(\omega)=A(\omega)^\perp$.

定理 8. 设 $A:\Sigma^*\to l$ 为任一量子递归语言,则 $A^\perp$ 也是一量子递归语言.

证明. 由定理 7,不妨设 $R(A)=\{r_1,r_2,\cdots,r_n\}$ 且 $R(A)$ 中仅有 $k(\leq n)$ 个元素 $r_{i_1},\cdots,r_{i_k}$ 满足 $0<r_{i_1},\cdots,r_{i_k}<1$,则 $A_{[r_i]}=L_i$,L_i 为量子递归语言,且 $R(A^\perp)=\{r_{i_1}^\perp,\cdots,r_{i_k}^\perp,1\}$ 或 $R(A^\perp)=\{r_{i_1}^\perp,\cdots,r_{i_k}^\perp\}$. 由 $A^\perp$ 的定义知,$A^\perp_{[1]}=\Sigma^*-\bigcup_{i=1}^n L_i$,$A^\perp_{[r_{i_k}^\perp]}=L_{i_k}(k=1,\cdots,m)$. 由于递归语言关于并运算及补运算都封闭,故 $A^\perp_{[1]}$,$A^\perp_{[r_{i_k}^\perp]}(k=1,\cdots,m)$ 均为递归语言,由定理 7 知,$A^\perp$ 也是量子递归语言. 证毕.

注 4. 量子递归可枚举语言关于补运算不封闭,反例如下.

例 2. 令 $A=rec_{M'}^{[D]}$(见例 1),A 是量子递归可枚举语言,则 $A^\perp:\Sigma^*\to l$,定义为

$$A^\perp(\omega)=\begin{cases}0, & \omega\in L\\ a^\perp, & \omega\in\Sigma^*-L\end{cases}.$$

显然,$R(A^\perp)=\{a^\perp\}$,$A^\perp_{a^\perp}=\Sigma^*-L$,因 L 是递归可枚举语言但非递归语言,故 $A^\perp_{a^\perp}=\Sigma^*-L$ 不是递归可枚举语言. 由定理 5 知,$A^\perp$ 不是量子递归可枚举语言.

4 基于量子逻辑的图灵机的通用性

我们知道通用图灵机是存在的,通用图灵机可以模拟所有的图灵机. 根据丘奇-图灵论题,通用图灵机是现代计算机的形式化模型. 对于基于量子逻辑的图灵机,是否也存在这样一个图灵机,它可以模拟其它任意类型的基于量子逻辑的图灵机的计算,这就是基于量子逻辑的通用图灵机. 正如通用图灵机奠定了计算机科学的基础一样,基于量子逻辑的通用图灵机的存在性对基于量子逻辑的图灵机的实现是非常重要的. 下面我们将讨论这个重要问题. 本节只对基于深度优先方式识别语言的量子图灵机进行研究,关于宽度优先的量子图灵机的通用性见结论部分的讨论.

一个基于量子逻辑的通用图灵机(l-VTM)M_u 是指当考察一个输入串 $\omega\in\Sigma^*$ 是否可以被一个给定的量子图灵机 M 接受时,就将这个给定的量子图灵机的编码和相应的输入串的编码 ω 连接起来,记作 $\langle M,\omega\rangle$,($\langle M,\omega\rangle\in\{0,1\}^*$,至于具体怎样进行编码后面会详细介绍)作为基于量子逻辑的通用图灵机 l-VTM M_u 的输入,由 M_u 去模拟给定的量子图灵机 M 的运行. 即有 $\overset{l}{\models} rec_{M_u}^{[D]}(\langle M,\omega\rangle)\leftrightarrow rec_M^{[D]}(\omega)$.

注意到,对任意的 $\omega\in\Sigma^*$,$a\in l$,一定存在一个量子图灵机 M 使得 $\lceil rec_M^{[D]}(\omega)\rceil=a$,因此,一旦所有的 l-VTMs 的编码都给定,不管 M_u 是否存在,$A_u:\{0,1\}^*\to l$ 都是存在的(定义好的),其定义如下:$\forall s\in\{0,1\}^*$,

$$A_u(s)=\begin{cases}\lceil rec_M^{[D]}(\omega)\rceil, & \text{若 } M \text{ 为 } l\text{-VTM 且 } s=\langle M,\omega\rangle\\ 0, & \text{否则}\end{cases}.$$

我们称 $A_u:\{0,1\}^*\to l$ 为 l-VTMs 的通用量子语言. 由上述讨论,显然有如下定理.

定理 9. $R(A_u)=\{A_u(s)\in l:A_u(s)>0,s\in 0,1^*\}=\{a\in l:A_u(s)=a=\lceil rec_M^{[D]}(\omega)\rceil>0,M$ 为 l-VTM 且 $s=\langle M,\omega\rangle,\omega\in\Sigma^*\}=l-\{0\}$.

由定理 9 知,若 l 是无限的,则 $R(A_u)$ 也是无限的,由定理 5 知,A_u 不能被任何量子图灵机所识别,从而 l-VTM M_u 不存在. 即如下结论.

推论 1. 若正交模格 l 是无限的,则 l-VTM M_u 不存在. 即不存在 l-VTM M_u,使得 $rec_{M_u}^{[D]}=A_u$.

那么正交模格 l 是有限的情况下 l-VTM M_u 存在吗?为方便起见,以下不妨假设 $l=\{0,a_1,\cdots,a_r\}$,其中 $a_r=1$. 由定理 2 知,对任何 l-VTM 都存在一个与之等价的 l-VTMc,因此,我们下面只考虑 l-VTMc. 为了给出其合理编码,需要如下引理来保证.

引理 5. 设 $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$ 为一个 l-VTMc,则存在 l-VTMc $M'=(Q',\Sigma,\Gamma,\delta',p_1,B,F')$ 满足如下条件:(1)Q' 至少包含 $r+1$ 个元素,使得 p_1 为初始状态,$p_2,\cdots,p_{r+1}$ 为接受状态,且 $F'(p_{i+1})=a_i,i=1,\cdots,r$;(2)对任意的 $\omega\in\Sigma^*$,都有 $\overset{l}{\models} rec_{M'}^{[D]}(\omega)\leftrightarrow rec_M^{[D]}(\omega)$.

证明. 假设 $p_2,\cdots,p_{r+1}\notin Q$,令 $Q'=Q\cup\{p_2,\cdots,p_{r+1}\}$,$p_1=q_0$,而 $F':Q'\to l$ 定义为,$F'(p_{i+1})=a_i(i=1,\cdots,r)$,$F'(q)=0(q\in Q)$. 转移函数 $\delta:Q'\times\Gamma\to Q'\times\Gamma\times\{L,R\}$ 定义为 $\forall(q,x)\in Q'\times\Gamma$,

$$\delta'(q,x)=\begin{cases}\delta(q,x), & F(q)=0 \text{ 且 } \delta(q,x) \text{ 有定义}\\ (p_i,x,R), & F(q)=a_i\\ \text{无定义}, & \text{否则}\end{cases}.$$

易验证，这样得到的 l-VTMc $M'=(Q',\Sigma,\Gamma,\delta',p_1,B,F')$ 满足条件(1)，(2).

引理 6. 设 l-VTMc $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$，$M_1=(Q,\Sigma,\Gamma\cup\{X_2\},\delta,q_0,B,F)$，且 $X_2\notin\Gamma$(显然 $X_2\notin\Sigma$)则有 $\forall\omega\in\Sigma^*,\overset{l}{\models}rec_{M_1}^{[D]}(\omega)\leftrightarrow rec_M^{[D]}(\omega)$.

证明. 显然 M_1 的运行和带符号 X_2 无关，和 M 的运行完全一致，从而结论成立.

我们知道，对一个量子图灵机，状态用什么符号表示不重要，关键是状态个数，因此我们可以给状态编号，均为 $\{q_1,q_2,\cdots,q_n\}$，又由引理 5 和 6，我们不妨设

$$M=(\{q_1,q_2,\cdots,q_n\},\Sigma,\Gamma\cup\{X_2\},\delta,q_1,B,F)$$

为任意一个量子图灵机，其中 $X_2\notin\Gamma$，$q_2,\cdots,q_{r+1}$ 都是接受状态，且 $F(q_{i+1})=a_i(i=1,\cdots,r)$.

下面介绍一种编码系统，思路为：

(1) 用 0 和 1 对除空白符以外的其它带符号进行编码，具体操作如下：首先给带符号排序，不妨设 $\Gamma\cup\{X_2\}=\{X_1,X_2,\cdots,X_m\}$，我们约定，$X_1$ 对应 0，X_2 对应分隔符 1，X_3 对应空白符 B，其它带符号(若还有的话)X_i 用编码 0^i(i 个 0)来表示.

(2) 用 0 和 1 对任意的输入 $\omega\in\Sigma^*$ 进行编码，具体操作如下：因 $X_2\notin\Sigma\subseteq\Gamma$，从而 $\forall\omega=X_{k_1},\cdots,X_{k_p}\in\Sigma^*$，其中 $X_{k_1},\cdots,X_{k_p}\in\Sigma$，$\omega$ 可用如下编码表示

$$0^{k_1}10^{k_2}1\cdots10^{k_p}\tag{3}$$

注意到，这种形式的编码和 Σ 上的字符串是一一对应的.

(3) 用 0 和 1 对图灵机的移动函数进行编码，具体操作如下：

用 D_1，D_2 分别表示 R，L，那么一个移动 $(q_k,X_l,D_h)\in\delta(q_i,X_j)$ 可以用如下编码表示

$$0^i10^j10^k10^l10^h\tag{4}$$

注意到，$i=1,r+1,\cdots,n$；$k=1,2,\cdots,n$；$j,l=1,2,\cdots,m$；$h=1,2$. 这样，可以用如下字符串表示 M

$$111code_1\,11code_2\,11\cdots11code_t111\tag{5}$$

其中，每个 $code_i$ 都是形如式(4)的字符串，且字符串(5)中包含 M 的所有移动编码. 进一步，M 的编码和它的一个输入串 ω 的编码连接，记作 $\langle M,\omega\rangle$，可以表示成

$$111code_1\,11code_2\,11\cdots11code_t1110^{k_1}10^{k_2}1\cdots10^{k_p}\tag{6}$$

可见，按照上述编码系统，无论 l 有限与否，我们都可以用 $\{0,1\}$ 上的串来表示所有的量子图灵机和输入，且因 $X_2\notin\Sigma$ 保证了分隔符 1 不会出现在任何量子图灵机的输入中. 当给定一个 l-VTMc M 及一个输入 ω 时，其编码 $\langle M,\omega\rangle$ 不止一个，但都具有形式

$$111code_{i_1}\,11code_{i_2}\,11\cdots11code_{i_t}1110^{k_1}10^{k_2}1\cdots10^{k_p},$$

其中 $i_1,i_2,\cdots,i_t$ 是 $1,2,\cdots,t$ 的一个排列. 而 $\{0,1\}$ 上的一个串最多表示一个 l-VTMc 及一个输入 ω 的编码，有的串不是任何 l-VTMc 及一个输入 ω 的编码. 为了区分开来，$\forall s\in\{0,1\}^*$，若 s 表示某个 l-VTMc 及一个输入 ω 的编码，则称 s 为有效编码，否则为无效编码.

例 3. 设 l 是一个正交模格 M_{O_6}(如图 1)，l-VTMc $M=(\{q_1,q_2,q_3,q_4\},\{X_1,X_4\},\{X_1,X_2,X_3,X_4\},\delta,q_1,B,\{(q_3,a_2),(q_4,1)\})$，其中，$\delta$ 定义为：$\delta(q_1,X_1)=\{(q_2,X_2,D_1),(q_3,X_1,D_2)\}$，$\delta(q_2,X_2)=\{(q_3,X_1,D_1),(q_4,X_1,D_1)\}$，其它情况 δ 取值为 0.

则 M 的编码如下：

$$11101010^210^2101101010^31010^2110^210^210^3$$
$$1010110^210^210^41010111.$$

输入 $X_1^2X_4=X_1X_1X_4$ 的编码如下：

$$01010000.$$

编码 $\langle M,\omega\rangle$ 为

$$11101010^210^2101101010^31010^2110^210^210^31010110^2$$
$$10^210^41010111010100000.$$

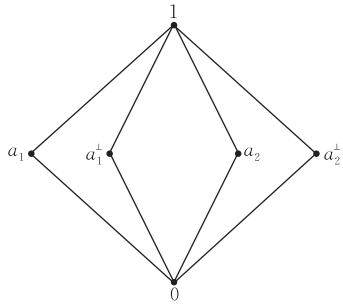


图 1　正交模格 M_{O_6}

注 5. 由 l-VTM 与 l-VTMc 的等价性知，l-VTMs 的通用量子语言定义为

$$A_u(s)=\begin{cases}\lceil rec_M^{[D]}(\omega)\rceil, & M\text{ 为 }l\text{-VTMc 且}\\ & s=\langle M,\omega\rangle\\ 0, & \text{否则}\end{cases}.$$

这样，在 l 有限的情况下，有了上述编码系统及引理 5 和 6，我们可以构造一个 l-VTMc M_u，其输入符号集就是 $\{0,1\}$，使得 M_u 能够识别 A_u. 即如下定理.

定理 10. 若正交模格 l 有限，则 A_u 是量子递归可枚举语言，即存在一个 l-VTMc M_u，使得 $rec_{M_u}^{[D]}=$

A_u. 即当正交模格 l 有限时，通用 l-VTMc 是存在的.

证明. 我们先构造一个接受 A_u 的 3 带 l-VTMc $M_u^3=(\{q_1,\cdots,q_n\},\{0,1\},\Gamma,\delta,q_1,B,F_u)$，接受态为 $q_2,\cdots,q_{r+1}$ 且 $F_u(q_{i+1})=a_i(i=1,\cdots,r)$. 它的第 1 条带为输入带，开始时，输入 $s\in\{0,1\}^*$ 只出现在第 1 条带上. 第 2 条带用来保存所模拟的 M 的带，使用与 M 的编码相同的格式. 也就是说，M 的带符号 X_i 表示成 0^i，带符号用单个间隔符 1 分隔. M_u^3 的第 3 条带保存 M 运行过程中的状态，状态 q_i 表示成 0^i. M_u^3 的运行如下：

(1) 检查输入 s，若 s 是某个 l-VTMc M 及其一个输入 ω 的有效编码，即 $s=\langle M,\omega\rangle$，则进行(2). 否则，停机不接受.

(2) 初始化第 2 条带，即以编码的形式将 M 的输入串 ω 抄写在第 2 条带上，也即将第 1 条带上出现第 2 个子串 111 以后的串抄写在第 2 条带上，进行(3).

(3) 把 0(M 的初始状态)写在第 3 条带上，把 M_u^3 的第 2 条带的读头移到最左端第 1 个被模拟单元，进行(4).

(4) 当第 3 条带当前的符号串为 0^i，第 2 个带头扫描的符号串为 0^j1 时. M_u^3 扫描第 1 条带从最左端直到第 2 个子串 111，查找以 110^i10^j1 开头的子串. 可能有不止 1 个子串以 110^i10^j1 开头. 利用 M_u^3 的非确定性，M_u^3 将和 M 在这一步所做的那样选择其中的 1 个这样的子串. 若这样的子串找到了，不妨设为 $110^i10^j10^k10^l10^h$，这时 M_u^3 把第 3 条带上字符改写为 B，然后第 3 条带上的读头回到最左端重新写上 0^k，让第 2 条带的读头写下符号 0^l 并按方向 D_h 移动该读头，进行(5). 否则，说明 M 没有这样的移动，这时 M_u^3 也停机不接受.

(5) 第 3 条带当前的符号串为 0^k，若 $2\leq k\leq r+1$，则 M_u^3 进入接受状态 q_k 停机，且 q_k 为接受状态的真值为 a_k. 否则进行(4).

在上述运行中 M_u^3 模拟了 M 在输入 ω 下的运行，即 $\forall s\in\{0,1\}^*$，若 $s=\langle M,\omega\rangle$，则

$c\in ID(M_u^3)^+$，$b(c)=(q_1,\langle M,\omega\rangle,B,B)$，$e(c)=(q_{i+1},u_1\ v_1,u_2\ v_2,u_3\ v_3)$，$F_u(q_{i+1})=a_i\Leftrightarrow c_M\in ID(M)^+$，$b(c)=q_1\omega$，$e(c)=\alpha_1q_{i+1}\alpha_2$，$F(q_{i+1})=a_i$ 且 $\alpha_1\alpha_2$ 的编码恰为 u_2v_2.

因此有 $\lceil rec_{M_u^3}^{[D]}(\langle M,\omega\rangle)\rceil=\lceil rec_M^{[D]}(\omega)\rceil$. 否则，$\lceil rec_{M_u^3}^{[D]}(s)\rceil=0$.

又由定理 3 知，存在一个 l-VTMc M_u 与 l-VTMc M_u^3 等价，从而有 $rec_{M_u}^{[D]}=A_u$.

由注 5 易知有如下推论.

推论 2. 当正交模格 l 有限时，通用 l-VTM 是存在的.

类似地，可以给出 l-VDTMs 的通用量子语言：$A_{ud}:\{0,1\}^*\to l$ 定义如下，$\forall s\in\{0,1\}^*$，

$$A_{ud}(s)=\begin{cases}\lceil rec_M^{[D]}(\omega)\rceil, & M\text{ 为 }l\text{-VDTM 且}\\ & s=\langle M,\omega\rangle\\ 0, & \text{否则}\end{cases}.$$

当正交模格 l 有限时，类似地有如下定理.

定理 11. 若正交模格 l 有限，则存在一个 l-VDTM M_{ud}，它可以模拟任何一个 l-VDTM. 即 $rec_{M_{ud}}^{[D]}=A_{ud}$. 我们称 M_{ud} 为通用 l-VDTM.

5 结论与讨论

本文研究了基于量子逻辑的图灵机. 由结论可以看出，基于量子逻辑的图灵机相较经典图灵机来说，一方面，它们所依赖的逻辑不同，一个是量子逻辑，一个是经典逻辑，因而所得结论既有相似的地方，又有本质的不同. 比如：在经典图灵机中，基于深度优先和基于宽度优先两种方式识别的语言是一致的，但在基于量子逻辑的图灵机中，这两种方式不再等价. 本文主要就基于深度优先的量子图灵机的有关性质进行了研究，研究表明：不同于经典图灵机，在基于量子逻辑的图灵机中，确定型图灵机与不确定型图灵机一般不是等价的，而且，通用图灵机只有在真值有限时才存在，一般情况下不再存在. 另一方面，本文结果也表明了基于量子逻辑的图灵机可以通过经典的图灵机和带有量子特性的终状态加以实现，量子逻辑意义下的图灵机与经典图灵机具有紧密联系与本质区别. 另外，在量子逻辑意义下，如定理 1 所表明的那样，基于宽度优先识别语言与基于深度优先识别语言一般不再等价，下面我们简单罗列一些基于宽度优先的量子图灵机的性质(这些都可以从论文有关基于深度优先量子图灵机的论证中推导出来)：定理 2 与定理 3 一般不再成立，基于宽度优先的量子图灵机一般不再能够用具有分明转移的量子图灵机模拟，其表现可能会更复杂一些. 因此，定理 5 一般也不再成立，这也就是说，基于宽度优先量子图灵机一般不再有如定理 5 那样好的层次刻画. 当然，定理 6 显然是成立的，从而也表明在基

于宽度优先的量子图灵机中，确定型图灵机与不确定型图灵机仍然是不等价的. 定理 4 对宽度优先的量子图灵机也是成立的. 通过适当的编码系统，定理 10 与定理 11 对基于宽度优先量子图灵机也是成立的，但对于一般的真值集(正交模格)，基于宽度优先量子图灵机的通用性还有待于进一步的讨论. 有关基于宽度优先量子图灵机的进一步性质，我们将另文讨论. 另外，本文的结论也表明，基于量子逻辑的图灵机(在深度优先识别语言情形下)与模糊图灵机有很多的相似性(参见文献[16-17])，量子逻辑意义下的计算模型与模糊计算模型具有很紧密的联系. 量子逻辑的核心作用除在宽度优先与深度优先方面的作用外还有待于进一步研究，这也构成另一个研究课题.

另外，基于传统的 Hilbert 空间上的量子图灵机早在 1985 年就由牛津大学的 Deutsch[18] 教授提出，后续的 Bernstern、Vazirani[19] 以及 Gudder[20] 做了更为深刻的研究. 这些研究揭示了量子计算的一些本质属性，成为目前量子计算的一个典型计算模型. 而本文提出的基于量子逻辑的图灵机理论是继应明生教授提出基于量子逻辑的计算模型的又一补充. 正如基于量子逻辑的自动机理论和基于传统的 Hilbert 空间上的量子自动机有紧密联系又有所区别，传统的 Hilbert 空间上的量子图灵机和基于量子逻辑的深度优先图灵机关系更为紧密，而与基于量子逻辑的宽度优先图灵机会有较大的区别. 我们将在后续工作中继续深入研究它们之间的关系.

参　考　文　献

[1] Nielsen M A, Chuang I L. Quantum Computation and Quantum Information. Cambridge: Cambridge University, 2000

[2] Ying M S. A theory of computation based on quantum logic (I). Theoretical Computer Science, 2005, 344: 134-207

[3] Ying M S. Quantum logic and automata theory. In Engesser K, Gabbay D M, Lehmann D eds. Handbook of Quantum Logic and Quantum Structures: Quantum Structures. Amsterdam: Elsevier, 2007: 619-754

[4] Ying M S. Automata theory based on quantum logic (I). International Journal of Theoretical Physics, 2000, 39: 981-991

[5] Ying M S. Automata theory based on quantum logic (II). International Journal of Theoretical Physics, 2000, 39: 2545-2557

[6] Qiu Dao-Wen. Notes on automata theory based on quantum logic. Science China Information Science, 2007, 37(6): 723-737(in Chinese)
(邱道文. 基于量子逻辑的自动机理论的一些注记. 中国科学 E 辑：信息科学，2007，37(6)：723-737)

[7] Qiu D W. Automata theory based on quantum logic: Reversibilities and pushdown automata. Theoretical Computer Science, 2007, 386: 38-56

[8] Lu R Q, Zheng H. Lattices of quantum automata. International Journal of Theoretical Physics, 2003, 42: 1425-1449

[9] Shang Y, Xian L, Lu R Q. Automata theory based on unsharp quantum logic. Mathematical Structures in Computer Science, 2009, 19(4): 737-756

[10] Li Yong-Ming. Finite automata based on quantum logic and monadic second-order quantum logic. Science China Information Sciences, 2009, 39(11): 1135-1145(in Chinese)
(李永明. 基于量子逻辑的有穷自动机与单体二阶量子逻辑. 中国科学 F 辑：信息科学，2009，39(11)：1135-1145)

[11] Birkhoff G, von Neumann J. The logic of quantum machines. Annals of Mathematics, 1936, 37: 823-843

[12] Kalmbach G. Orthomodular Lattices. London: Academic Press, 1983

[13] Li Y M, Li Z H. Free semilattices and strongly free semilattices generated by partially ordered sets. Northeastern Mathematical Journal, 1993, 9(3): 359-366

[14] Hopcroft J E, Ullman J D. Introduction to Automata Theory, Languages and Computation. New York: Addison-Wesley, 1979

[15] Sipser M. Translated by Zhang Li-Ang, Wang Han-Pin, Huang Xiong. Introduction to Computation Theory. Beijing: China Machine Press, 2002(in Chinese)
(Sipser M. 张立昂，王悍贫，黄雄译. 计算理论导引. 北京：机械工业出版社，2002)

[16] Li Yong-Ming. Approximation and universality of fuzzy Turing machines. Science China Information Sciences, 2008, 38(8): 1139-1352(in Chinese)
(李永明. 模糊图灵机的逼近性与通用性. 中国科学 E 辑：信息科学，2008，38(8)：1139-1352)

[17] Li Y M. Fuzzy Turing machines: Variants and universality. IEEE Transactions on Fuzzy Systems, 2008, 16(6): 1491-1502

[18] Deutsch D. Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A, 1985, 400: 97-117

[19] Bernstern E, Vazirani L. Quantum complexity theory. SIAM Journal on Computing, 1997, 26: 1411-1473

[20] Gudder S. Quantum automata: An overview. International Journal of Theoretical Physics, 1999, 38: 2261-2282

LI Yong-Ming, born in 1966, Ph.D., professor. His research interests include non-classical computation theory, computational intelligence, quantum computing and quantum information, topology over lattices.

LI Ping, born in 1979, Ph. D., lecturer. Her research interests include non-classical computation theory and computational intelligence.

Background

The ideas of quantum computing come from the connections between physics and computation. In particular, in 1994, Shor discovered a polynomial-time algorithm for prime factorization on quantum computers. And in 1996, Grover found a quantum algorithm for searching item in an unsorted database in square root of the time. Since then, quantum computation has attracted more and more attention in the research community. In this field, the computing models of quantum computation is still one of the most important topic to study. The theory of automata based on quantum logic (also called orthomodular lattice-valued automata), established by Mingsheng Ying et al., is an important research direction on the computing models of quantum computation. Presently, some results different from those based on classical logic have been obtained . And these researchers seek to reveal the logical basis of quantum computing.

Some properties of finite automata based on quantum logic have been discussed, and some properties of pushdown automata based on quantum logic have been studied, which all have obtained some results different from those based on classical logic. But, there is no any research on Turing Machines based on quantum logic at present. Therefore, this paper will study some properties of Turing Machines based on quantum logic(also called quantum Turing Machines).

It is well known that classical Turing Machines is equivalent to many of its modified versions. Just because of this, Turing Machines can be viewed as a general model of computation. And there is the universal Turing Machine (programmable Turing Machine), the universal Turing Machine can simulate any Turing Machine, which is the basic reason that the universal Turing Machine can be viewed as a formal model of the modern computers. And we know, the properties of finite automata based on quantum logic is different from those of classical automata. This is mainly because that the distributive law may not hold in the lattices of their truth values. At the same time, some properties hold, such as Kleene Theorem, Büchi-Elgot Theorem, and Schützenberger Theorem. But, the related constructions all possess quantum logic characteristics. For example, based on quantum logic, deterministic finite automata and non-deterministic finite automata are still equivalent, however, the construction method by classical subsets is not applicable. Therefore, we give the construction method by subsets based on quantum logic, and it is proved the equivalence between deterministic finite automata and non-deterministic finite automata. This construction method is not dependent on the distributive law. So, we further want to know, under the condition that the lattices of truth values do not satisfy the distributive law, what about the computing power of quantum Turing Machines and its modified versions? Is there the universal quantum Turing Machines? The two problems are the main research contents in this paper. And we will give the related constructions based on quantum logic. Then we further obtain some conclusions different essence from that based on classical logic.

This paper is the authors' original research results, signature without controversy. And citing others results have indicate the source.

模糊知识的三种否定及其集合基础

潘正华
(江南大学理学院 江苏 无锡 214122)

摘 要 对于模糊知识中"否定"的认知与处理,文中从概念层面上区分模糊知识中的矛盾否定关系与对立否定关系,研究发现了模糊知识中存在一规律:一对对立的概念为模糊概念,则它们之间必然存在"中介"的模糊概念;反之,如果一对对立的概念之间存在中介的模糊概念,则对立的概念必然是模糊概念.因此,作者提出在模糊知识的否定关系中存在三种不同的否定关系,即矛盾否定关系、对立否定关系和中介否定关系,并给出它们的形式定义.为了能够刻画这些关系的内在性质与联系,作者提出了一种新的具有矛盾否定、对立否定和中介否定的模糊集 FSCOM,并讨论了 FSCOM 的特征、FSCOM 的基本运算与性质以及 FSCOM 与 Zadeh 模糊集的关系等.在后续文中将表明,FSCOM 是一种处理实际中的模糊知识及其各种否定的有效方法.

关键词 模糊知识;模糊集;矛盾否定关系;对立否定关系;中介否定关系
中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2012.01421

Three Kinds of Negation of Fuzzy Knowledge and Their Base of Set

PAN Zheng-Hua
(*School of Science, Jiangnan University, Wuxi, Jiangsu* 214122)

Abstract Negative relation in fuzzy knowledge is differentiated as contradictory negative relation and opposite negative relation in this paper, and a character of fuzzy knowledge is discovered: If two opposite concepts are fuzzy concepts then must exist a "medium" fuzzy concept between them, contrarily if there is medium fuzzy concept between the two concepts then two concepts must be fuzzy concepts. We thus propose that negative relations in fuzzy knowledge included contradictory, opposite and medium negative relation. In order to describe intrinsic properties of these relations, we define a new fuzzy set FSCOM with contradictory negation, opposite negation and medium negation, discuss that characteristics of FSCOM, operations and their properties in FSCOM, as well as relationship with Zadeh' fuzzy set and so on. In next paper which will shows that FSCOM is an effective method for handling the various negations of fuzzy information.

Keywords fuzzy knowledge; fuzzy sets; contradictory negative relation; opposite negative relation; medium negative relation

1 引 言

知识的否定在知识处理中、尤其在模糊知识处理中扮演了重要角色.随着知识研究的发展,对于模糊知识中"否定"的认知与研究,近年来一些学者主张知识处理需要不同的否定.Wagner 等人认为,从逻辑观点看,在知识推理、自然语言、逻辑程序设计(Prolog)、语义网(Semantic Web)、数据库查询语言 SQL 以及产生式规则系统(如 CLIPS 和 Jess)等

收稿日期:2010-03-16;最终修改稿收到日期:2011-05-21.本课题得到国家自然科学基金(60973156)和江南大学创新研究团队计划资助.
潘正华,男,1957 年生,教授,中国计算机学会(CCF)高级会员,主要研究领域为知识表示与知识推理、经典逻辑与非经典逻辑等. E-mail: pan_zhenghua@163.com.

领域中，否定是一个非清晰的概念，并提出在所有这些计算信息处理系统中要区分强否定(strong negation)和弱否定(weak negation)，强否定表示明确的假(explicit falsity)，弱否定表示非-真(non-truth)[1-7]. 2006 年，Ferré[8]提出一种认识的扩充，区分否定中的外延否定和内涵否定，既将基于模态逻辑 AIK(All I Know)的一种逻辑转化运用于逻辑概念分析 LCA(Logical Concept Analysis)框架中，而且这种认识的扩充不需失去 LCA 的普遍性. 2007 年，Kaneiwa 主张在描述逻辑中区分两种否定，提出一个带有经典否定和强否定扩展的描述逻辑 $ALC_{\sim}$，$ALC_{\sim}$ 中用经典否定(如 not happy)和强否定(如 unhappy)描述 contraries，contradictories 以及 subcontraries 等概念，并期望将这些概念形式化后能够提供一个改进的适合解释经典否定与强否定以及各种结合的语义，从而表明这种语义对 $ALC_{\sim}$ 中的概念保持矛盾性(contradictoriness)和反对性(contrariness)[9]. 2006 年，潘正华等人从概念层面上提出在知识处理中区分知识的矛盾关系和对立关系，研究指出了清晰性知识和模糊性知识中存在五种矛盾否定关系与对立否定关系，对这五种关系给出一种逻辑描述，并运用在知识表示与知识推理中[10-14].

对于模糊知识的否定的认知，本文研究并提出模糊知识中存在三种否定关系既“矛盾”否定关系、“对立”否定关系和“中介”否定关系，给出了三种否定关系的形式定义. 为了能够刻画这些关系的内在性质与联系，进一步研究了它们的集合基础，定义了一种新的具有矛盾否定、对立否定和中介否定的模糊集 FS$_{\mathrm{COM}}$，讨论了 FS$_{\mathrm{COM}}$ 的特征、FS$_{\mathrm{COM}}$ 的一些基本运算及其性质以及与 Zadeh 模糊集的关系等.

2 模糊概念中的各种否定关系

在知识中，概念是知识构成的基本成分，是一种元知识. 在形式逻辑中，概念之间的关系是指概念外延的关系，它区分为相容关系和不相容关系. 概念 A 与 B 之间的不相容关系，是指 A 与 B 两个概念的外延(外延用一个矩形框表示)之间没有任何一部分重合的关系(图 1). 例如：“白”与“非白”，“青年”与“老年”，“导体”与“绝缘体”等等.

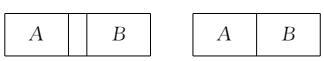


图 1 不相容概念 A 与 B 的外延没有重合部分

自 Aristotle 以来，形式逻辑将概念的不相容关系区分为矛盾关系和对立关系. 概念的矛盾关系，是指在同一个属概念下的两个种概念之间的不相容关系，它们的外延互相排斥，外延之和等于属概念的外延；概念的对立关系，是指在同一个属概念之下的两个种概念之间的不相容关系，它们的外延互相排斥，外延之和小于属概念的外延. 一个概念与其“否定”之间的关系就是一种不相容关系，因而，一个概念与其否定概念之间的关系包括了矛盾否定关系和对立否定关系.

概念的模糊性，即是概念在外延上的不分明性. 对于一个模糊概念与其否定的关系，我们认为存在下列三种情形.

2.1 模糊概念中的三种否定关系

(1)模糊概念中的矛盾否定关系 CFC(Contradictory negative relation in Fuzzy Concepts)

关系特征：“外延界限不分明，非此即彼”.

例如：属概念“人”下的种概念“青年人”与“非青年人”的关系，属概念“速度”下的种概念“快”与“不快”的关系等(图 2).

图 2 模糊概念“青年”与其矛盾否定“非青年”的外延关系

(2)模糊概念中的对立否定关系 OFC(Opposite negative relation in Fuzzy Concepts)

关系特征：“外延界限不分明，不非此即彼”.

例如：属概念“人”下的种概念“青年人”与“老年人”的关系，属概念“速度”下的种概念“快”与“慢”的关系等(图 3).

图 3 模糊概念“青年”与其对立否定“老年”的外延关系

在现实世界的各种知识中，许多对立的概念之间存在具有“中介”特征的概念. 所谓对立概念之间的中介概念，即指在同一个属概念下，两个对立的种概念之间呈现出“过渡状态”的另一个种概念. 对于对立的模糊概念，通过对大量的客观实例进行研究后我们发现，对立的模糊概念中存在如下规律：

“如果一对对立概念为模糊概念，则对立概念之间必然存在中介的模糊概念；反之，如果一对对立概念之间存在中介的模糊概念，则对立概念一定是模糊概念. 换言之，对立概念之间存在中介的模糊概念，当且仅当对立概念为模糊概念”.

这种存在于对立的模糊概念之间的中介模糊概念，从它的内涵和外延可知，它与对立的模糊概念的关系是一种否定关系. 对此我们称为“中介”否定关系.

(3) 模糊概念中的中介否定关系 MFC(Medium negative relation in Fuzzy Concepts)

关系特征：“外延界限不分明，彼与此的中介”.

例如：“中年人”是对立概念“青年人”与“老年人”之间的中介概念，中年人与青年人(或老年人)之间的关系是中介否定关系；“黄昏”或“黎明”是对立概念“白昼”与“黑夜”之间的中介概念，黄昏(或黎明)与白昼(或黑夜)之间的关系是中介否定关系；“半导体”是“导体”与“绝缘体”之间的中介概念，半导体与导体(或绝缘体)之间的关系是中介否定关系等(图 4).

图 4　对立的模糊概念“青年”和“老年”与其中介否定“中年”的外延关系

因而我们提出，在模糊概念中存在三种不同的否定关系，即矛盾否定关系 CFC、对立否定关系 OFC 和中介否定关系 MFC.

2.2　模糊概念中的三种否定关系的形式定义

既然外延为概念所反映的对象范围、概念之间的关系为概念外延的关系，因而从概念的外延角度，可给出 CFC、OFC 和 MFC 的形式定义.

定义 1.　设 $U(\neq\varnothing)$ 为论域(对象域)，$X(X\subseteq U)$ 为关于 U 中对象的一个概念. 对于任何 X，若存在一个划分 $\xi:\{X_1,X_2,\cdots,X_n\}$，$X_i\subseteq X$，$X_i\neq\varnothing$，$\bigcup_{i=1}^{n}X_i=X$，则称 X 为 $X_1,X_2,\cdots,X_n$ 的属概念，$X_i(i=1,2,\cdots,n)$ 为 X 的种概念；其中，若 $X_i\cap X_j=\varnothing(i\neq j,\ i,j=1,2,\cdots,n)$，则称种概念 X_i,X_j 为清晰概念，若 $X_i\cap X_j\neq\varnothing$，则称种概念 X_i,X_j 为模糊概念.

由于任何一对具有矛盾否定关系的概念和一对具有对立否定关系的概念，都是同一个属概念下的一对种概念，所以，CFC 和 OFC 分别是同一个属概念下的两个种概念之间的关系. 由上述定义可知，它们应分别是 $X\times X$ 上的二元关系，即 $X\times X$ 的不同子集. 因此，对于 CFC 和 OFC 的形式表达，可定义如下.

定义 2.　设一个属概念为 $A=\bigcup_{i=1}^{n}A_i$，其中 A_i 为 A 的种概念. 对于一个 $A_i(i\in\{1,2,\cdots,n\})$，若存在 A 的种概念 A_j 和 $A_k(A_j\neq A_k\neq A_i)$，A_i、A_j 和 A_k 是模糊概念，并且 A_i 与 A_j 具有矛盾否定关系，A_i 与 A_k 具有对立否定关系，则

$$\begin{aligned}\mathrm{CFC}=\{&(A_i,A_j)\mid A_i\neq A_j,A_i\cap A_j\neq\varnothing,\\&A_i\cup A_j=A\}\subset A\times A;\end{aligned}$$

$$\begin{aligned}\mathrm{OFC}=\{&(A_i,A_k)\mid A_i\neq A_k,A_i\neq A_j,A_k\neq A_j,\\&A_i\cap A_k\neq\varnothing,A_i\cup A_k\subseteq A\}\subset A\times A.\end{aligned}$$

我们已知，当对立否定概念是模糊概念时，它们之间存在中介(否定)概念. 因此，一对对立否定的模糊概念与其中介否定的关系 MFC 应是 $(X\times X)\times X$ 的一个子集.

定义 3.　设一个属概念 $B=\bigcup_{i=1}^{n}B_i$，B_i 是 B 的种概念. 若 $B_i,B_j\subseteq B(i\neq j)$ 是具有对立否定关系的模糊概念，则存在 $B_m\subseteq B(m\neq i,m\neq j)$，有

$$\begin{aligned}\mathrm{MFC}=\{&((B_i,B_j),B_m)\mid B_i\neq B_j,B_i\cap B_m\neq\varnothing,\\&B_j\cap B_m\neq\varnothing,B_i\cup B_j\cup B_m\subseteq B\}\subset\\&(B\times B)\times B.\end{aligned}$$

由以上定义，容易证明 CFC、OFC 和 MFC 具有如下性质：

(1) CFC、OFC 和 MFC 互不相同；

(2) CFC、OFC 具有对称性，不具有自反性、传递性；

(3) MFC 不具有对称性、自反性、传递性.

3　模糊知识及其三种否定的一种集合基础

集合及其方法，是从数学角度描述知识及其规律的最基本的抽象概念和手段. 如何对模糊概念及其三种不同否定关系 CFC、OFC 和 MFC 进行刻画，我们提出如下一种新的模糊集.

3.1　区分矛盾否定、对立否定和中介否定的模糊集 FSCOM

定义 4[15].　设 U 是论域. 映射

$$\Psi_A:U\to[0,1]$$

确定了 U 上的模糊子集 A. 映射 Ψ_A 称为 A 的隶属函数，$\Psi_A(x)$ 称为 x 对 A 的隶属程度(简称隶属度)，记为 $A(x)$.

定义 5.　设 A 是 U 上的模糊子集，$\lambda\in(0,1)$.

(1) 映射

$$\Psi^{\lambda}:\{A(x)\mid x\in U\}\to[0,1]$$

若满足 $\Psi^{\neg}(A(x))=1-A(x)$，则映射 $\Psi^{\neg}$ 确定了 U 上的一模糊子集，记作 $A^{\neg}$，$A^{\neg}(x)=\Psi^{\neg}(A(x))$. $A^{\neg}$ 称为 A 的对立否定集.

(2) 映射

$$\Psi^{\sim}:\{A(x)\mid x\in U\}\to[0,1]$$

若满足

$$\Psi^{\sim}(A(x))=\begin{cases}\dfrac{2\lambda-1}{1-\lambda}(A(x)-\lambda)+1-\lambda, & \lambda\in[1/2,1)\text{且}A(x)\in(\lambda,1] \quad (1)\\ \dfrac{2\lambda-1}{1-\lambda}A(x)+1-\lambda, & \lambda\in[1/2,1)\text{且}A(x)\in[0,1-\lambda) \quad (2)\\ \dfrac{1-2\lambda}{\lambda}A(x)+\lambda, & \lambda\in(0,1/2]\text{且}A(x)\in[0,\lambda) \quad (3)\\ \dfrac{1-2\lambda}{\lambda}(A(x)+\lambda-1)+\lambda, & \lambda\in(0,1/2]\text{且}A(x)\in(1-\lambda,1] \quad (4)\\ A(x), & \text{其它} \quad (5)\end{cases}$$

则映射 $\Psi^{\sim}$ 确定了 U 上的一模糊子集，记作 $A^{\sim}$，$A^{\sim}(x)=\Psi^{\sim}(A(x))$. $A^{\sim}$ 称为 A 的中介否定集.

(3) 映射

$$\Psi^{\neg}:\{A(x)\mid x\in U\}\to[0,1]$$

若满足 $\Psi^{\neg}(A(x))=\max(A^{\neg}(x),A^{\sim}(x))$，则 $\Psi^{\neg}$ 确定了 U 上的一模糊子集，记作 $A^{\neg}$，$A^{\neg}(x)=\Psi^{\neg}(A(x))$. $A^{\neg}$ 称为 A 的矛盾否定集.

以上定义的论域 U 上的模糊子集，称为“区分矛盾否定、对立否定和中介否定的模糊集”，简记为 FSCOM(Fuzzy Sets with Contradictory negation, Opposite negation and Medium negation).

3.2 FSCOM 的特征

由 FSCOM 的定义可看出，在 FSCOM 中，模糊集 A 与 A 的对立否定集 $A^{\neg}$、中介否定集 $A^{\sim}$ 以及矛盾否定集 $A^{\neg}$ 具有如下关系及特点：

(1) 对于任意的 $x\in U$，$A(x)$，$A^{\neg}(x)$，$A^{\sim}(x)$，$A^{\neg}(x)\in[0,1]$；

(2) 矛盾否定由对立否定和中介否定确定，即 $A^{\neg}(x)=\max(A^{\neg}(x),A^{\sim}(x))$；

(3) 由于 λ 是可变的，所以 λ 的大小以及变化，决定了 x 对 A、$A^{\neg}$ 和 $A^{\sim}$ 的隶属度 $A(x)$、$A^{\neg}(x)$ 和 $A^{\sim}(x)$ 的取值范围的大小和变化. 其中

当 $\lambda\geq1/2$ 时，$A(x)\in(\lambda,1]$，或 $A(x)\in[0,1-\lambda)$. 其中，若 $A(x)\in(\lambda,1]$，则有 $A^{\sim}(x)\in[1-\lambda,\lambda]$ 与 $A^{\neg}(x)\in[0,1-\lambda)$，若 $A(x)\in[0,1-\lambda)$，则有 $A^{\sim}(x)\in[1-\lambda,\lambda]$ 与 $A^{\neg}(x)\in(\lambda,1]$；

当 $\lambda\leq1/2$ 时，$A(x)\in(1-\lambda,1]$，或 $A(x)\in[0,\lambda)$. 其中，若 $A(x)\in(1-\lambda,1]$，则有 $A^{\sim}(x)[\lambda,1-\lambda]$ 与 $A^{\neg}(x)\in[0,\lambda)$，若 $A(x)\in[0,\lambda)$，则有 $A^{\sim}(x)\in[\lambda,1-\lambda]$ 与 $A^{\neg}(x)\in(1-\lambda,1]$；

关于 $A(x)$、$A^{\neg}(x)$ 和 $A^{\sim}(x)$ 之间的关系，可用图 5、图 6 描述(图中符号“•”与“◦”分别表示一个区间的闭端点和开端点).

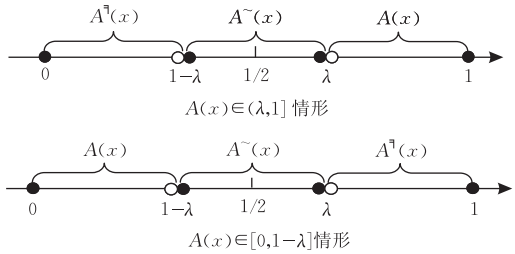


图 5 当 $\lambda\geq1/2$ 时，$A(x)$，$A^{\neg}(x)$，$A^{\sim}(x)$ 在 $[0,1]$ 中的关系

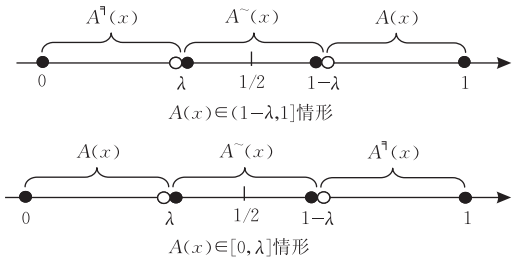


图 6 当 $\lambda\leq1/2$ 时，$A(x)$，$A^{\neg}(x)$，$A^{\sim}(x)$ 在 $[0,1]$ 中的关系

由 FSCOM 的定义，容易验证 FSCOM 具有下列性质.

命题 1. 设 A 是一个 FSCOM 模糊集. 则 $A(x)\geq A^{\sim}(x)\geq A^{\neg}(x)$，当且仅当 $A(x)\in(\lambda,1]$，$A^{\neg}(x)\geq A^{\sim}(x)\geq A(x)$，当且仅当 $A(x)\in[0,1-\lambda)$.

由于当 $A(x)\geq A^{\sim}(x)$ 时，$A^{\sim}(x)\geq A^{\neg}(x)$；当 $A^{\sim}(x)\geq A(x)$ 时，$A^{\neg}(x)\geq A^{\sim}(x)$，所以，有如下命题.

命题 2. 设 A 是一个 FSCOM 模糊集. 则 $A(x)>A^{\sim}(x)>A^{\neg}(x)$，或者 $A^{\neg}(x)\geq A^{\sim}(x)\geq A(x)$.

在 FSCOM 中，因为 $a\leq b(a,b\in[0,1])$ 时，$\Psi^{\neg}(a)\geq\Psi^{\neg}(b)$，$\Psi^{\sim}(a)\leq\Psi^{\sim}(b)$，所以，FSCOM 具有如下性质.

命题 3. 在 FSCOM 中，$\Psi^{\neg}$ 是减函数，$\Psi^{\sim}$ 是增函数.

3.3 FSCOM 与 Zadeh 模糊集的关系比较

在现有的能够刻画事物模糊性的集合论中，有 Zadeh[15] 模糊集 FS 以及随着模糊知识处理技术的发展而出现的 FS 的各种扩展[16]，如直觉模糊集(Intuitionistic Fuzzy Sets)[17]、区间值模糊集(Interval valued fuzzy sets)[18]、Vague 集[19]和粗糙集(Rough Sets)[20]，我们认为这些集合论关于否定的认识思想与传统集合一样，它们都只有一种否定即矛盾否定，只是否定的定义形式不同，没有区分模糊性事物之间的矛盾关系与对立关系. 因此，这些集合理论不具有区分、表达模糊概念中的矛盾否定、对立否定以及中介否定的能力. 对此，我们可作以下归纳比较(表 1).

表 1　关于模糊概念 A 的否定的各种认知思想与表示

认知思想	A 的否定 1	A 的否定 2	A 的其它否定
传统集合	矛盾否定：$\neg A$	×	×
模糊集	矛盾否定：$\bar{A}$	×	×
直觉模糊集	矛盾否定：$-A$	×	×
粗糙集	矛盾否定：$-A$	×	×
Wagner G	弱否定：$-A$	强否定：$\sim A$	×
Ferré S	外延否定：$\neg A$	内涵否定：mal-A	×
Kaneiwa K	经典否定：$\neg A$	强否定：$\sim A$	×
FSCOM	矛盾否定：$A^{\neg}$	对立否定：$A^{\urcorner}$	A 与 $A^{\urcorner}$ 的中介否定：$A^{\sim}$

在 FSCOM 中，一个模糊集 A 的否定区分为矛盾否定 $A^{\neg}$、对立否定 $A^{\urcorner}$ 以及中介否定 $A^{\sim}$. 在 Zadeh 模糊集中，A 的(矛盾)否定定义为 $\bar{A}(x)=1-A(x)$. 由 FSCOM 的定义和 Zadeh 模糊集定义可知，Zadeh 模糊集中 A 的否定在 FSCOM 中由对立否定和中介否定共同确定：$A^{\neg}(x)=\max(A^{\urcorner}(x),A^{\sim}(x))$. 由 FSCOM 定义和命题 2 可知，当 $A(x)<1/2$ 时，$A^{\neg}(x)=\max(A^{\urcorner}(x),A^{\sim}(x))=A^{\urcorner}(x)$，即对立否定与矛盾否定相同；当 $A(x)>1/2$ 时，$A^{\neg}(x)=\max(A^{\urcorner}(x),A^{\sim}(x))=A^{\sim}(x)$，即中介否定与矛盾否定相同；当 $A(x)=1/2$ 时，$A^{\neg}(x)=A^{\urcorner}(x)=A^{\sim}(x)=1/2$，即矛盾否定、对立否定和中介否定三者相同. 因而，FSCOM 中的对立否定和中介否定与 Zadeh 模糊集中否定的关系如下.

命题 4.　设 A 是一个模糊集.

当 $A(x)<1/2$ 时，FSCOM 模糊集 A 的对立否定 $A^{\urcorner}$ 与 Zadeh 模糊集 A 的否定 $A^{\neg}$ 相同；

当 $A(x)>1/2$ 时，FSCOM 模糊集 A 的中介否定 $A^{\sim}$ 与 Zadeh 模糊集 A 的否定 $A^{\neg}$ 相同；

当 $A(x)=1/2$ 时，FSCOM 模糊集 A 的对立否定 $A^{\urcorner}$、中介否定 $A^{\sim}$ 与 Zadeh 模糊集 A 的否定 $A^{\neg}$ 三者相同.

3.4 FSCOM 的运算及其性质

定义 6.　设 A,B 是 FSCOM 模糊集 $\forall x\in U$

$A\subseteq B$，当且仅当 $A(x)\leq B(x)$；

$A=B$，当且仅当 $A(x)=B(x)$.

定义 7.　设 A,B 是 FSCOM 模糊集. 称 $A\cup B$ 为 A 与 B 的并集，$A\cap B$ 为 A 与 B 的交集，若

$(A\cup B)(x)=\max(A(x),B(x))$；

$(A\cap B)(x)=\min(A(x),B(x))$.

根据以上定义，易证下列性质.

性质 1.　设 A,B 是 FSCOM 模糊集. 则

(1) 幂等律. $A\cup A=A, A\cap A=A$；

(2) 交换律. $A\cup B=B\cup A, A\cap B=B\cap A$；

(3) 结合律. $(A\cup B)\cup C=A\cup(B\cup C)$，$(A\cap B)\cap C=A\cap(B\cap C)$；

(4) 吸收律. $A\cap(A\cup B)=A$，$A\cup(A\cap B)=A$；

(5) 分配律. $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$，$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$；

(6) 0-1 律. $A\cup\varnothing=A, A\cap\varnothing=\varnothing, U\cup A=U$，$U\cap A=A$.

证明.　只证(5)，其余同理可证.

(5) 因 $(A\cup(B\cap C))(x)=\max(A(x),\min(B(x),C(x)))$，$((A\cup B)\cap(A\cup C))(x)=\min(\max(A(x),B(x)),\max(A(x),C(x)))$，其中

若 $A(x)>\max(B(x),C(x))$，则 $(A\cup(B\cap C))(x)=((A\cup B)\cap(A\cup C))(x)=A(x)$；

若 $A(x)\leq\max(B(x),C(x))$，则存在两种情形：① 当 $B(x)>C(x)$ 时，有 $(A\cup(B\cap C))(x)=\max(A(x),C(x))$，$((A\cup B)\cap(A\cup C))(x)=\min(B(x),\max(A(x),C(x)))=\max(A(x),C(x))$；② 当 $B(x)\leq C(x)$ 时，有 $(A\cup(B\cap C))(x)=\max(A(x),B(x))$，$((A\cup B)\cap(A\cup C))(x)=\min(\max(A(x),B(x)),C(x))=\max(A(x),B(x))$；即当①和②时，均有 $(A\cup(B\cap C))(x)=((A\cup B)\cap(A\cup C))(x)$.

所以，根据定义 6，$A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$. 同理，可证 $A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$.

证毕.

性质 2.　设 A,B 和 C 是 FSCOM 模糊集. 则

(1) $A^{\urcorner\urcorner}=A$；

(2) $A^{\sim}=A^{\urcorner\sim}$；

(3) $A^{\neg}=A^{\urcorner}\cup A^{\sim}$；

(4) $A^{\sim}=A^{\neg}\cap A^{\urcorner\neg}$；

(5) $A^{\urcorner\neg}=A\cup A^{\sim}$；

(6) $(A\cup B)^{\urcorner}=A^{\urcorner}\cap B^{\urcorner}$；

(7) $(A\cap B)^{\urcorner}=A^{\urcorner}\cup B^{\urcorner}$.

证明.

(1) 由 FSCOM 定义,$A^{\neg\neg}(x)=1-A^{\neg}(x)=1-(1-A(x))=A(x)$,所以,$A^{\neg\neg}=A$ 得证.

(2) 如果 $A^{\sim}(x)>A^{\neg\sim}(x)$,则 $(A^{\neg})^{\sim}(x)>(A^{\neg})^{\neg\sim}(x)=A^{\neg\neg\sim}(x)$,因 $A^{\neg\neg}=A$,所以 $A^{\neg\sim}(x)>A^{\sim}(x)$;反之,如果 $A^{\sim}(x)<A^{\neg\sim}(x)$,则 $(A^{\neg})^{\sim}(x)<(A^{\neg})^{\neg\sim}(x)=A^{\neg\neg\sim}(x)$,即 $A^{\neg\sim}(x)<A^{\sim}(x)$;因此,有 $A^{\sim}(x)=A^{\neg\sim}(x)$. 由定义 6,$A^{\neg\neg}=A$ 得证.

(3) 根据 FSCOM 定义与定义 7,$A^{\neg}(x)=\max(A^{\neg}(x),A^{\sim}(x))=(A^{\neg}\cup A^{\sim})(x)$,由定义 6,$A^{\neg}=A^{\neg}\cup A^{\sim}$ 得证.

(4) 由定义 7 和 FSCOM 定义,$(A^{\neg}\cap A^{\neg\neg})(x)=\min(A^{\neg}(x),A^{\neg\neg}(x))=\min(\max(A^{\neg}(x),A^{\sim}(x)),\max(A^{\neg\neg}(x),A^{\neg\sim}(x)))=\min(\max(A^{\neg}(x),A^{\sim}(x)),\max(A(x),A^{\sim}(x)))$. 其中,若 $A(x)>A^{\sim}(x)$,根据命题 2,则有 $A^{\sim}(x)>A^{\neg}(x)$,所以,$(A^{\neg}\cap A^{\neg\neg})(x)=A^{\sim}(x)$;若 $A(x)\leq A^{\sim}(x)$,根据命题 2,则有 $A^{\sim}(x)<A^{\neg}(x)$,所以,$(A^{\neg}\cap A^{\neg\neg})(x)=A^{\sim}(x)$. 由定义 6,$A^{\sim}=A^{\neg}\cap A^{\neg\neg}$ 得证.

(5) 由(3),$A^{\neg}=A^{\neg}\cup A^{\sim}$,有 $A^{\neg\neg}=A^{\neg\neg}\cup A^{\neg\sim}$;再由(1)和(2),得 $A^{\neg\neg}=A\cup A^{\sim}$.

(6) 根据 FSCOM 定义,$(A\cup B)^{\neg}(x)=1-(A\cup B)(x)=1-\max(A(x),B(x))$,$(A^{\neg}\cap B^{\neg})(x)=\min(1-A(x),1-B(x))$. 其中,若 $A(x)\geq B(x)$,则有 $(A\cup B)^{\neg}(x)=(A^{\neg}\cap B^{\neg})(x)=1-A(x)$;若 $A(x)<B(x)$,则有 $(A\cup B)^{\neg}(x)=(A^{\neg}\cap B^{\neg})(x)=1-B(x)$;所以,$(A\cup B)^{\neg}(x)=(A^{\neg}\cap B^{\neg})(x)$. 由定义 6,$(A\cup B)^{\neg}=A^{\neg}\cap B^{\neg}$ 得证.

(7) 根据 FSCOM 定义,$(A\cap B)^{\neg}(x)=1-(A\cap B)(x)=1-\min(A(x),B(x))$,$(A^{\neg}\cup B^{\neg})(x)=\max(1-A(x),1-B(x))$. 其中,若 $A(x)\geq B(x)$,则有 $(A\cap B)^{\neg}(x)=(A^{\neg}\cup B^{\neg})(x)=1-B(x)$;若 $A(x)<B(x)$,则有 $(A\cap B)^{\neg}(x)=(A^{\neg}\cup B^{\neg})(x)=1-A(x)$;所以,$(A\cap B)^{\neg}(x)=(A^{\neg}\cup B^{\neg})(x)$. 由定义 6,$(A\cap B)^{\neg}=A^{\neg}\cup B^{\neg}$ 得证. 证毕.

性质 3. 设 A,B 是 FSCOM 模糊集. 则

(1) $A\subseteq B\Leftrightarrow B^{\neg}\subseteq A^{\neg}$;

(2) $A\subseteq B\Leftrightarrow A^{\sim}\subseteq B^{\sim}$;

(3) $A^{\sim}\subseteq B^{\sim}\Leftrightarrow A^{\sim\sim}\subseteq B^{\sim\sim}$;

(4) $A\subseteq B^{\neg}\Leftrightarrow B\subseteq A^{\neg}$;

(5) $A^{\neg}\subseteq B\Leftrightarrow B^{\neg}\subseteq A$.

证明.

(1) 若 $A\subseteq B$,由定义 6,有 $A(x)\leq B(x)$,即 $1-B(x)\leq 1-A(x)$. 据定义 5,则有 $B^{\neg}(x)\leq A^{\neg}(x)$,即 $B^{\neg}\subseteq A^{\neg}$. 反之,同理可证.

(2) 若 $A\subseteq B$,由定义 6,有 $A(x)\leq B(x)$. 据命题 3,$\Psi^{\sim}$ 是增函数,故 $\Psi^{\sim}(A(x))\leq\Psi^{\sim}(B(x))$,即 $A^{\sim}(x)\leq B^{\sim}(x)$(据定义 5),由定义 6,则 $A^{\sim}\subseteq B^{\sim}$. 反之,同理可证.

(3) 若 $A^{\sim}\subseteq B^{\sim}$,由定义 6,$\Psi^{\sim}(A(x))\leq\Psi^{\sim}(B(x))$. 因 $\Psi^{\sim}$ 有正序性(据命题 3),故 $\Psi^{\sim}(\Psi^{\sim}(A(x)))\leq\Psi^{\sim}(\Psi^{\sim}(B(x)))$,即 $A^{\sim\sim}\subseteq B^{\sim\sim}$.

(4) 若 $A\subseteq B^{\neg}$,由定义 6,有 $A(x)\leq B^{\neg}(x)$,即 $A(x)\leq 1-B(x)$. 故 $B(x)\leq 1-A(x)$,即 $B(x)\leq A^{\neg}(x)$. 由定义 6,有 $B\subseteq A^{\neg}$. 反之,同理可证.

(5) 同(4)证可得. 证毕.

由定义 7,易证 FSCOM 具有下列结论.

性质 4. 设 A 是一个 FSCOM 模糊集,$\Delta,\nabla\in\{\neg,\sim,\neg\}$. 则

$$A\cup A^{\Delta}=U,\ A\cup A^{\nabla}=U,\ A^{\Delta}\cup A^{\nabla}=U,$$
$$A\cap A^{\Delta}=\varnothing,\ A\cap A^{\nabla}=\varnothing,\ A^{\Delta}\cap A^{\nabla}=\varnothing$$

都不成立.

性质 4 表明,在 FSCOM 中,排中律和矛盾律都不成立.

4 应 用

为了表明 FSCOM 处理客观实际问题的适用性,我们研究了 FSCOM 在模糊决策、模式识别中的应用[21-23]. 在文献[21-22]中,针对长江三角洲区域收入高(低)和存款多(少)者(人、家庭)的投资决策实例,运用 FSCOM 研究投资决策中的模糊知识及其不同否定的表达、推理与实现,并比较了 FSCOM 与 Zadeh 模糊集和直觉模糊集在该实例中的应用效果. 文献[23]中提出了 FSCOM 的模糊度,贴近度,距离贴近度和格贴近度定义,并讨论了在模式识别实例中的应用.

5 结 论

(1) 对于模糊知识中“否定”的认知与研究,本文提出区分模糊知识中的矛盾否定关系与对立否定关系的思想. 由此发现了对立的模糊知识中存在一规律:如果一对对立的概念为模糊概念,则它们之间必然存在中介的模糊概念,反之,如果一对对立的概念之间存在中介的模糊概念,则对立的概念必然是

模糊概念.

(2)基于以上认识,从概念层面上确立了模糊知识中存在的3种不同的否定关系,即矛盾否定关系CFC、对立否定关系OFC以及中介否定关系MFC,并给出了它们的形式定义.

(3)为了对模糊知识与其3种不同否定关系的内在性质和联系予以描述,定义了一种新的能够完全刻画模糊知识与其矛盾否定、对立否定和中介否定的模糊集FSCOM,并研究了它具有的特征、运算和性质以及与Zadeh模糊集的关系等.

参 考 文 献

[1] Wagner G. A database needs two kinds of negation//Lecture Notes in Computer Science 495. Springer, 1991: 357-371

[2] Wagner G. Web rules need two kinds of negation//Lecture Notes in Computer Science 2901. Springer, 2003: 33-50

[3] Analyti A, Antoniou G, Damasio C V, Wagner G. Negation and negative information in the W3C resource description framework. Annals of Mathematics, Computing & Teleinformatics, 2004, 1(2): 25-34

[4] Minker J, Ruiz C. Semantics for disjunctive logic programs with explicit and default negation. Fundamenta Informaticae, 1994, 20(3/4): 145-192

[5] Dung P M, Mancarella P. Production systems need negation as failure. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(2): 336-353

[6] Beeson M, Veroff R, Wos L. Double-negation elimination in some propositional logics. Studia Logica, 2005, 80(2/3): 195-234

[7] Vakarelov D. Nelson's negation on the base of weaker versions of intuitionistic negation. Studia Logica, 2005, 80(2/3): 393-430

[8] Ferré S. Negation, opposition, and possibility in logical concept analysis//Lecture Notes in Artificial Intelligence 3874. Springer, 2006: 130-145

[9] Kaneiwa K. Description logic with contraries, contradictories, and subcontraries. New Generation Computing, 2007, 25(4): 443-468

[10] Pan Zhenghua, Zhu Wujia. A new cognition and processing on contradictory knowledge//Proceedings of the IEEE International Conference on Machine Learning and Cybernetics. Qingdao, 2006, 3: 1532-1537

[11] Pan Zhenghua, Zhang Shengli. Five kinds of contradictory relations and opposite relations in inconsistent knowledge//Proceedings of the 4th IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'07). Shanghai, 2007, 4: 761-764

[12] Pan Zhenghua, Zhang Shengli. Differentiation and processing on contradictory relation and opposite relation in knowledge//Proceedings of the 3rd IEEE International Conference on Natural Computation (ICNC'07). Shanghai, 2007, 4: 334-338

[13] Pan Zhenghua. A logic description on different negation relation in knowledge. Lecture Notes in Computer Science, 2008, 5227: 815-823

[14] Wang Cen, Pan Zhenghua. Searching method of fuzzy knowledge reasoning based medium logic//Lecture Notes in Computer Science 5227. Springer, 2008: 401-409

[15] Zadeh L A. Fuzzy sets. Information and Control, 1965, 8(3): 338-353

[16] Atanassov K. Intuitionistic fuzzy sets. Fuzzy Sets and Systems, 1986, 20(1): 87-96

[17] Yager R R. Connectives and quantifiers in fuzzy sets. Fuzzy Sets and Systems, 1991, 40(1): 39-75

[18] Gau Wen-Lung, Buehrer D J. Vague sets. IEEE Transactions on Systems, Man, and Cybernetics, 1993, 23(2): 610-614

[19] Pawlak Z. Rough sets. International Journal of Computer and Information Sciences, 1982, 11: 341-356

[20] Lei Ying-Jie, Sun Jin-Ping, Wang Bao-Shu. Some extension of fuzzy knowledge processing and fuzzy sets. Journal of Air Force Engineering University, 2004, 5(3): 40-44 (in Chinese)
(雷英杰,孙金萍,王宝树. 模糊知识处理与模糊集理论的若干拓展. 空军工程大学学报, 2004, 5(3): 40-44)

[21] Pan Zhenghua, Zhang Lijuan. A new fuzzy set with three kinds of negations and applications to decision making in financial investment. Journal of Human and Ecological Risk Assessment, 2011, 17(4): 795-780

[22] Xu Jiang, Pan Zhenghua. Representation and reasoning of fuzzy knowledge with its different negations in financial investment decision making. Computer Applications and Software, 2011, 28(3): 37-40(in Chinese)
(徐江,潘正华. 金融投资决策中的模糊知识及其不同否定的表示与推理. 计算机应用与软件, 2011, 28(3): 37-40)

[23] Yang Lei, Pan Zhenghua. Fuzzy degree, closeness degree and applications of fuzzy set with three kinds of negations. Computer Applications and Software, 2011, 29(1): 51-59(in Chinese)
(杨磊,潘正华. 具有三种否定的模糊集FSCOM的模糊度与贴近度及其应用. 计算机应用与软件, 2011, 29(1): 51-59)

PAN Zheng-Hua, born in 1957, professor. His research interests include knowledge representation and knowledge reasoning, classical logic and nonclassical logics, etc.

Background

Negation in knowledge processing is an important notion, especially in fuzzy knowledge processing. The concept of negation plays a special role in nonclassical logics and in knowledge representation formalisms, in which the negative information has been taking into account par with positive information. Some scholars suggested that uncertain information processing needed different negations in various domains[1-14]. Wagner considered that negation is not a clean concept, there are (at least) two kinds of negation in these domains: a weak negation expressing non-truth (in the sense of "she doesn't like snow" or "he doesn't trust you"), and a strong negation expressing explicit falsity (in the sense of "she dislikes snow" or "he distrusts you")[1-3]. Ferré introduced an epistemic extension for the concept of negation in Logical Concept Analysis and Natural Language, the aim is to allow for the distinction between negation, opposition, and possibility in a unique formalism, and proposed that there are extensional negation and intentional negation[8]. Kaneiwa proposed that description logic $ALC_{\sim}$ with classical negation and strong negation, the classical negation $\neg$ represents the negation of a statement, the strong negation $\sim$ may be more suitable for expressing explicit negative information (or negative facts), in other worlds, $\sim$ indicates information that is directly opposite and exclusive to a statement rather than its complementary negation[9]. Since 2006, we introduced an epistemic extension for the concept of negation in knowledge processing, proposed that negative relations in knowledge should differentiate contradictory relation and opposite relation, and described to these relations using the medium logic, as well as application to fuzzy information representation[10-14]. However, for all of FS (Zadeh's Fuzzy Set) and various extensions of FS such as Intuitionistic Fuzzy Set, Vague Sets and Rough Set, their ideas as CS (Classical Set) for negative concepts, in which there is only a negation in theories, only definiens form is difference. Therefore, they can not describe to contradictory negation, opposite negation and medium negation of fuzzy knowledge. This paper aim to investigate base of set which deal with contradictory negation, opposite negation and medium negation of fuzzy knowledge.

This work was supported by the National Natural Science Foundation of China (60973156) and the Program for Innovative Research Team of Jiangnan University.

(l,d)-模体识别问题的遗传优化算法

霍红卫 郭丹丹 于 强 张懿璞 牛 伟
(西安电子科技大学计算机学院 西安 710071)

摘 要 转录因子结合位点识别在基因表达调控过程中起着重要的作用.文中提出了一种贝叶斯模型驱动的模体识别的遗传优化算法 GOBMD(Genetic Optimization with Bayesian Model for Motif Discovery).GOBMD 首先使用一个基于位置加权散列的投影过程,将输入序列中的 l-mers 投影到 k 维($k<l$)子空间,找出 DNA 序列中的起始良好候选模体,作为遗传算法的初始群体,以进一步求精.在遗传迭代过程中,采用结合贝叶斯模型的适应度函数指导进化过程.模拟数据的实验结果表明,与 Gibbs、WINNOWER、SP-STAR、PROJECTION 这些模体识别算法相比,GOBMD 在对植入(l,d)-模体识别时有较好的性能,能够解决大部分挑战性的植入(l,d)-模体识别问题.此外,作者用 Boxplot 显示了上述模体识别算法在模拟数据识别上的性能系数分布,结果表明 GOBMD 具有较好的效率.针对真实生物序列的实验结果同样表明了 GOBMD 算法的有效性.

关键词 模体识别;遗传算法;贝叶斯模型;散列;投影
中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2012.01429

Genetic Optimization for (l,d)-Motif Discovery

HUO Hong-Wei GUO Dan-Dan YU Qiang ZHANG Yi-Pu NIU Wei
(*School of Computer Science and Technology, Xidian University, Xi'an* 710071)

Abstract Transcription factor binding site (TFBS) detection plays an important role in gene finding and understanding gene regulation relationship. Motifs are weakly conserved and motif discovery is a challenging problem. We propose a new approach called Genetic Optimization with Bayesian model for Motif Discovery (GOBMD). GRBMA first uses a position-weight hashing based projection, which mapping the l-mers in DNA sequences into some k-demission subspaces ($k<l$), to find good starting candidates motifs. GOBMD then employs an effective genetic refinement to evolve the candidate motifs for further optimization. GOBMD also incorporates the Bayesian formula and relative entropy in its fitness to find the best configuration of sites locations. Experimental results on simulated data show that GOBMD can compete with Gibbs, WINNOWER, SP-STAR, PROJECTION on most implanted (l,d)-motif finding problems. We compare the performance coefficient scores for identifying (l,d)-motif finding problems by making separate box plots for each of the algorithms listed above. The experimental results on realistic biological data by identifying a number of known transcriptional regulatory motifs in eukaryotes also show that GOBMD can predict the TFBSs efficiently.

Keywords motif identification; genetic algorithm; Bayesian model; hashing; projection

收稿日期:2011-05-20;最终修改稿收到日期:2011-10-31.本课题得到国家自然科学基金(69601003)、博士点基金(20100203110010)和青年科学基金(60705004)资助.**霍红卫**,女,1963 年生,博士,教授,主要研究领域为算法设计与分析、并行算法、生物信息学算法.E-mail:hwhuo@mail.xidian.edu.cn.**郭丹丹**,女,1984 年生,硕士,主要研究方向为大规模应用问题的算法及软件、生物信息学算法.**于 强**,男,1983 年生,博士研究生,主要研究方向为生物信息学算法、并行算法.**张懿璞**,男,1985 年生,博士研究生,主要研究方向为生物信息学算法、并行算法.**牛 伟**,男,1987 年生,硕士,主要研究方向为大规模应用问题的算法及软件、生物信息学算法.

1 引 言

识别未经比对的 DNA 序列中的模式(也称模体)是生物信息学中的一个基本问题,对于发现调控信号和破解基因组中的调控编码有着重要的意义.虽然近 20 年来,出现了解决该问题的很多算法,但由于模式通常较短、退化以及规律性较差,使得设计一种识别隐含在给定未比对序列中模体的方法仍然是一个难题[1].模体识别算法可以分为 3 大类[2]:(1)基于单个基因组的共调控基因的启动子序列识别模体的算法;(2)基于多个物种的单个基因的直系启动子序列(即进化足迹)的识别模体的算法;(3)基于共调控基因和进化足迹的启动子序列识别模体的算法.早期的模体识别算法分为两类:词枚举法和位置权重矩阵(Position Weight Matrix,PWM)更新法.词枚举法表示简单,利于使用统计方法准确枚举模体在基因组中的生物学意义,并能保证全局最优性.但这种方法只适合于较短模体[3].YMF[4]是基于词枚举的方法,它使用三阶马尔可夫模型来对背景序列建模,输出 Z-得分最大的模体.其它方法还有基于统计推理的 WORDUP[5]以及基于后缀树的 WEEDER[6].Marsan 和 Sagot[7]扩展了这种方法,采用优化的数据结构(如后缀树)改进了这种方法的性能,并用于复合模体的识别.MITRA 算法[8]是以 Pevzner 和 Sze[9]提出的 WINNOWER 算法为基础,采用不匹配树作为数据结构,提出了修剪模式空间的新方法 MITRA,该方法可以更有效地使用相似对信息,可以解决像二联体这样的复合模式识别问题,也能求解(15,4)问题.基于谱的方法[10-11]在实际中非常有用,但不能保证收敛到最佳二分体信号,且在检测较弱的二分体信号时可能失效.

基于概率模型的方法需要使用最大似然估计或贝叶斯推理来估计其参数.在基于概率模型的方法中,是使用位置权重矩阵表示模体模型.虽然基于概率模型的方法要求较少的参数,但是这种方法依赖于调控区域的概率模型,对于输入数据的微小变化非常敏感.这种方法能够定量捕获一组 DNA 位点的变异信息.最有代表性的方法有 BioProspector[11]、CONSENSUS[12]和 MEME[13].

BioProspector 是一种基于吉布斯采样的方法,它使用 0～3 阶的马尔可夫背景模型,模型参数由用户给定,或者从一些特定序列估计而得,模体的统计重要性由 Monte Carlo 方法所估计的得分分布来确定.CONSENSUS 是一种基于贪心概率模型的方法,它根据大方差统计法来估计一个给定的信息容量的统计意义,并输出具有最大信息容量的位点.MEME 扩展了 Lawrence 和 Reilly[14]提出的模体发现的 EM 算法,将每个长为 l 的子序列 l-mer转化为一个字符概率矩阵,并使用此矩阵作为 EM 算法的起点.在模体已知信息较少的情况下,可以识别未经比对的生物高分子序列中的新的模体.算法输出具有最强统计意义的模体(E-值).这种方法几乎保证了 EM 算法的良好起始点.然而,对于各种长度的 l-mers(如 6～30),检查所有可能的起始点对于大型数据集而言,计算复杂度较高,计算时间较长.AlignACE[15]是一种基于吉布斯采样的方法,它使用最大先验对数似然(Maximum a Priori log-likelihood,MAP)得分来评价所采样的不同模体.PROJECTION 算法[16]通过投影模板将输入序列中的每个 l-mer 投影到一个较小的空间中,再使用 EM 算法进行求精.Xie 等人[17]首先枚举出长为 12～22 的 l-mer 的候选列表,然后对于人类基因组的一组非编码保守元素,统计匹配上的实例个数.算法 MDScan[18]首先搜索 ChIP-array 高度丰富的片段,得到多个候选模体模式,然后通过贝叶斯统计公式导出的统计得分函数来指导更新过程,不断更新和求精候选模体.Zhang 等人[19]提出了一种结构模体提取算法 EXMOTIF.

最近,Bi[20]提出了在局部比对空间中进行随机采样的模体识别的蒙特卡罗 EM 算法 MCEMDA.MCEMDA 从初始模型开始,然后迭代执行蒙特卡罗模拟,并进行参数更新,直到收敛.Davila 等人[21]提出了植入(l,d)-模体识别的快速精确算法,可以解(17,6)和(19,7)这样的难题.

Zare-Mirakabad 等人[22]提出了基于遗传算法识别二联体模式的方法,建立了以 SP 值、匹配数和信息容量多个目标的适应度函数.Li[23]提出了耦合 EM 算法的遗传算法来识别二联体的方法.Wei 和 Jensen[24]使用位置权重矩阵作为模体描述模型,并在标准遗传算法的基础上,引入了面向模体识别问题的两个特定的遗传算子.Huo 等人[25-26]提出了模体识别的优化遗传算法.

由于遗传算法在进行全局搜索时,还维持着一个候选解组成的群体,这样遗传算法就比那些局部搜索和单点搜索方法更有效,然而,基于遗传算法的

模体识别方法，由于表示候选模体的初始群体是随机产生的，加之搜索空间巨大，需要较大的群体规模和迭代次数，这样就会对遗传算法的速度和有效性产生一定的影响. 本文中我们提出了模体识别的新算法 GOBMD(Genetic Optimization with Bayesian model for Motif Discovery)，并将该算法应用于转录因子结合位点的识别. GOBMD 结合随机投影策略和散列法计算出候选模体，作为遗传算法的初始群体进行优化，在适应度函数中结合信息熵和贝叶斯模型，更准确地表征了每个候选模体的特征. 此外，对结果采用局部优化技术，提高了识别性能. 模拟数据的实验结果表明，与 Gibbs、WINNOWER、SP-STAR、PROJECTION 这些经典的模体识别算法相比，GOBMD 在进行植入模体识别时有较好的性能，能够解决大部分挑战性的植入模体问题. 同时，我们利用图盒对这些方法的性能系数的统计意义做了阐述，结果表明 GOBMD 具有较好的效率. 针对真实数据的实验结果同样表明了 GOBMD 算法的有效性.

2　方法概述

2.1　植入(l,d)-模体识别问题

所谓植入模体识别问题，是指在含有植入模体实例的 DNA 序列中定位出植入模体实例的过程. 由于基因变异的影响，植入(l,d)-模体的各个植入模体实例并不完全匹配，而是在部分碱基位上存在变异. 植入(l,d)-模体识别问题(Planted (l,d)-Motif Problem，PMP)的形式定义如下.

定义 1.　给定含有植入模体实例的 DNA 序列的一个集合 $S=\{s_1,s_2,\cdots,s_t\}$，$|s_i|=n$，$i=1,2,\cdots,t$，及未知模体 M 的长度 l，满足 $0\leq d<l<n$. (l,d)-模体识别问题定义为：确定一个满足 $|x|=l$ 的 l-mer x(称为模体)，使得 s_i 中存在长为 l 的模体实例 x_i，满足 x 与 x_i 至多在 d 个位置不同，$i=1,2,\cdots,t$.

生物信息学中通常用 l-mer 表示长为 l 的序列. 如果植入模体实例是完全保守的，即模体位点未发生变异，那么植入(l,d)-模体识别问题可以简化为计算长度为 l 的子串在 DNA 序列中出现次数的问题. 如果植入模体实例不是完全保守的，那么植入(l,d)-模体识别问题的实质是一种相似比对. Buhler 和 Tompa[16] 对植入(l,d)-模体识别问题的难度进行了详细的概率分析，并且导出在每条 DNA 序列至少包含一个植入模体实例的情况下，植入模体实例个数的期望值为 $E(l,d)$.

$$E(l,d)=4^l(1-(1-P_d)(n-l+1))^t \quad (1)$$

其中 P_d 是含有 d 个变异的植入模体实例出现在 DNA 序列的某个位置的概率，n 是 DNA 序列的长度，t 是 DNA 序列的个数. 数学期望 E 的意义在于能够事先估计求解植入(l,d)-模体识别问题的难度，这个计算结果表示随机生成的 t 条长度为 n 的 DNA 序列中可能存在的植入模体实例个数的期望值.

2.2　GOBMD 算法

优化遗传算法 GOBMD 首先利用随机投影算法 RPS[25] 所构造的模体候选集作为初始群体 P_0；然后，执行遗传算法的迭代过程，在适应度函数中结合了信息熵及贝叶斯模型，用以评价每个个体，并通过交叉算子和变异算子的作用，更新每一代群体，不断重复这个过程，直到收敛；最后，执行局部优化过程，消除遗传迭代后结果可能出现的漂移问题. GOBMD 模体发现算法如下所示.

算法.　GOBMD$(t,n,l,d,k,\theta,m,p_c,p_m)$.

1. create the initial population P_0 by RPS
2. evaluate the population in P_0 with Bayesian scoring function
3. while (stopping criteria is not satisfied) do
4. 　　perform selection operator
5. 　　perform mutation operator
6. 　　perform crossover operator
7. 　　evaluate current population
8. 　　perform the elitist strategy
9. local optimization
10. return

其中，k 为投影规模；θ 为桶的阈值；m 为投影次数；p_c、p_m 分别为交叉概率和变异概率，我们将在 4.1 节详细讨论这些参数的选取和计算.

2.3　基于散列的随机投影

在随机投影算法 RPS[25] 的每一次投影中，序列中的每个 l-mer 按照投影模板被投影到相应的桶中，计算每个合格桶(桶中序列数达到给定阈值的那些桶)的位置权重矩阵 $\boldsymbol{W}$，进而计算出每条输入序列中在该矩阵下具有最大似然比的一个 l-mer，并记录其所在位置. 从而得到 t 个 l-mer，由此计算出其频率权重矩阵 $\boldsymbol{FWM}$ 及其一致序列，该一致序列即为初始群体的一个个体. 图 1 给出了一个一致序列的构造过程.

比对矩阵

	A	G	G	T	A	C	T	T
	C	C	A	T	A	C	G	T
	A	C	G	T	T	A	G	T
	A	G	G	T	C	C	A	T
	C	C	G	T	A	C	G	G

FWM

A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4

一致序列

	A	C	G	T	A	C	G	T

图 1 一致序列构造

从每条 DNA 序列中选出一个位置，由所有这些位置处的 l-mer 可构造出一个比对矩阵. 图 1 中的比对矩阵是 5 个 8-mer 构成的比对矩阵实例. 然后，统计出这个矩阵中每列中字符的频率，选择每列中频率最大的字符，即可得到一致序列. 对于每个合格桶，均可构造出这样的一个个体. 基于随机投影策略的初始群体的构造过程如图 2 所示.

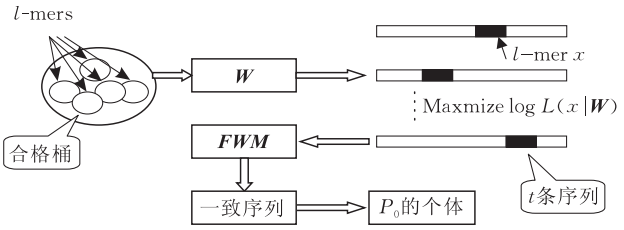


图 2 初始群体构造

投影算法 RPS 在每一次投影前随机选取 $1\sim l$ 个碱基位中的 $k(k\leq l)$ 个无重复的投影位置组成的集合 p，$p=\{p_i \mid 1\leq i\leq k<l, 1\leq p_i\leq l\}$. 然后，算法通过一个投影函数 f，根据投影位置集 p 将 DNA 序列中的每个 l-mer 投影到相应的桶中，接着进入下一次投影. 经过 m 次投影之后，就会发现某些桶内的 l-mer 和模体的一致序列在许多碱基位上是相同的，这样就为进一步搜索提供了较好的初始点，同时，缩小了搜索空间. 为了保证经过 m 次 k 个位置的投影之后，能以高概率(95%以上)得到至少一个较好的初始点，文献[16]给出了成为合格桶的阈值 θ 的经验值 4，以及投影大小 k 和投影次数 m 的具体的计算方法.

在同一个合格桶中的 l-mer 的共同的特征是：在投影的相应位置上的碱基都相同. 对于合格桶中的 l-mer，可以计算出其一致序列，在此基础上，再对合格桶施加一个逐步求精的迭代步骤就可以还原出原始模体.

根据投影位置集 p 及对应位置上的碱基，可以构造一个散列函数，来实现这个投影过程，并将散列函数值作为桶的标识.

桶的标识计算如下：对 l-mer 在投影位置上的碱基进行数字编码，A 为 1，C 为 2，G 为 3，T 为 4，在 k 个投影位置之外的碱基编码为 0，即投影时不考虑其它位置上的碱基. 可以看出这是一个 5 进制数的编码.

给定投影位置集 p 以及 l-mer，定义 l-mer x 的第 j 个位置的权值 $w(j)$ 为

$$w(j)=b^{j-1} \tag{2}$$

其中，b 为 b 进制的基数，对于DNA序列，b 取值为 5. 定义 l-mer x 在投影 p 下的散列值 $h_p(x)$ 为其上每个位置权值与其对应碱基编码乘积之和模 B 的结果：

$$h_p(x)=\Big(\sum_{j\in p} x_j \cdot w(j)\Big) \bmod B \tag{3}$$

其中，p 为投影位置集；x_j 是与 l-mer x 的第 j 个位置的碱基所对应的数字编码，范围为[1，4]，B 是散列表的大小.

将有相同散列值的 l-mer 存放到同一个桶中，桶的标识即为该散列值，如图 3 所示.

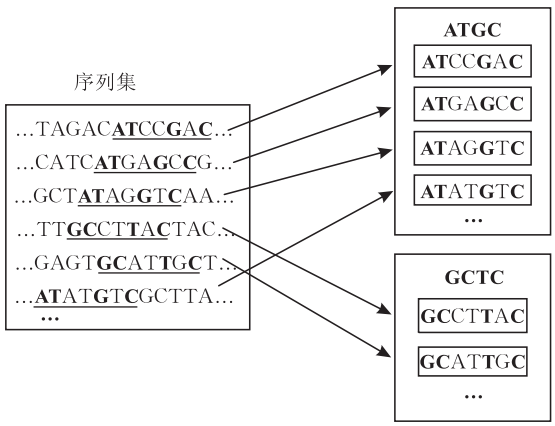


图 3 随机投影(其中 $l=7, k=4, p=(1,2,5,7)$)

当 m 次投影结束之后，由合格桶的集合构造遗传迭代需要的初始群体 P_0，具体的构造方法是根据合格桶内的所有 l-mer 形成该桶的位置权重矩阵 $\boldsymbol{W}$，在每条输入序列中寻找一个使 $L(x|\boldsymbol{W})$ 达到最大的 l-mer x，将找到的 t 个 l-mer 形成它们的一致序列和频度权值矩阵 ***FWM***，将该一致序列作为个体放入初始群体集合 P_0 中.

$$L(x|\boldsymbol{W})=\sum_{1\leq i\leq l} \log P(x_i|\boldsymbol{W}) \tag{4}$$

其中 $P(x_i|\boldsymbol{W})$ 为碱基 x_i 在位置权重矩阵 $\boldsymbol{W}$ 的第 i 列出现的概率.

3 编码及适应度函数

在每一次投影结束之后，保存产生的合格桶到

一个集合中，该集合内的元素在 m 次投影之后全部为较优的初始模体（合格桶），在该集合中挑选部分或者全部的合格桶形成初始群体，在初始群体的基础上执行遗传算法的迭代过程，终止时最优的个体为所查找的模体.

3.1　个体表示

由于初始群体是由存放合格桶的集合形成的，那么每个合格桶就对应着初始群体中的一个独立个体（这里按照初始群体对个体数目不做限制进行讨论）. 首先将每个合格桶数据结构转化为初始群体中的每个个体. 采用如下方法：首先由合格桶内的所有 l-mers 计算出该桶的位置权重矩阵 $\boldsymbol{W}$；然后，在每条输入序列中找出一个使得似然 $L(x|\boldsymbol{W})$ 达到最大的 l-mer x. 对于 t 条输入序列，可以找出 t 个这样的 l-mer，由这些 l-mer 所在相应序列中的位置可构成一个 t-元组 $(b_1,b_2,\cdots,b_t)$，如图 4 所示，其中 b_i 表示模体在第 i 个序列中起始位置. 将每个这样的 t 元组作为群体中的一个个体. 重复这个过程，由每个合格桶都可得到一个 t-元组，从而也就形成了遗传算法的初始群体.

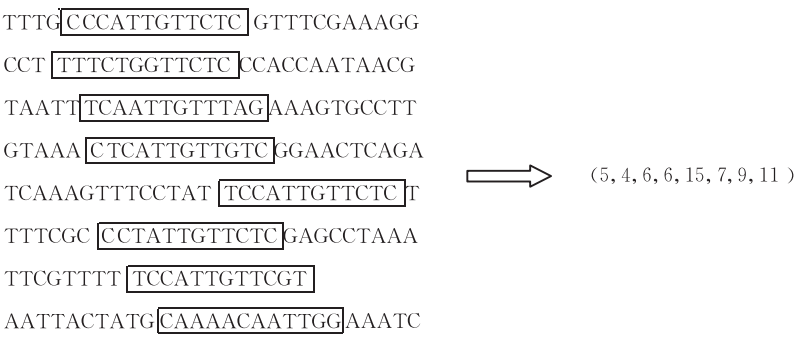


图 4　一个 8 元组的构造过程

3.2　适应度函数

首先通过 RPS 构造出候选模体，将其作为遗传算法的初始群体，其中每个个体对应着一个位置权重矩阵. 然后，通过交叉算子、变异算子等的操作，产生新一代群体，新群体中的每个个体所对应的位置权重矩阵为新的位置权重矩阵，使用更新后的位置权重矩阵扫描输入数据的结合位点. 最后，结合贝叶斯模型[24,27]，计算比对结合位点（矩阵）的得分值，作为结合位点适应度的评价. 如式(5)所示.

$$\psi_{\text{ent}}(A)=|A|\left(\log\left(\frac{\hat{p}_0}{1-\hat{p}_0}\right)-1+\sum_{j=1}^{w}\sum_{k=1}^{4}\hat{\theta}_{jk}\log\left(\frac{\hat{\theta}_{jk}}{\theta_{0k}}\right)\right) \tag{5}$$

其中，$|A|$ 是预测的位点数，即在序列集合里出现的植入模体实例个数之和；$\hat{p}_0=|A|/L$ 是预测的模体实例出现的频率，其中 $L=\Sigma_i(l_i-w+1)$ 表示模体 A 中潜在的模体实例个数之和；$\sum_{j=1}^{w}\sum_{k=1}^{4}\hat{\theta}_{jk}\log(\hat{\theta}_{jk}/\theta_{0k})$ 是预测模体的矩阵频率 $\hat{\theta}_{jk}$ 和背景频率 θ_{0k} 之间的相对熵. 通常将少量的伪数目 β 加到预测模体矩阵的每个元素上，以确保模体的位置权重矩阵中的所有元素 $\hat{\theta}_{jk}$ 均不为零.

3.3　遗传算子

变异算子作为一种局部随机搜索算子，可以保证遗传算法群体的多样性，它与交叉/选择算子结合在一起，保证了遗传算法的有效性. 本文算法采用单点变异，变异算子在个体 $(b_1,\cdots,b_i\cdots,b_t)$ 中随机选择一个位置 i，然后以概率 p_m 对该位置上的值 b_i 进行变异，发生单点变异后的个体为 $(b_1,\cdots,b_i'\cdots,b_t)$，即序列 i 中的植入模体实例起始位点由序列 i 的第 b_i 个位点变成了序列 i 的第 b_i' 个位点，如图 5 所示. 在图 5 中，元组(4,3,5,9,2)的第 3 个位置上的值发生了变异，即在 DNA 序列集 S 中的第 3 条序列的模体实例的起始位点发生了变化，变异前模体起始位点为第 5 个位置，变异后模体起始位点为第 4 个位置，相应的植入模体实例从 ATCCGT 变成了 CATCCG.

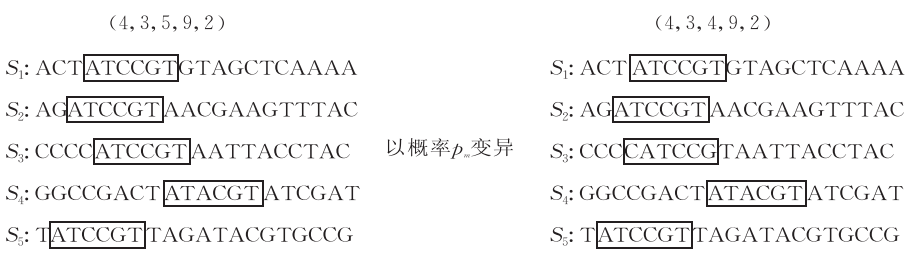


图 5　变异算子

本文算法采用单点交叉.首先将初始群体中的个体随机配对.对于每对个体$(a_1,\cdots,a_t)$和$(b_1,\cdots,b_t)$,随机选择交叉点,执行交叉操作,生成的两个新个体$(a_1,\cdots,a_c,b_{c+1},\cdots,b_t)$和$(b_1,\cdots,b_c,a_{c+1},\cdots,a_t)$,如图6所示.在图6中,交叉前的一对个体是(4,3,5,9,2)和(3,2,4,1,6),随机选择的交叉点是元组的第3个位置,单点交叉操作后,产生两个新个体(4,3,5,1,6)和(3,2,4,9,2).

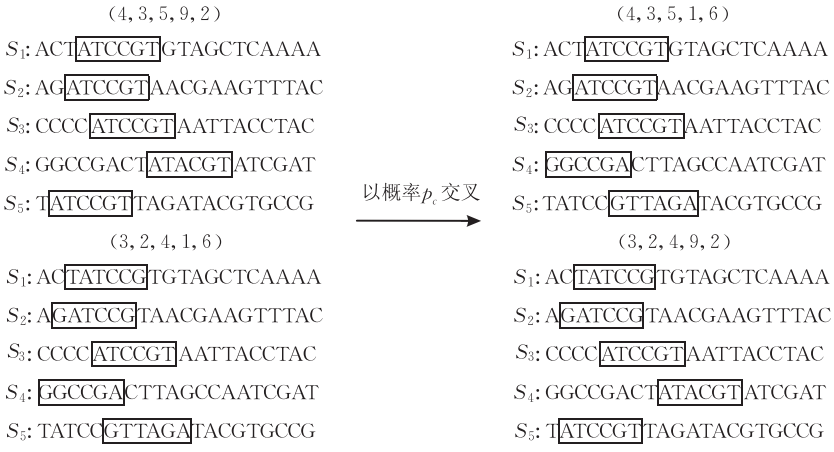


图6 交叉算子

选择算子在当前群体中随机选择M次,每次选择两个个体组成竞赛小组(tournament),比较组内两个个体的适应度,将适应度高的个体遗传到下一代群体当中.除此之外为了保证适应度最高的个体不出现退化且不会使算法迅速收敛到局部最优,GOBMD算法保存当前群体中适应度最高的个体,使它不参与交叉运算与变异运算,最后用它替换本代群体经过交叉、变异操作后所产生的适应度最低的个体.

4 实验结果与分析

4.1 参数设置

在序列集合中,t表示DNA序列个数,序列长度用n表示,在模拟数据中t和n对所有模体识别问题都相同;在真实生物数据中,t和n的值与具体数据有关.l表示模体或模体实例的长度,d表示不匹配碱基数.除此之外,GOBMD在使用投影策略生成初始解空间时,还需要确定3个重要参数:投影规模k、投影次数m和桶的阈值θ.

4.1.1 投影规模

投影规模k必须大小合适才能够保证算法正常运行.如果k过大,会导致l-mer的投影过于分散,投影到每个桶中的l-mer数量非常少,不容易构造出良好的候选模体,不利于迭代算法进一步求精;反之如果k取值过小,会造成投影效果过于集中,即很多l-mer都被投影到同一个桶中,使得桶中含有很多噪声(无用信息),同样不利于算法查找正确的模体.

假设将$t(n-l+1)$个来自背景序列的l-mer随机投影到总共4^k个桶中,设E为平均每个桶含有的l-mer的个数,则得

$$E=\frac{t(n-l+1)}{4^k} \tag{6}$$

我们希望每个桶中来自于背景序列的l-mer个数越少越好,因为背景序列是随机生成的,不含植入模体的信息,局部搜索算法如果针对随机生成的数据展开计算,不仅增加计算上的负担,而且影响算法进一步求精的准确性,干扰正常的局部搜索.所以代表背景序列l-mer平均到每个桶中的个数E越小越好,通常要求$E<1$,这里取$E\leq 0.90$,则上式可以写成

$$0.90\geq\frac{t(n-l+1)}{4^k} \tag{7}$$

由上式解出k,可得

$$k\geq\log_4\left(\frac{t(n-l+1)}{0.90}\right) \tag{8}$$

本文中k取值为$\left\lceil\log_4\left(\frac{t(n-l+1)}{0.90}\right)\right\rceil$.

4.1.2 投影次数

接下来确定算法的投影次数m.由于投影是随机且均匀地在$1\sim l$个位置中选择k个投影位置.而模体在l个碱基位置中的d个位置发生变异得到模体实例,那么模体实例被投影到存放植入模体的桶的概率为$p(l,d,k)$为

$$p(l,d,k)=\frac{C_{l-d}^{k}}{C_{l}^{k}} \tag{9}$$

存放植入模体的桶是指桶内存放的l-mer大多数是模体实例,并且有足够的数量来启动一个针对

该桶的局部搜索过程.通常认为这样的桶经过求精步骤最终得到植入模体的可能性很大.

设 $B_{t,p}(x)$ 表示在 t 次独立的伯努利实验中，至多有 x 次成功的概率，每次实验成功的概率为 p.则在算法执行的每一次循环(投影)中，有不超过 θ 个模体实例投影到存放植入模体桶中的概率是 $B_{t,p(l,d,k)}(\theta)$.

算法总共执行 m 次循环(投影)，则在一次循环中，至少有 θ 个模体实例投影到植入模体的桶中的概率为 q，执行的每次投影中至多有 θ 个模体实例投影到存放植入模体桶中的概率是 B，令 $q\geq 0.95$，则

$$m=\left\lceil \frac{\log(1-q)}{\log B} \right\rceil \tag{10}$$

$$q=1-[B_{t,p(l,d,k)}(\theta)]^m \tag{11}$$

为了保证植入模体桶中有足够多的模体实例，令 $q\geq 0.95$，保证以高概率启动一个局部搜索过程，则上式可写为

$$0.95\leq 1-[B_{t,p(l,d,k)}(\theta)]^m \tag{12}$$

解出上式中的 m，可得

$$m\leq \frac{\log(1-q)}{\log(B_{t,p(l,d,k)}(\sigma))} \tag{13}$$

本文中 m 取值为 $\left\lfloor \frac{\log(1-q)}{\log(B_{t,p(l,d,k)}(\sigma))} \right\rfloor$.

4.1.3　桶的阈值

算法需要选择 3 个关键的参数，即投影规模 k、投影次数 m 以及桶的阈值 θ.选择合适的参数，可以减少对无意义的数据进行处理的时间，同时可以避免随机生成的背景数据对模体发现算法的干扰，从而提高局部搜索算法的正确率.因为要将 $t(n-l+1)$ 个 l-mer 散列到 4^k(一种实现方式，所选取的散列表大小)个桶中，如果 $4^k>t(n-l+1)$，桶中所含 l-mer 的平均数将会小于 1.对于 2.1 节描述的模拟数据的挑战性植入模体问题，以及关于它的难度的分类[25]，可以选择足够大的 k，来满足这个低噪声的条件，且不违反约束条件 $k<l-d$.因为它们所含的模体实例总数不超过 20，因而，我们不能期望在合理的投影次数中，有太多的实例散列到同一个桶中.因此，桶的阈值大小设置为 3～4.实验中 θ 取值 4.

4.2　在模拟数据上的实验结果与分析

按照文献[16]提供的数据生成方法来构造模拟数据集.该集合包含 20 个序列，每条序列 600 个碱基.序列中的背景序列随机生成，然后，在每条序列中随机选择一个植入模体的起始位点，把有 d 个变异位点的植入模体实例 l-mer 植入到背景碱基序列中.每次运行的投影位置数 $k=7$，桶的阈值 $\theta=4$，交叉概率 $p_c=0.3$，变异概率 $p_m=0.001$，投影次数 m 是根据公式求得的.针对每种模体识别问题，运行程序 20 次，取 20 次的平均值作为算法在模拟数据上的最终识别结果.用性能系数 $|K\cap P|/|K\cup P|$ 作为度量对算法识别的正确率进行评估[9]，其中 K 表示在 t 个植入模体实例中的 $t\times l$ 个残基位置集合，P 表示算法预测的 t 个模体实例所对应的残基位置集合.表 1 中列出的结果是 Gibbs、WINNOWER、SP-STAR、PROJECTION 和 GOBMD 算法求解植入 (l,d)-模体识别问题时，20 次运行结果的平均性能系数，其中第 2～5 列结果来自文献[16]，m 为投影次数.

表 1　识别信号较弱模体的平均性能系数比较

(l,d)	平均性能系数					m
	Gibbs	WINNOWER	SP-STAR	PROJECTION	GOBMD	
(10,2)	0.20	0.78	0.56	0.82	0.94	72
(11,2)	0.68	0.90	0.84	0.91	0.986	16
(14,4)	0.02	0.02	0.20	0.77	0.932	647
(15,4)	0.19	0.92	0.73	0.93	0.98	172
(16,5)	0.02	0.03	0.04	0.70	0.725	1292
(17,5)	0.28	0.03	0.69	0.93	0.976	378
(18,6)	0.03	0.03	0.03	0.74	0.814	2217
(19,6)	0.05	0.03	0.40	0.96	0.997	711

从表 1 中可以看出，WINNOWER 和 SP-STAR 算法[9]成功解决了(15,4)-模体识别问题，但对于信号强度较(15,4)弱一些的模式，如(14,4)-,(16,5)-和(18,6)-模体识别问题却无能为力.PROJECTION[16]和 GOBMD 可以解决信号较弱的植入模体识别问题.虽然 PROJECTION 和 GOBMD 均能求解诸如(9,2),(11,3),(13,4),(15,5)和(17,6)这样的信号极弱的模体识别问题，但实验结果表明，GOBMD 的性能更好一些，如表 2 所示.

表 2　识别信号极弱模体的平均性能系数比较

(l,d)	平均性能系数	
	PROJECTION	GOBMD
(9,2)	0.280	0.306
(11,3)	0.026	0.195
(13,4)	0.062	0.231
(15,5)	0.018	0.203
(17,6)	0.022	0.250

图 7 用 Boxplots① 显示了上述模体识别算法在模拟数据识别上的性能系数分布，其中纵坐标表示

① Lane D. Box Plots, Connexions Web site. http://cnx.org/content/m10215/2.8/, Apr 20, 2008

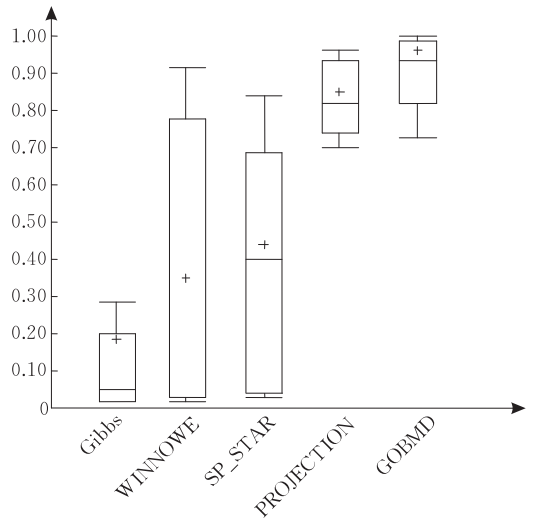


图 7 几种模体识别算法的性能系数分布图

性能系数.图 7 中的加号(+)表示算法的性能系数的平均值,组成方格的上面一条横线是 75%的性能分布线,下面的一条横线是 25%的性能分布线,在 25%分布线和 75%分布线之间的那条线是 50%的性能分布线.例如,在 GOBMD 算法的方格图中,25%的分布线对应的性能值是 0.814,算法识别的问题中有 25%的性能系数小于等于 0.814.除了基本的方格以外,图中给每个方格上下还标识了附加线,以表明性能系数数据分布的额外信息,加水平线是为了使垂直线更明显.从图 7 可以看出,比之 Gibbs、WINNOWER、SP-STAR、PROJECTION 这些算法,GOBMD 的平均性能系数较大,识别效果较好,尤其是对于(11,2),(15,4),(17,5),(19,6)这样的问题,它们的性能系数接近为 1.对于(10,2),(11,2)这样的模体识别问题,GOBMD 算法也有优势.但是,在背景序列中有一些模体实例,它们比信号微小模体更保守,GOBMD 可能把它们作为更高优先级的识别结果,这样就会出现在背景序列中识别信号微小模体时,无法准确地找到该模体,导致识别率极低的情况.

4.3 在真实数据上的实验结果与分析

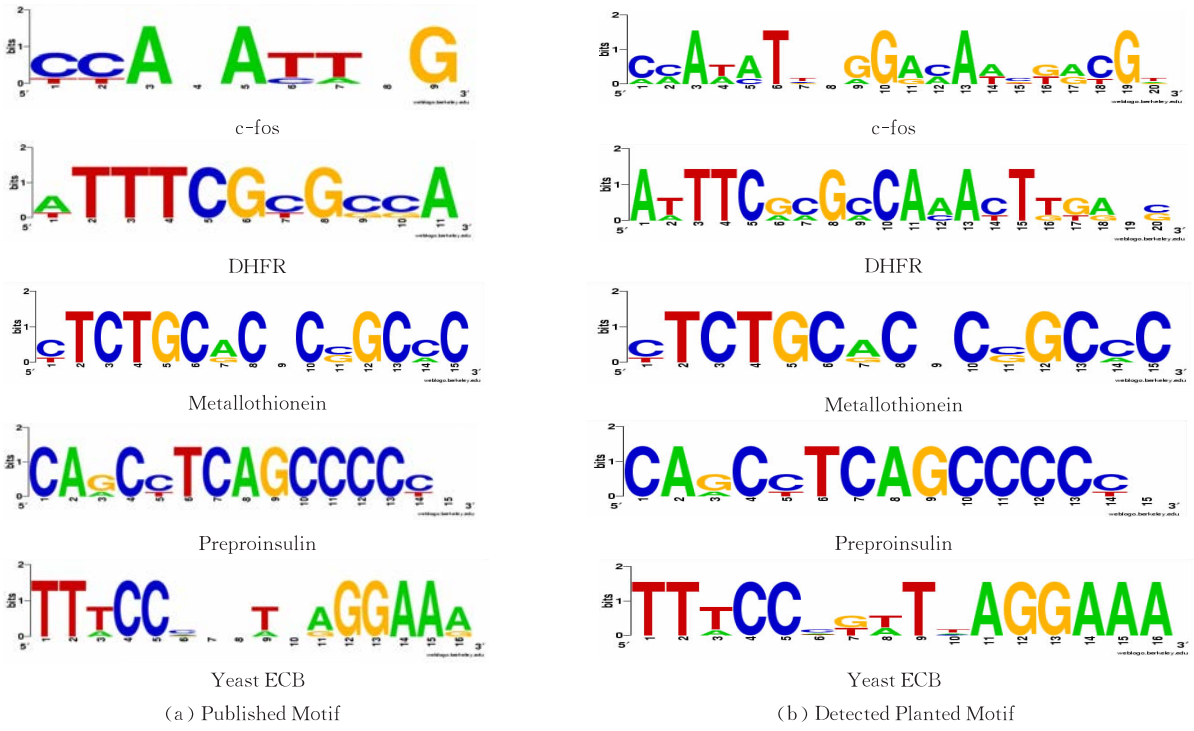
我们使用 GOBMD 算法来查找几个真核基因上游的已知转录调控元件以测试其有效性.其中同源序列取自以下 4 种类型基因的上游区域的各种生物体[28].这 4 种类型的基因是 preproinsulin、DHFR、metallothionein 和 c-fos.已知它们包含了特定转录因子的结合位点.我们还测试了取自酵母 S. cerevisiae 的一组启动子区域,已知它们包含一个共享细胞周期依赖性启动子.这些序列的信息如表 3 所示.算法中使用的参数设置如下:交叉概率为 1,变异概率为 0.0001,终止代数设为 30 代,群体大小设为 500.

表 3 真实生物序列信息

Gene(Bases)	Existing Motif	Species/genes	Input Size(Bases)
preproinsulin(11429)	CAGCCTCAGCCCCCA	AOTUSTRIVIRGATUS	2113
	CAACCTCAGCCCCCT	CAVIAPORCELLUS	1472
	CAGCCTCAGCCCCCA	HUMAN	4992
	CAGCTTCAGCCCCTC	RAT	2852
DHFR(5843)	ATTTCGCGCCA	CRIGR	1521
	TTTTCGTGGGA	MELANOGASTER	1954
	ATTTCGCGCCA	HUMAN	1133
	ATTTCGCGCCA	MOUSE	1235
metallothionein(13627)	CTCTGCGCCCGGCCC	BOVIN	2749
	CTCTGCACCCCGCCC	GRIGR	1380
	TTCTGCACACGGCAC	TROUT	1147
	CTCTGCACTCCGCCC	RAT	8351
c-fos(19554)	CCATATTAG	HUMAN	6210
	CCAAACTGG	CHICK	3858
	CCATATTAG	MOUSE	3967
	CTACATTTG	FUGRU	2522
	TCAAATATG	TEFL	2997
Yeast ECB(2631)	CCCGTTTAGGAAA	SWI4	587
	CCCTTTTAGGAAA	CDC46	263
	CCCAGAAAGGAAA	CDC47	317
	CCCACTTAGGAAA	CDC6	325
	CCCGTTTAGGAAA	CLN3	1139

图 8 显示了 GOBMD 算法预测的 preproinsulin、DHFR、metallothionein、c-fos 和 Yeast ECB 的模体序列 logo 图以及已公布的模体序列 logo 图.

表 4 给出了 GOBMD 所找到的同源序列、已公开的结果[28]以及识别的正确率.



(a) Published Motif　　(b) Detected Planted Motif

图 8　序列 logo 图

表 4　真实生物数据识别结果

Gene	Detected Planted Motif	Published Motif	(l,d)	正确率
preproinsulin	CAGCCTCAGCCCCC	CAGCCTCAGCCCCC	(15,2)	1
DHFR	TCGCGCCAAAC	ATTTCGCGCCA	(11,2)	0.8125
metallothionein	CTCTGCACRCCGCCC	CTCTGCACRCCGCCC	(15,2)	1
c-fos	CCATATTNGGACAACGACGT	CCANATTNG	(20,2)	0.63
Yeast ECB	TTTCCCGTTTAGGAAA	TTTCCCNNTNAGGAAA	(16,5)	0.8875

5　结　论

本文提出了一种最优化遗传算法，它能够有效地解决植入(l,d)-模体识别问题. 基于随机投影策略，为最优化遗传算法生成初始解空间，使最优化遗传算法有个好的起点，接下来利用贝叶斯打分函数作为适应度函数，引导遗传算法做一系列进化迭代，最终识别出来的最优个体 A_{opt} 作为最优化遗传算法识别出的最优植入模体. 从随机投影中获得的结果为遗传算法提供了一个好的起点. 这种随机投影策略也能够很好地优化其它启发式算法，例如模拟退火算法、Gibbs 采样算法.

虽然 GOBMD 在很多种情况下效果很好，但是在真实生物数据中，背景核苷分布完全不同于独立随机分布模式. 如果样本中的核苷分布是偏移的，模体识别将变得非常困难，也就是说，在真实数据中，一些核苷比其它核苷更频繁出现. 算法 GOBMD 通过适度的移动，改变最优模体的偏移现象.

GOBMD 可以扩展到未知模体宽度，也可以在大的生物数据集上测试 GOBMD，而且在处理各种模体模式时，GOBMD 可以变得更加灵活. 另外，可以设计新的散列函数式的投影部分变得更加有效，也可以对投影部分和遗传算法的进化部分实现并行，提高算法的运行速度.

参　考　文　献

[1] Tompa M et al. Assessing computational tools for the discovery of transcription factor binding sites. Nature Biotechnology, 2005, 23(1): 137-144

[2] Das Modan K, Dai Ho-Kwok. A survey of DNA motif finding algorithms. BMC Bioinformatics, 2007, 8(Suppl 7): S21

[3] GuhaThakurta D. Computational identification of transcriptional regulatory elements in DNA sequence. Nucleic Acids Research, 2006, 34(12): 3585-3598

[4] Sinha S, Tompa M. YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Research, 2003, 31(13): 3586-3588

[5] Pesole G, Prunella N, Liuni S, Attimonelli M, Saccone C.

WORDUP: An efficient algorithm for discovering statistically significant patterns in DNA sequences. Nucleic Acids Research, 1992, 20(11): 2871-2875

[6] Pavesi G, Mauri G, Pesole G. An algorithm for finding signals of unknown length in DNA sequences. Bioinformatics, 2001, 17(1): S207-S214

[7] Marsan L, Sagot M-F. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. Journal of Computational Biology, 2000, 7(3-4): 345-362

[8] Eskin E, Pevzner P A. Finding composite regulatory patterns in DNA sequences. Bioinformatics, 2002, 18(1): 354-363

[9] Pevzner P A, Sze S H. Combinatorial approaches to finding subtle signals in DNA sequences//Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB). Price Center, UC San Diego, La Jolla, California, 2000, 8: 269-278

[10] GuhaThakurta D, Stormo G D. Identifying target sites for cooperatively binding factors. Bioinformatics, 2001, 17(7): 608-621

[11] Liu X, Brutlag D L, Liu J S. BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes//Proceedings of the 6th Pacific Symposium on Biocomputing(PSB). Hawaii, 2001, 6: 127-138

[12] Stormo G D, Hartzell G W. Identifying protein-binding sites from unaligned DNA fragments. Proceedings of the National Academy of Sciences of the United States of America, 1989, 86(4): 1183-1187

[13] Bailey T L, Elkan C. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine Learning, 1995, 21(1-2): 51-80

[14] Lawrence C E, Reilly A A. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequence. Proteins: Structure, Function, and Bioinformatics, 1990, 7(1): 41-51

[15] Roth F P, Hughes J D, Estep P W, Church G M. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nature Biotechnology, 1998, 16(10): 939-945

[16] Buhler J, Tompa M. Finding motifs using random projections. Journal of Computational Biology, 2001, 9(2): 225-242

[17] Xie Xiaohui, Mikkelsen Tarjei S, Gnirke Andreas, Lindblad-Toh Kerstin, Kellis Manolis, Lander Eric S. Systematic discovery of regulatory motifs in conserved regions of the human genome, including thousands of CTCF insulator sites. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104(17): 7145-7150

[18] Liu X S, Brutlag D L, Liu J S. An algorithm for finding protein—DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. Nature Biotechnology, 2002, 20(8): 835-839

[19] Zhang Yongqiang, Zaki Mohammed J. EXMOTIF: Efficient structured motif extraction. Algorithms for Molecular Biology, 2006, 1: 21

[20] Bi Chengpeng. A Monte Carlo EM algorithm for de novo motif discovery in biomolecular sequences. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2009, 6(3): 370-386

[21] Davila J, Balla S, Rajasekaran S. Fast and practical algorithms for planted (l,d) motif search. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2007, 4(4): 544-552

[22] Zare-Mirakabad F, Ahrabian H, Sadeghi M, Hashemifar S, Nowzari-Dalini A, Goliaei B. Genetic algorithm for dyad pattern finding in DNA sequences. Genes & Genetic Systems, 2009, 84(1): 81-93

[23] Li Liping. GADEM: A genetic algorithm guided formation of spaced dyads coupled with an EM algorithm for motif discovery. Journal of Computational Biology, 2009, 16(2): 317-29

[24] Wei Zhi, Jensen Shane T. GAME: Detecting cis_regulatory elements using a genetic algorithm. Bioinformatics, 2006, 22(13): 1577-1584

[25] Huo Hong-Wei, Zhao Zhen-Hua, Stojkovic V, Liu Li-Fang, Optimizing genetic algorithm for motif discovery. Mathematical and Computer Modelling, 2010, 52(11-12): 2011-2020

[26] Huo Hong-Wei, Wang Xiao-Wu. An adaptive suffix tree based algorithm for repeats identification in a DNA sequence. Chinese Journal of Computers, 2010, 33(4): 747-754 (in Chinese)
(霍红卫, 王小武. DNA 序列中基于适应性后缀树的重复体识别算法. 计算机学报, 2010, 33(4): 747-754)

[27] Jensen S T, Liu J S. BioOptimizer: A Bayesian scoring function approach to motif discovery. Bioinformatics, 2004, 20(10): 1557-1564

[28] Blanchette M, Schwikowski B, Tompa M. Algorithms for phylogenetic footprinting. Journal of Computational Biology, 2002, 9(2): 211-223

HUO Hong-Wei, born in 1963, Ph. D., professor. Her research interests include design and analysis of algorithms, parallel algorithms, bioinformatics algorithms.

GUO Dan-Dan, born in 1984, M. S.. Her research interests include algorithms and software for large-scale applications and bioinformatics algorithms.

YU Qiang, born in 1983, Ph. D. candidate. His research interests include bioinformatics algorithms and parallel algorithms.

ZHANG Yi-Pu, born in 1985, Ph. D. candidate. His research interests include bioinformatics algorithms and parallel algorithms.

NIU Wei, born in 1987, M. S.. His research interests include algorithms and software for large-scale applications and bioinformatics algorithms.

Background

Motif discovery in unaligned DNA sequences is a challenging problem in computer science and molecular biology. Motifs can be used to determine evolutionary and functional relationships. Over the past few years, many motif discovery tools have been designed and make available to public. They vary each other mostly in their definition of what constitutes a motif, what constitutes statistical overrepresentation of a motif and what method has been used to find statistically overrepresented motifs. Benchmark experiments of some motif discovery tools reveal that the nucleotide and the binding site level accuracy are very low.

Experimental methods such as DNase footprinting, gelshift, reporter construct assays, ChIP have been used to determine DNA sites that are bounded by the TFs. However, determining hundreds or thousands of potential binding sites only using some of these methods is costly and time consuming. Also, some of the binding sites are inaccurate. As a result, some computational methods are given to help the DNA sequences' analysis. From the first approach when Pribnow discovered TATA box of saccharomycete using the early multiple sequence alignments to the newest approaches based on evolutionary algorithms for motif finding, various methods for identifying motifs have been proposed.

The brute force approach to solve the planted motif finding problem requires time $l(n-l+1)^t$, i. e., $O(\ln^t)$, because there are $(n-l+1)$ varying positions in each of the t sequences and it needs to check $(n-l+1)^t$ sets of starting positions, where t is the number of sequences with planted motif instances, n is the length of each sequence.

Recently, Bi presents a Monte Carlo EM motif discovery algorithm based on the position weight matrix updating technique. Davila et al. propose a sequence of exact algorithms for (l,d)-motif discovery problems and solve the challenging instances, (17,6) and (19,7) with little space needs. A few TFBS discovery methods using evolutionary algorithms have been recognized due to their advantages of synthesizing local search and global search of evolutionary algorithms. The evolutionary algorithms can overcome the disadvantages of local search and therefore give the better result than those algorithms that are apt to trap into a local optimum.

We propose a new approach called Genetic Optimization with Bayesian model for Motif Discovery(GOBMD). GRBMA first uses a position-weight hashing based projection, which mapping the l-mers in DNA sequences into some k-demission subspaces ($k<l$), to find good starting candidates motifs. GOBMD then employs an effective genetic refinement to evolve the candidate motifs for further optimization. GOBMD also incorporates the Bayesian formula and relative entropy in its fitness to find the best configuration of sites locations. Experimental results on simulated data show that GOBMD can compete with Gibbs, WINNOWER, SP-STAR, PROJECTION on most implanted (l,d)-motif finding problems. We compare the performance coefficient scores for identifying (l,d)-motif finding problems by making separate box plots for each of the algorithms listed above. The experimental results on realistic biological data by identifying a number of known transcriptional regulatory motifs in eukaryotes also show that GOBMD can predict the TFBSs efficiently.

量子搜索算法的多相位关系研究

金文梁

(扬州大学信息工程学院 江苏 扬州 225127)

摘 要 假设给定一个总数为 N 的无序数据库,极其复杂的计算使得几乎不可能建立一个精确的数学公式来描述这个结论:在二维复子空间中,对于一个等幅分布的初始态,存在两个定义在实数域上的相位旋转角集合以使得唯一的目标态能以 100%的成功概率找到;文中采取了一种近似的计算方法,通过归纳法推导出了多相位匹配方程.倘若其中一个相位旋转角集合中的元素个数 j 相对于 N(N 足够大)较小,则该方程就能保证唯一的目标态以较高的成功概率找到.接着,通过文中推导出的一个递推关系式,对任意给定的 $j>2$,分析了 Long 算法的计算复杂性.最后,通过一些数值模拟的实例进一步验证了多相位匹配方程的有效性.

关键词 Grover 量子搜索算法;Long 算法;Long 算子;二维复子空间;多相位匹配方程

中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2012.01440

Investigation of Multiphase Relationship for Quantum Search Algorithm

JIN Wen-Liang

(*College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu* 225127)

Abstract Suppose we are given an unsorted database of size N. Whereas the extremely complicated calculations make it almost impossible to establish a precise mathematical formulation to describe the conclusion that for a uniform initial amplitude distribution, there exist two sets of the phase rotation angles defined in the real domain such that a unique desired state can be found with certainty in a two-dimensional complex subspace, we resorted to an approximate computational method for simplifying the calculations and thus derived the multiphase matching equation by induction. This equation guarantees that a unique desired state can be found with high success probability provided the number j of elements in one of the sets of the phase rotation angles is relatively small compared to N (N is sufficiently large). In this case, for any given $j>2$, we analyze the computational complexity of Long algorithm by exploiting a recurrence relation derived in this paper. Finally, we further verify the validity of the multiphase matching equation by some examples of numerical simulation.

Keywords Grover quantum search algorithm; Long algorithm; Long operator; two-dimensional complex subspace; multiphase matching equation

1 引 言

在一个大型无序数据库中,原先的 Grover 量子搜索算法[1]能以数量级为 $O(\sqrt{N})$ 的叠代次数找到唯一的目标态.并且该算法已经被证明是最优的[2-3].迄今为止主要是从以下的方面对该量子搜索算法进行了扩展:(1)假设有多个目标态[4-5];(2)以任意的酉变换来代替 Walsh-Hadamard 变换[5-6];(3)引入了概率幅扩大的思想[5,7];(4)通过并行的

收稿日期:2011-01-06;最终修改稿收到日期:2011-12-19.**金文梁**,男,1968 年生,博士,主要研究方向为量子算法、量子信息论. E-mail: wenliang20021@yahoo.cn.

量子计算方式来进一步降低搜索次数[8]；(5) 以任意的相位旋转代替反方向的相位旋转[9]（这种推广的量子搜索算法称之为 Long 算法并给出了相位匹配条件 $\theta=\phi$）；(6) 初始态是任意的复概率幅分布，而不再是等概率幅分布[10-11]；(7) 讨论了任意的纠缠初始态[12]；(8) 将量子搜索空间由二维复子空间扩展至三维复子空间[13].

原先的 Grover 量子搜索算法[1]不能以 100% 的概率找到唯一的目标态. Long 等人[14]首先指出相位不匹配不能构造量子搜索算法. 为确保以 100% 的概率找到所指定的目标态，在二维复子空间条件下国内外的研究工作者给出了不同形式的相位旋转角公式[9,15-17]. 相位匹配在三维空间中有一个直观的几何解释，这在文献[18]中已给出. 相位匹配已经被 3 个实验所证实[19-21]. 在文献[22]中，Long 等人对有关量子搜索算法作了全面详尽的论述. 此外，Li 等人[23]指出 Long 算法中的算子 $Q=-I_\gamma U^{-1} I_\tau U$ 中的 U^{-1}（U^{-1} 是 U 的逆算子）可被任意的酉算子 V 所代替，即 $Q=-I_\gamma U^{-1} I_\tau U$ 可写成 $Q=-I_\gamma V I_\tau U$ 的形式. 我们证明了在三维复子空间情形下（指偏角 $0<\psi<\pi/2$），初始态（纯态）不管以何种形式出现，目标态都不能以 100% 的概率找到[13]；但是只要无序数据库中总个数 N 足够大，那么在满足相位匹配条件 $\theta=\phi\in(0,\pi]$ 的条件下找到目标态的最大成功概率近似等于 $\cos^2\psi$[13,24]. 本文提出了这样一个推测，即在二维复子空间中，如果无序数据库中总的个数 N 足够大（这里假设只有一个目标态，初始态为平均叠加态）且正整数 $j(j\geq 2)$ 的值相对于 N 而言较小，那么对定义在实数域 $\mathbb{R}$ 上的两个任意相位角集合 $\{\theta_j,\theta_{j-1},\cdots,\theta_1\}$ 和 $\{\phi_j,\phi_{j-1},\cdots,\phi_1\}$，如果满足多相位匹配方程 $\sum_{m=1}^{j}\theta_m=\sum_{m=1}^{j}\phi_m$，则能以较高的成功概率找到目标态（在文献[25]中只讨论了 $j=2$ 的情形）. 这个结论的有效性由下面的一个工作提供了支持：Long 等[26]发现在量子搜索算法中，相位系统误差是主要误差，这样的结果导致两个相位转角的累积和不同，而随机误差不重要，这样导致两个相角的累计和基本相等.

2　二维酉矩阵表象

假设量子系统中目标态和非目标态的总个数 N 足够大，要找的唯一目标态是 $|\tau\rangle$. 单位矢量 $|\alpha\rangle$ 定义为

$$|\alpha\rangle=\frac{1}{\sqrt{N-1}}\sum_{x\notin\tau}|x\rangle \tag{1}$$

任意给定两个不为 $2k'\pi$（k' 为任意整数）的实数 ϕ 和 θ（两个相位旋转角），定义以下两个酉算子

$$U_\phi=I+(e^{i\phi}-1)|\tau\rangle\langle\tau| \tag{2}$$

$$U_\theta=I+(e^{i\theta}-1)|\eta\rangle\langle\eta| \tag{3}$$

其中 $i=\sqrt{-1}$，$|\eta\rangle$ 是由所有的目标态和非目标态所张成的 N 维复 Hilbert 空间中的一个单位向量且满足下面的关系式 (5)，那么 Long 算子 $G=-AU_\theta A^{-1}U_\phi$（$A$ 是任意的酉算子，A^{-1} 是 A 的逆算子）可表示为

$$G=-(I+(e^{i\theta}-1)|\mu\rangle\langle\mu|)(I+(e^{i\phi}-1)|\tau\rangle\langle\tau|) \tag{4}$$

I 表示单位算子，

$$|\mu\rangle=A|\eta\rangle \tag{5}$$

如果 $|\mu\rangle\in L$（L 表示由正交基 $\{|\alpha\rangle,|\tau\rangle\}$ 所张成的二维复平面），那么 $|\mu\rangle$ 可以表示成下面最一般的情形

$$|\mu\rangle=e^{it_1}\cos t|\alpha\rangle+\sin t\,e^{it_2}|\tau\rangle \tag{6}$$

$t_1,t_2\in\mathbb{R}$，$0<t\leq\beta_0$，其中

$$\beta_0=\arcsin(\sqrt{1/N}) \tag{7}$$

根据式 (4) 和 (6)，通过计算可得到

$$G\{|\alpha\rangle,|\tau\rangle\}=\{|\alpha\rangle,|\tau\rangle\}\boldsymbol{Q} \tag{8}$$

上式中

$$\boldsymbol{Q}=-\begin{pmatrix}Q_{11}&Q_{12}\\Q_{21}&Q_{22}\end{pmatrix} \tag{9}$$

$$Q_{11}=e^{i\theta}\cos^2 t+\sin^2 t,$$

$$Q_{12}=e^{i\phi}e^{-i\Lambda}(e^{i\theta}-1)\frac{\sin(2t)}{2},$$

$$Q_{21}=(e^{i\theta}-1)e^{i\Lambda}\frac{\sin(2t)}{2},$$

$$Q_{22}=e^{i\phi}(e^{i\theta}\sin^2 t+\cos^2 t),$$

参数 $\Lambda=t_2-t_1$.

3　Long 算法的多相位匹配问题

对任意给定的两个集合 $\{\theta_j,\theta_{j-1},\cdots,\theta_1\}$ 和 $\{\phi_j,\phi_{j-1},\cdots,\phi_1\}$（$j\geq 2$），定义算子

$$\widetilde{G}_j=G(\phi_j,\theta_j)G(\phi_{j-1},\theta_{j-1})\cdots G(\phi_1,\theta_1) \tag{10}$$

此算子 $\widetilde{G}_j$ 在基 $\{|\alpha\rangle,|\tau\rangle\}$ 下所对应的矩阵形式是

$$\widetilde{\boldsymbol{Q}}_j=Q(\phi_j,\theta_j)Q(\phi_{j-1},\theta_{j-1})\cdots Q(\phi_1,\theta_1) \tag{11}$$

该酉矩阵 $\widetilde{\boldsymbol{Q}}_j$ 最终可表示成下面包含 4 个实参数的最一般的矩阵形式

$$\widetilde{\boldsymbol{Q}}_j=\begin{pmatrix}\cos(\Phi_j)e^{i\alpha_j}&\sin(\Phi_j)e^{i\gamma_j}\\-\sin(\Phi_j)e^{i(\beta_j-\gamma_j)}&\cos(\Phi_j)e^{i(\beta_j-\alpha_j)}\end{pmatrix} \tag{12}$$

$\Phi_j,\alpha_j,\beta_j,\gamma_j$ 为 4 个实参数（任意的一个二阶酉矩阵都可以表示成式 (12) 的形式，参见文献[27]中等

式(5.50)). 当 $\alpha_j=\beta_j-\alpha_j$ 时,式(12)可简化为

$$\widetilde{\boldsymbol{Q}}_j=\mathrm{e}^{\mathrm{i}\alpha_j}\begin{pmatrix}\cos(\Phi_j) & \sin(\Phi_j)\mathrm{e}^{\mathrm{i}\lambda_j}\\ -\sin(\Phi_j)\mathrm{e}^{-\mathrm{i}\lambda_j} & \cos(\Phi_j)\end{pmatrix},$$

其中 $\lambda_j=\gamma_j-\alpha_j$. 当满足条件

$$\lambda_{j_n}=\lambda_{j_{n-1}}\cdots=\lambda_{j_1}=\lambda \tag{13}$$

时(λ 是一个实数),可得

$$\boldsymbol{X}_{k=\sum_{s=1}^{n}j_s}=\widetilde{\boldsymbol{Q}}_{j_n}\widetilde{\boldsymbol{Q}}_{j_{n-1}}\cdots\widetilde{\boldsymbol{Q}}_{j_1}$$

$$=\mathrm{e}^{\mathrm{i}\left(\sum_{s=1}^{n}\alpha_{j_s}\right)}\times\begin{pmatrix}\cos\left(\sum_{s=1}^{n}\Phi_{j_s}\right) & \sin\left(\sum_{s=1}^{n}\Phi_{j_s}\right)\mathrm{e}^{\mathrm{i}\lambda}\\ -\sin\left(\sum_{s=1}^{n}\Phi_{j_s}\right)\mathrm{e}^{-\mathrm{i}\lambda} & \cos\left(\sum_{s=1}^{n}\Phi_{j_s}\right)\end{pmatrix} \tag{14}$$

其中正整数 $n\geq1$,$k=\sum_{s=1}^{n}j_s$ 是总的叠代次数,整数 $j_s\geq2$.

现作一个基变换:

$$\{|\beta\rangle,|\tau\rangle\}=\{|\alpha\rangle,|\tau\rangle\}\boldsymbol{V} \tag{15}$$

其中过渡矩阵

$$\boldsymbol{V}=\begin{pmatrix}\mathrm{e}^{\mathrm{i}\lambda} & 0\\ 0 & 1\end{pmatrix} \tag{16}$$

则在基$\{|\beta\rangle,|\tau\rangle\}$下,式(14)可改写为

$$\boldsymbol{X}'_{k=\sum_{s=1}^{n}j_s}=\boldsymbol{V}^{-1}\boldsymbol{X}_{k=\sum_{s=1}^{n}j_s}\boldsymbol{V}$$

$$=\mathrm{e}^{\mathrm{i}\left(\sum_{s=1}^{n}\alpha_{j_s}\right)}\begin{pmatrix}\cos\left(\sum_{s=1}^{n}\Phi_{j_s}\right) & \sin\left(\sum_{s=1}^{n}\Phi_{j_s}\right)\\ -\sin\left(\sum_{s=1}^{n}\Phi_{j_s}\right) & \cos\left(\sum_{s=1}^{n}\Phi_{j_s}\right)\end{pmatrix} \tag{17}$$

现假设初始态$|\gamma_0\rangle=\cos\beta_0|\beta\rangle+\sin\beta_0|\tau\rangle$. 经过 Long 算子对初始态$|\gamma_0\rangle$总数为 k 的叠代作用,此时目标态$|\tau\rangle$前的概率幅变成

$$d_{k=\sum_{s=1}^{n}j_s}=\mathrm{e}^{\mathrm{i}\left(\sum_{s=1}^{n}\alpha_{j_s}\right)}\sin(\Theta+\beta_0) \tag{18}$$

其中 $\Theta=-\sum_{s=1}^{n}\Phi_{j_s}>0$. 为避免确定关系式(13),令 $\widetilde{\boldsymbol{Q}}_{j_n}=\widetilde{\boldsymbol{Q}}_{j_{n-1}}\cdots=\widetilde{\boldsymbol{Q}}_{j_1}=\widetilde{\boldsymbol{Q}}_j$,则式(17)和(18)分别变成

$$\widetilde{\boldsymbol{Q}}_2\doteq\begin{pmatrix}\mathrm{e}^{\mathrm{i}(\theta_1+\theta_2)} & t\,\mathrm{e}^{-\mathrm{i}\Lambda}\left[\mathrm{e}^{\mathrm{i}\theta_2}\mathrm{e}^{\mathrm{i}\phi_1}(\mathrm{e}^{\mathrm{i}\theta_1}-1)+\mathrm{e}^{\mathrm{i}\phi_1}\mathrm{e}^{\mathrm{i}\phi_2}(\mathrm{e}^{\mathrm{i}\theta_2}-1)\right]\\ t\,\mathrm{e}^{\mathrm{i}\Lambda}\left[\mathrm{e}^{\mathrm{i}\theta_2}(\mathrm{e}^{\mathrm{i}\theta_1}-1)+\mathrm{e}^{\mathrm{i}\phi_2}(\mathrm{e}^{\mathrm{i}\theta_2}-1)\right] & \mathrm{e}^{\mathrm{i}(\phi_1+\phi_2)}\end{pmatrix}$$

如果 $\theta_1+\theta_2=\phi_1+\phi_2=\Delta$,上式变为

$$\boldsymbol{X}'_{k=nj}=\boldsymbol{V}^{-1}(\widetilde{\boldsymbol{Q}}_j)^n\boldsymbol{V}$$

$$=\mathrm{e}^{\mathrm{i}(n\alpha_j)}\begin{pmatrix}\cos(-n\Phi_j) & -\sin(-n\Phi_j)\\ \sin(-n\Phi_j) & \cos(-n\Phi_j)\end{pmatrix} \tag{19}$$

和

$$d_{k=nj}=\mathrm{e}^{\mathrm{i}(nj)}\sin(\Theta'+\beta_0) \tag{20}$$

其中 $\Theta'=-n\Phi_j>0$. 当 $\sin(\Theta'+\beta_0)=1$,就能以 100% 的最大成功概率找到目标态$|\tau\rangle$.

4 多相位匹配方程

上节讨论了在理论上存在着两个集合$\{\theta_j,\theta_{j-1},\cdots,\theta_1\}$和$\{\phi_j,\phi_{j-1},\cdots,\phi_1\}$($j\geq2$),在基$\{|\beta\rangle,|\tau\rangle\}$下,经过式(10)中所定义的酉算子 $\widetilde{G}_j$ 对初始态$|\gamma_0\rangle=\sqrt{N-1/N}|\beta\rangle+\sqrt{1/N}|\tau\rangle$的 n 次叠代作用,目标态$|\tau\rangle$能以 100% 的最大成功概率找到. 但要建立这两个集合之间精确的数学关系式却是件非常困难的事情. 现在自然就有这样的一个问题产生:如果把条件 $\sin(\Theta'+\beta_0)=1$ 放宽为 $\sin(\Theta'+\beta_0)\approx1$,那么集合$\{\theta_j,\theta_{j-1},\cdots,\theta_1\}$和$\{\phi_j,\phi_{j-1},\cdots,\phi_1\}$应大致满足什么样的条件才可使得 $\sin(\Theta'+\beta_0)\approx1$ 成立? 对于这个问题,我们有下面的一个推测(conjecture).

推测:假设 N 较大. $\{\theta_j,\theta_{j-1},\cdots,\theta_1\}$ 和 $\{\phi_j,\phi_{j-1},\cdots,\phi_1\}$($\theta_m,\phi_m\neq2k'\pi$,$1\leq m\leq j$,$k'$是任意的整数)是定义在实数域 $\mathbb{R}$ 上的两个任意的相位旋转角集合. 如果整数 $j\geq2$ 的值相对于 N 较小且这两个相位旋转角集合满足下面的多相位匹配方程

$$\sum_{m=1}^{j}\theta_m=\sum_{m=1}^{j}\phi_m=\Delta \tag{21}$$

则总能够通过选择某个合适的整数 $n\geq1$ 以使得初始态 $|\gamma_0\rangle=\cos\beta_0|\beta\rangle+\sin\beta_0|\tau\rangle$ 在经过定义的式(10)中的算子 $\widetilde{G}_j$ 的 n 次叠代作用后,目标态$|\tau\rangle$以较高的成功概率找到.

证明. 忽略 t 的高阶项,式(9)可近似表示为

$$\boldsymbol{Q}\doteq-\begin{pmatrix}\mathrm{e}^{\mathrm{i}\theta} & \mathrm{e}^{\mathrm{i}\phi}\mathrm{e}^{-\mathrm{i}\Lambda}(\mathrm{e}^{\mathrm{i}\theta}-1)t\\ (\mathrm{e}^{\mathrm{i}\theta}-1)\mathrm{e}^{\mathrm{i}\Lambda}t & \mathrm{e}^{\mathrm{i}\phi}\end{pmatrix} \tag{22}$$

(1) $j=2$ 的情形

当 $j=2$ 时,忽略 t 的高阶项,式(11)近似成为

$$\widetilde{\boldsymbol{Q}}_2 \doteq \mathrm{e}^{\mathrm{i}\Delta}\begin{pmatrix} 1 & t\left[\mathrm{e}^{-\mathrm{i}(\Lambda-\phi_1)}-\mathrm{e}^{-\mathrm{i}(\Lambda+\phi_2-\theta_2)}+\mathrm{e}^{-\mathrm{i}(\Lambda-\theta_2)}-\mathrm{e}^{-\mathrm{i}\Lambda}\right] \\ t\left[-\mathrm{e}^{\mathrm{i}(\Lambda-\phi_1)}+\mathrm{e}^{\mathrm{i}(\Lambda+\phi_2-\theta_2)}-\mathrm{e}^{\mathrm{i}(\Lambda-\theta_2)}+\mathrm{e}^{\mathrm{i}\Lambda}\right] & 1 \end{pmatrix} \tag{24}$$

忽略对最后结果没有影响的总的相位因子 $\mathrm{e}^{\mathrm{i}\Delta}$ 并且观察到上面矩阵副对角线上的两个元素满足关系式 $\widetilde{\boldsymbol{Q}}_{2(21)}=-\overline{(\widetilde{\boldsymbol{Q}}_{2(12)})}$（$\overline{(\widetilde{\boldsymbol{Q}}_{2(12)})}$ 是 $\widetilde{\boldsymbol{Q}}_{2(12)}$ 的共轭复数），式(24)可简化为

$$\widetilde{\boldsymbol{Q}}_2 \doteq \begin{pmatrix} 1 & tr\mathrm{e}^{\mathrm{i}\delta} \\ -tr\mathrm{e}^{-\mathrm{i}\delta} & 1 \end{pmatrix} = \boldsymbol{I}+tr\boldsymbol{T} \approx \mathrm{e}^{t'\boldsymbol{T}} \tag{25}$$

$\boldsymbol{I}$ 表示单位矩阵，实数 $r>0$ 和实数 δ 通过关系式 $r\mathrm{e}^{\mathrm{i}\delta}=\mathrm{e}^{-\mathrm{i}(\Lambda-\phi_1)}-\mathrm{e}^{-\mathrm{i}(\Lambda+\phi_2-\theta_2)}+\mathrm{e}^{-\mathrm{i}(\Lambda-\theta_2)}-\mathrm{e}^{-\mathrm{i}\Lambda}$ 来定义，即 $r=|\mathrm{e}^{\mathrm{i}\phi_1}-\mathrm{e}^{\mathrm{i}(\theta_2-\phi_2)}+\mathrm{e}^{\mathrm{i}\theta_2}-1|$（符号 $|\cdot|$ 表示一个复数的模），$t'=tr$，$\boldsymbol{T}=\begin{pmatrix} 0 & \mathrm{e}^{\mathrm{i}\delta} \\ -\mathrm{e}^{-\mathrm{i}\delta} & 0 \end{pmatrix}$. 利用性质 $\boldsymbol{T}^2=-\boldsymbol{I}$，$\boldsymbol{T}^3=-\boldsymbol{T}$，$\boldsymbol{T}^4=\boldsymbol{I}$，$\boldsymbol{T}^5=\boldsymbol{T}$，…，可得

$$(\widetilde{\boldsymbol{Q}}_2)^n \doteq \mathrm{e}^{nt'\boldsymbol{T}}=\cos(nt')\boldsymbol{I}+\sin(nt')\boldsymbol{T} = \begin{pmatrix} \cos(nt') & \mathrm{e}^{\mathrm{i}\delta}\sin(nt') \\ -\mathrm{e}^{-\mathrm{i}\delta}\sin(nt') & \cos(nt') \end{pmatrix} \tag{26}$$

根据式(15)给出的基变换 $\{|\beta\rangle,|\tau\rangle\}=\{|\alpha\rangle,|\tau\rangle\}\boldsymbol{V}$，在基 $\{|\beta\rangle,|\tau\rangle\}$ 下，式(26)可重新表示为

$$(\widetilde{\boldsymbol{Q}}_2)^n=\begin{pmatrix} \cos(nt') & -\sin(nt') \\ \sin(nt') & \cos(nt') \end{pmatrix},$$

注意此时式(16)中的过渡矩阵取为

$$\boldsymbol{V}=\begin{pmatrix} -\mathrm{e}^{-\mathrm{i}\delta} & 0 \\ 0 & 1 \end{pmatrix}.$$

因此初始态 $|\gamma_0\rangle=\cos\beta_0|\beta\rangle+\sin\beta_0|\tau\rangle$ 经过算子 $\widetilde{G}_2=G(\phi_2,\theta_2)G(\phi_1,\theta_1)$ 的 n 次叠代作用（此时实际总的叠代次数为 $k=2n$），$|\beta\rangle$ 前的概率幅变成 $d_{k=2n}=\sin(nt'+\beta_0)$. 当 $\sin(nt'+\beta_0)\approx 1$，即 $n\approx\lceil(\pi/2-\beta_0)/t'\rceil$ 时，就能以较高的成功概率找到目标态，符号 $\lceil\cdot\rceil$ 表示按四舍五入的方式取符号内实数的整数.

(2) $j\geq 3$ 的情形

令 $j=j'-1$，$j'\geq 4$ 是一个相对于 N 较小的正整数. 假设在忽略 t 的高阶项的情况下矩阵 $\widetilde{\boldsymbol{Q}}_{j'-1}$ 可表示成下列形式

$$\widetilde{\boldsymbol{Q}}_{j'-1} \doteq (-1)^{j'-1}\begin{pmatrix} \mathrm{e}^{\mathrm{i}(\theta_1+\theta_2+\cdots+\theta_{j'-1})} & t\mathrm{e}^{-\mathrm{i}\Lambda}z_1 \\ t\mathrm{e}^{\mathrm{i}\Lambda}z_2 & \mathrm{e}^{\mathrm{i}(\phi_1+\phi_2+\cdots+\phi_{j'-1})} \end{pmatrix} \tag{27}$$

且其与式(23)中的矩阵 $\widetilde{\boldsymbol{Q}}_2$ 有着完全相同的性质（即对所有的 $2\leq j\leq j'-1$，矩阵 $\widetilde{\boldsymbol{Q}}_j=\boldsymbol{Q}(\phi_j,\theta_j)\boldsymbol{Q}(\phi_{j-1},\theta_{j-1})\cdots\boldsymbol{Q}(\phi_1,\theta_1)$ 的两个列矢量互相正交且该两个列矢量所对应的范数相等），这里的 z_1 和 z_2 是两个复数. 那么当 $j=j'$ 时有

$$\begin{aligned}\widetilde{\boldsymbol{Q}}_{j'} &\doteq \begin{pmatrix} \mathrm{e}^{\mathrm{i}\theta_{j'}} & \mathrm{e}^{\mathrm{i}\phi_{j'}}\mathrm{e}^{-\mathrm{i}\Lambda}(\mathrm{e}^{\mathrm{i}\theta_{j'}}-1)t \\ (\mathrm{e}^{\mathrm{i}\theta_{j'}}-1)\mathrm{e}^{\mathrm{i}\Lambda}t & \mathrm{e}^{\mathrm{i}\phi_{j'}} \end{pmatrix}\times \begin{pmatrix} \mathrm{e}^{\mathrm{i}(\theta_1+\theta_2+\cdots+\theta_{j'-1})} & t\mathrm{e}^{-\mathrm{i}\Lambda}z_1 \\ t\mathrm{e}^{\mathrm{i}\Lambda}z_2 & \mathrm{e}^{\mathrm{i}(\phi_1+\phi_2+\cdots+\phi_{j'-1})} \end{pmatrix} \\ &\doteq (-1)^{j'}\begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}\end{aligned} \tag{28}$$

其中

$p_{11}=\mathrm{e}^{\mathrm{i}(\theta_1+\theta_2+\cdots+\theta_{j'})}$，

$p_{12}=t\mathrm{e}^{-\mathrm{i}\Lambda}[\mathrm{e}^{\mathrm{i}\theta_{j'}}z_1+\mathrm{e}^{\mathrm{i}(\phi_1+\phi_2+\cdots+\phi_{j'-1})}\mathrm{e}^{\mathrm{i}\phi_{j'}}(\mathrm{e}^{\mathrm{i}\theta_{j'}}-1)]$，

$p_{21}=t\mathrm{e}^{\mathrm{i}\Lambda}[\mathrm{e}^{\mathrm{i}\phi_{j'}}z_2+\mathrm{e}^{\mathrm{i}(\theta_1+\theta_2+\cdots+\theta_{j'-1})}(\mathrm{e}^{\mathrm{i}\theta_{j'}}-1)]$，

$p_{22}=\mathrm{e}^{\mathrm{i}(\phi_1+\phi_2+\cdots+\phi_{j'})}$.

根据归纳假设很容易证明上述矩阵 $\widetilde{\boldsymbol{Q}}_{j'}$ 中的两个列矢量也互相正交且它们各自所对应的范数也相等. 从而可以得到以下结论：对任意给定的一个相对于 N 较小的正整数 j'，在忽略 t 的高阶项的情况下，所有矩阵 $\widetilde{\boldsymbol{Q}}_j(2\leq j\leq j')$ 中的两个列向量互相正交且它们所对应的范数相等.

现令 $\sum_{m=1}^{j'}\theta_m=\sum_{m=1}^{j'}\phi_m=\Delta$ 且忽略对最后结果没有影响的总的复数因子 $\mathrm{e}^{\mathrm{i}\Delta}$，则可将式(28)化简为下列形式

$$\widetilde{\boldsymbol{Q}}_{j'} \doteq (-1)^{j'}\begin{pmatrix} p'_{11} & p'_{12} \\ p'_{21} & p'_{22} \end{pmatrix} \tag{29}$$

其中

$p'_{11}=1$，$p'_{12}=t\mathrm{e}^{-\mathrm{i}\Lambda}[\mathrm{e}^{\mathrm{i}(\theta_{j'}-\Delta)}z_1+\mathrm{e}^{\mathrm{i}\theta_{j'}}-1]$，

$p'_{21}=t\mathrm{e}^{\mathrm{i}\Lambda}[\mathrm{e}^{\mathrm{i}(\phi_{j'}-\Delta)}z_2-\mathrm{e}^{-\mathrm{i}\theta_{j'}}+1]$，$p'_{22}=1$.

利用式(27)中矩阵 $\widetilde{\boldsymbol{Q}}_{j'-1}$ 两个列向量的正交性条件可得到关系式

$$p'_{12}=-(\overline{p'_{21}}) \tag{30}$$

利用式(30)可将式(29)化简为

$$\widetilde{\boldsymbol{Q}}_{j'} \doteq (-1)^{j'}\begin{pmatrix} 1 & t\tau'\mathrm{e}^{\mathrm{i}\sigma} \\ -t\tau'\mathrm{e}^{-\mathrm{i}\sigma} & 1 \end{pmatrix} \tag{31}$$

这里的正实数 τ' 和实数 σ 通过 $\tau'\mathrm{e}^{\mathrm{i}\sigma}=\mathrm{e}^{-\mathrm{i}\Lambda}[\mathrm{e}^{\mathrm{i}(\theta_{j'}-\Delta)}z_1+\mathrm{e}^{\mathrm{i}\theta_{j'}}-1]$ 来定义，即

$$\tau'=|\mathrm{e}^{\mathrm{i}(\theta_{j'}-\Delta)}z_1+\mathrm{e}^{\mathrm{i}\theta_{j'}}-1| \tag{32}$$

如果 N 足够大，那么对所有相对于 N 较小的正整数 $j'(j'\geq 4)$，式(31)可以近似表示为

$$\begin{pmatrix} 1 & t\tau' e^{i\sigma} \\ -t\tau' e^{-i\sigma} & 1 \end{pmatrix} = (-1)^{j'}(\boldsymbol{I}+t\tau'\boldsymbol{T}')$$

$$\approx (-1)^{j'}\left[\boldsymbol{I}+\sum_{N'=1}^{\infty}\frac{1}{N'!}(\xi\boldsymbol{T}')^{N'}\right] = (-1)^{j'}e^{\xi\boldsymbol{T}'} \quad (33)$$

其中

$$\xi = t\tau', \quad \boldsymbol{T}' = \begin{pmatrix} 0 & e^{i\sigma} \\ -e^{-i\sigma} & 0 \end{pmatrix}.$$

矩阵 $\boldsymbol{T}'$ 同样具有性质：$(\boldsymbol{T}')^2 = -\boldsymbol{I}$，$(\boldsymbol{T}')^3 = -\boldsymbol{T}'$，$(\boldsymbol{T}')^4 = \boldsymbol{I}$，$(\boldsymbol{T}')^5 = \boldsymbol{T}'$，…. 下面的证明步骤和 $j=2$ 时的情形类似. 重新取过渡矩阵

$$\boldsymbol{V} = \begin{pmatrix} -e^{-i\sigma} & 0 \\ 0 & 1 \end{pmatrix},$$

经过算子 $\widetilde{G}_{j'} = G(\phi_{j'}, \theta_{j'})\cdots G(\phi_2, \theta_2)G(\phi_1, \theta_1)$ 对初始态 $|\gamma_0\rangle = \cos\beta_0|\beta\rangle + \sin\beta_0|\tau\rangle$ 的

$$n \approx \lceil(\pi/2-\beta_0)/\xi\rceil \quad (34)$$

次叠代作用，就能以较高的成功概率找到目标态. 最后把符号 j' 换成 j 即得推测中的结论.

从上面的证明过程可以看出，式(21)是个近似公式，在量子搜索算法中它不是最佳的关系式，只是说明二维复子空间中，在满足该式且整数 $j\geq 2$ 相对于 N 较小的情况下，目标态(初始态是平均叠加态)仍然能以较高的成功概率找到. 现在分析对任意给定的两个相位角集合 $\{\theta_j, \theta_{j-1}, \cdots, \theta_1\}$ 和 $\{\phi_j, \phi_{j-1}, \cdots, \phi_1\}$ $(j\geq 3)$，当满足多相位匹配方程 $\sum_{m=1}^{j}\theta_m = \sum_{m=1}^{j}\phi_m = \Delta$ 时，如何来确定式(34). 当出现在式(6)中的参数 t 确定时，$\xi = t\tau'$ 可由式(32)中定义的 τ' 来予以确定. 比较矩阵(27)和(28)中的第 1 行第 2 列中的两个矩阵元素，可得以下递推关系式

$$z_{1(j-1)} = e^{i\theta_{j-1}}z_{1(j-2)} + e^{i(\phi_1+\phi_2+\cdots+\phi_{j-2})}e^{i\phi_{j-1}}(e^{i\theta_{j-1}}-1) \quad (35)$$

同样通过比较矩阵(23)和(27)中第 1 行第 2 列中的两个矩阵元素，可得

$$z_{1(j=2)} = e^{i\theta_2}e^{i\phi_1}(e^{i\theta_1}-1) + e^{i\phi_1}e^{i\phi_2}(e^{i\theta_2}-1) \quad (36)$$

对任意的 $\{\theta_{j-1}, \theta_{j-2}, \cdots, \theta_1\}$ 和 $\{\phi_{j-1}, \phi_{j-2}, \cdots, \phi_1\}$，从式(36)出发，利用递推关系式(35)，可最终求得 $z_{1(j-1)}$，最后利用矩阵(31)中第 1 行第 2 列元素的因子定义式 $\tau' e^{i\sigma} = e^{-i\Lambda}\left[e^{i(\theta_{j'}-\Delta)}z_1 + e^{i\theta_{j'}} - 1\right]$，通过式(32)来求得 τ'. 从而利用 $\xi = t\tau'$ 和式(34)可近似得到叠代次数 n，注意总的叠代次数是 $k = jn$.

显然在基 $\{|\beta\rangle, |\tau\rangle\}$ 下，可将矢量关系式 $|\gamma_n'\rangle = (\widetilde{G}_j)^n|\gamma_0\rangle$ 写成下面的矩阵形式

$$\begin{pmatrix} u_n \\ v_n \end{pmatrix} = (\widetilde{\boldsymbol{Q}}_j)^n \begin{pmatrix} \cos\beta_0 \\ \sin\beta_0 \end{pmatrix} \quad (37)$$

对于所有的整数 $n\geq 1$，成功概率由

$$P_n = |v_n|^2 \quad (38)$$

来确定.

注释：最后，我们需要指出以下两点：

(1) 从矩阵(22)中第 1 行第 2 列中的元素可得

$$z_{1(j=1)} = e^{i\phi_1}(e^{i\theta_1}-1) \quad (39)$$

这实际上就是文献[9]中所对应的 $\theta_1 = \phi_1$ 情形. 根据递归关系式(35)容易得到式(36). 因此，一方面式(21)可视为对相位匹配条件 $\theta_1 = \phi_1$[9] 的进一步推广；另一方面，却有可能出现下面的情况：当 N 不是太大且满足式(21)时，在式(37)中的 $n=1$ 情形下，此时的成功概率 $P_1 = |v_1|^2$ 也有可能较大. 这意味着当总的叠代次数 $k=j$ 时，即便任意的两个相位旋转角 θ_m 和 ϕ_m 完全独立且所有的相位对 (θ_m, ϕ_m) 都不相同 $(1\leq m\leq j=k)$，在满足多相位匹配方程(21)条件下，Long 算法也依然有效. 对这种情况，实际上可通过结合多相位匹配方程(21)和经典穷举算法，在对两个相位角集合 $\{\theta_j, \theta_{j-1}, \cdots, \theta_1\}$ 和 $\{\phi_j, \phi_{j-1}, \cdots, \phi_1\}$ 作出适当调整后，最后是仍然能够以几乎接近 100% 的成功概率找到目标态的. 由于篇幅关系，这个问题在这里就不作详述了.

(2) 当 j 的值较小时，比如：$j=1, 2$ 情形下，Long 算法的复杂性较容易确定. 但随着 j 值的逐步增大，针对不同的 j 值，该算法的复杂性以统一的公式给出却是非常困难的. 这是因为对于不同的 j 值，算法的复杂性公式完全不同. 但对于某个给定的 j 值而言，却总能够从初始关系式(39)出发，按照递推关系式(35)来确定与该 j 值所对应的 z_1 值，从而最终确定其复杂性.

5 多相位匹配方程有效性的仿真验证

本节将通过给出下面的具体算例来进一步验证多相位匹配方程(21)的有效性. 图 1～图 5 是按下列方式得出的成功概率仿真曲线图. 首先设定必要的参数：$N=10000$，$t=\beta_0=\arcsin(1/100)$，$\Lambda=2\pi/3$. 然后针对每个确定的参数 $j=2, 3, 10, 20, 50$，由计算机系统随机任意产生 $0<\phi_m, \theta_m<50$ $(m=1, 2, \cdots, j)$(调整前). 接着将 (ϕ_j, θ_j) 改变为 $\phi_j = \Delta - \sum_{m=1}^{j-1}\phi_m$，$\theta_j = \Delta - \sum_{m=1}^{j-1}\theta_m$，$\Delta = \{-4\pi/5, 0, 2\pi/3\}$(调整后 3 种情形). 成功概率 P_n 由式(38)确定.

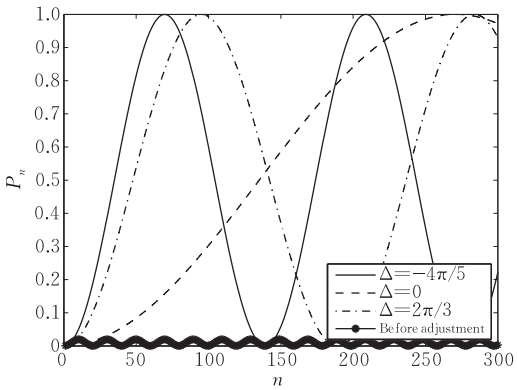


图 1　$j=2$ 时调整前和调整后成功概率 P_n 和叠代次数 n 关系曲线图($N=10^4$, $j=2$)

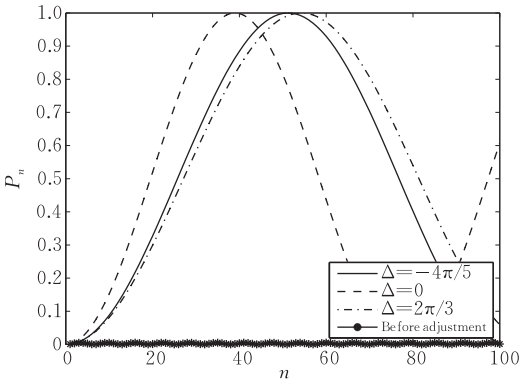


图 2　$j=3$ 时调整前和调整后成功概率 P_n 和叠代次数 n 关系曲线图($N=10^4$, $j=3$)

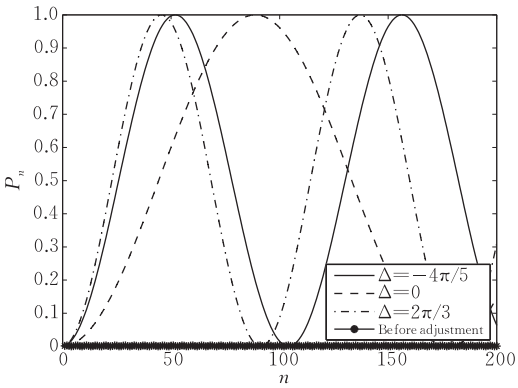


图 3　$j=10$ 时调整前和调整后成功概率 P_n 和叠代次数 n 关系曲线图($N=10^4$, $j=10$)

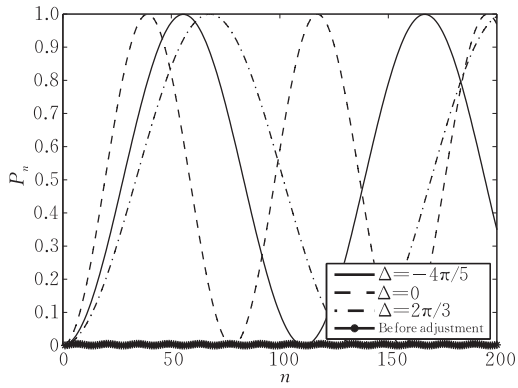


图 4　$j=20$ 时调整前和调整后成功概率 P_n 和叠代次数 n 关系曲线图($N=10^4$, $j=20$)

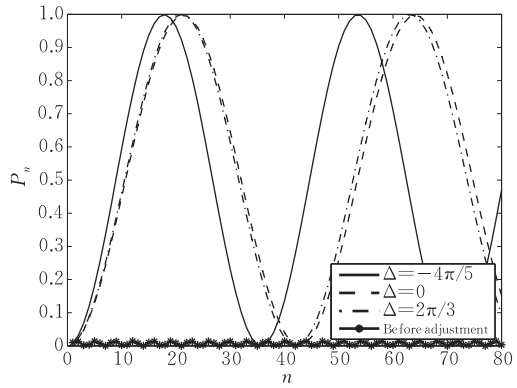


图 5　$j=50$ 时调整前和调整后成功概率 P_n 和叠代次数 n 关系曲线图($N=10^4$, $j=50$)

6　结　论

提出了二维复子空间中 Long 算法的多相位匹配问题，即假设初始态为

$|\gamma_0\rangle=\sqrt{N-1/N}|\beta\rangle+\sqrt{1/N}|\tau\rangle$($N$ 足够大)，在酉算子 $\tilde{G}_j=G(\phi_j,\theta_j)G(\phi_{j-1},\theta_{j-1})\cdots G(\phi_1,\theta_1)$ 作用下，在理论上存在着两个集合 $\{\theta_j,\theta_{j-1},\cdots,\theta_1\}$ 和 $\{\phi_j,\phi_{j-1},\cdots,\phi_1\}$，目标态 $|\tau\rangle$ 能以 100% 的最大成功概率找到. 同时给出了一个推测：如果这两个相位角集合满足多相位匹配方程 $\sum_{m=1}^{j}\theta_m=\sum_{m=1}^{j}\phi_m=\Delta$ 且整数 $j\geq 2$ 相对于 N 较小，则目标态 $|\tau\rangle$ 能以较高的成功概率找到. 尽管该多相位匹配方程的推导过程和文献[9]中推导单相位匹配条件 $\theta=\phi$ 的过程有些类似，但他们的基本思想出发点是不一致的. 一方面，多相位匹配方程是针对多相位匹配问题而给出的一个近似公式(对应于 $j\geq 2$ 的情形). 在文献[15]中证明了单相位匹配条件 $\theta=\phi$ 也是一个精确的相位公式；然而，随着 j 值的增大以及对于不同的 j 值($j\geq 2$)，要进一步证明多相位匹配方程是否也是个精确公式却是异常困难的. 另一方面，却给予了我们以下启示：当 N 不是太大且满足多相位匹配方程时，有可能出现每次叠代的相位旋转角 θ_m 和 ϕ_m 完全独立且所有的相位对 (θ_m,ϕ_m) 都不相同($1\leq m\leq j=k$，k 表示总的叠代次数)的情况下，Long 算法也依然有效.

致　谢　作者感谢审稿专家所提供的帮助和提出的宝贵意见及为此文付出劳动的编辑老师！

参 考 文 献

[1] Grover L K. Quantum mechanics helps in searching for a needle in a haystack. Physical Review Letters, 1997, 79(2): 325-328

[2] Bennett C H, Bernstein E, Brassard G et al. Strengths and weaknesses of quantum computing. SIAM Journal on Computing, 1997, 26(5): 1510-1523

[3] Zalka C. Grover's quantum searching algorithm is optimal. Physical Review A, 1997, 60(4): 2746-2751

[4] Boyer M, Brassard G, Hoyer P et al. Tight bounds on quantum searching. Fortschritte Der Physik-Progress of Physics, 1996, 46(4-5): 493-506

[5] Grover L K. Quantum computers can search rapidly by using almost any transformation. Physical Review Letters, 1998, 80(19): 4329-4332

[6] Brassard G, Hoyer P, Tapp A. Quantum counting//Proceedings of the 25th ICALP, Aalborg, Denmark. Lecture Notes in Computer Science 1443. Springer, 1998: 820-831

[7] Brassard G, Hoyer P, Mosca M et al. Quantum amplitude amplification and estimation. Quantum Computation and Quantum Information: AMS Contemporary Mathematics Series Millennium Volume, 2002, 305: 53-74

[8] Gingrich R, Williams C P, Cerf N. Generalized quantum search with parallelism. Physical Review A, 2000, 61(5): 1-8

[9] Long G L, Li Y S, Zhang W L et al. Phase matching in quantum searching. Physics Letters A, 1999, 262(1): 27-34

[10] Biham E, Biham O, Biron D et al. Grover's quantum search algorithm for an arbitrary initial amplitude distribution. Physical Review A, 1999, 60(4): 2742-2745

[11] Biham E, Biham O, Biron D et al. Analysis of generalized Grover quantum search algorithms using recursion equations. Physical Review A, 2000, 63(1): 2742-2749

[12] Carlini A, Hosoya A. Quantum computers and unstructured search: Finding and counting items with an arbitrarily entangled initial state. Physics Letters A, 2001, 280(3): 114-120

[13] Jin W L, Chen X D. A desired state can not be found with certainty for Grover's algorithm in a possible three-dimensional complex subspace. Quantum Information Processing, 2011, 10(3): 419-429

[14] Long G L, Zhang W L, Li Y S et al. Arbitrary phase rotation of the marked state can not be used for Grover's quantum search algorithm. Communications in Theoretical Physics, 1999, 32(3): 335-338

[15] Long G L, Xiao L, Sun Y. Phase matching condition for quantum search with a generalized quantum database. Physics Letters A, 2002, 294(3-4): 143-152

[16] Long G L. Grover algorithm with zero theoretical failure rate. Physical Review A, 2001, 64(2): 022307

[17] Hoyer P. Arbitrary phases in quantum amplitude amplification. Physical Review A, 2000, 62(5): 052304

[18] Long G L, Tu C C, Li Y S et al. An SO(3) picture for quantum searching. Journal of Physics A: Mathematical and General, 2001, 34(4): 861-866

[19] Long G L, Yan H Y, Li Y S et al. Experimental NMR realization of a generalized quantum search algorithm. Physics Letters A, 2001, 286(2): 121-126

[20] Bhattacharya N, van den Heuvell H B V, Spreeuw R J C. Implementation of quantum search algorithm using classical fourier optics. Physical Review Letters, 2002, 88(13): 137901

[21] Puentes G, Mela C L, Ledesma S et al. Optical simulation of quantum algorithms using programmable liquid-crystal displays. Physical Review A, 2004, 69(4): 042319

[22] Long G L, Liu Y. Search an unsorted database with quantum mechanics. Frontiers of Computer Science in China, 2007, 1(3): 247-271

[23] Li D F, Li X X. More general quantum search algorithm $Q=-I_7VI_\tau U$ and the precise formula for the amplitude and the non-symmetric effects of different rotating angles. Physics Letters A, 2001, 287(5-6): 304-316

[24] Jin W L. Quantum search in a possible three-dimensional complex subspace. Quantum Information Processing, 2012, 11(1): 41-54

[25] Tulsi A. Quantum computers can search rapidly by using almost any selective transformation. Physical Review A, 2008, 78(2): 022332

[26] Long G L, Li Y S, Zhang W L et al. Dominant gate imperfection in Grover's quantum search algorithm. Physical Review A, 2000, 61(4): 042305

[27] Joshi A W. Matrices and Tensors in Physics. New Delhi: New Age International (P) Ltd., Reprint, 2005, 69

JIN Wen-Liang, born in 1968, Ph. D.. His research interests include quantum algorithm, quantum information theory.

Background

Grover's search algorithm is one of the key algorithms in the field of quantum computing. It allows one to find a single desired state in a large unsorted database of size N using merely a number of queries in $O(N^{1/2})$, compared to the classical $O(N)$, and thus provides a quadratic speedup. There have been various improvements made to the original Grover's search algorithm over the past decade. To guarantee the effectiveness of the Long algorithm, which is one of the generalized Grover's algorithms, the phase matching condition (i. e. identical rotation angles) shall be observed by using an approximate method. Biham et al. also thought that in order for the Long algorithm to apply, the two rotation angles must be equal. Subsequently, Long et al. further showed that this phase condition is actually an exact relation. By now the Grover's search algorithm has been verified experimentally in NMR (nuclear magnetic resonance) system with a few qubits, and the experimental implementation of the Long algorithm has been successfully completed in optical device and in NMR system.

When the search space is extended to a possible three-dimensional complex subspace, we proved that no matter what the initial superposition may be, a desired state cannot be found with certainty, and demonstrated that if N is sufficiently large then corresponding to the case of identical rotation angles, the maximum success probability of finding a unique desired state is approximately equal to the square of the cosine function of a deflection angle. Furthermore, we also showed that in this case, as the difference between the two rotation angles increases, the Long algorithm fails to enhance the probability of measuring a unique desired state irrespective of whether or not a deflection angle is small.

Most of the previous work on quantum search focused primarily on the study of the single-phase matching. In the present paper we made a remarkable advance beyond this limit by introducing the notion of the multiphase matching, so as to make the Long algorithm more generally applicable. Our analysis showed that a unique desired state can be found with high probability of success provided the multiphase matching equation is satisfied for an equally-weighted initial superposition of all basis states in a two-dimensional complex subspace.

We have another purpose in this paper to reveal that when N is not too large, maybe the multiphase matching equation is able to provide the following particular situation: in the case that any two phase rotation angles corresponding to each iteration of Long operator are completely independent and all of the pairs of phases are different from one another, the Long algorithm can still be effective.

一种适合于频繁位置更新的网络受限移动对象轨迹索引

丁治明
(中国科学院软件研究所基础软件国家工程研究中心 北京 100190)

摘 要 移动对象索引是支持海量移动对象管理的一项关键技术.目前的移动对象时空轨迹索引方法如STR-Tree、TB-Tree、FNR-Tree、MON-Tree等均直接以轨迹单元作为基本的索引记录单位,在位置更新时需要频繁地在索引中插入新的记录,从而严重地影响了数据库的总体性能.为了解决上述问题,文中提出一种网络受限移动对象的动态概略化轨迹R树索引(DSTR-Tree).DSTR-Tree将索引空间划分成等距格栅,并通过格栅单元对每一条移动对象轨迹进行概略化,然后以概略化轨迹单元为基本索引记录单位建立R树索引.由于概略化轨迹的粒度大大粗于原始轨迹,因此移动对象不需要在每次位置更新的同时触发索引更新,而仅需要在轨迹跨越当前格栅单元时才进行索引更新,从而显著地降低了索引更新的代价.实验结果表明,DSTR-Tree在移动对象数据库频繁位置更新的实际运行条件下,提供了良好的索引维护及总体查询处理性能.

关键词 移动对象;数据库;时空轨迹;概略化;索引
中图法分类号 TP309 **DOI号**: 10.3724/SP.J.1016.2012.01448

An Index Structure for Frequently Updated Network-Constrained Moving Object Trajectories

DING Zhi-Ming
(*National Fundamental Software Research Center, Institute of Software, Chinese Academy of Sciences, Beijing* 100190)

Abstract Index is a key technology to improve the query processing performance of moving objects databases. However, current index methods for moving object trajectories, such as STR-Tree, TB-Tree, FNR-Tree, and MON-Tree, take trajectory units as the basic index records, and therefore, frequent insertions are needed when location updates occur in order to keep the consistency between the index and the trajectories in database, which greatly affects the overall performance of moving objects databases. To solve this problem, we propose a new index method, Dynamic Sketched-Trajectory R-Tree for Network-constrained Moving Objects (DSTR-Tree) in this paper. The DSTR-Tree divides the spatial-temporal space into equal-sized grid cells, transforms every trajectory into a sketched trajectory with each unit connecting two centers of the grid cells that the moving object travels through, and indices the sketched trajectory units as an R-Tree. Since the sketched trajectory has much coarser granularity than the original trajectory, the index updating cost can be greatly reduced — when a location update occurs, even though the original trajectory needs to be changed, the sketched trajectory may remain unchanged so that the DSTR-Tree does not need to be changed either. The experimental results show that the DSTR-Tree outperforms the previously proposed trajectory index methods in running moving objects databases with frequent location updates.

Keywords moving objects; database; spatial-temporal trajectory; sketched; index

收稿日期:2011-02-17;最终修改稿收到日期:2011-06-07.本课题得到国家自然科学基金项目(91124001,60970030)资助.**丁治明**,男,1966年生,博士,研究员,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为数据库与知识库系统、移动计算、时空数据库与数据挖掘、信息检索、物联网与云计算等. E-mail: zhiming@iscas.ac.cn.

1 引　言

近年来，移动对象数据库（Moving Objects Databases，MOD）成为了一个热点研究领域并得到了国内外研究人员的广泛关注. 移动对象数据库属于时空数据库（Spatio-Temporal Database）的范畴，是指对位置不断移动的物体或目标（如汽车、飞机、轮船、行人等）的动态位置及其它相关属性进行表示与管理的数据库[1]. 越来越多的应用要求对移动对象进行管理，而定位技术和无线通信技术的发展使得跟踪和记录移动对象的位置成为可能.

在典型的移动对象数据库系统中，通常存放着海量移动对象的时空数据. 例如，一个大中型城市的移动对象数目可以达到数百万甚至更多. 为了支持对这些移动对象过去及当前位置的查询，有效的索引手段是需要解决的关键问题.

在移动对象索引方面，人们已经进行了大量的工作，这些工作可以分为两大类：针对移动对象当前位置的索引和针对移动对象完整时空轨迹的索引. 移动对象当前位置索引的典型代表是 TPR-Tree[2]，其基本思想是在 R* 树索引的基础上，允许最小包容矩形（MBR）包含速度和方向等参数信息，从而使 MBR 能随时间参数进行变化，而不需要随着移动对象位置的变化频繁地修改索引记录. TPR-Tree 提出后，人们在此基础上进行了大量的改进工作，提出了许多 TPR-Tree 的变种树如 R^{EXP}-Tree[3]、TPR*-Tree[4]、Bulk-loading TPR-Tree[5]、HTPR-Tree[6]等. 此外，文献[7-9]还针对移动对象当前位置索引中由于频繁位置更新所导致的写代价过高问题进行了优化. 但是，所有上述索引均只能支持对移动对象当前时刻位置的查询，而不能支持对过去位置的查询.

移动对象完整时空轨迹的索引包含了移动对象过去及当前的位置信息，因此比移动对象当前位置索引具有更为广泛的用途. 这方面的研究工作又可以分为两大类：基于 Euclidean 空间的轨迹索引和基于交通网络的轨迹索引.

基于 Euclidean 空间的轨迹索引以 Euclidean 轨迹单元为索引记录的基本单位进行索引的组织[10-13]，其中每个 Euclidean 轨迹单元是 $X\times Y\times T$ 空间中的一个直线段，这些直线段可以组织成 R 树、Quad 树、Grid File 等形式，从而支持对移动对象的快速搜索. 上述方法的缺点在于 Euclidean 轨迹单元是直线，因此需要大量的轨迹单元来刻画复杂的移动对象轨迹曲线，效率较低.

为了进一步提高效率，人们越来越多地转向基于交通网络的移动对象轨迹索引，并提出了多种方法[14-17]. 基于交通网络的移动对象轨迹索引一般采用双层结构，其中上层是一个 2 维的 R 树，用于对固定的道路网络进行索引；下层是一系列的 R 树，每个下层 R 树与一条道路相对应，用于对移动对象在该条道路中提交的轨迹单元进行索引. 其中，每个轨迹单元对应于移动对象在曲线道路上的一段匀速行驶过程，可以刻画 $X\times Y\times T$ 空间中的一个曲线段. 与基于 Euclidean 空间的轨迹索引相比，基于交通网络的轨迹索引减少了轨迹单元的数目并降低了存储开销.

尽管在移动对象轨迹索引方面人们已经取得了一些重要的进展，但是目前的索引方法还存在着诸多缺陷：

（1）几乎所有的移动对象轨迹索引均以轨迹单元作为索引记录的基本单位，由于索引记录的粒度太细，每次当移动对象发生位置更新并产生新的轨迹单元时，均需要在索引中插入相应的索引记录，因此索引更新的频率等同于位置更新的频率，从而造成极大的索引更新开销；

（2）目前已经提出的基于网络的移动对象轨迹索引方法均采用了双层索引结构，这种结构缺乏通用性，仅适合于在专用系统中实现，很难在通用的可扩充数据库系统（如 PostgreSQL）中实现；

（3）基于网络的移动对象轨迹索引仅能处理移动对象与路网匹配的情况，不能表示移动对象与路网匹配不上（如移动对象在电子地图之外的小路上行驶）的情况，缺乏灵活性.

为了解决上述问题，本文提出一种新的移动对象轨迹索引方法：网络受限移动对象的动态概略化轨迹 R 树索引（Dynamic Sketched-Trajectory R-Tree for Network-constrained Moving Objects，简称 DSTR-Tree）. DSTR-Tree 将索引空间 $X\times Y\times T$ 划分成等距格栅，通过格栅单元对每一条时空轨迹进行概略化，并以概略化轨迹单元为基本索引记录单位建立 R 树索引. 这种方法实际上对移动对象时空轨迹进行了粒度粗化. 当发生位置更新时，如果新产生的轨迹单元没有跨越移动对象上次位置更新时所对应的格栅单元（该格栅单元称为移动对象的“活动格栅单元”），则不需要对索引进行任何修改；仅当新产生的轨迹单元跨越了活动格栅单元时才需要在

索引中插入新的记录,因此极大地降低了索引更新的代价,符合移动对象频繁位置更新的现实情况.此外,DSTR-Tree 采用了一种典型的单层树型结构,可以轻易地在通用数据库系统如 PostgreSQL 中实现.最后,DSTR-Tree 可以兼容移动对象在路网中和路网之外行驶的情况,具有充分的灵活性和实用性.

本文第 2 节给出网络受限移动对象通用时空轨迹数据模型;第 3 节描述 DSTR-Tree 的基本结构及相关算法;第 4 节对实验结果进行分析;第 5 节给出相关结论.

2 网络受限移动对象的通用时空轨迹模型

从抽象模型的角度来看,移动对象的时空轨迹对应于 $X\times Y\times T$ 空间中的一条曲线.在移动对象数据库中,需要对上述曲线进行离散化才能被计算机处理.本节讨论时空轨迹的离散化表示方法.

在移动对象轨迹的模型化方面,早期的工作采用的是直接基于 Euclidean 空间的表示方法[10,18],即通过 $X\times Y\times T$ 空间中的一系列直线线段来表示轨迹.为了刻画复杂的轨迹曲线,通常需要大量的直线线段,这意味着移动对象数据库需要更多的位置更新和更多的存储空间来生成和管理这些轨迹.近几年来,越来越多的工作转向了基于路网的轨迹表示方法[15,19-20],即通过一系列基于路网的轨迹单元来表示轨迹,每个基于路网的轨迹单元刻画移动对象在曲线道路上的一段匀速行使过程.由于基于路网的轨迹单元对应于 $X\times Y\times T$ 空间中的一段曲线,因此与基于 Euclidean 的表示方法相比,上述方法极大地降低了轨迹单元的数量和位置更新的代价.其缺点是无法表示移动对象与路网匹配不上的情况,如当移动对象在电子地图之外的小路或广场上行驶,或者当电子地图没有及时更新导致与实际道路不符时,均有可能出现移动对象与路网匹配不上的情况.

为了克服目前移动对象轨迹模型化方法的缺陷,增加轨迹表示的通用性和灵活性,我们在本节给出一种通用的移动对象轨迹模型,作为 DSTR-Tree 索引的基础.该模型在主要考虑移动对象路网受限运动的前提下,兼容与路网匹配不上的特殊情况.

为了讨论方便,我们假设移动对象在与路网匹配时采用文献[21]提出的基于路网的位置更新策略,即当移动对象跨越不同的道路时触发 IDTLU 位置更新;当移动对象的计算位置与 GPS 位置的差值达到规定的距离阈值 ξ_d 时触发 DTTLU 位置更新;当移动对象的行驶速度与上次位置更新时的速度差值达到规定的速度阈值 ξ_s 时触发 STTLU 位置更新;当移动对象与路网不能匹配时,采用基于固定时间间隔的位置更新策略(FTLU),即每隔规定的时间阈值 ξ_t 触发一次位置更新.此外,当移动对象的路网匹配状态发生改变(由匹配状态变成不匹配状态,或反之)时需要立刻进行位置更新.

定义 1(交通网络). 交通网络 N 定义为

$$N=(R,J),$$

其中,R 是道路的集合,J 是交叉路口的集合.

定义 2(道路). 道路 r 定义为如下形式:

$$r=(rid,geo,len,(jid_j,pos_j)_{j=1}^{m}),$$

其中,rid 是道路标识;geo 是该道路的地理几何形状;len 是道路的长度;$(jid_j,pos_j)_{j=1}^{m}$ 描述该条道路所包含的各个交叉路口(见定义 3)的标识以及它们在道路中的相对位置(设每条道路的总长度为 1,则该条道路中的任一相对位置 pos 可以用[0,1]之间的一个实数表示).

在上述定义中,道路的几何形状 geo 用一条折线 pl 表示,pl 是由 $X\times Y$ 平面中的一组点所组成的序列,即

$$pl=(x_i,y_i)_{i=1}^{n}(n\geq 2),$$

其中,(x_1,y_1) 为 pl 的起点,称为 r 的"0-端点",(x_n,y_n) 为 pl 的终点,称为 r 的"1-端点".

定义 3(交叉路口). 交叉路口 j 与实际交通网络中的交叉路口、道路出入口或道路端点相对应,定义为如下形式:

$$j=(jid,loc,((rid_i,pos_i))_{i=1}^{n},m),$$

其中 jid 是交叉路口的标识;loc 是 j 的地理位置,用其(x,y)坐标表示;$(rid_i,pos_i)(1\leq i\leq n)$描述 j 所连接的第 i 条道路,其中 rid_i 是道路标识,$pos_i\in[0,1]$是 j 在该条道路中的相对位置;m 是 j 的连接矩阵,用以描述该交叉路口的各交通流之间的连通关系[19].

定义 4(网络位置). 交通网络 N 中的任意一个网络位置 $npos$ 可以包括两种情况:① 位于某个交叉路口,此时可以直接以交叉路口的标识 jid 表示;② 位于道路中,此时可以用其所在的道路标识 rid 及在该道路中的相对位置 pos 表示.因此网络位置 $npos$ 定义为

$$npos=\begin{cases}jid, & npos\text{ 位于交叉路口}\\ (rid,pos), & npos\text{ 位于道路中}\end{cases}.$$

定义 5(移动对象的运行矢量). 移动对象 mo 在 t 时刻的运行矢量 $\boldsymbol{mv}$ 定义为

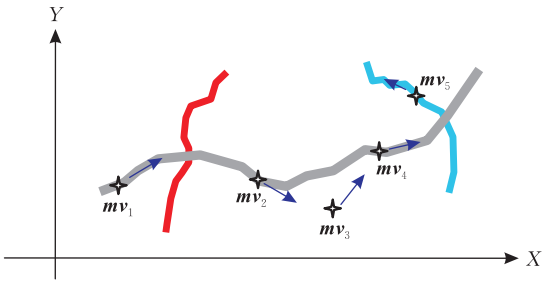
$$\boldsymbol{mv}=(t,(x,y),v,d,npos),$$

其中,t 是采集该运行矢量的时刻;(x,y)、v、d 分别是移动对象在 t 时刻的位置、速度、方向;$npos$ 是 $\boldsymbol{mv}$ 对应的网络位置(当移动对象的位置与道路网络匹配不上时,$npos$ 取空值). 在 $\boldsymbol{mv}$ 的各个参数中,t,(x,y),v,d 是由 GPS 采样得到的,而 $npos$ 是通过路网匹配得到的. 如果 $npos$ 为空,则 $\boldsymbol{mv}$ 称为 Eulidean 运行矢量;如果 $npos$ 有具体的网络位置匹配值,则 $\boldsymbol{mv}$ 称为路网匹配的运行矢量.

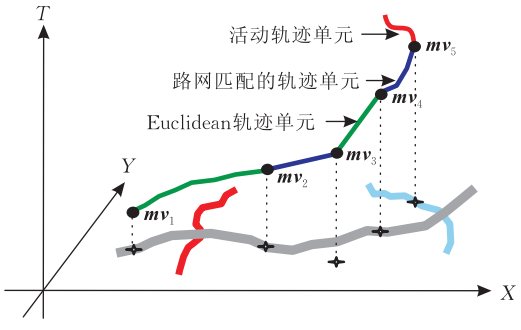
定义 6(移动对象的时空轨迹). 移动对象 mo 的时空轨迹 $traj$ 是 mo 在行驶过程中通过位置更新操作所生成的一组运行矢量的序列,用以描述 mo 的位置随着时间变化的过程,定义为如下形式

$$traj=(\boldsymbol{mv}_i)_{i=1}^{n}=((t_i,(x_i,y_i),v_i,d_i,npos_i))_{i=1}^{n}.$$

一条时空轨迹可以看成是若干个轨迹单元(见定义 7)所组成的序列,对应于 $X\times Y\times T$ 空间中的一条曲线. 图 1 给出了一个移动对象在路网中的行驶过程及对应得时空轨迹曲线.



(a) 移动对象在路网中的行驶过程



(b) 对应的移动对象时空轨迹曲线

图 1　移动对象的行驶过程以及对应的时空轨迹

移动对象的轨迹是通过位置更新操作得到的. 目前已经提出了多种不同的位置更新方法[21-24]. 在位置更新中,移动对象原始提交的参数一般包含运行矢量中的 t、(x,y)、v、d,而 $npos$ 的匹配可以在移动对象端或者在服务器端完成. 移动对象在行驶的过程中不断采集新的运行矢量并将之发送给服务器,因此服务器上保存的轨迹是不断增长的. 每隔一段时间(如两个星期),数据库需要对轨迹数据进行转储,并生成新的移动对象关系表. 一个理想的位置更新方法需要同时保证如下两个条件:

(1) 所生成的时空轨迹应精确地刻画移动对象实际的当前位置及历史行驶过程;

(2) 所生成的时空轨迹应包含尽可能少的运行矢量(或轨迹单元),并在其生成过程中采用尽可能少的位置更新操作.

定义 7(移动对象的轨迹单元). 时空轨迹 $traj=(\boldsymbol{mv}_i)_{i=1}^{n}=((t_i,(x_i,y_i),v_i,d_i,npos_i))_{i=1}^{n}$ 中的任意两个相邻的运行矢量 $\boldsymbol{mv}_i$ 和 $\boldsymbol{mv}_{i+1}$ $(1\leq i<n)$ 构成一个轨迹单元,记为 $\mu(\boldsymbol{mv}_i,\boldsymbol{mv}_{i+1})$. 此外,$traj$ 的最后一个运行矢量 $\boldsymbol{mv}_n$ 也对应着一个轨迹单元,记为 $\mu(\boldsymbol{mv}_n)$(我们称 $\mu(\boldsymbol{mv}_n)$ 为该移动对象的活动轨迹单元).

根据不同的路网匹配情况,轨迹单元 $\mu(\boldsymbol{mv}_i,\boldsymbol{mv}_{i+1})$ $(1\leq i<n)$ 可以被解析成不同的几何形态:

(1) 如果 $\boldsymbol{mv}_i$ 和 $\boldsymbol{mv}_{i+1}$ 均为路网匹配的运行矢量,则 $\mu(\boldsymbol{mv}_i,\boldsymbol{mv}_{i+1})$ 称为路网匹配的轨迹单元,对应于 $X\times Y\times T$ 空间中的一条曲线线段,该曲线线段反映移动对象沿着路网从 $\boldsymbol{mv}_i$ 到 $\boldsymbol{mv}_{i+1}$ 匀速行驶的时空过程(如图 1(b)中的 $\mu(\boldsymbol{mv}_1,\boldsymbol{mv}_2)$、$\mu(\boldsymbol{mv}_4,\boldsymbol{mv}_5)$);

(2) 如果 $\boldsymbol{mv}_i$ 和 $\boldsymbol{mv}_{i+1}$ 之一或二者皆为 Eulidean 运行矢量,则 $\mu(\boldsymbol{mv}_i,\boldsymbol{mv}_{i+1})$ 称为 Euclidean 轨迹单元,对应于 $X\times Y\times T$ 空间中的一条直线线段,该直线线段反映移动对象从 $\boldsymbol{mv}_i$ 到 $\boldsymbol{mv}_{i+1}$ 匀速直线行驶的时空过程(如图 1(b)中的 $\mu(\boldsymbol{mv}_2,\boldsymbol{mv}_3)$、$\mu(\boldsymbol{mv}_3,\boldsymbol{mv}_4)$).

对于活动轨迹单元 $\mu(\boldsymbol{mv}_n)$,其几何形状可以通过如下方式求得(设每个移动对象有一个活动状态标记 $activeFlag$,表示其是否处于联机行驶状态):

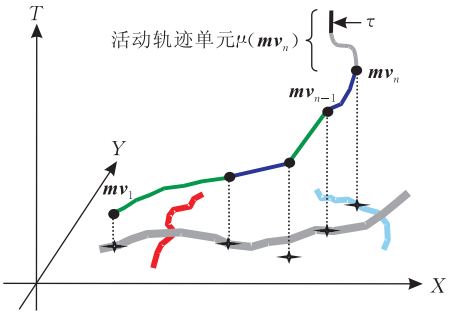
(1) 如果 $activeFlag=$false,则移动对象处于离线关闭状态,此时 $\mu(\boldsymbol{mv}_n)$ 为空值;

(2) 如果 $activeFlag=$true,则移动对象处于活动行驶状态,此时可以进一步细分为两种情况:

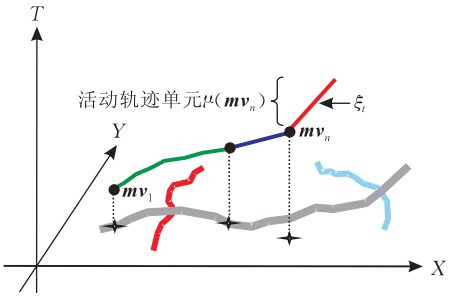
(2.1) 如果 $\boldsymbol{mv}_n$ 是路网匹配的运行矢量,则 $\mu(\boldsymbol{mv}_n)$ 对应于 $X\times Y\times T$ 空间中的一条曲线线段,该曲线线段反映移动对象沿当前道路按照 $\boldsymbol{mv}_n$ 中描述的速度、方向匀速行驶至道路终点(见图 2(a)中 $\mu(\boldsymbol{mv}_n)$ 的灰色曲线部分),并继续停留在道路终点 τ

时间(见图 2(a)中 $\mu(\boldsymbol{mv}_n)$的黑色垂直线部分. τ 的计算方法将在下面进一步分析)的行驶过程. 上述 $\mu(\boldsymbol{mv}_n)$的几何曲线反映的是移动对象在 $\boldsymbol{mv}_n$ 之后的计算位置. 根据文献[21-22]的分析,移动对象的计算位置在按照 $\boldsymbol{mv}_n$ 中的行驶参数到达当前道路终点之后,将继续停留在道路终点.

(2.2)如果 $\boldsymbol{mv}_n$ 是Euclidean运行矢量,则 $\mu(\boldsymbol{mv}_n)$ 对应于 $X\times Y\times T$ 空间中的一条直线线段,该直线线段反映移动对象从 $\boldsymbol{mv}_n$ 开始,按照 $\boldsymbol{mv}_n$ 中的速度、方向等运行参数匀速直线行驶 ξ_t 时间的行驶过程(见图 2(b)).



(a) $\boldsymbol{mv}_n$为路网匹配的运行矢量



(b) $\boldsymbol{mv}_n$为Euclidean运行矢量

图 2 活动轨迹单元 $\mu(\boldsymbol{mv}_n)$对应的几何曲线

通过分析可知,$\mu(\boldsymbol{mv}_n)$所对应的上述几何曲线的时间跨度能够保证下一次位置更新之前的所有计算位置都包含在该曲线中了. 如果 $\boldsymbol{mv}_n$ 为路网匹配的运行矢量(见上述(2.1)),则 $\mu(\boldsymbol{mv}_n)$几何曲线的时间跨度为按照最慢可能的速度到达道路终点所需要的时间;如果 $\boldsymbol{mv}_n$ 为 Euclidean 运行矢量(见(2.2)),则 $\mu(\boldsymbol{mv}_n)$几何曲线的时间跨度为 FTLU 位置更新的时间间隔. 因此可以保证,在上述时间跨度用完之前,必然会有新的位置更新产生.

下面讨论上述(2.1)中 τ 的计算方法. 如图 2(a)所示,活动运行矢量 $\boldsymbol{mv}_n$ 为路网匹配的运行矢量. 设 $\boldsymbol{mv}_n=(t_n,(x_n,y_n),v_n,d_n,npos_n)$,且 $npos_n=(rid_n,pos_n)$. 根据 $\boldsymbol{mv}_n$ 中的参数,可以计算出移动对象在 t_n 时刻离道路终点的距离为 $(1-pos_n)\times length(rid_n)$,其中 $length(rid_n)$ 为道路 rid_n 的长度. 移动对象按照 v_n 的速度行驶至道路终点所需要的时间为

$$t_{\text{norm}}=((1-pos_n)\times length(rid_n))/v_n,$$

而移动对象在不触发 STTLU 的情况下的最慢可能速度为 $(v_n-\xi_s)$,因此在不触发位置更新的前提下最晚到达道路终点的时间为

$$t_{\text{slow}}=((1-pos_n)\times length(rid_n))/(v_n-\xi_s),$$

所以有

$$\tau=t_{\text{slow}}-t_{\text{norm}}.$$

3 DSTR-Tree 索引的结构及相关算法

本节首先讨论 DSTR-Tree 的基本结构,然后给出 DSTR-Tree 索引的初始建立、动态维护及查询处理算法.

3.1 移动对象轨迹的概略化及 DSTR-Tree 索引的结构

为了对轨迹进行概略化,首先需要对 $X\times Y\times T$ 空间进行格栅化. 设移动对象数据库的应用时空空间为 $I_x\times I_y\times I_t$,其中,$I_x=[x_0,x_1]$,$I_y=[y_0,y_1]$,$I_t=[t_0,\perp]$(I_t的终点为$\perp$(未定义),因为当前时刻是不断增长的). 在进行格栅化时,可以将 I_x 划分成 n 个等大小的子区域,每个子区域的大小为

$$\zeta_x=\frac{x_1-x_0}{n}.$$

同理可以将 I_y 划分成 m 个等大小的子区域,每个子区域的大小为

$$\zeta_y=\frac{y_1-y_0}{m}.$$

对于 I_t,由于其终点未定义,可以将之划分成大小为 Δt 的等长时间段,即

$$\zeta_t=\Delta t.$$

通过上述方法,$I_x\times I_y\times I_t$ 空间被划分成了等距格栅. 格栅单元用其编号(N_x,N_y,N_t)来表示,其中 N_x,N_y,N_t 分别为该单元在 X,Y,T 轴所对应的子区域编号,例如图 3 中右上角标记灰色的单元编号为(4,3,2).

将 $I_x\times I_y\times I_t$ 空间进行格栅化之后,接下来需要对轨迹进行概略化,从而形成概略化的轨迹(Sketched Trajectory). 为了方便叙述,称进行概略化之前的轨迹为"原始轨迹".

定义 8(移动对象的概略化轨迹). 设 $traj=((t_i,(x_i,y_i),v_i,d_i,npos_i))_{i=1}^{n}$ 是移动对象的轨迹,其概略化轨迹 $sketch(traj)$ 是将 $traj$ 轨迹曲线所穿

行的格栅单元的中心点通过直线线段连接起来所形成的轨迹，定义为如下形式：

$$sketch(traj)=(c_j)_{j=1}^{k}=((t_j,x_j,y_j))_{j=1}^{k},$$

其中 $c_j=(t_j,x_j,y_j)(1\leq j\leq k)$ 是 $traj$ 所穿行的第 j 个格栅单元的中心点的坐标. $sketch(traj)$ 中两个相邻的中心点 c_j 和 c_{j+1} 构成一个概略化轨迹单元(Sketched-Trajectory Unit)，记为 $\hat{\mu}(c_j,c_{j+1})$，它对应于 $X\times Y\times T$ 空间中的一条连接 c_j 和 c_{j+1} 的直线线段. 注意，概略化轨迹 $sketch(traj)$ 中已经包含了 $traj$ 的活动轨迹单元 $\mu(\boldsymbol{mv}_n)$ 的概略化信息. 图 3 给出了对 $I_x\times I_y\times I_t$ 空间进行格栅化及对原始轨迹进行概略化的例子.

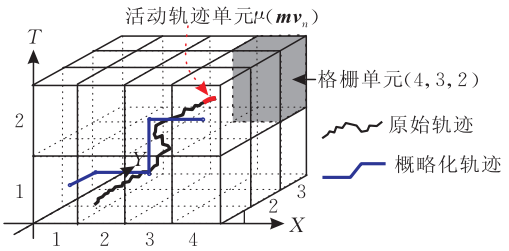


图 3　时空应用区域 $I_x\times I_y\times I_t$ 的格栅化及原始轨迹的概略化

如图 3 所示，概略化轨迹用大为减少的线段近似表示了原始轨迹的几何形状. 同一条原始时空轨迹所对应的概略化轨迹的单元数目取决于格栅单元的大小：格栅单元越大，则概略化轨迹的单元数目越少，反之亦然.

算法 1 描述了根据移动对象的原始时空轨迹生成概略化轨迹的过程. 其中，函数 $getCellLocated(\boldsymbol{mv})$ 计算运行矢量 $\boldsymbol{mv}$ 所位于的格栅单元，$getCellsTravelled(\mu)$ 计算轨迹单元 μ 所穿行的格栅单元序列，$extractCell(cellseq,i)$ 从格栅单元序列 $cellseq$ 中提取第 i 个格栅单元，$getCenter(cell)$ 计算格栅单元 $cell$ 的中心点坐标，$|cellseq|$ 返回格栅单元序列 $cellseq$ 中所包含的格栅单元数目，apppend($sketchTraj$, $center$) 将中心点坐标 $center$ 添加到 $sketchTraj$ 的最后，$doNothing()$ 不做任何事情而直接返回到主程序.

算法 1.　移动对象概略化轨迹生成算法.

全局变量：$I_x\times I_y\times I_t$　//应用空间区域

　　ζ_x,ζ_y,ζ_t　//格栅单元的划分粒度

输入：$traj=(\boldsymbol{mv}_i)_{i=1}^{n}=((t_i,(x_i,y_i),v_i,d_i,npos_i))_{i=1}^{n}$

　　$activeflag$　//移动对象的活动标记

输出：$sketchTraj=((t_j,x_j,y_j))_{j=1}^{k}$

```
1.  sketchTraj=NULL;
2.  startingCell=getCellLocated(mv_1);
3.  apppend(sketchTraj,getCenter(startingCell));
4.  IF n=1 THEN
5.   IF activeflag=FALSE THEN
6.    Return(sketchTraj);
7.   ELSE
8.    cellsTravelled=getCellsTravelled(μ(mv_1));
9.    IF (|cellsTravelled|=1) AND
        (extractCell(cellsTravelled,1)=
         startingCell) THEN
10.    Return(sketchTraj);
11.   ELSE
12.    FOR j=2 to |cellsTravelled| DO
13.     apppend(sketchTraj,getCenter
              (extractCell(cellsTravelled,j)));
14.    ENDFOR
15.   ENDIF
16.  ENDIF
17. ELSE  //n>1
18.  currentCell=startingCell;
19.  FOR i=1 to n-1 DO
     //处理非活动轨迹单元 μ(mv_i,mv_{i+1})(1≤i≤n-1)
20.   cellsTravelled=getCellsTravelled(μ(mv_i,mv_{i+1}));
21.   IF (|cellsTravelled|=1) AND
        (extractCell(cellsTravelled,1)=currentCell)
        THEN
22.    doNothing();
23.   ELSE //μ(mv_i,mv_{i+1})穿越了多个格栅单元
24.    FOR j=2 to |CellsTravelled| DO
25.     apppend(sketchTraj,getCenter
              (extractCell(cellsTravelled,j)));
26.    ENDFOR;
27.    currentCell=extactCell(cellsTravelled,
                              |cellsTravelled|);
28.   ENDIF
29.  ENDFOR
     //处理活动轨迹单元 μ(mv_n)
30.  IF activeflag=TRUE THEN
31.   cellsTravelled=getCellsTravelled(μ(mv_n));
32.   IF (|cellsTravelled|=1) AND
        (extractCell(cellsTravelled,1)=currentCell)
        THEN
33.    doNothing();
34.   ELSE
35.    FOR j=2 to |CellsTravelled| DO
36.     apppend(sketchTraj,getCenter
              (extractCell(cellsTravelled,j)));
37.    ENDFOR
38.   ENDIF
39.  ENDIF
40.  Return(sketchTraj);
41. ENDIF
```

在算法 1 中，函数 $getCellsTravelled(\mu)$ 的返回值可以包含一个或多个格栅单元，这些格栅单元的中心点被依次添加到 $sketchTraj$ 中.

通过算法 1 可以看出，当移动对象的原始轨迹单元不跨越格栅单元时，可以直接跳过该单元而不需要做任何处理；仅当原始轨迹单元跨越一个或多个格栅单元时，才需要生成相应的概略化轨迹单元. 在特殊情况下（如移动对象沿某条道路长时间匀速行驶时），原始轨迹单元可能跨越多个格栅单元，此时需要将这些格栅单元的中心连线依次加入到概略化轨迹中（见图 4(b)中的情况(3)）. 由于算法对活动轨迹单元也进行了相同的处理（见算法 1 第 30～39 行，其中活动轨迹单元 $\mu(\boldsymbol{mv}_n)$ 是根据移动对象最后一次位置更新时提交的行驶参数进行预测并实体化所得到的单元，如图 2 所示），从而使得移动对象即使长时间不进行位置更新时，DSTR-Tree 中也预先包含了相应的概略化轨迹单元. 这些预先计算的、与 $\mu(\boldsymbol{mv}_n)$ 相对应的概略化轨迹单元需要在下一次位置更新时进行调整和重新预测（见算法 3）.

图 4 分析了在处理一个新的轨迹单元 μ 时会出现的 3 种典型情况，其中标记灰色的单元是 $currentCell$ 所对应的格栅单元. 在 3 种典型情况中，(1)和(2)较为常见，而(3)仅在偶然的情况下才会出现，这是因为格栅单元的粒度通常远大于轨迹单元的粒度，所以不会经常出现 μ 跨越 3 个或更多格栅单元的情况.

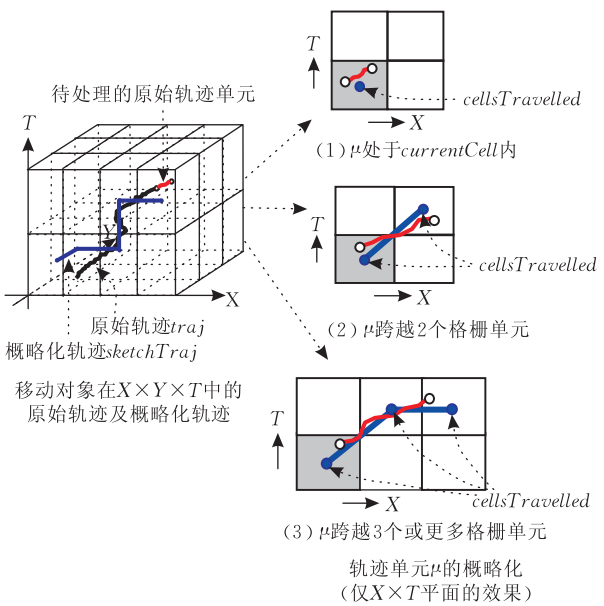


图 4 对原始轨迹单元 μ 的概略化处理

通过图 4 可以看出，在 DSTR-Tree 中，选取合适的格栅单元大小是至关重要的. 如果格栅单元过小，则会频繁地出现一个原始轨迹单元跨越多个格栅单元的情况，从而使得格栅化得不偿失；反之，如果格栅单元过大，又会导致索引块中无用记录增多，并进一步导致元组求精时间的增加. 为了获得良好的总体性能，在确定格栅单元大小时，可以参照原始轨迹单元的大小按比例放大，使得一个格栅单元能够平均容纳 φ 个原始轨迹单元.

根据具体的系统情况，φ 可以取不同的值（φ 还可以取小数，如 1.5），见本文的实验部分. 例如，设系统中移动对象的平均位置更新时间间隔为 30 s，平均速度为 60 km/h，30 s 对应的行驶距离为 500 m，所以原始轨迹单元的平均大小为 500 m×500 m×30 s. 如果取 $\varphi=5$，则格栅单元的大小可以确定为 2500 m×2500 m×150 s.

在完成 $I_x\times I_y\times I_t$ 空间的格栅化和原始时空轨迹的概略化之后，以每个概略化轨迹的各个轨迹单元为基本索引记录单位建立 R 树索引，即可得到 DSTR-Tree. 图 5 给出了 DSTR-Tree 的结构.

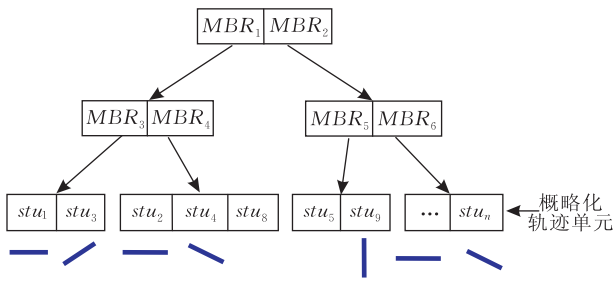


图 5 DSTR-Tree 索引的结构

在图 5 中，DSTR-Tree 叶子结点中的记录项结构为〈MBR，PT_{mo}，stu〉，其中 stu 为概略化轨迹单元，MBR 为该概略化轨迹单元的最小包容矩形，PT_{mo} 为指向实际移动对象记录的指针或记录标识. 中间结点的记录项为〈MBR，PT_{node}〉，其中 MBR 包含下层结点所有记录的最小包容矩形，PT_{node} 是指向下层结点的指针.

3.2 DSTR-Tree 索引的初始建立算法

在移动对象数据库中初始建立 DSTR-Tree 索引时，需要对数据库中的每条时空轨迹进行概略化，并将概略化轨迹中的每个轨迹单元逐一插入到 DSTR-Tree 中. 算法 2 给出了 DSTR-Tree 的初始建立算法. 在 DSTR-Tree 对数据库中的轨迹数据进行处理的同时，移动对象数据库还会继续接收到新的位置更新消息. 这些新位置更新消息被暂时缓存在 $buffer$ 中，等数据库中的轨迹数据被处理完毕之后，再调用动态维护算法（算法 3）处理这些新收到

的位置更新消息.

算法 2.　DSTR-Tree 的初始建立算法.

输入：*MObjs*　　//移动对象的集合

输出：*dstrTree*　　//DSTR-Tree

1. *dstrTtree*=NULL;
2. *buffer*=∅;
3. asynExec(*receiveLUMsg*(*buffer*));
   //启动另一线程接受新的位置更新并存入 *buffer*;
4. FOR EACH *mo*∈*MObjs* DO
5. *sketchTraj*=*skecth*(*mo.traj*,*mo.actviveflag*);
   //调用算法 1 计算概略化轨迹;
6. FOR EACH *skecthTrajUnit*∈*getUnits*(*sketchTraj*) DO
7. insert(*dstrTree*,*indexRec*(*mo.id*, *skecthTrajUnit*));
8. ENDFOR
9. ENDFOR
10. WHILE *buffer*≠∅ DO
11. *lumsg*=*getLUMsg*(*buffer*);
12. IF notIndexed(*lumsg*) THEN
13. 调用算法 3 根据 *lumsg* 对 *dstrTree* 进行维护;
14. ENDIF
15. ENDWHILE
16. Return(*dstrTree*).

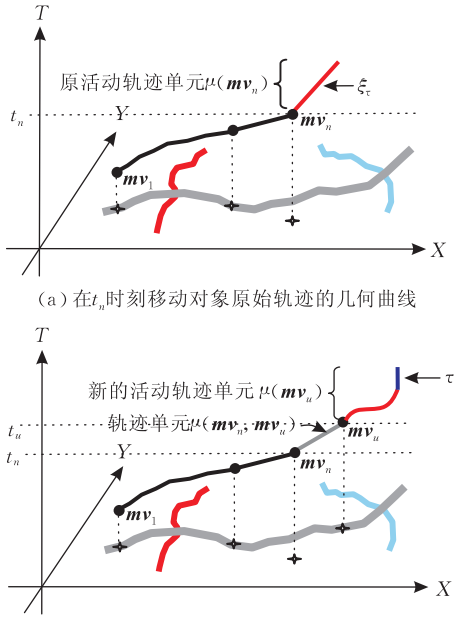
在算法 2 中，函数 asynExec(*func*)通过启动另一线程异步地执行函数 *func*，*receiveLUMsg*(*buffer*)接收位置更新消息并存入到 *buffer* 中，*getUnits*(*sketchTraj*)返回 *sketchTraj* 中的所有轨迹单元所组成的集合，*indexRec*(*moid*,*stu*)根据移动对象标识 *moid* 和概略化轨迹单元 *stu* 生成相应的索引记录，*getLUMsg*(*buffer*)从 *buffer* 中取出(同时从 *buffer* 中删除)一个位置更新消息，notIndexed(*lumsg*)判断 *lumsg* 是否在相应的移动对象被处理之前已经写入其原始轨迹中了，这种情况下 *lumsg* 的信息已经包含在 *dstrTree* 中了，不需要再进行处理.

需要注意的是，概略化轨迹及概略化轨迹单元仅在建立索引和维护索引的过程中使用，并不需要永久地保存在数据库中.

3.3　DSTR-Tree 索引在位置更新时的动态维护算法

在 DSTR-Tree 索引初始建立之后，当发生位置更新时，不仅需要在数据库中将新的运行矢量附加到轨迹中，而且还需要对 DSTR-Tree 进行动态维护. 由于移动对象在行驶过程中不断地通过位置更新操作向数据库服务器发送新的运行矢量，因此其时空轨迹是随着新的运行矢量的加入而不断增长的. 当原始轨迹的增长导致概略化轨迹的变化时，就需要同时对 DSTR-Tree 索引进行相应的修改.

更具体地说，对于一个活动移动对象 *mo*，设其原始时空轨迹为 $traj=(\boldsymbol{mv}_i)_{i=1}^{n}=((t_i,(x_i,y_i),v_i,d_i,npos_i))_{i=1}^{n}$，对应的概略化轨迹为 $sketch(traj)=((t_j,x_j,y_j))_{j=1}^{k}$. 当服务器新接收到 *mo* 发送来的运行矢量 $\boldsymbol{mv}_u$ 时，需要将 $\boldsymbol{mv}_u$ 附加到 *traj* 中. 上述过程对应着原始轨迹的几何曲线的变化(见图 6)，部分情况下原始轨迹的变化也导致概略化轨迹的变化，此时需要对 DSTR-Tree 进行相应的修改. 但由于概略化轨迹单元的粒度远大于原始轨迹移动单元的粒度，大部分情况下 $\boldsymbol{mv}_u$ 加入到 *traj* 中不会对 *sketch*(*traj*)造成任何改变，此时就不需要对索引进行任何修改.



(a) 在 t_n 时刻移动对象原始轨迹的几何曲线

(b) 在 t_u 时刻移动对象原始轨迹的几何曲线

图 6　在发生位置更新时移动对象原始轨迹几何曲线的改变

数据库服务器在接收到一个位置更新消息时对 DSTR-Tree 索引进行维护的过程如算法 3 所示. 在该算法中，函数 *getTrajectory*(*moid*)根据移动对象的标识在数据库中检索指定移动对象的轨迹(为了加快检索速度，可以基于轨迹数据建立以 *moid* 为关键字的 B^+ 树索引)，*final*(*traj*)提取轨迹 *traj* 的最后一个运行矢量，*getCellsTravelledExt*($\mu_1,\mu_2,\cdots$)计算轨迹单元序列 $\mu_1,\mu_2,\cdots$ 所穿行的格栅单元序列，*getSTUnits*(*cellseq*)返回格栅单元序列 *cellseq* 对应的所有轨迹单元所组成的集合.

在算法 3 中，$cellsTravelled_{\text{old}}$ 是原活动轨迹单元 $\mu(\boldsymbol{mv}_n)$ 所穿过的格栅单元的序列，如〈*cell*3,

$cell4\rangle.cellsTravelled_{new}$是 $\mu(\boldsymbol{mv}_n,\boldsymbol{mv}_u)$和 $\mu(\boldsymbol{mv}_u)$所穿行的格栅单元序列，如$\langle cell3,cell5,cell4\rangle$如果二者不相等，则需要对 DSTR-Tree 进行调整，通过插入 *unitsInsert*，并删除 *unitsDelete*，可以将 DSTR-Tree 调整为与新概略化轨迹相对应的状态.

算法 3. 位置更新时 DSTR-Tree 索引的维护算法.

输入：$LUMsg=(moid,t,(x,y),v,d,npos)$；
//收到的位置更新消息
dstrTree //DSTR-Tree 索引

1. $\boldsymbol{mv}_u=(t,(x,y),v,d,npos)$；
2. $\boldsymbol{mv}_n=final(getTrajectory(moid))$；
3. $cellsTravelled_{old}=getCellsTravelled(\mu(\boldsymbol{mv}_n))$；
4. $cellsTravelled_{new}=getCellsTravelledExt(\mu(\boldsymbol{mv}_n,\boldsymbol{mv}_u),\mu(\boldsymbol{mv}_u))$；
5. IF $cellsTravelled_{old}=cellsTravelled_{new}$ THEN
//概略化轨迹没有发生变化
6. doNothing()；
7. ELSE //概略化轨迹发生了变化
8. $unitsInsert=getSTUnits(cellsTravelled_{new})-getSTUnits(cellsTravelled_{old})$；
9. $unitsDelete=getSTUnits(cellsTravelled_{old})-getSTUnits(cellsTravelled_{new})$；
10. FOR $skecthUnit\in unitsDelete$ DO
11. delete($dstrTree,indexRec(moid,skecthUnit)$)；
12. ENDFOR
13. FOR $skecthUnit\in unitsInsert$ DO
14. insert($dstrTree,indexRec(moid,skecthUnit)$)；
15. ENDFOR
16. ENDIF

在移动对象数据库中，每次收到一个新的位置更新消息 *LUMsg* 时，均需要将 *LUMsg* 插入到相应移动对象的轨迹中，同时调用算法 3 对索引进行维护. 由于概略化轨迹的粒度远大于原始轨迹的粒度，大部分情况下算法 3 并不需要对索引作任何修改（见第 5～6 行），从而有效地降低了索引更新的代价.

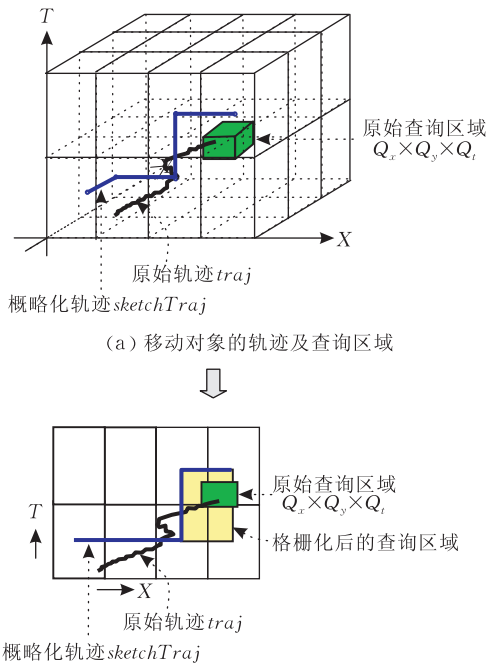
3.4 基于 DSTR-Tree 的移动对象查询处理算法

本节分析基于 DSTR-Tree 的移动对象查询处理方法. 设有任一查询 Q，其查询区域为

$$\text{range}(Q)=Q_x\times Q_y\times Q_t,$$

其中 $Q_x=[q_x^0,q_x^1]$，$Q_y=[q_y^0,q_y^1]$，$Q_t=[q_t^0,q_t^1]$.

在处理上述查询时，需要首先对查询区域 $range(Q)$进行格栅化，即要将查询区域修正为以格栅单元的中心作为区域的起止点，如图 7 所示（为了方便叙述，称 $range(Q)$为原始查询区域）.



（a）移动对象的轨迹及查询区域

（b）对查询区域的格栅化（仅$X\times T$平面的效果）

图 7 对查询区域的格栅化

在图 7 中，移动对象的原始轨迹和原始查询区域 $range(Q)$相交，因此该移动对象理应作为索引的查询结果输出. 但是，由于在 DSTR-Tree 中存放的是概略化轨迹，且该概略化轨迹与查询区域 $range(Q)$并不相交，因此如果不对查询区域进行格栅化，则会漏检该移动对象. 通过分析可知，由于所有经过相应格栅单元的原始轨迹都被映射到格栅单元的中心了，因此只要查询区域包含了格栅单元的中心即不会产生漏检. 对查询区域格栅化的目标即是保证所有其原始轨迹与原始查询区域相交的移动对象，必然有其概略轨迹与格栅化查询区域相交.

下面以 $Q_x=[q_x^0,q_x^1]$的变换为例，分析查询区域格栅化的过程. 对 q_x^0和 q_x^1分别进行如下变换，可以将其起始点分别对应到相应格栅单元的中心：

$$q_x^0=x_0+\left(\left\lfloor\frac{q_x^0-x_0}{\xi_x}\right\rfloor+0.5\right)\times\xi_x,$$

$$q_x^1=x_0+\left(\left\lfloor\frac{q_x^1-x_0}{\xi_x}\right\rfloor+0.5\right)\times\xi_x.$$

类似地，可以对 $Q_y=[q_y^0,q_y^1]$，$Q_t=[q_t^0,q_t^1]$进行相应的变换，从而得到格栅化之后的查询区域.

注意，格栅化之后的查询区域尽管在形式上表示为一个矩形区域，但由于区域的起始点可以相等，因此实际上可以对应于 $X\times Y\times T$ 空间中的一条直线或者一个点.

在完成上述变换之后，可以根据格栅化后的查询区域对 DSTR-Tree 进行查询，得到一组候选移动对象的集合(见算法 4 中的 *filterResult*). 然后系统对 *filterResult* 中的移动对象按照查询条件逐个进行计算，并输出满足查询条件的移动对象作为最终查询结果.

基于 DSTR-Tree 的查询处理算法如算法 4 所示. 在该算法中，函数 *gridCenterAlign*(*range*)对查询区域 *range* 进行格栅化，*evaluate*(*moTuple*, *Q*)根据查询 *Q* 对元组 *moTuple* 进行计算，如果元组不满足查询条件则返回 NULL；否则返回查询计算的结果.

算法 4.　基于 DSTR-Tree 的移动对象查询处理算法.

输入：查询 *Q*，其查询区域为 $range(Q)=Q_x\times Q_y\times Q_t$；*dstrTree*

输出：*refineResult*　//*Q* 的查询处理结果

1. $filterResult \leftarrow$ Search(*dstrTree*, *gridCenterAlign*($Q_x\times Q_y\times Q_t$));
2. $refineResult=\varnothing$;
3. FOR EACH $PT_{mo}\in filterResult$ DO
4. $moTuple=getTuple(PT_{mo})$;
5. IF $evaluate(moTuple,Q)\neq$NULL THEN
6. $refineResult=refineResult\cup evaluate(moTuple,Q)$;
7. ENDIF
8. ENDFOR
9. Return(*refineResult*).

如算法 4 所示，基于 DSTR-Tree 的查询分为两个阶段：

(1) 索引筛选(Filtering)阶段，通过 DSTR-Tree 索引筛选出查询区域所对应的移动对象集合 *filterResult*(该集合大于实际满足查询条件的移动对象集合)；

(2) 元组求精(Refinement)阶段，对 *filterResult* 集合中包含的移动对象的数据库元组进行查询计算，并将最终结果 *refineResult* 返回给查询用户.

通过算法 4 可以看出，数据库系统查询处理的总时间却取决于索引筛选时间和元组求精时间. 其中，索引筛选时间又受到索引更新代价及索引查询代价的双重影响. 通过分析可知，系统中格栅单元的粒度越大，则索引更新的代价越低，而 *filterResult* 中包含的无用记录越多，导致元组求精的代价越大；反之，系统中格栅单元的粒度越小，则 *filterResult* 中包含的无用记录越少，元组求精的代价也越小，但是索引更新的代价却越高，也会间接影响总体查询性能. 由此可见，为了达到最佳的总体查询性能，需要选择合适的格栅单元粒度.

4　实验比较与分析

在“基于路网的移动对象数据库及交通流统计分析系统(NMOD-TFSA)”[25] 中，我们实现了本文提出的 DSTR-Tree 索引模块. NMOD-TFSA 是在 PostgreSQL 8.2.4 数据库内核及其空间数据扩展 PostGIS 的基础上实现的一个移动对象数据库系统，支持完整的空间数据类型、交通网络数据类型、移动对象时空轨迹数据类型以及丰富的时空查询与位置更新操作. NMOD-TFSA 系统的结构如图 8 所示.

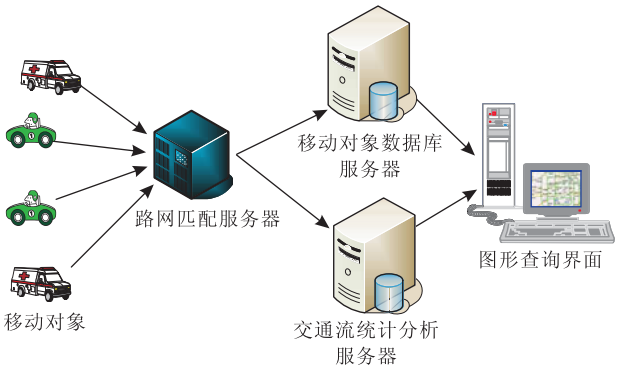


图 8　NMOD-TFSA 移动对象数据库系统的结构

DSTR-Tree 是在 PostgreSQL 所提供的通用索引框架 GIST 的基础上实现的. GIST 是一个可扩充的通用树形索引框架，通过开放相应的索引记录结构定义以及排序、插入、删除、查询等函数，可以实现用户自定义的单层树形索引结构，如 R 树、R* 树等. 由于 DSTR-Tree 是一种典型的单层树形结构，因此可以通过 GIST 无缝地在 PostgreSQL 中实现，这是与其它基于路网的移动对象索引相比所具有的优点之一. MON-Tree[15]、FNR-Tree[14]、NDTR-Tree[17] 等均采用了双层结构，因此无法直接在 GIST 中实现.

为了分析相关索引的性能，我们将 DSTR-Tree 与目前主流的移动对象轨迹索引方法进行了比较. 由于已经提出的移动对象轨迹索引方法如 STR-Tree、TB-Tree、MON-Tree、NDTR-Tree 等均以原始轨迹单元作为索引记录的基本单位，因此它们在位置更新时具有相同量级的索引更新频率；此外，由于 MON-Tree 和 NDTR-Tree 采用双层结构，不能

在 PostgreSQL 中基于 GIST 直接实现，而 DSTR-Tree 的主要设计目的之一即通过单层通用的索引数据结构(详见第 1 节)，来克服双层索引结构不易在通用 DBMS 中实现且无法兼容路网之外移动对象的局限，因此本文选择上述方法中最具有代表性的单层索引 TB-Tree 作为比较对象.

由于 TB-Tree 是基于 Euclidean 空间的，而 DSTR-Tree 可以同时兼容 Euclidean 运行矢量和基于路网的运行矢量两种情况，为了保证实验结果的公正性，在实验中强制规定 DSTR-Tree 和 TB-Tree 均处理完全相同的基于 Euclidean 的轨迹，从而使得二者在相同的位置更新频率下进行比较和分析. 此外，TB-Tree 只能处理移动对象的历史轨迹信息，不能依据活动轨迹单元信息对当前位置进行估算. 为了使之能对当前位置进行预测计算处理，将活动轨迹单元的插入与修正过程(见第 3.3 节)也加入到 TB-Tree 中.

实验中的交通网络数据采用的是北京市的真实 GIS 地图数据(为了实验方便，我们仅选择了三级以上的道路)，并通过一个数据转换程序，将 GIS 数据转换成了本文所定义的格式. 移动对象的轨迹数据则采用了北京星通联华交通科技有限公司提供的北京市出租汽车 GPS 历史数据，通过一个回放程序重新再现移动对象的位置更新及轨迹生成过程.

实验的硬件平台是 Genuine Intel(R) CPU 2140 处理器，1.6 GHz 主频，1 GB 内存，运行 Linux 操作系统. 表 1 列出了实验中的主要参数.

表 1 模拟实验的主要参数

参数名称	参数值(单位)	参数含义
N_{mo}	1000～9000	移动对象的个数
N_{routes}	38577	路网中道路的数量
N_{juncts}	15584	路网中交叉路口的数量
Lon-range	116.125～116.625	地图的经度范围
Lat-range	39.75～40.083	地图的纬度范围
Duration	10800(s)	每一轮模拟持续的时间
ξ	65(s)	移动对象触发位置更新的时间间隔
tuSize	1000 m×1000 m×65 s	原始轨迹单元平均大小
φ	2～10	格栅单元大小与原始单元大小的倍数关系
cellSize	*tuSize*×φ(φ=2～10)	DSTR-Tree 进行空间划分时的格栅单元大小，为 *tuSize* 的 φ 倍

在实验数据集中，原始轨迹单元的平均大小大约为 *tuSize*=1 km×1 km×65 s. DSTR-Tree 格栅单元的大小取其 φ 倍，如当 φ=5 时，格栅单元的大小为 *cellSize*=5 km×5 km×325 s. 为了实验方便，统一取 φ 为 2、4、6，即比较 *cellSize* 分别取值为 2 km×2 km×130 s、4 km×4 km×260 s、6 km×6 km×390 s 时的实验结果.

在实验中比较的性能指标包括：

(1) 索引更新代价(以索引记录插入和删除的总操作数来表示)；

(2) 索引所占用的存储空间大小；

(3) 数据库的查询处理时间；

(4) 数据库的查询处理时间与格栅单元大小之间的关系.

索引更新代价的实验结果如图 9 所示. 从图中可以看出，DSTR-Tree 在各种参数条件下的索引更新代价都低于 TR-Tree. 由于 DSTR-Tree 加大了索引记录的粒度，不需要像 TB-Tree 那样在每次位置更新时都要往索引中写入新记录，因此减少了索引更新的次数. 另外可以看出，DSTR-Tree 的格栅粒度越大其索引更新代价越低，因为移动对象仅在其时空轨迹跨越格栅单元时才需要进行索引更新.

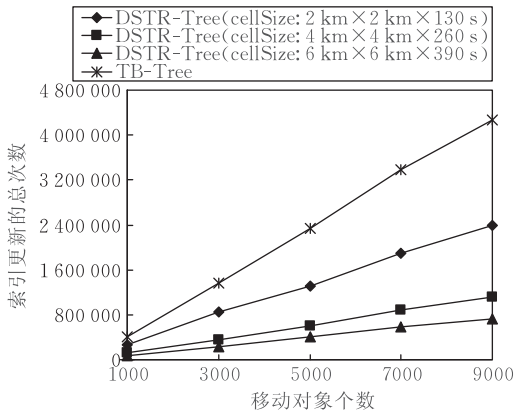


图 9 索引更新代价

图 10 给出了索引存储空间消耗的实验结果. 从图中可以看出，DSTR-Tree 在各种参数条件下的索引存储空间普遍低于 TB-Tree. 这是因为 DSTR-Tree 索引以概略化轨迹单元为索引记录的基本单位，而 TB-Tree 以原始轨迹单元为索引的记录单位. 对于同一个移动对象，其概略化轨迹单元的数目远小于原始轨迹单元的数目，因此 DSTR-Tree 的记录数比 TB-Tree 要少，索引存储空间也小于 TB-Tree.

图 11 是移动对象数据库在频繁位置更新的动态运行条件下进行查询处理的时间. 如前所述，查询处理时间由索引查找时间和元组求精时间组成. 为了显示查询处理各阶段时间的细节，在图 11(a)和图 11(b)分别给出索引查询时间和元组求精时间的实验结果，在图 11(c)中给出总体查询处理时间.

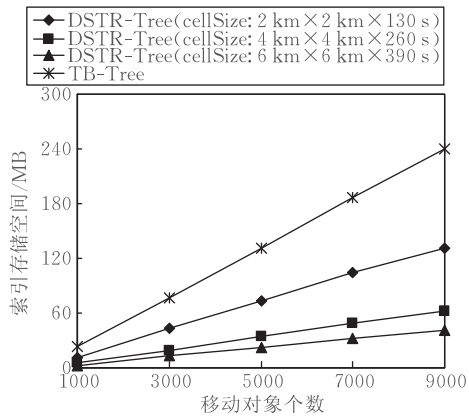
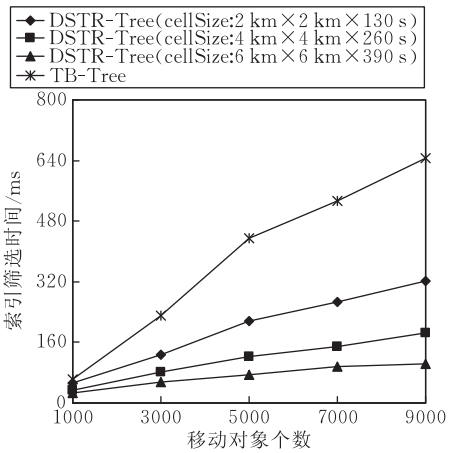


图 10　索引存储空间

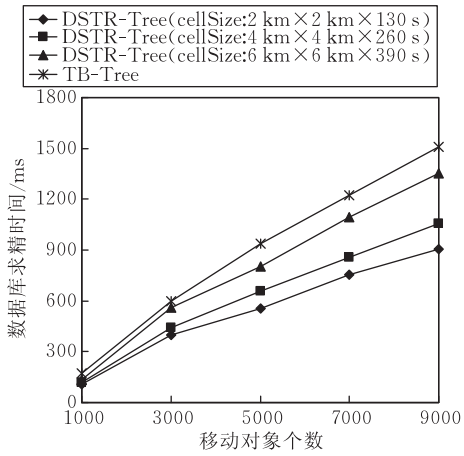
从图 11(a)可以看出，DSTR-Tree 的索引查找时间总体上低于 TB-Tree，这是因为一方面 TB-Tree 比DSTR-Tree要大，因此索引查找需要扫描更多的记录；另一方面，TB-Tree 在索引查找的同时需要处理频繁的索引更新操作，而 DSTR-Tree 的索引更新代价要小得多.

如图 11(b)所示，总体上讲，DSTR-Tree 的元组求精时间也少于 TB-Tree. 尽管 TB-Tree 比 DSTR-Tree 具有更高的准确率(即 filterResult 中包含的无用记录少)，但是这一优势被频繁的索引更新操作抵消掉了.

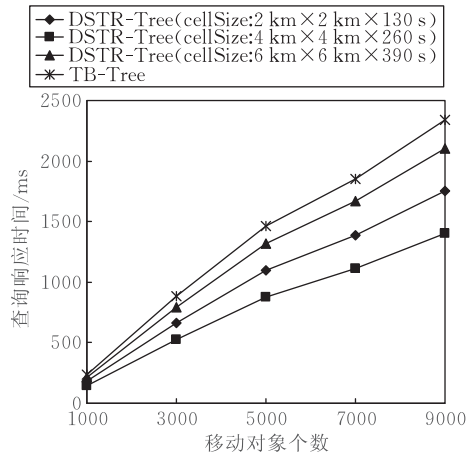
从图 11(c)可以看出，在频繁位置更新的条件下，DSTR-Tree 比 TB-Tree 具有更快的动态查询处理时间. 查询处理时间是索引查找时间与元组求精时间之和，再加上一些额外的开销，它同样极大地受到索引更新代价的影响. 此外，从图中还可以看出，DSTR-Tree 的查询响应时间与格栅的大小不呈单调关系.



(a) 索引筛选时间



(b) 数据库求精时间



(c) 查询响应时间

图 11　DSTR-Tree 与 TB-Tree 的性能比较

为了更好地分析查询响应时间与格栅单元大小的关系，变化 φ 的取值，从而得到格栅大小与查询响应时间的关系如图 12 所示(实验中移动对象的个数固定为 5000 个).

从图中可以出看，在本文的实验环境下，当格栅大小是原始轨迹单元大小的 4 倍(即 $\varphi=4$)时，系统能达到最佳的总体查询性能. 格栅单元大小的选择与实验环境密切相关，通常的影响因素包括数据库

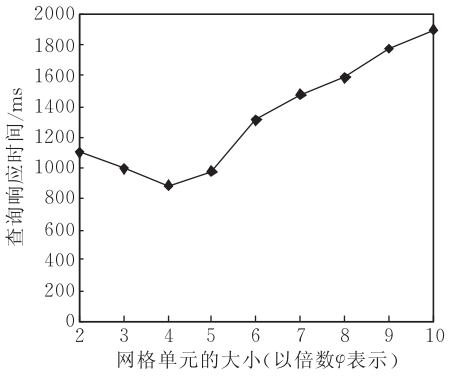


图 12 查询响应时间与格栅单元大小的关系

服务器的读写负荷比例、元组求精计算速度、数据库写速度等.

综合以上分析,与 TB-Tree 等以移动对象原始轨迹单元为基本索引单位的索引方法相比,DSTR-Tree 在移动对象数据库频繁位置更新的现实动态运行条件下具有更好的总体性能.

5 结 论

移动对象索引是支持海量移动对象管理的一项关键技术.然而,目前的移动对象轨迹索引方法如 STR-Tree、TB-Tree、FNR-Tree、MON-Tree、NDTR-Tree 等均以轨迹单元作为索引的基本记录单位,每次位置更新的同时需要进行一次索引更新,从而极大地增加了索引更新的代价,限制了数据库系统所能支持的移动对象的规模.此外,现有的网络受限移动对象的轨迹索引只能支持移动对象与路网完全匹配的情况,且采用双层结构,缺乏必要的灵活性,影响了它们在实际应用系统中的广泛使用.

为了解决上述问题,本文提出了一种网络受限移动对象的动态概略化轨迹 R 树索引(DSTR-Tree).DSTR-Tree 以移动对象的概略化轨迹单元作为索引记录的基本单位,当移动对象不跨越格栅单元时不需要对索引进行更新,因此降低了索引更新的代价,有效地提高了移动对象数据库的动态查询性能.此外,DSTR-Tree 采用典型的单层树形结构,可以无缝地在通用数据库框架如 PostgreSQL 中实现.最后,DSTR-Tree 能够同时支持移动对象在路网内和在路网之外移动的情况,具有充分的灵活性和实用性.实验结果表明,DSTR-Tree 在实际移动对象数据库频繁位置更新的现实条件下,提供了优秀的索引维护及总体查询处理性能.

致 谢 感谢中国科学院软件研究所的郭黎敏博士在 NMOD-TFSA 系统实现及 DSTR-Tree 索引实验方面所作的大量工作!

参 考 文 献

[1] Wolfson O, Xu B, Chamberlain S et al. Moving object databases: Issues and solutions//Proceedings of the 10th SSDBM. Capri, Italy, 1998: 111-122

[2] Saltenis S, Jensen C S, Leutenegger S T et al. Indexing the position of continuously moving objects//Proceedings of the ACM SIGMOD 2000. Texas, USA, 2000: 331-342

[3] Saltenis S, Jensen C S. Indexing of moving objects for location-based services//Proceedings of the 18th ICDE. San Jose, CA, 2002: 463-472

[4] Tao Y, Papadias D, Sun J. The TPR*-tree: An optimized spatio-temporal access method for predictive queries//Proceedings of the 29th International Conference on VLDB. Berlin, Germany, 2003: 790-801

[5] Lin B, Su J. On bulk loading TPR-tree//Proceeding of the 5th International Conference on Mobile Data Management (MDM'2004). Berkeley, USA, 2004: 114-124

[6] Tung H D T, Jung Y J, Lee E J et al. Moving point indexing for future location query//Proceedings of the ER Workshops'2004. Shanghai, China, 2004: 79-90

[7] Chen N, Shou L, Chen G et al. Adaptive indexing of moving objects with highly variable update frequencies. Journal of Computer Science and Technology, 2008, 23(6): 998-1014

[8] Chen J, Meng X. Update-efficient indexing of moving objects in road networks. Geoinformatica, 2009, 13(4): 397-424

[9] Kwon D, Lee S, Lee S. Indexing the current positions of moving objects using the lazy update R-tree//Proceedings of the 3rd International Conference on Mobile Data Management (MDM'2002). Singapore, 2002: 113-120

[10] Pfoser D, Jensen C S, Theodoridis Y. Novel approach to the indexing of moving object trajectories//Proceedings of the 26th International Conference on VLDB. Cairo, Egypt, 2000: 395-406

[11] Pfoser D. Indexing the trajectories of moving objects. IEEE Data Engineering Bulletin, 2002, 25(2): 3-9

[12] Tao Y, Papadias D. MV3R-tree: A spatio-temporal access method for timestamp and interval queries//Proceeding of the 27th International Conference on VLDB. Roma, Italy, 2001: 431-440

[13] Ding R, Meng X, Bai X. Efficient index update for moving objects with future trajectories//Proceedings of the 8th International Conference on DASFAA, Kyoto, Japan, 2003: 183-194

[14] Frentzos E. Indexing objects moving on fixed networks//Proceedings of the 8th International Symposium SSTD. Santorini Island, Greece, 2003: 289-305

[15] Almeida V T, Güting R H. Indexing the trajectories of moving objects in networks. GeoInformatica, 2005, 9(1): 33-60

[16] Chen J, Meng X. Indexing future trajectories of moving objects in a constrained network. Journal of Computer Science and Technology, 2007, 22(2): 245-251

[17] Ding Zhi-Ming, Li Xiao-Nan, Yu Bo. Indexing the historical, current, and future locations of network-constrained moving objects. Journal of Software, 2009, 20(12): 3193-3204(in Chinese)
(丁治明，李肖南，余波. 网络受限移动对象过去、现在及将来位置的索引. 软件学报，2009，20(12)：3193-3204)

[18] Güting R H, Böhlen M H, Erwig M et al. A foundation for representing and querying moving objects. ACM Transactions on Database Systems, 2000, 25(1): 1-42

[19] Ding Z, Güting R H. Managing moving objects on dynamic transportation networks//Proceedings of the 16th International Conference on SSDBM. Santorini Island, Greece, 2004: 287-296

[20] Güting R H, Almeida V T, Ding Z. Modeling and querying moving objects in networks. VLDB Journal, 2006, 15(2): 165-190

[21] Ding Z, Zhou X. Location update strategies for network-constrained moving objects//Proceedings of the 13th International Conference on DASFAA. New Delhi, India, 2008: 644-652

[22] Wolfson O, Yin H. Accuracy and resource consumption in tracking and location prediction//Proceedings of the 8th International Symposium SSTD. Santorini Island, Greece, 2003: 325-343

[23] Civilis A, Jensen C S, Pakalnis S. Techniques for efficient road-network-based tracking of moving objects. IEEE Transactions Knowledge and Data Engineering, 2005, 17(5): 698-712

[24] Tiesyte D, Jensen C S. Efficient cost-based tracking of scheduled vehicle journeys//Proceedings of the 9th International Conference on MDM. Beijing, China, 2008: 9-16

[25] Ding Z, Huang G. Real-time traffic flow statistical analysis based on network-constrained moving object trajectories//Proceedings of the 20th International Conference on DEXA. Linz, Austria, 2009: 173-183

DING Zhi-Ming, born in 1966, Ph. D., professor, Ph. D. supervisor. His main research interests include database systems, mobile computing, embedded systems, spatial-temporal database/data mining, and information retrieval.

Background

The focus of this paper is on the trajectory index problem in Moving Objects Databases (MOD). MOD is a database which can track and manage the dynamically changing locations of moving objects such as cars, ships, flights, and pedestrians. An MOD system can manage huge numbers of moving objects so that index is crucial to query them efficiently.

In recent years, the moving object trajectory index problem has been intensely studied with a lot of methods proposed, such as STR-Tree, TB-Tree, FNR-Tree, and MON-Tree. However, nearly all existing trajectory indices take trajectory units as the basic index records, whose granularity is too fine. As a result, frequent index updates are needed when location updates occur so that the efficiency can be greatly affected.

To solve this problem, we propose a new index method, Dynamic Sketched-Trajectory R-Tree for Network-constrained Moving Objects (DSTR-Tree) in this paper. The DSTR-Tree divides the spatial-temporal space into equal-sized grid cells, transforms every trajectory into a sketched trajectory with each unit connecting two centers of the grid cells that the moving object travels through, and indices the sketched trajectory units as an R-Tree. Since the sketched trajectory has much coarser granularity than the original trajectory, the index updating cost can be greatly reduced — when a location update occurs, even though the original trajectory needs to be changed, the sketched trajectory may remain unchanged so that the DSTR-Tree does not need to be changed either.

We have implemented the DSTR-Tree in an object-relational database, PostgreSQL, and compared it with other trajectory indices in terms of index update costs, storage consumption, query response time, and selectivity. The experimental results show that the DSTR-Tree outperforms the previously proposed trajectory index methods in running moving objects databases with frequent location updates.

The paper was partially supported by National Natural Science Foundation of China under Grant Nos. 91124001 and 60970030.

The research team has been conducting research on Moving Objects Databases for about 10 years, with a series of results achieved such as the MODTN database model for network-constrained moving objects, the UTR-Tree and NDTR-Tree index methods for moving object trajectories, the Net-LUM and the ANLUM location update mechanisms for moving object tracking. We have also implemented a moving objects database system called NMOD-TFSA, which can manage trajectories of moving objects and extract traffic conditions through statistical analysis.

The work of the paper is focusing on the management of huge moving objects in databases, and can be used to support quick query processing of trajectories.

HybridHP：一种轻型的内核完整性监控方案及其形式化验证

钱振江[1),2),3)] 刘 苇[1),2)] 黄 皓[1),2)]

[1)](南京大学软件新技术国家重点实验室 南京 210046)

[2)](南京大学计算机科学与技术系 南京 210046)

[3)](常熟理工学院计算机科学与工程学院 江苏 常熟 215500)

摘 要 虽然传统的虚拟化监控方法可以在一定程度上保障操作系统安全.然而,虚拟监控器VMM中管理域Domain0的存在以及操作系统级的切换所带来的性能损失是很多具有大型应用的操作系统所不能接受的.注重硬件虚拟化技术的监控能力而摒弃其不必要的虚拟化能力,提出了一个新型的通用的虚拟化监控框架HybridHP,并实现其原型.HybridHP将管理域和虚拟机监控机制两者整合到被监控操作系统的地址空间,具有很好的获取被监控系统操作语义的能力.利用Isabelle/HOL形式化辅助证明工具验证HybridHP的隔离性、安全性和监控能力.最后对HybridHP进行了攻击实验和性能评估,结果显示HybridHP提供了和传统的虚拟化监控方案相同的安全保障,并具有很好的系统性能.

关键词 硬件虚拟化;内核完整性;安全监控;安全攻击;Isabelle形式化验证

中图法分类号 TP309 **DOI号**: 10.3724/SP.J.1016.2012.01462

HybridHP: A Verified Lightweight Approach to Provide Lifetime Kernel Integrity Surveillance

QIAN Zhen-Jiang[1),2),3)] LIU Wei[1),2)] HUANG Hao[1),2)]

[1)](*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210046)

[2)](*Department of Computer Science and Technology, Nanjing University, Nanjing* 210046)

[3)](*School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, Jiangsu* 215500)

Abstract Although traditional virtualization monitoring can help ensure security, the existence of management domain (such as Domain0) and performance loss caused by OS-level switches make these approaches unsuitable for many OSs with large applications. In this paper, focusing on monitoring capability of the hardware virtualization technology without the unnecessary virtualization functionality, we propose HybridHP, a new general-purpose framework of virtualization monitoring, and implement the prototype. HybridHP merges the management domain and virtual machine monitoring functionality into the monitored system, and has strong ability to obtain operational semantics of the monitored system. We use the formal theorem prover Isabelle/HOL to verify isolation, security and monitoring capability of HybridHP. With the systemic experiments and performance evaluation for HybridHP, we show that HybridHP provides at least the same

收稿日期:2011-04-18;最终修改稿收到日期:2012-02-13.本课题得到国家"八六三"高技术研究发展计划项目基金(2007AA01Z409)、江苏省科技支撑计划自然科学基金(BE2008124)以及江苏省"六大人才高峰"高层次人才项目(2011-DZXX-035)资助.**钱振江**,男,1982年生,博士研究生,讲师,中国计算机学会(CCF)会员,主要研究方向为操作系统安全、形式化验证和嵌入式系统.E-mail: zhenjiang.qian@gmail.com. **刘 苇**,男,1986年生,硕士研究生,主要研究方向为虚拟机监控、形式化验证.**黄 皓**,男,1957年生,博士,教授,博士生导师,主要研究领域为系统软件、信息安全.

security guarantees as what can be achieved by the traditional virtualization monitoring approaches, and has well system performance.

Keywords　hardware virtualization; kernel integrity; security monitoring; security attacks; Isabelle formal verification

1　引　言

内核级的攻击和恶意程序可以轻易地破坏整个操作系统的完整性，而传统监控方案的本质缺陷在于恶意程序可能获得和内核一样的权限，导致操作系统没有保护自身的能力. 近年来硬件虚拟化技术(如 Intel-VT 和 AMD-SVM 等)的发展为基于虚拟机(如 XEN[1] 和 KVM[2] 等)的监控方式提供了底层的支持和保障. 总的来说，由于虚拟机监控器位于内核的底层，比内核具有更高的权限，所以可以中断内核的执行，对内核的执行进行审查和检验，从而保证内核的完整性. 这种实时性也是虚拟机监控器区别于其它监控方案的最大优势. 很多的学者对此进行了研究[3-4]. 基于虚拟机的监控器能够捕捉到包括内存在内的资源访问动作，使得在内核执行过程中能够根据内核执行路径中所访问的对象进行选择性的事件触发，从而提高了监控器监控的实时性.

已有的虚拟化监控方案[3,5-11]，普遍存在这么几个方面的问题：

(1) 性能损失. 这些方案注重虚拟化机制，而非监控本身. 在监控单独的操作系统时，它们往往需要付出很大的额外代价，例如采用 XEN 进行监控的方式，需要有单独的管理域 Domain0，像 I/O 之类的操作需要管理域 Domain0 的干预处理，这极大地影响了被监控系统的性能；

(2) 可信基(Trusted Computing Base，TCB)膨胀. 虚拟机自身的代码量往往过于庞大(如 XEN 3.4.1 拥有 230 K SLOC 的代码量)，再加上管理域 Domain0 的系统代码量，我们面临着虚拟机本身以及相应的管理域 Domain0 的正确性问题，很难保证监控方案自身不会引入程序错误；

(3) 语义获取难. 引用监控器位于被监控系统地址空间之外，因此引用监控器看到的视图和被监控系统的是有差别的，很难去理解被监控系统中对某个内存地址的具体操作语义，管理域进行监控的过程需要对被监控系统的具体操作语义进行转化，而这一过程是很难做到实时而全面的. 虽然 Sharif 等人在被监控系统中构建了一个监控器 SIM[12]，以便于取得被监控系统的语义，但是 SIM 与被监控系统的隔离性仍需要依赖于另一个监控器的保护，而本文提出的内嵌式监控器的自我保护能力不依赖于其它任何安全机制.

在本文中，给出了一个通用的虚拟化框架 HybridHP，其特点如下：

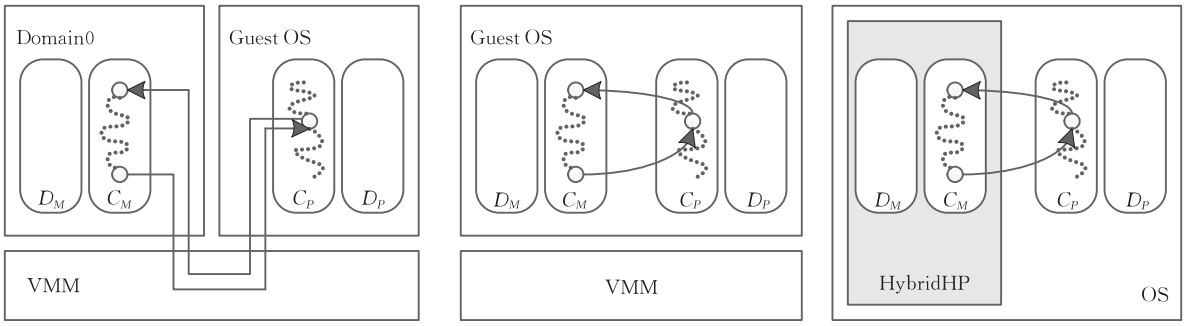
(1) HybridHP 注重于监控本身而非虚拟化技术. HybridHP 没有 Domain0 之类的单独管理域概念，HybridHP 以模块的方式嵌入到被监控系统的内核空间中进行监控服务. HybridHP 只负责监控系统的运行，能够截获被监控系统中的特权操作、异常和对寄存器、内存、I/O 等的访问，且不受原有内核模块的影响和破坏，并且不会干预其 I/O 的操作，为此被监控系统的性能不会受到很大的影响；

(2) 由于采用模块的方式运行，HybridHP 看到的内存视图和被监控系统看到的是一致的，这样容易获得被监控系统的操作语义，监控的正确率和效率可以得到很大的提高；

(3) 由于 HybridHP 监控功能的单一性，其代码量可以控制在很小的范围，这便于对其正确性进行证明，我们利用 Isabelle[13] 形式化辅助证明工具对其进行了正确性证明.

总的来说，HybridHP 是被监控系统内部的一个模块，它利用硬件虚拟化技术，使自身独立于被监控系统，具有比被监控系统中的原有模块更高的权限. 图 1 显示了 HybridHP 与已有虚拟化监控方案的区别，其中 C_M、D_M 是监控器的代码段和数据段，C_P、D_P 是被监控系统的代码段和数据段. 区别于已有的虚拟化监控方案，由于 HybridHP 以模块方式运行，其自身的代码和数据信息暴露在被监控系统的视图下，很容易受到其它恶意模块的破坏，所以它自身的安全性和隔离性是我们成功与否的关键. 为此，我们提出了相应的保护机制，并且在高层使用形式化的方法并借助 Isabelle 证明其安全性和隔离性.

HybridHP 是一种完全内嵌式的监控方案，不借助其它任何的辅助监控措施，并且其正确性经过



(a) 传统的虚拟化监控框架 (b) SIM的监控框架 (c) HybridHP的监控框架

图 1 HybridHP 与已有的虚拟化监控方案的区别

严格的形式化逻辑验证.

本文第 2 节阐述 HybridHP 的整体框架设计;第 3 节说明 HybridHP 原型的具体实现细节和关键技术;第 4 节给出 HybridHP 正确性的 Isabelle 形式化证明;第 5 节阐述对 HybridHP 的测试结果和系统性能评估;第 6 节对本文进行总结和对未来的工作进行展望.

2 HybridHP 的设计

2.1 HybridHP 的目标

HybridHP 的设计目标是能够利用硬件虚拟化技术对操作系统实施监控,但不会对操作系统的性能和效率造成太大的影响,为此需要实现以下几个具体目标:

(1) 监控程序本身是安全的. 为了实现这个目标,要求监控程序非常小,其本身是可以通过形式化验证的,这样才能保证它自身的安全可靠和可控;

(2) 监控程序自我保护. 监控程序本身要有自我保护的能力,防止其它恶意模块的破坏;

(3) 监控程序不可旁路. 监控程序能够截获被监控系统中的特权操作和对关键内存对象、寄存器、I/O 设备的访问等. 恶意模块不能绕过监控程序执行非法操作和修改受保护的关键数据;

(4) 机制与策略分离. 监控程序本身提供的只是监控机制,具体的策略是可以实时更新的;

(5) 被监控系统的性能不会受到太大的影响. 已有的监控方案正是由于架构的原因,极大地损失了系统的性能,所以并不是很适用;

(6) 易于获得被监控系统的语义. 获得操作系统的语义是监控程序一个很重要的任务,而已有的虚拟机监控器由于虚拟机看到的内存视图和被监控系统看到的是有差别的,很难去理解被监控系统的具体操作语义.

2.2 HybridHP 的整体框架

HybridHP 的整体框架如图 2 所示. HybridHP 的核心特点在于 HybridHP 以模块的方式在被监控系统内核空间中运行,由于使用了硬件虚拟化技术,比其它的内核模块具有更高的权限. 整个内核空间处于 0 级模式(Ring 0),但是 Intel-VT 又将 0 级分成两种子模式:VMX Root 模式和 VMX Non-Root 模式. VMX Root 是真正的 0 级,具有所有权限. VMX

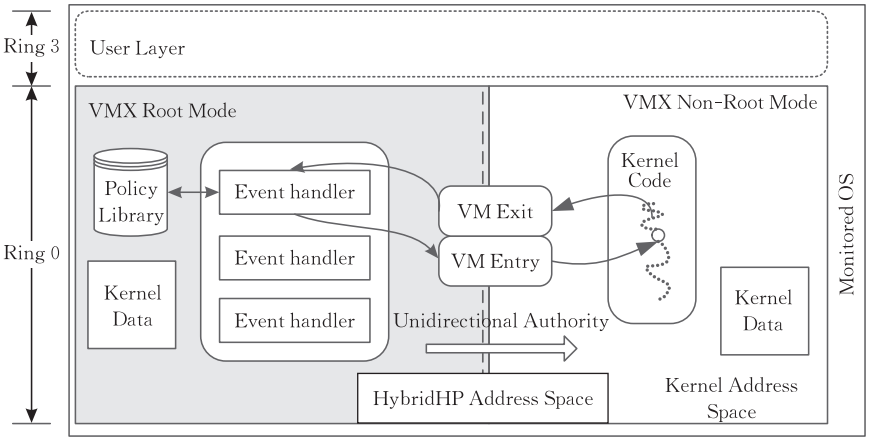


图 2 HybridHP 框架

Root 模式的操作方式和没有开启硬件虚拟化机制(VMX)时的操作基本上是相同的，不同之处在于 VMX Root 模式中能够调用 VMX 的指令，同时，一些寄存器中能够装载的值会受到一定的限制. 在本文的框架中，HybridHP 运行在 VMX Root 模式. VMX Non-Root 是受限的 0 级，其中很多的特权操作和事件会触发控制权转移到 VMX Root 模式的监控程序. 对应地，框架中的被监控系统运行在 VMX Non-Root 模式.

HybridHP 本身不提供虚拟化，没有管理域和驱动域的概念. 对于系统的驱动部分(如 I/O 访问)，被监控系统的驱动模块直接和硬件交互，不需要 HybridHP 的干预，这样对系统的性能不会造成太大的影响. 当然，如果需要 HybridHP 进行监控和管理的话，也可以通过对 HybridHP 增加相应的策略来达到目的. 对于监控能力，HybridHP 只提供监控的机制，采用机制和策略相分离的方式. 为此，HybridHP 启动之后，首先从外部环境获得策略信息. 为了避免外部环境的不可信以及恶意策略对 HybridHP 完整性的破坏，HybridHP 使用数字签名的方式来进行策略的安全更新. HybridHP 根据策略对内核的运行过程进行实时监控，如关键数据的保护、异常行为触发等. 运行在 VMX Non-Root 模式的被监控系统触发受 HybridHP 监控的事件，通过两种模式之间唯一的入口 VM Exit 进入 VMX Root 模式，由 HybridHP 根据策略信息进行处理，处理完成后通过 VM Entry 返回 VMX Non-Root 模式.

由于 HybridHP 和被监控系统采用的是相同的页表视图，HybridHP 对于被监控系统的对内存地址的操作语义不需要经过转换而可以直接理解，提高监控处理的效率. 同时，HybridHP 具有自我保护的能力，被监控系统虽然可以看到 HybridHP 的页表视图，但是无法对其进行修改，而 HybridHP 拥有修改被监控系统内存视图的权限. 因此，从地址空间的角度来讲，被监控系统内核的地址空间其实是 HybridHP 地址空间的子集，HybridHP 和内核其它模块对于地址空间视图的权限是单向的.

3　HybridHP 的实现

为了验证本文的方案，实现了 HybridHP 的原型. 本节对 HybridHP 的实现和主要关键技术进行阐述.

HybridHP 以模块的方式在被监控系统内核空间中运行，区别于系统中的其它模块，在 HybridHP 中开启硬件虚拟化(VMX)，HybridHP 运行在 VMX Root 模式，而整个被监控系统运行在 VMX Non-Root 模式，所以 HybridHP 具有比其它的内核模块更高的权限. 如何做到在 HybridHP 框架中开启硬件虚拟化，使得 HybridHP 运行于 VMX Root 模式，而被监控系统运行于 VMX Non-Root 模式是需要解决的第 1 个问题.

在 HybridHP 框架中，控制 VMX Root 和 VMX Non-Root 运行模式的关键是 VMX 控制结构 VMCS，主要包括 3 个部分：

(1) 被监控系统的状态区(Guest State Area)，用于在 VMX Non-Root 向 VMX Root 模式切换时保存被监控系统的状态信息，如段寄存器、CR 寄存器、指令指针 IP、栈顶指针 SP 等，以及从 VMX Root 返回 VMX Non-Root 模式时的 VM Entry 入口地址；

(2) 主机状态区(Host State Area)，存放监控系统自身的状态信息. 在 HybridHP 框架中，主机状态区主要包括 HybridHP 运行过程的各种状态信息，如与 HybridHP 运行相关的段寄存器、CR 寄存器、HybridHP 栈顶指针 SP，以及从 VMX Non-Root 向 VMX Root 模式切换时的 VM Exit 处理程序的入口地址；

(3) HybridHP 执行控制区(Execution Control Area)，主要包括 HybridHP 监控过程的配置信息，如需要保护的寄存器信息(CR0 等)、HybridHP 需要捕获的特权指令(如 CPUID 等)和异常(如 page-fault 等)信息以及相应的处理方式和处理函数入口地址. 同时在 HybridHP 执行控制区中还存放了产生 VMX Non-Root 到 VMX Root 模式切换的原因，包括错误码、触发切换的特权指令以及被监控系统期望修改的寄存器或者对象的地址等信息.

在被监控系统启动后，以模块的方式加载 HybridHP. HybridHP 启动的初始化流程包括：

(1) 分配 HybridHP 的内存空间，主要包括内核栈、动态数据区、以及 VMX 控制结构 VMCS 所需的空间；

(2) 通过 VMXON 指令开启硬件虚拟化，设置被监控系统的状态区；

(3) 设置 VMCS 中的主机状态区，其中 VM Exit 的入口地址指向 HybridHP 的处理函数入口，页目录地址寄存器 CR3 设置为与被监控系统相同，

使得 HybridHP 与被监控系统采用相同的页表视图；

(4)设置 HybridHP 执行控制区，包括需要保护的寄存器(如 CR0 等)、需要捕获的特权指令(如 CPUID 等)以及异常(如 pagefault 等)信息，这是 HybridHP 监控的核心，同时根据策略信息将关键数据区域所在的页面以及被监控系统页表所在的页等数据设置成在 VMX Non-Root 模式下只读，开启对数据的保护；

(5)通过 VMLANCH 指令返回被监控系统 VMX Non-Root 模式，使得被监控系统开始正常运行.

下面对 HybridHP 框架的具体实现和关键技术进行详细阐述.

3.1 HybridHP 的关键技术之一：关键数据保护

为了尽量地压缩 HybridHP 的代码量，HybridHP 框架没有单独的管理域 Domain0 的概念，更没有像传统虚拟机监控器方案中自身实现的虚拟存储管理功能，HybridHP 和被监控系统处于相同的页表视图下，被监控系统甚至可以看到 HybridHP 的存在，如何实现对关键数据(如被监控系统的页表和代码所在页、需要保护的特定数据页等)的保护是需要解决的关键问题.

HybridHP 的监控能力主要体现在两个方面：特权指令的捕获和关键数据区域的保护.

对特权指令的实时捕获是 HybridHP 作为轻型内核完整性监控方案的基础. HybridHP 通过自身的初始化过程在 VMCS 的 HybridHP 执行控制区中对需要监控捕获的特权指令(如 CPUID 等)进行配置. 在被监控系统的运行过程中，由于被监控系统运行于 VMX Non-Root 模式，这些特权指令的运行将触发指令陷入(Trap)事件，被监控系统的运行将被打断，通过 VM Exit 的入口地址进入 HybridHP 的相应处理程序，运行模式也从 VMX Non-Root 切换到 VMX Root 模式，控制权转移至 HybridHP，从而实现对特权指令的捕获. HybridHP 在处理完指令陷入之后，通过 VM Entry 接口返回被监控系统.

对于关键数据区域的保护，涉及到 HybridHP 框架中对寄存器和异常等的监控事件类型的捕获，主要由以下技术提供保证：

(1)始终开启 CPU 的页保护机制. 开启页保护机制是 HybridHP 监控的基础，因为如果让操作系统具有去除页保护机制的能力，那么内核恶意模块就可以不通过页保护机制而更改任意的内存，那么也就没有了关键数据区域的概念. 在 X86 架构中，页保护机制由寄存器 CR0 的 WP 位控制. 在 HybridHP 的控制策略中，设置对 CR0 寄存器的访问控制，同时在 VMCS 的 HybridHP 执行控制区设置对 CR0 寄存器的保护. 在 HybridHP 的监控框架下，被监控系统的运行过程中如果出现修改 CR0 寄存器的操作，将触发 CRA(Control Register Accesses)异常，运行模式从 VMX Non-Root 切换到 VMX Root 模式，同时控制权转移到 HybridHP. HybridHP 通过对被监控系统的操作语义进行分析并根据控制策略进行判断，对于非法的 CR0 修改动作，HybridHP 禁止该动作的执行，从而使得被监控系统中的恶意模块没有禁止页保护机制的能力，即始终开启页保护机制(CR0.WP=1).

(2)在上述(1)的基础上，HybridHP 根据策略将需要保护的关键数据在内存中的页设置成在 VMX Non-Root 模式下只读. 一旦发生对这些数据的修改操作就会触发 pagefault 异常，通过模式转移 VM Exit 从 VMX Non-Root 模式切换为 VMX Root 模式，控制权交由 HybridHP，通过和策略信息比较，判断修改动作的合法性，如果合法，HybridHP 负责将该修改动作执行，否则，通过注入返回的方式，向被监控系统发送错误警告.

(3) HybridHP 将被监控系统的页表所在的页设为在 VMX Non-Root 模式下只读，那么恶意模块试图通过去掉页表中相应页的保护属性而修改内存的操作必然会触发 pagefault 异常，被 HybridHP 所捕获，处理方式与(2)类似.

从以上的技术可以看出，对关键数据的修改动作不会绕过 HybridHP 的监控处理，满足不可旁路的要求.

3.2 HybridHP 的关键技术之二：自我保护

基于 XEN 之类的虚拟机监控框架中，由于使用了影子页表[1]，虚拟机监控器以及管理域与被监控系统采用的是不同的页表，为此监控器部分的相关数据结构是不在被监控系统的内存视图之内的，所以监控程序不会受到被监控系统的破坏和干扰. 也就是说由于 XEN 中有自己的虚拟存储管理，所以可以做到监控器自身的数据对客户机完全不可见. HybridHP 以模块化的方式嵌入在被监控系统中运行，采用了与被监控系统相同的页表视图，没有对被监控系统完全隐藏物理内存的能力，为此 HybridHP 如何达到自我保护也是需要解决的关键问题.

HybridHP 框架通过将自己的代码页、数据页和策略所在页等数据设置为在 VMX Non-Root 模式下只读来起到自我保护. 也就是说，HybridHP 将

自身作为关键数据的一部分进行监控保护，那么即使自身暴露在被监控系统的内存视图中，利用 3.1 节中的关键数据保护技术，如果被监控系统中的恶意模块试图修改 HybridHP 的代码、数据或者策略信息，都会被 HybridHP 捕获到，HybridHP 可以否决所有这些非法的修改.

总的来说，HybridHP 的自我保护能力使得虽然自身的所有数据暴露于被监控系统的视图之内，但是被监控系统没有对其进行修改的权限，所以可以保证自身的安全.

3.3　HybridHP 的关键技术之三：策略更新

为了实现监控策略和机制的分离以及策略的更新，HybridHP 本身实现的只是监控的机制，监控策略需要由外部环境提供. 在传统的基于虚拟机的监控方案中，虚拟机监控器所采用的监控策略由管理域 Domain0 提供，为此策略的安全性由 Domain0 保证. 在 HybridHP 框架中，HybridHP 自身是一个可信计算基(TCB)，而被监控系统是不可信的，如果依赖被监控系统为其提供策略信息，被监控系统中的恶意模块可以修改策略信息或者任意封装恶意的策略并发送给 HybridHP，那么整个监控框架是不可信的. 同时，HybridHP 没有管理域 Domain0 的概念，为此需要解决 HybridHP 策略信息的来源问题，以及策略信息从被监控系统的用户层输入到 HybridHP 空间这样一条安全的策略输入路径的问题.

利用一套独立的可信平台作为策略中心，其中存放了为 HybridHP 所准备的各种策略信息，该策略中心作为 HybridHP 策略信息的来源保证了策略信息源头的安全性. 同时，采用数字签名加密认证的方法来解决上述的策略安全输入路径问题. 对从策略中心获得的 HybridHP 策略信息文件使用 Hash 算法(如 MD5、SHA 算法等)计算 Hash 值，即做“数字摘要”，再对数字摘要用签名私钥做非对称加密(如 RSA、ElGamal① 等)，即做“数字签名”. 之后将以上的签名和策略信息文件原文一起进行封装，被监控系统在用户层将这一封装的结果通过 VMX 系统调用 Hypercall 的形式向 HybridHP 发送，HybridHP 对接收的经过加密的策略信息进行验证. HybridHP 收到的数字签名结果，其中包括数字签名和策略信息文件原文. HybridHP 首先用发方公钥解密数字签名，导出数字摘要，并对策略信息文件原文做同样 Hash 算法得出一个新的数字摘要，并将这两个摘要进行结果比较，结果相同则签名得到验证，否则，说明策略在输入过程存在被恶意的模块修改的情况，HybridHP 判定输入的策略信息为无效，不予采用. 对于经过验证的策略信息文件，HybridHP 还需检查其是否为预定义的策略格式，如三元组(主体，动作，客体)等. 如果是正确的策略信息，那么将其加入策略库中，并进行相应的关键数据保护设置，否则忽略这些信息，这样可以确保 HybridHP 得到的策略一定是管理人员所希望的策略，整个框架如图 3 所示.

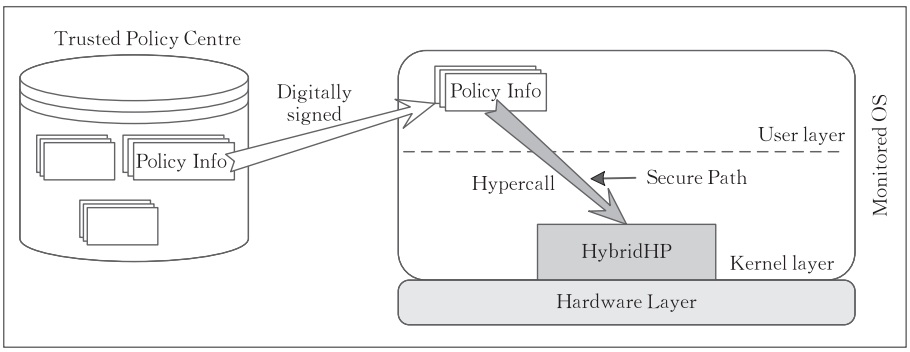


图 3　HybridHP 策略更新

由于 Hash 算法(MD5、SHA 等)存在安全性的问题，不可信的被监控系统可以生成一个与可能的策略文件具有相同“数字摘要”的恶意策略文件，并传递给 HybridHP，但其内容只有极小的可能性也是一段策略内容，而更多的是一段乱码，或者说不是预定义的格式，如三元组(主体，动作，客体)等，对此 HybridHP 仍能将其识别. 同时，在保证签名私钥安全不泄露的情况下，外部不可信的被监控系统无法将恶意的策略信息进行封装并发送给 HybridHP. 因此，恶意策略无法破坏 HybridHP 的完整性，保证了策略信息的机密性、不可修改性和不可伪造性.

HybridHP 只提供监控的机制，采用机制和策略相分离的方式，具体的策略由外部环境提供，这给具体的策略配置提供了便利. 在本文中，可以采用三

① ElGmal Encryption. http://en.wikipedia.org/wiki/ElGamal_encryption

元组(主体,动作,客体)的格式来定义策略,如表 1 所示.其中表头的第 2 行表示客体.

表 1 HybridHP 策略示例表

主体	动作					
	CR0	PageTable	IDT	GDT	LDT	Predefined CriticalData
可信模块	R	R	R/W	R/W	R/W	R/W
不可信模块	R	R	R	R	R	R
内核	R	R	R/W	R/W	R/W	R/W
HybridHP	R/W	R/W	R/W	R/W	R/W	R/W

HybridHP 策略主体可以分为可信模块、不可信模块、内核以及 HybridHP.可信模块主要包括得到第三方(工业界)认证的各种设备驱动程序,如显卡和网卡驱动等;不可信模块主要包括上层用户的各种第三方应用程序,这些程序模块未得到有效的认证,因而是不可信的,它们对于系统资源的访问是只读的权限;被监控系统的内核部分在可信的情况下对系统资源具有大部分的权限;在 HybridHP 框架中,HybridHP 自身作为可信基,对系统资源具有所有的访问权限.

3.4 HybridHP 的关键技术之四:可信启动

HybridHP 保证了在其启动以后的时间内监控系统的安全性,但是在 HybridHP 启动之前的阶段无法保证系统的安全性.被监控系统中的恶意模块可以在 HybridHP 启动之前破坏 HybridHP,或者禁止 HybridHP 的启动.

在监控框架中,HybridHP 启动之前的安全保护由基于 TPM[①] 的可信启动来负责,如图 4 所示.

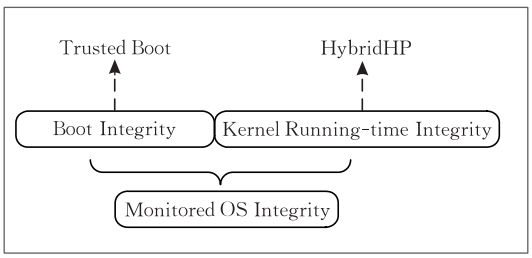


图 4 系统完整性保护

在 HybridHP 原型中整合了 tboot[②] 软件,通过监控启动过程中所有关键数据的完整性,能够有效地防止系统的核心组件被替换,从而保证系统启动过程的完整性,这也是 HybridHP 监控服务的前提.

4 HybridHP 的验证

HybridHP 代码量控制在 8K SLOC.本节使用 Isabelle 形式化辅助证明工具阐述 HybridHP 框架的隔离性、安全性和监控能力的验证,并进行安全分析.

验证部分包括:HybridHP 对"去掉页保护机制"、"修改页表"和"修改关键数据"动作的不可旁路性,HybridHP 自我保护机制,以及 HybridHP 策略更新.

4.1 Isabelle 的符号表示

这一小节描述验证过程用到的符号表示方法.采用 Isabelle 系统的符号表示,这种符号表示方法与传统数理逻辑中的方式基本相同.Isabelle 可以实现对计算系统的形式化描述和验证,采用的 Isabelle/HOL 是 Isabelle 中对高阶逻辑(Higher Order Logic,HOL)的支持和实现.

HOL 作为一种类型化的逻辑系统,其类型系统与函数式编程语言(如 ML、Haskell 等)是类似的.类型变量可以用$'a$,$'b$ 等符号进行表示.对于类型项,可以用如 $x::'a$ 定义方式.Isabelle 对类型支持构造子操作,例如 *nat list* 用于定义由自然数组成的列表,*int set* 定义由整形变量组成的集合.

在 HybridHP 的描述和验证的过程中,对于新类型的构造主要采用 3 种定义方式:datatype、types 和 record. datatype 实现对代数数据类型的构造,例如,对于寄存器类型定义为

datatype *reg*=CR0|CR1|CR2|CR3|CR4.

types 表示类型的简化记号,如 types *nat_set*=*nat set*,定义了新的类型 *nat_set*,它表示自然数组成的集合类型.

record 用于构造带名称的元组类型,例如,对于 3.3 节描述的策略信息定义为

record *policy*=*subject*::*string*,
cond::*string set*,
object::*string*.

新类型 *policy* 表示策略信息,它含有 3 个成员:*subject* 表示策略信息的动作主体,*cond* 表示需要对对象设置的操作语义信息,*object* 表示策略信息中的具体客体对象.对于 record,引用成员信息可以表达为如 *subject policy*,表示引用 *policy* 中的 *subject* 成员.假设 *policy* 拥有值(|*subject*=HybridHP,*cond*=READ_ONLY,*object*=CR0|),

① Trusted Computing Group. http://www.trustedcomputinggroup.org/developers/trusted_platform_module/specifications

② Trusted Boot. http://tboot.sourceforge.net

更新操作可以表达为如 *policy*(|*cond* := READ_WRITE|)，表示将 *policy* 中的 *cond* 成员修改成 READ_WRITE，但 *subject* 和 *object* 成员保持不变.

对于函数运算，使用“⇒”符号表示函数定义域和值域的映射对应关系. 函数更新操作使用如 $g(x:=y)$方式来表达. 函数在集合上的值域运算使用如 $g'Z \equiv \{y \mid \exists x \in Z.\ y = gx\}$表达，表示函数 g 以集合 Z 为定义域的值域.

4.2　HybridHP 对象模型

HybridHP 以模块的方式在被监控系统的内核空间中运行，对被监控系统的行为进行监视，根据预定义的策略信息对系统中的操作进行判断. 被监控系统中其它模块不直接与 HybridHP 交互，并不需要知道 HybridHP 的存在. 系统中各种动作的执行效果，可以看成是对系统状态的改变或者迁移，HybridHP 在此过程中起到监控和决定的作用，如图 5 所示.

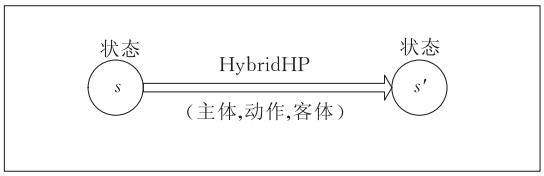


图 5　状态转换图

文中使用 Isabelle 验证工具对 HybridHP 的安全性进行验证，需要对 HybridHP 建立模型. 以对象模型的方式建立 HybridHP 的语义模型，将操作的主体和客体作为对象来看待，同时将系统状态之间的转换看成是操作的主客体对象以及 HybridHP 相互作用的结果. 在这样一个场景下，在下面的章节中使用 Isabelle 对 HybridHP 对象模型进行符号化表示，并在此基础上阐述验证 HybridHP 对“去掉页保护机制”、“修改页表”、和“修改关键数据”动作的不可旁路性，HybridHP 自我保护机制，以及 HybridHP 策略更新行为的正确性定理.

4.3　HybridHP 系统模型形式化描述

系统执行进程标识：types *entity_id* = *nat*.

系统的执行进程使用 record 定义：

record *process* = *pid* :: *entity_id*,
pcb :: *pcb_struct*,

其中 *pcb_struct* 为进程控制块 PCB 结构.

系统的状态定义为

record *state* = *mem* :: *entity_id* ⇒ *process*,
regs :: *reg set*,
pagetable :: *pageitem set*,
next_id :: *entity_id*,
curr_mode :: *string*,

其中，*mem* 表示系统中内存视图，通过进程的标识可以找到所有的进程个体；*regs* 表示系统中的寄存器集合；*pagetable* 表示系统的页表，是由页表项 *pageitem* 组成的集合，*pageitem* 是虚拟地址 *va* 和物理地址 *pa* 组成的 record；*next_id* 表示系统中可用的进程标识；*curr_mode* 指明系统目前的模式，取值如 *VMX_Non_Root* 和 *VMX_Root*.

针对要验证的动作类型定义如下：

datatype *action* = Write CR0
| Write PageTable
| Write CriticalData
| Write HybridHPData
| PolicyUpdate *policy_info*
| Null_Action.

action 包含对 CR0、页表、关键数据和 HybridHP 数据信息的修改动作、策略更新动作以及空操作.

HybridHP 的策略集合定义为 *PolicySet* :: *policy set*.

系统所有的状态集合定义为 *S* :: *state set*.

获得动作主体的语义函数 *SubjectofAction*：*action* ⇒ *subject*.

获得动作客体的语义函数 *ObjectofAction*：*action* ⇒ *object*.

获得动作的操作语义函数 *CondofAction*：*action* ⇒ *cond*.

HybridHP 策略监控判定语义定义：

policy_judge :: *action* ⇒ *policy set* ⇒ *bool*,
policy_judge a ps ≡
(|*subject* = *SubjectofAction a*,
cond = *CondofAction a*,
object = *ObjectofAction a*|) ∈ *ps*.

policy_judge 判断动作是否符合预定的策略.

执行单个动作引起的系统状态转化语义函数 *step*：*state* ⇒ *action* ⇒ *state*.

4.4　形式化验证

本节从攻击的角度来说明修改系统属性的操作，阐述 HybridHP 如何防范这些修改方式，并验证其正确性，同时对策略更新进行验证.

定理 1.　CR0 保护.

Theorem 1.　$\forall s \in S.$ [|*action* = Write CR0; (*policy_judge action PolicySet*) = False; *s'* = *step*

s action|] → CR0 (*regs s'*) = CR0 (*regs s*).

第 1 种攻击方式试图去掉系统的页保护机制(CR0.WP 位),从而可以修改任意的内存页.由于 HybridHP 对 CR0 进行保护,这种对 CR0 的写操作立即引起 CRA 异常,HybridHP 通过 VMX 的控制结构 VMCS 读取触发原因,此时 CR2 寄存器的值指向写数据错误的地址.通过 CR3、task_struct 等获得执行主体的信息,即对操作语义进行解析,并根据策略信息判断出这种操作的违法性,然后使用注入返回的方式对被监控系统进行错误警告.执行该操作前后,系统状态中 CR0 保持不变.定理 1 说明在系统的运行过程中,如果当前动作为修改 CR0(Write CR0),而在策略判断中,该动作的主体没有权限修改 CR0,此时策略检测(*policy_judge action PolicySet*)无法通过(False),那么系统在该动作后状态(s')环境中的 CR0 寄存器保持不变.如此可以保证 HybridHP 监视系统中所有对 CR0 的操作,使得被监控系统没有禁止页保护机制的能力,即始终开启页保护机制.

定理 2. 页表保护.

Theorem2. $\forall s \in S.$ [|*action* = Write PageTable; (*policy_judge action PolicySet*) = False; s' = *step s action*|] → *pagetable s'* = *pagetable s*.

这种攻击方式试图通过修改页表中的页表项的只读属性,从而对系统的内存页进行修改.对于这种攻击方式,没有 HybridHP 监控的系统中很容易受到破坏.HybridHP 框架根据策略信息对系统中的页表所在的页进行保护,因此修改页表的操作将引起 pagefault 异常,由 HybridHP 在 VMX Root 模式下进行处理,此时 CR2 寄存器的值指向页表所在的页.定理 2 说明 HybridHP 监控对页表的修改动作,如果当前动作为修改页表(Write PageTable),而在策略判断中,该动作的主体没有权限修改页表,此时策略检测(*policy_judge action PolicySet*)也无法通过(False),那么系统在该动作后状态(s')环境中的页表项保持不变,从而保证了对页表的保护.

定理 3. 关键数据保护.

Theorem 3. $\forall s \in S.$ [|*action* = Write CriticalData; (*policy_judge action PolicySet*) = False; s' = *step s action*|] → *mem s'* = *mem s*.

Proof apply (rule Theorem1, rule Theorem2)

第 3 种攻击方式对内存中的关键数据页进行修改,而这些页受到 HybridHP 的策略保护.定理 1 保证了外部攻击无法去除系统的页保护机制,定理 2 保证了外部攻击无法修改内存页在页表中的保护属性,因此这第 3 种攻击方式必然会触发 HybridHP 对 pagefault 异常的捕获,无法绕过 HybridHP 的监控.为此,定理 3 的证明过程需要借助定理 1 和定理 2.

定理 1、定理 2、定理 3 阐述了 HybridHP 监控的不可旁路性.

定理 4. 自我保护.

Theorem4. $\forall s \in S.$ [|*action* = Write HybridHP-Data; (*policy_judge action PolicySet*) = False; s' = *step s action*; *curr_mode s* = VMX_Non_Root|] → (*mem s'*) HybridHP = (*mem s*) HybridHP.

Proof apply (*case_tac* (*curr_mode s*) and rule Theorem1, rule Theorem2, rule Theorem3).

最后 1 种攻击方式假设通过页表视图和系统模块信息找到 HybridHP 代码和数据以及策略信息所在的内存页,对 HybridHP 进行破坏.HybridHP 框架的自我保护能力是通过将自身的代码段、数据段以及策略信息的内存页设置成在 VMX Non-Root 模式下只读来达到的.同时,系统中的恶意模块运行在 VMX Non-Root 模式,因此这种修改操作必然受到 HybridHP 的干预.为此,定理 4 说明,如果当前状态(s)的运行模式为受限模式(VMX_Non_Root),策略检测(*policy_judge action PolicySet*)对修改 HybridHP 自身信息的非法动作判断为没有相应的权限(False),那么该非法动作无法修改 HybridHP 的信息,即 HybridHP 的数据信息保持不变.我们可以看出定理 4 的证明需要定理 1、定理 2、定理 3 的辅助,并对系统的当前的状态模式采用分情况的证明策略 *case_tac* 来拆解.

定理 5. HybridHP 策略更新.

Theorem5.

$$\begin{aligned}&\forall s \in S.\ [|action = \text{PolicyUpdate } policy\_info|] \rightarrow\\ &\quad (validate\_policy(policy\_info) = \text{True} \wedge\\ &\quad (policy\_judge\ action\ PolicySet) = \text{True} \rightarrow\\ &\qquad PolicySet = PolicySet \cup policy\_info) \wedge\\ &\quad (validate\_policy(policy\_info) = \text{False} \vee\\ &\qquad (policy\_judge\ action\ PolicySet) = \text{False} \rightarrow\\ &\qquad s' = step\ s\ Null\_Action \wedge\\ &\qquad PolicySet \ominus s = PolicySet \ominus s').\end{aligned}$$

HybridHP 采用数字签名加密认证的方式进行策略更新,主要是为了避免外部环境的不可信对策略信息安全性的影响.PolicyUpdate 是策略更新动作,策略信息定义为 *policy_info* :: *policy set*.本文对主要的加密算法(如 MD5、SHA 算法以及 RSA、ElGamal 等)和数字签名过程进行了 Isabelle 建模验证,并将其封装在库中,对外提供 *validate_policy*

接口进行解码验证和策略信息的预定义格式(如三元组)识别. 如果策略信息通过验证,那么 HybridHP 认为这些策略信息是可取的,并加入系统的策略集合中;如果策略信息没有通过验证($validate_policy(policy_info)=$False),或者不允许恶意的策略更新动作(($policy\_judge\ action\ PolicySet$)=False),那么此更新动作无法执行,以空操作 $Null\_Action$ 来表示. 同时,策略信息保持不变,其中 $PolicySet\ \Theta s$ 表示状态 s 下的策略集. 定理 5 阐述了 3.3 节中描述的在被监控系统不可信的环境下,策略更新的安全性以及恶意策略不会破坏 HybridHP 的完整性.

上述的定理都依赖于策略信息的安全性,即其正确性和所构造的 $PolicySet$ 有关. $PolicySet$ 可以看成是 3.3 节中从外部环境输入的策略信息,而策略信息源头的安全性和策略更新的安全路径保证了输入到 HybridHP 中的策略信息的合法性,因此这些定理的正确性得到了保证.

Isabelle 的验证环境配置为 Studio XPS 9100 台式电脑,2.8GHz Intel i7 930 处理器,3GB 内存,操作系统采用 openSUSE Desktop 11.3 版本,Isabelle 采用 Isabelle2009-2_bundle_x86-linux 版本. 整个 Isabelle 验证工程代码量大概在 15K SLOC 左右,完整的验证耗时 40min 左右.

Isabelle 的验证结果如图 6 所示. "No subgoals" 说明 Isabelle 验证逻辑完整,不存在任何未证明的子目标.

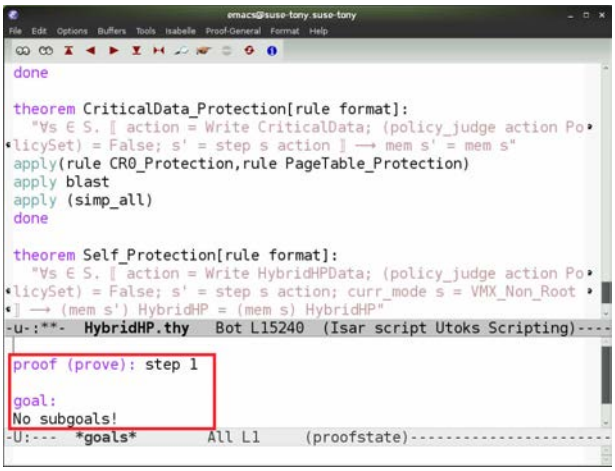


图 6　Isabelle 验证结果图

5　实验和系统性能评估

HybridHP 对操作系统的性能影响及其监控的有效性是实验关注的重点.

在传统的虚拟化监控方案中,性能评估往往关注于监控机制本身所带来的性能损失(将这部分损失的性能定量记为 ε),而忽视被监控系统所获得的真实物理性能(定量记为 α). 我们认为在传统的虚拟化监控框架下,需要引入考虑虚拟框架(包括管理域如 Domain0 以及虚拟机如 XEN)耗用的性能(定量记为 β). 因此,整个监控框架下整体的性能(定量记为 θ)应该是被监控系统所获得的物理性能(α)、虚拟框架耗用的性能(β)和监控机制导致的损失性能(ε)这 3 个部分之和,即 $\theta=\alpha+\beta+\varepsilon$,取 θ 的值为 100%.

被监控系统所得到的真实性能是关注的重点,所以在性能测试中,加入了这一个评估指标. 主要从以下两个角度进行性能评估:(1)被监控系统所获得的真实性能(α);(2)监控本身所带来的性能损失(ε).

为了评估 HybridHP 的性能,选择了几个 Linux 的基准测试程序,包括 UnixBench① 以及其它的一些真实应用软件,通过和传统的基于 XEN 的监控框架以及 SIM 的监控方案进行对比来说明 HybridHP 的性能. 测试平台是 Studio XPS 9100 台式电脑,2.8GHz Intel i7 930 处理器,3GB 内存. 被监控系统的各种软件配置信息如表 2 所示.

表 2　被监控系统软件配置信息

名称	版本	配置
openSUSE Desktop	11.3	Standard Installation
UnixBench	4.1.0	Make run
Kernel build	2.6.34.12	Make world
Bzip2	1.0.4	tar −zxvf ⟨kernel file⟩

此外,为了有效地进行性能对比,本实验还将被监控系统作为客户机,部署到传统的基于 XEN 的以及 SIM 的监控框架中. 其中管理域 Domain0 采用半虚拟化(Paravirtualization)配置,XEN 采用标准配置,软件配置信息如表 3 所示.

表 3　Domain0 和 XEN 软件配置信息

名称	版本	配置
Domain 0	OpenSUSE 11.3	Standard Paravirtualization
XEN	3.4	Standard Installation

图 7 说明了被监控系统在 3 种框架下所获得的性能(α)对比,其中"解压文件"操作是面向 CPU 计算的,"编译"是面向 I/O 操作的,而 UnixBench 是

① UnixBench. http://ftp.tux.org/pub/benchmarks/System/unixbench

一个综合测试.与传统的基于 XEN 的监控方案和 SIM 不同,HybridHP 没有管理域 Domain0 的存在,因此在这 3 个对比测试中,采用 HybridHP 框架的被监控系统所获得性能均接近于物理主机性能.特别是在"编译"测试中,由于 HybridHP 不对被监控系统的 I/O 操作进行干预,系统的 I/O 性能基本没有影响.当然,HybridHP 对于系统页表操作的影响,和 XEN 中影子页表操作对系统的性能影响类似,主要的时间开销包括:模式转换产生的上下文切换、策略搜索时间、HybridHP 对异常处理的时间以及页表切换(CR3)的时间.

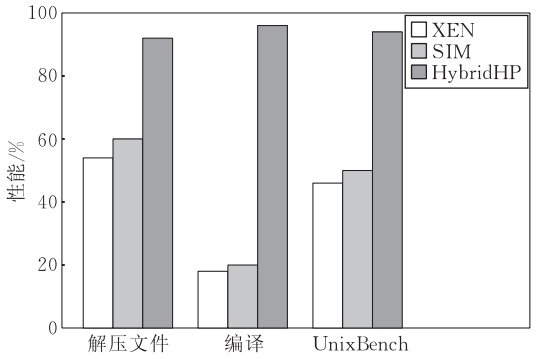


图 7 被监控系统所获得的性能(α)对比图

表 4 说明了以创建进程操作为例,3 种框架下监控机制本身所带来的性能损失(ε).可以看出,监控性能损失接近并略低于 SIM 的性能损失,HybridHP 对被监控系统的性能影响控制在很小的范围,满足 2.1 节第 5 点对 HybridHP 的性能要求,这也说明 HybridHP 是一种保证内核完整性的轻型方案.

表 4 创建进程操作的监控性能损失(ε)对比

监控类型	平均时间/μs	相对性能损失/%
裸机	3.487	NULL
XEN	28.039	704.1
SIM	3.967	13.7
HybridHP	3.853	10.5

为了验证 HybridHP 监控能力的有效性,我们选择 4 种不同类型的 rootkit 攻击方式[14]:Mood-nt、adore-ng、Synapsys 和 SucKIT2,并对它们的攻击目标进行了配置,使其适应 HybridHP 监控框架.第 1 种攻击方式试图通过写 CR0 寄存器来设置 WP 为 0,从而去掉页保护机制;第 2 种攻击方式试图修改系统的页表中某一项的只读权限,从而可以任意修改内存页;第 3 种攻击方式试图修改系统调用表,模拟对系统关键数据的修改;第 4 种攻击方式查找 HybridHP 所在的页,试图修改 HybridHP 的代码和数据页.这 4 种攻击代表了公认系统漏洞数据库(NVD①)目前典型的内核攻击方式.实现结果表明,HybridHP 可以有效地监控和阻止这 4 种攻击方式.

6 结束语

本文提出了一种用于监控内核完整性的轻型方案 HybridHP,它以模块的方式在被监控系统的内核空间中实施内核完整性监控服务.HybridHP 的功能由 4 个关键技术保证:关键数据保护、自我保护、策略更新和可信启动.关键数据保护技术对由策略信息设定的关键数据实施保护,被监控系统运行于 VMX Non-Root 模式,HybridHP 运行于 VMX Root 模式,保证了 HybridHP 监控的不可旁路性.HybridHP 和被监控系统拥有相同的页表视图,便于获得被监控系统的操作语义,提高监控处理的效率.但 HybridHP 的代码和数据信息暴露于被监控系统的内存视图下,为此加入了 HybridHP 的自我保护功能,防止被监控系统中恶意模块对其进行破坏.HybridHP 采用监控机制和策略分离的思想,本身只提供监控能力,通过策略更新的方式支持策略信息的动态修改,考虑到被监控系统的不可信问题,使用数字签名加密的技术保证策略更新路径的安全性以及策略信息的机密性、不可修改性和不可伪造性.可信启动技术提供在 HybridHP 对被监控系统实施监控服务之前的安全性,保证被监控系统启动过程中不会对 HybridHP 进行破坏.实现了 HybridHP 的原型,并使用 Isabelle 对其正确性进行了证明.同时,本文对 HybridHP 进行了攻击测试和系统性能评估,结果表明 HybridHP 能有效地监控针对内核完整性的攻击,同时对被监控系统的性能影响控制在很小的范围.

接下来的工作计划将 HybridHP 在多核处理器平台进行改进和实现,使得 HybridHP 运行在独立的处理器核上,加快监控处理的速度.HybridHP 提供了监控内核完整性的轻型方案,如何和各种具体的安全策略模型(如 BLP、Lattice 和 Biba 完整性模型等)结合以及如何构建更加有效的安全策略以保证尽可能多的攻击动作和攻击动作序列被检测也是将来研究的一个重要方向.

① National Vulnerability Database. http://nvd.nist.gov/

致　谢　本文作者感谢所有本文的匿名审稿者，感谢您们对本文提出宝贵的意见！

参　考　文　献

[1] Barham P, Dragovic B, Fraser K, Hand S, Harris T L, Ho A, Neugebauer R, Pratt I, Warfield A. XEN and the art of virtualization//Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03). New York, 2003: 164-177

[2] Kivity A, Kamay Y, Laor D, Lublin U, Liguori A. KVM: The Linux virtual machine monitor//Proceedings of the 2007 Ottawa Linux Symposium. Ottawa, 2007: 225-230

[3] Garfinkel T, Rosenblum M. A virtual machine introspection based architecture for intrusion detection//Proceedings of the 10th Network and Distributed System Security Symposium (NDSS'03). San Diego, 2003: 191-206

[4] Grizzard J. Towards self-healing systems: Re-establishing trust in compromised systems[Ph. D. dissertation]. Georgia Institute of Technology, Atlanta, Georgia, 2006

[5] Jiang X, Wang X, Xu D. Stealthy malware detection through VMM-based "Out-Of-the-Box" semantic view reconstruction//Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07). Alexandria, VA, 2007: 128-138

[6] Lanzi A, Sharif M, Lee W. K-Tracer: A system for extracting kernel malware behavior//Proceedings of the 16th Network and Distributed System Security Symposium (NDSS'09). San Diego, 2009

[7] Payne B D, Carbone M, Sharif M I, Lee W. Lares: An architecture for secure active monitoring using virtualization//Proceedings of the 29th IEEE Symposium on Security and Privacy (S&P'08). Oakland, California, 2008: 233-247

[8] Riley R, Jiang X, Xu D. Guest-transparent prevention of kernel rootkits with VMM-based memory shadowing//Proceedings of the 11th Recent Advances in Intrusion Detection (RAID'08). Boston, MA, 2008: 1-20

[9] Seshadri A, Luk M, Qu N, Perrig A. SecVisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity OSes//Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP'07). Stevenson, WA, 2007: 335-350

[10] Wang Z, Jiang X, Cui W, Ning P. Countering kernel rootkits with lightweight hook protection//Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09). Chicago, IL, 2009: 545-554

[11] Yin H, Liang Z, Song D. HookFinder: Identifying and understanding malware hooking behaviors//Proceedings of the 16th Network and Distributed System Security Symposium (NDSS 2008). San Diego, 2008

[12] Sharif M, Lee W, Cui W, Lanzi A. Secure In-VM monitoring using hardware virtualization//Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09). Chicago, IL, 2009: 477-487

[13] Nipkow T, Paulson L C. Isabelle/HOL: A Proof Assistant for Higher-Order Logic. Berlin: Springer, 2002

[14] Petroni N L. Property-based integrity monitoring of operating system kernels[Ph. D. dissertation]. University of Maryland, College Park, 2008

QIAN Zhen-Jiang, born in 1982, Ph. D. candidate, lecturer. His current research interests include operating system security, formal verification and embedded system.

LIU Wei, born in 1986, M. S. candidate. His current research interests include virtual machine surveillance and formal verification.

HUANG Hao, born in 1957, Ph. D., professor, Ph. D. supervisor. His current research interests include system software and information security.

Background

This work is mainly supported by the National High Technology Research and Development Program (863 Program) of China under grant No. 2007AA01Z409, the Jiangsu Provincial Natural Science Foundation of Science and Technology Support Program under grant No. BE2008124, and the "Six Talents Peak" High-Level Personnel Project of Jiangsu Province under grant No. 2011-DZXX-035. They all aim to develop a secure and trusted microkernel operating system in which novel security technologies and virtual monitoring are researched, system design are formally specified and some of the security properties are verified formally or informally. In past years, the group has done a lot of related

works including a trusted boot revolution, an operating system integrity surveillance prototype based on VMM (OSISS), an operating system object semantics model (OSOSM) and its partial formal specifications and verification. This paper reports the authors' works on how to design and implement lifetime kernel integrity surveillance for OSOSM.

This paper aims to resolve how to provide lifetime kernel integrity surveillance using virtual technology, but without loss of performance of monitored OS. In this paper, the authors explain the existing problems of the traditional virtual machine monitoring approaches, such as system performance loss, trusted computing base (TCB) expansion of the monitoring programs, and difficulty of obtaining the operational semantics of the monitored system. This paper proposes and implements HybridHP, a lightweight approach to provide lifetime kernel integrity surveillance, which resolves the problems above. To the best of our knowledge, HybridHP is the first system that merges the management domain (e.g. Domain0) and virtual machine monitoring functionality into the monitored system, and provides monitoring services in the way of embedded module. The memory view of HybridHP is consistent with the view of the monitored system, so it is easy to obtain the operational semantics of the monitored system. HybridHP does not intervene in the I/O operation of the monitored system, and the performance loss is little. HybridHP has the small amount of codes with 8K SLOC, and is easy to be verified formally. In this paper, the correctness of HybridHP is verified in strict formal logic with the theorem prover Isabelle/HOL. The authors also introduce the systemic experiments and performance evaluation for HybridHP. HybridHP can effectively monitor the kernel integrity, and has good overall system performance.

对等点播系统中节点搜索机制研究

张铁赢[1)] 刘 悦[1)] 钟运琴[1),2)] 程学旗[1)]

[1)](中国科学院计算技术研究所 北京 100190)

[2)](中国科学院研究生院 北京 100190)

摘 要 对等点播系统(P2P-VoD)中的跳转操作需要高效的节点搜索,如何快速查找到"合适"的节点是个挑战."合适"包含两方面因素:(1)内容匹配;(2)物理性能匹配.而传统的方法大部分只涉及对前者的研究.文中提出了一种层次化的搜索模型(简称Mediacoop),不仅可以让搜索到的节点在内容上满足要求,而且在物理性能上也能满足要求.具体而言,Mediacoop首先利用播放距离来索引全部节点,再利用延迟特征优选内容上已经符合要求的节点.在NS2模拟器上的实验表明,Mediacoop在用户体验和系统开销上均优于传统的方法.同时,在实际系统CoolFish中的部署和运行也验证了Mediacoop的实用性.

关键词 对等;视频点播;节点搜索;流媒体

中图法分类号 TP309 **DOI号**: 10.3724/SP.J.1016.2012.01475

Lookup Mechanism for Peer-to-Peer Video-on-Demand

ZHANG Tie-Ying[1)] LIU Yue[1)] ZHONG Yun-Qin[1),2)] CHENG Xue-Qi[1)]

[1)](*Institute of Computing Technology, Chinese Academic of Sciences, Beijing* 100190)

[2)](*Graduate University of Chinese Academic of Sciences, Beijing* 100190)

Abstract A fundamental challenge in P2P-VoD system is how to provide random seeking function. To address this problem, in this paper, we propose Mediacoop, a novel structured lookup service which can find peers to provide required data with good quality. In Mediacoop, we exploit the unchanged playpoint distance between neighbors to avoid publishing large number of sharing messages. In addition, Mediacoop considers the underlying network in order to find close supplying peers. Theoretical analysis and extensive simulations show that Mediacoop outperforms traditional methods with less overhead. We have also implemented a real-world P2P system based on Mediacoop, called CoolFish. The running results also prove the effectiveness of our design.

Keywords Peer-to-Peer; Video-on-Demand; peer search; streaming

1 引 言

随着互联网宽带接入的普及,对等视频点播服务(P2P-VoD)已经成为了最流行的互联网应用之一.点播的最大特点在于用户可以随意跳转,即从当前位置跳转到前面或者后面进行观看.但是,跳转到新位置后,当前邻居节点很可能没有所需数据,造成了邻居节点的失效.因此,我们的目标是如何快速高效地查找到"合适"的邻居节点来提供数据.

"合适"的邻居节点包含两方面特征.一是内容匹配,即该节点拥有查找者所需内容,这一点也是最

收稿日期:2010-12-17;最终修改稿收到日期:2012-04-10.本课题得到国家"八六三"高技术研究发展计划项目基金(2006AA010105-02)、国家"九七三"重点基础研究发展规划项目基金(2004CB318109)和国家自然科学基金(60933005)资助.**张铁赢**,男,1982年生,助理研究员,主要研究兴趣为计算机网络、分布式计算、P2P流媒体、网络安全.E-mail:zhangtiey@software.ict.ac.cn.**刘 悦**,女,1971年生,博士,副研究员,主要研究兴趣为信息检索、社会计算.**钟运琴**,男,1984年生,博士研究生,主要研究方向为海量数据管理、时空数据库、云计算.**程学旗**,男,1971年生,研究员,博士生导师,主要研究领域为信息检索、社会计算、分布式计算.

基本的要求.二是质量匹配,即查该节点拥有较好的物理性能,例如高带宽和低延迟.质量匹配不仅仅是查询优化的问题,它对播放质量起到了至关重要的作用.Huang[1]、Pucha[2]以及 Hefeeda 等人[3]均指出,邻居节点的物理性能较差是导致无法及时获得数据的重要原因.衡量物理性能最重要的两个指标是带宽和延迟[4],本文中使用延迟作为衡量指标,因为带宽只能在建立连接后通过实际传输的数据测得[3-4],而我们的目标是建立连接前进行节点选择(即节点查找),所以带宽这一指标不适用于本文.详见第 2 节和第 4 节.

通常,针对 P2P-VoD 的搜索方法只涉及内容搜索.典型的方法是使用分布式哈希表(DHT),把缓冲好的内容对应的信息发布到网络中等待查询(如文献[3,5]).但是,这种方法应用在视频服务中会造成大量的网络开销.因为每个节点缓冲的内容是随着观看进度发生变化的,节点需要不断地发布对应的信息,这就造成了巨大的网络开销.尽管文献[3]作者对节点物理性能方面进行了探索,但是他们提出的方法并不能应用在 P2P-VoD 中.文献[6-7]作者提出了非 DHT 的结构化搜索方法以避免大量的网络开销,但是他们都没有对节点的物理性能进行研究.

在本文中,针对 P2P-VoD 的特征,我们提出了一个基于结构化的层次搜索模型(简称 Mediacoop).本方法的一次查找能够同时满足内容匹配和质量匹配的双重要求.具体而言 Mediacoop 的查找过程被分为两个阶段:

(1)使用播放距离来定位拥有所需内容的节点.对于给定影片,各节点的播放速度是相同的,这样,节点间的播放距离是不变的(除非节点进行跳转和暂停操作).因此,避免了传统方法的巨额网络开销;

(2)把备选节点索引为一个树形覆盖网(overlay),该过程是以 AS 域间延迟及 IP 前缀为基础进行索引,其结果是能够找到与查询者延迟最小的节点子集,以完成节点质量匹配.

我们在理论上证明了:以不低于传统 DHT 的查找效率 $O(\log N)$,Mediacoop 能够同时完成上述两阶段查找.我们在 NS2 模拟器上做了大量对比实验,结果表明 Mediacoop 在启动时间、跳转延迟、播放连贯度和网络开销上的优势十分明显.同时,在实际 P2P-VoD 系统 CoolFish 上实现了 Mediacoop,其运行结果也验证了算法的实用性.

本文第2节介绍相关工作;第3节描述 Mediacoop 的基本模型;第 4 节详细讨论 Mediacoop 的设计细节;第 5 节给出理论上和实验上的验证结果,并介绍实际系统的运行状况;第 6 节对全文进行总结.

2 相关工作

P2P-VoD 网络一般分为树形结构和网状结构两类.树形结构被较早提出,但是由于树形结构并不适用于动态性很强的 P2P 网络,目前主要的研究方向集中在网状结构.我们先简单介绍树形结构的已有方法.P2Cast[8]是一个典型的单播树系统,它使用补丁技术进行数据流分发.然而,在这种分发模型中,一个播放节点只有唯一的父亲节点提供数据,这对于异构网络来说是远远不够的.另外,在动态性很强的 P2P 环境下维护树形结构是很困难的.作为改进,P2VoD[9]把分发树组织为层次结构.当节点离开的时候会自动由上层节点替代,新加入系统的节点被分配到最低层.但是,P2VoD 并没有提出适合点播的动态交互操作,例如跳转等.作为网络结构上的极大改进,网状结构是目前主要的研究方向.

PROP 作为网状结构的代表,以 DHT 作为基础提供 P2P-VoD 服务.在 PROP 中,只要节点获取到一个数据块,它便把该数据块的信息发布到网络中.当节点需要数据的时候,它先去查询 DHT 获得拥有所请求内容的节点,再向这些节点请求数据.这种方法带来一个严重的问题,在观看过程中,节点收到的内容随播放进度发生变化,这样,节点就要不断地发布新收到的内容信息.这种频繁的更新消息带来了巨大的网络开销.不仅如此,当缓冲区中陈旧数据被丢弃的时候,也要发布相应的删除信息.

PROMISE[3]在节点的物理性能上进行了探索,基于网络带宽提出了一种节点选择方法.在播放过程中,它利用实际传输的媒体数据计算邻居节点可以提供的可用带宽,从而进行节点的优化筛选.但是,这种方法是在节点间建立网络连接后进行数据传输的时候进行的,也就是说是此时的资源节点(或称为邻居节点)已经存在,目标是进行节点"选择"而不是"查找".因此,物理带宽这一指标并不适用于节点查找这种场景.

和我们工作较为相关的主要是 OBN[6]和 RINDY[7].OBN 减少了上述 DHT 方法带来的网络开销,它利用节点间缓冲区的重叠关系构建非 DHT 的结构化 overlay.但是,节点物理性能的研究依然没有在 OBN 中得以实现.RINDY 使用类似的方法

构建了一个多环的查找网络.查询者缩小环形范围直至找到候选节点集合,再使用 Gossip 协议定位拥有所需内容节点.但是,Gossip 协议会带来较大的网络开销并且这种方法依然没有考虑节点物理性能上的优劣.

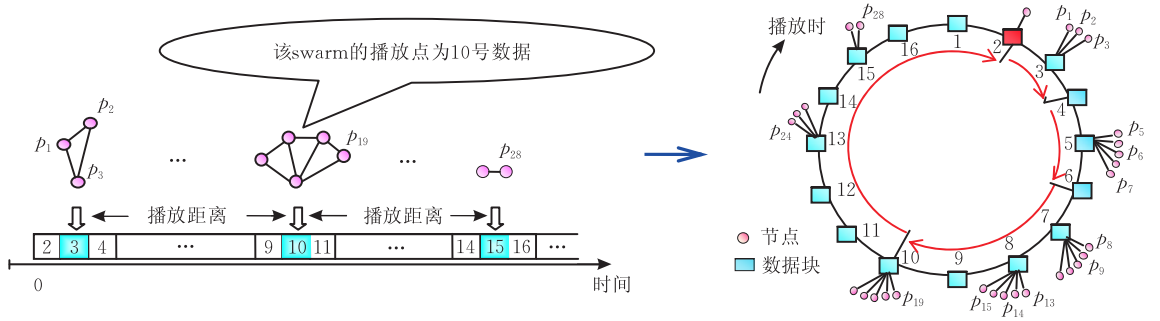
3　Mediacoop 的系统模型

本节主要讨论 Mediacoop 的基本思路,具体细节设计在第 4 节中介绍.Mediacoop 的两大中心任务:(1)如何提供高效的内容查询;(2)如何找到网络延迟较低的节点.相应地,Mediacoop 的搜索过程分为两个阶段,这两个阶段对应着两个不同的覆盖网,基本思想由 3.1 节和 3.2 节分别介绍.

3.1　播放点覆盖网(Playpoint Overlay)

本节介绍第 1 阶段查询的基本思想.

第 1 阶段的目标是内容匹配查询,我们把一部电影分为 M 个数据块,数据块按播放顺序编号为 $block_1, block_2, \cdots, block_m$,每一个数据块对应一个播放点.这样,节点 p_i 的播放点(PlayPoint)PP_{p_i} 定义为节点 p_i 当前正在播放的数据块编号 $block_j$.拥有相同播放点的节点被归为同一个 swarm(如图 1(a)所示).对于给定影片,不同节点播放该影片的速度是一样的.因此,正常播放情况下,播放点间的距离是不发生变化的(除非节点进行跳转或暂停操作).在第 1 阶段,我们的想法是利用播放点距离把全部的 swarm 索引在一个环上(图 1(b)).一个 swarm 只要建立好索引信息(即路由表),即使在播放过程中,不用更新索引也可以获知其它 swarm 当前的状态,如图 2 所示.每个节点的路由表存储了其它 swarm 播放点的信息(详见 4.2 节),路由表中的每条表项由播放点距离计算得出.请求者可以方便地查找到拥有目标播放点的节点,整个查找过程类似于著名的 Chord 网络[10],不同之处在于我们使用了播放点距离而不是 Hash 值.因此,播放点距离的不变性使得节点不需要更新索引信息也获知其它 swarm 的状态.这个特性使 Mediacoop 大大减少了网络开销.播放点距离的数学定义在 4.2 节中给出.



(a) 播放点相同的节点归为同一个swarm,节点间的播放距离不变除非节点进行VCR操作

(b) 所有swarm被索引在首尾相接的环上,索引信息(路由表)根据播放点距离计算得出

图 1　播放示意

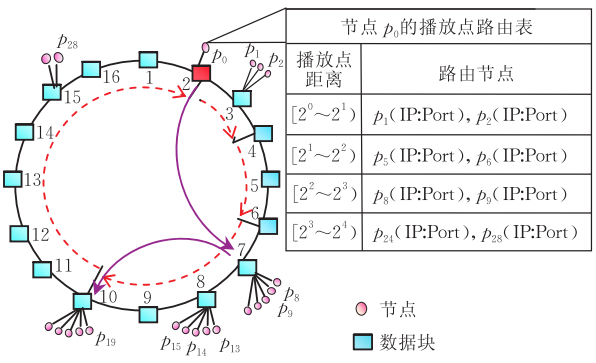


图 2　播放点覆盖网由节点的路由表构建而成

3.2　前缀树覆盖网(Prefix-Tree Overlay)

第 1 阶段查询的结果我们称之为种子节点,该种子节点满足内容匹配要求,在第 2 阶段中,我们的目标是从种子节点出发,找到网络延迟最小的节点子集.实际上,第 2 阶段查询就是在种子节点所属的 swarm 中进行的,因为根据第 1 阶段的查找过程,最后定位的 swarm 就是种子节点所在的节点集合.因此,如何把 swarm 内部的节点索引起来是第 2 阶段的首要任务.因为查询目标是延迟最小,而延迟和 AS 域的 IP 前缀又有密切关系[11],我们的基本想法是用 IP 前缀把同一个 swarm 中的节点索引起来,如图 3(b)所示的索引树.这样,由文献[11]的方法可以获得全局延迟表,根据该表便可得知延迟最小的 IP 前缀并作为查询目标,再根据图 3(b)所示的索引树便可找到属于目标 IP 前缀的节点.具体细节将在第 4 节加以介绍.

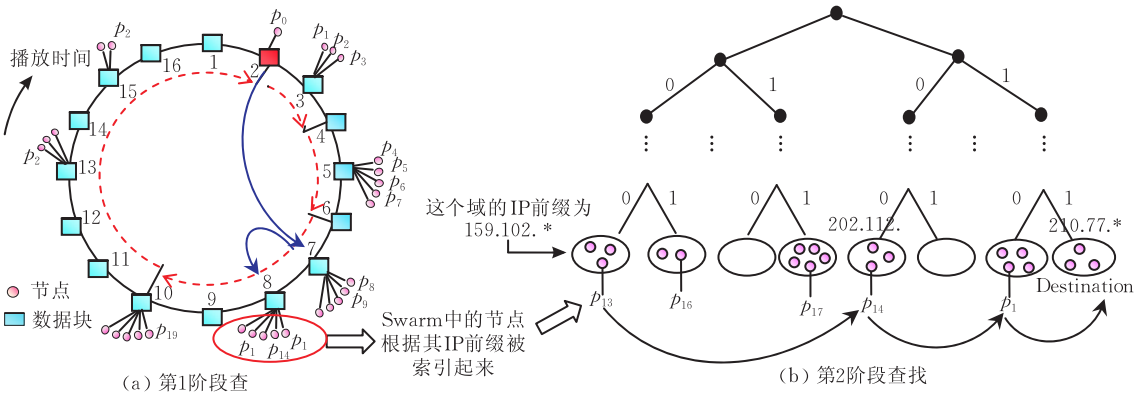


图 3 层次查找的例子

4 Mediacoop 的详细设计

在本节中,首先介绍为何使用网络延迟作为物理性能的衡量标准,以及如何探测全网 AS 域间延迟;接下来将详细介绍层次化索引信息(双层路由表)的构建,并介绍如何使用路由表进行查询;最后讨论系统的维护代价和系统开销.

4.1 节点物理性能的获取

作为交互性较强的应用,P2P 系统对端到端的延迟是很敏感的[11]. 特别是在点播服务中,用户的交互性操作频繁发生(例如启动、跳转、暂停等),如果节点间延迟很大,会导致较长的反应时间. 国际通信联盟 G114 号文件[12]建议,对于大多数交互性应用,以 150 ms 作为物理链路上端到端延迟的上限. 因此,在本文中,采用网络延迟作为衡量节点物理性能的指标,以 150 ms 作为端到端延迟的上限. 虽然端到端延迟可能随时间发生变化,但是十分精确的值对我们来说意义并不大. 我们只需要记录节点间的平均延迟值,选取延迟较小的作为邻居节点. 因此,我们最关心的是如何获得端到端延迟. 实际上,互联网是由多个 AS(Autonomous Systems)域组成,位于同一个 AS 内的节点距离较近,它们对外部 AS 表现出的网络延迟相似. 不同 AS 间的路由方式由边际网关协议(BGP)指定. AS 内部又分为更小的域(Cluster),由 IP 前缀分割开来. 这样只需要获得域间延迟,选择与查询者延迟最小的域中节点作为邻居. 我们使用文献[11]中的方法获得全覆盖网的域间延迟表,这样就可以从该表中获得延迟最小的域作为查询目标. 该延迟探测方法的概括性介绍在下段中给出(更多细节可以参考文献[11]).

图 4 描述了延迟探测的整个过程. 首先,可以从边际网关协议公共信息提供处获取 BGP 路由表及其更新信息,例如从 RouteViews (http://www.routeviews.org) 和 RIPERIS (http://www.ripe.net/projects/ris)均可获取. 由这些路由表,可以构建整个覆盖网的组成图,其组成单位是以 IP 前缀分割的各个域(Cluster). 我们以随机的方式从每个域中选取一个节点作为该域的"代表". 任意两个域之间的延迟是通过使用工具 King[13]测量对应"代表"的延迟获得. 最后得到两个表:(1) IP 与对应的域的映射表(ICMT);(2) 域与域之间的延迟表(CCDT). 由 ICMT,节点可以获知自己所属的域,再由 CCDT 获得与其所属的域延迟最小的目标域. 表 1 给出了 CoolFish 网络 CCDT 的例子. 接下来的核心任务就是如何把这些域组织起来提供高效搜索.

表 1 CoolFish 的 CCDT(时间: 2009.09.02, 15:00)

	延迟/ms			
	159.226.40.*	202.127.200.*	210.72.15.*	…
159.226.40.*	0	31	8	…
202.127.200.*	31	0	58	…
210.72.15.*	8	58	0	…
…				

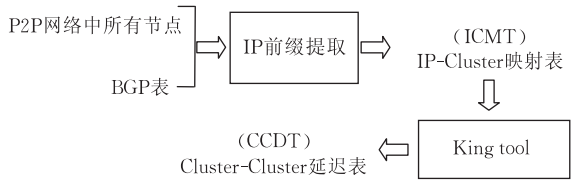


图 4 域间延迟探测示例

在本文方法中,所有的域被组织成二叉树的结构,每个叶子节点代表一个域. 这样,叶子节点的个数 K 和全部域的个数是相等的. 每个叶子节点以其物理地址作为标识(即 IP 前缀),在表现形式上是其对应的前缀码. 叶子节点 1 和叶子节点 2 的距离由他们前缀码的异或运算得出. 在形式上,距离计算如下所示:

$$D_{IPprefix}(n1,n2) = Prefix_{n1} \oplus Prefix_{n2} \quad (1)$$

注意,第 1 阶段得到的目标 swarm 中的节点就

分散在这些叶子节点中，如果节点稀少会导致个别叶子节点为空（如图 3(b)所示），这样的空叶子节点在实际操作中不会被索引起来. 有些人可能会提出质疑，为什么我们使用树形结构来组织 IP 前缀呢？原因其实很简单：二叉树叶子节点的距离计算和 IP 前缀的距离计算是一致的，都是异或操作.

4.2　双层路由表的构建

在 Mediacoop 中，一次查找由两阶段组成，分别在两个覆盖网中进行，相应的每个节点的路由表包含两个子表，分别是播放点路由表和 IP 前缀路由表.

播放点路由表：播放点路由表存储了其它 swarm 播放点的信息. 以节点 p 的播放点路由表为例，该表有 $\log M$（M 是数据块的数量）行，第 i 行（$0\leq i<\log M$）保存着 α 个节点的播放点信息，该 α 个节点与 p 播放距离为 $2^i\sim 2^{i+1}$. 播放距离（$p_1\rightarrow p_2$）的定义如下：顺时针沿播放点覆盖网（即图 1(b)所示的环）$p_1\sim p_2$ 的跳数. 其数学的形式化定义表示为

$$D_{\text{playpoint}}(p_1,p_2)=((PP_{p_2}-PP_{p_1})+M)\bmod M \tag{2}$$

式中，M 是数据块的个数，PP_{p_i} 是节点 p_i 的播放点. 图 5(a)给出了播放点路由表的结构. 每行记录了 $\alpha=2$ 个节点的信息（这里的 $\alpha=2$ 是我们的经验值. 为了保证路由节点的可用性，建议每条路由表项中存储的节点个数大于 1）.

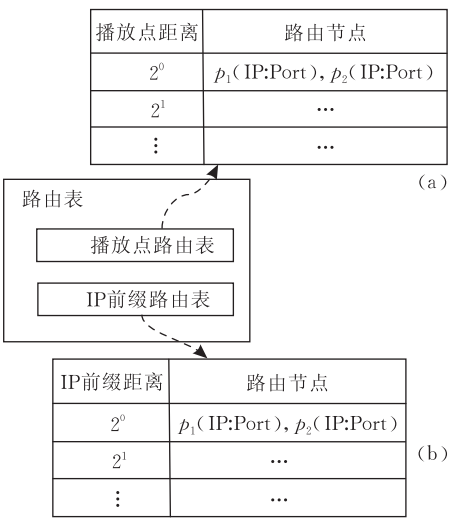


图 5　节点的双层路由表

IP 前缀路由表：以节点 p 为例，其 IP 前缀路由表存储了和自己所在同一 swarm 中的其它节点信息. IP 前缀路由表有 $\log K$（K 是叶子节点的数量）行. 第 j 行（$0\leq j<\log K$）保存着 β 个节点的 IP 前缀信息，该 β 节点与 p 的前缀码距离为 $2^j\sim 2^{j+1}$. 该距离的计算方法如式(1)所示. 图 5(b)给出了 IP 前缀路由表的结构示意图.

4.3　查询过程

内容匹配查询：即在播放点覆盖网中进行的第 1 阶段查询，图 6 给出了该查询算法的伪代码. 查询过程为递归方式，描述如下：当节点 A 收到查询消息，它首先根据式(2)计算自己与目标数据块的距离 D，然后该节点从自己的播放点路由表中选出与 D 最近的节点 B 并把该查询消息转发给 B. 该查询结束的条件为：(1) 发现节点 C 的距离等于 D，此时节点 C 作为种子节点被返回；(2) 已没有更近的节点可以转发，此时当前节点作为种子节点被返回.

```
//计算目标和当前节点播放点距离
D_pp=((PP_p−PP_A)+M)mod M
route_nodes=NULL;
//如果当前节点最近则直接直接返回
if D_pp==0 {
  return Node A;
} else {
  //查询路由表进行消息转发
  Let entry_key=⌈logD_pp⌉;
  //get_value(i)返回第 i 项里的节点
  route_nodes=get_value(entry_key);
  while (route_nodes==NULL
         && entry_key>0) {
    entry_key--;
    route_nodes=get_value(entry_key);
  }
  //找到最近节点
  if (route_nodes!=NULL) {
    forward to route_nodes;
  }
}
//如果找不到路由节点，返回当前节点
If (route_nodes==NULL) {
  return Node A;
}
```

图 6　内容匹配查询算法伪代码（Node A 收到内容查找消息后进行路由的过程）

质量匹配查询：即从第 1 阶段获得的种子节点出发，在 IP 前缀覆盖网中进行的第 2 阶段查询. 查询目标是发起者 p 的 CCTD 中延迟最小的域，然后发起者 p 向种子节点发送查询消息. 种子节点从自己的 IP 前缀路由表中找到距离目标最近的节点，并向其转发查询消息. 转发节点递归的进行上述操作直到没有更近的节点进行转发（注意这里和上述结束条件不同，我们的目的是获得更多的节点）. 在查询过程中，任意符合要求（IP 地址和目标 IP 前缀一致）的节点都将被返回. 如果最后仍没有返回任何节点，那么发起者 p 会要求种子节点随机的返回在同

一 swarm 中的节点即可，因为这些节点虽然没有最低的延迟，但是在内容上都是满足要求的. 图 7 给出了该查询算法的伪代码.

```
//计算目标和当前节点前缀距离
D_IPprefix = Prefix_p ⊕ Prefix_B;
route_nodes = NULL;
//如果查询就是当前节点直接返回
if D_IPprefix == 0 {
   return Node B;
} else {
   //查询路由表进行消息转发
   Let entry_key = ⌈logD_IPprefix⌉;
   route_nodes = get_value(entry_key);
   if (route_nodes != NULL) {
      forward to route_nodes;
   } else {
      forward to C∈RouterTable s. th.
      D_IPprefix(p,C) is minimal;
   }
}
//如果找不到路由节点，返回当前节点
If (route_nodes == NULL) {
   return Node B;
}
```

图 7 质量匹配查询算法伪代码(Node B 收到质量查找消息后进行路由的过程)

一个查询实例：图 3 给出了从发起者 p_0 开始的整个查询过程. 如图 3(a)所示，p_0 发起查询消息，目标是找到能够提供 8 号数据块的节点. p_0 首先计算自己和 8 号数据块的距离：$D_{\text{playpoint}}(p_0, block8)=6$，然后向最近的邻居 p_8 发出“内容匹配”查询消息. 以同样的方式，p_8 计算出自己和目标数据块的距离为 $D=1$ 然后发现节点 p_{13} 是属于目标 swarm 的. 因此，p_{13} 作为种子节点被返回. 接下来，p_0 从自己的域间延迟表中找到目标 IP 前缀 210.77.* 并发起第 2 阶段查询. p_0 向种子节点 p_{13} 发出 IP 前缀查询消息(如图 3(b))，收到查询消息后，p_{13} 从自己的 IP 前缀路由表中发现 p_{14} 最近并把消息转发给 p_{14}. 以同样的方式，p_{14} 把查询消息转发给 p_{15} 最后找到了位于 210.77.* 内的目标节点.

4.4 节点加入及系统维护

当节点 p 加入网络的时候，它首先联系一个业已存在的节点 J. p 以 J 的路由信息初始化自己路由表，然后节点 p 以自己为目标(播放点和 IP 前缀)进行一次完整的查询操作. 完成查询之后，节点 p 获得了最近的节点网络中距离自己最近的节点 Y，最后，p 使用 Y 的路由表内容来刷新自己的路由表. 注意，最后这次刷新的作用是：因为 Y 已经是距离 p 最近的节点，对于 p 来说就是如何获取比 Y 更远的节点信息，而 Y 的路由表刚好可以提供. 在路由表刷新的过程中，p 也把自己的信息插入到了其它节点的路由表中.

当节点 p 进行跳转、暂停、停止等 VCR 操作的时候，p 会通知邻居节点(路由表中的全部节点)自己新的播放点，同时邻居会更新相应的信息. 在我们的方法中，域间延迟探测程序运行在单独探测服务器上，并间隔地更新 ICMT 和 CCDT 以提供最近的数据. 有一点需要注意的是，CCDT 记录的是平均延迟. 当一个节点加入系统，它首先是向探测服务器请求 ICMT 和 CCDT 并以服务器的更新频率进行数据更新(在我们的系统中，更新频率是 30 min).

5 理论分析及实验

5.1 理论分析

在本节中，我们分析 Mediacoop 的查询效率. 因为该方法涉及到两阶段查询，可以建立如下分析模型：

$$P(M,K) = P_{\text{FirstStage}}(M) + P_{\text{SecondStage}}(K),$$

其中，$P(M,K)$ 是总的查询跳数，$P_{\text{FirstStage}}(M)$ 和 $P_{\text{SecondStage}}(K)$ 分别是第 1 阶段和第 2 阶段的查询跳数，M 是数据块的数量，K 是域(即叶子节点)的数量. 首先分析第 1 阶段的查询效率 $P_{\text{FirstStage}}(M)$. Mediacoop 是一个结构化的搜索方法，其搜索过程类似传统的 DHT 方法，例如 Chord. 但是，我们使用的是播放点而非节点标识来建立索引. 因此，本文方法的效率不是传统 DHT 方法的 $O(\log N)$，而是播放点覆盖网对应的环上的播放点数量 M. 但是如果节点很少会导致数据块为空(即数据块不对应节点)，比如当节点数少于数据块数量的时候，这种情况下 $K=N$. 因此查找效率应该取二者的最小值，即

$$P_{\text{FirstStage}}(M) = \min\{O(\log M), O(\log N)\},$$

这里，N 是网络中全部节点的数量. 总的来说，对于流行的 P2P 系统，节点的数量是十分庞大的，而一部影片的数据块的数量确实相当有限的. 举例来说，根据我们实际经验，对于时长为 2 h 的电影，720 个数据块就足矣了. 也就是说，$M \ll N$，因此我们有

$$P_{\text{FirstStage}}(M) = \min\{O(\log M), O(\log N)\} = O(\log M).$$

在第 2 阶段，搜索过程实际上是在二叉树上的折半查找. 因此，第 2 阶段的查找效率和折半查找的时间复杂度一致：

$$P_{\text{SecondStage}}(K) = O(\log K).$$

这里，K 是域的数量. 第 2 阶段的搜索过程在一个较小的范围中进行的，即第 1 阶段的目标 swarm. 而

平均来看，全部节点是均匀分布在每个数据块中的，即每个 swarm 的节点个数 $n=N/M$. 这样，K 小于或等于 n，K 取最大值 n 的条件是每个域正好只有一个节点. 因此，Mediacoop 总的搜索效率为

$$P(M,K)=P_{\text{FirstStage}}(M)+P_{\text{SecondStage}}(K)\leq O(\log M)+O(\log N/M)=O(\log N),$$

即

$$P(M,K)\leq O(\log N).$$

也就是说，在不低于传统 DHT 一次查询的效率下，可以进行两阶段查询，既满足了内容上的要求，也在物理性能上得以提升.

5.2　评价指标及实验参数

P2P-VoD 的评价指标分为两个方面：一是用户体验，二是系统扩展性. 前者主要指启动时间、跳转时间和播放连贯率[5,9]；后者是服务器压力和网络开销[5,9]. 在本文中，我们不仅对以上几点均进行验证. 还对搜索跳数进行了对比.

我们把 Mediacoop 分为两个版本进行对比，第 1 个是单纯的内容搜索，不具备延迟探测（简称 Mediacoop(no-DA)），即第 2 阶段没有搜索，取而代之的是使用 Gossip 协议定位节点；第 2 个版本是两阶段都有（简称为 Mediacoop(DA)）. 为了和目前较为流行的方法进行对比，我们实现了具有代表性的基于 DHT 的 P2P-VoD 系统 PROP[5]. 限于本文讨论的内容，我们没有实现 PROP 中的中心服务器功能. 但我们与传统的“缓存转发”系统 P2VoD[9] 进行对比①.

我们的对比试验是在 NS2 模拟器上进行的. 电影时长设置为 3600 s，码率为 500 Kbps，一个数据块对应的播放时长为 10 s. 使用拓扑结构生成器 GT-ITM[14] 生成了典型的 transit-stub 网络，其包含了 860 个路由器，之后随机选择 100 个 stub 节点作为域的分隔节点②. 每个 stub 节点之间的延迟为 10 ms～60 ms. 同时生成 8000 个节点以均匀分布的方式依附于每个域上. 整个实验分为 12 组进行，对应的节点数量从 100～8000 不等. 节点的加入以指数递减的方式进行[15]：

$$\lambda(t)=\lambda_0 e^{-\frac{t}{\tau}},$$

式中，λ_0 是初始加入速率，τ 为扩散参数. 相应地，设置节点平均加入时间间隔为 5 s，平均在线时间为 1800 s. 下载带宽为 1 Mbps，上行带宽可以支持 2 个并行流. 启动和跳转时的缓冲数据量为能够播放 5 s 的数据. 模拟试验程序运行在超级计算机 Dawning 4000A 上，总共运行时间大于为 4 d.

5.3　实验结果

（1）平均跳数. 在本项指标中，因为验证的是搜索到目标所进行的跳数，是针对结构化搜索方法的，因此并不涉及 P2VoD. 图 8 给出了 12 组实验对应的平均跳数的实验数据. 从结果可以看到，PROP 体现了典型的基于 DHT 方法的“$\log N$”法则，而 Mediacoop 两个版本的表现均强于 PROP. 虽然我们看到 Mediacoop(no-DA) 跳数少于 Mediacoop(DA)，那是因为前者没有第 2 阶段的搜索过程.

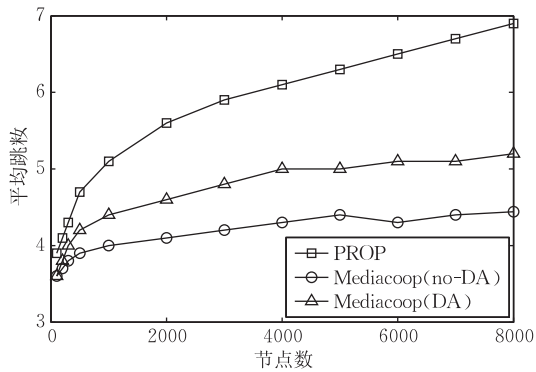


图 8　平均跳数随节点数的变化

（2）网络开销. 网络开销主要是指控制信息的数量，因为 P2VoD 是树形结构，并非网状，因此在本项指标中也不考虑 P2VoD. 如图 9 所示，PROP 在三者的比较中表现最差，因为它要不断地发布和删除内容信息，导致了大量的控制信息. Mediacoop (no-DA) 的网络开销虽然强于 PROP 但是要比 Mediacoop(DA) 大得多，因为它在第 2 阶段使用了 Gossip 协议，也带来了大量的控制信息. 相比之下，Mediacoop(DA) 能够减少 40%～70% 的网络开销.

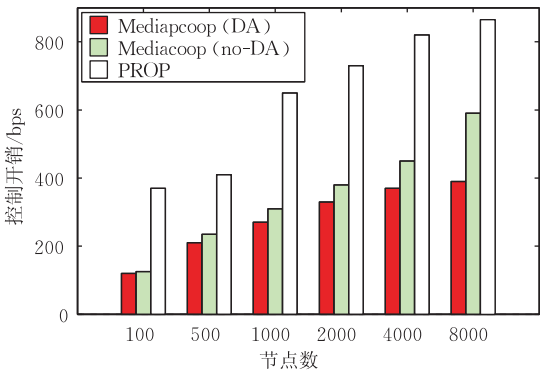


图 9　控制信息开销随节点数的变化

① P2VoD 没有跳转功能，在对比实验中我们对其加入了此项功能.

② GT-ITM 本身不能够区分 transit 节点和 stub 节点，我们在 GT-TIM 源码基础上开发了工具进行分离.

(3) 服务器压力. 在实验中,有一个具备1000 Mbps上能力的内容服务器. 如果一个节点没有及时收到其请求的数据,它就立即向内容服务器请求数据. 图 10 显示了随着节点数量的变化,内容服务器压力的状况. 对于 Mediacoop(no-DA),它的压力要大于 Mediacoop(DA),因为 Mediacoop(no-DA)没有第 2 阶段搜索,得到的节点延迟较大,造成了请求数据不能及时到达. 对于 PROP 随着数据缓冲区的不断更新,被抛弃的数据没有来得及更新,对这些数据的请求自然不能得到满足,造成了更大的服务器压力. 而在 P2VoD 中,服务器压力是最大的并且是线性增长的. 这是因为它是树形的组织结构,而上层节点的离开会造成它所有孩子节点缺失数据.

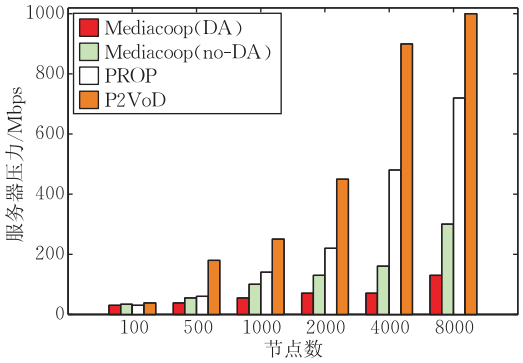


图 10 服务器压力随节点数的变化

(4) 播放连贯率. 图 11 显示了网络规模为 4000 个节点时,播放连贯率随时间的变化. Mediacoop 的表现要远远好于 RROP 和 P2VoD,其原因和服务器压力一节中的相同. 除此之外,PROP 这种基于 DHT 的方法必须要等到整个数据块都接收完成后才能发布信息,这样势必造成数据共享效率的低下.

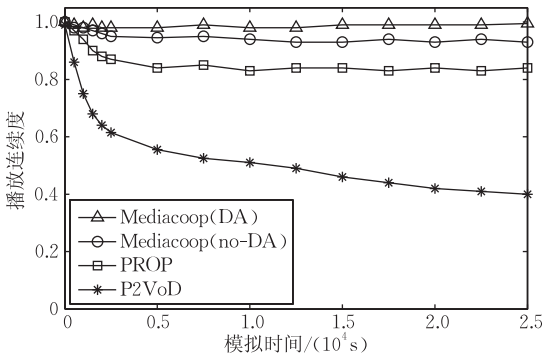


图 11 播放连贯度变化情况

(5) 启动和跳转时间. 这两个指标涉及到两部分的性能表现:① 查找速度;② 请求数据的速度. 实际上,前者就是第 1 项指标,搜索的平均跳数;而后者决定于搜索到的节点的质量. Mediacoop(DA)能够找到最近的节点,从而保证了请求的数据能够快速到达. 图 12 和图 13 给出了实验结果. 可以看到,对于 5 s 的数据缓冲区,Mediacoop(DA)平均只需要大于 3.5 s 的启动时间和 2 s 的跳转时间.

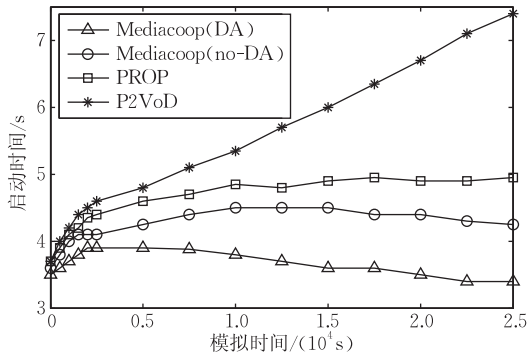


图 12 启动时间变化情况

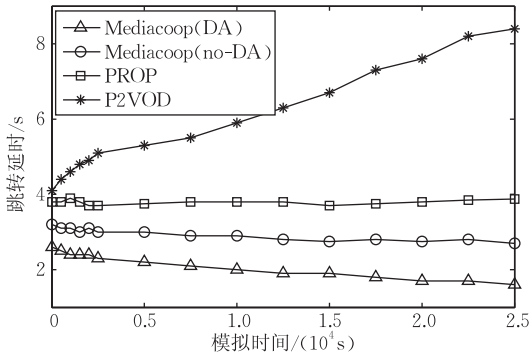


图 13 跳转延迟变化情况

5.4 真实系统上实现:CoolFish

我们实现了一个真实的P2P-VoD系统CoolFish①,并且已初具规模. CoolFish 的内容服务器位于中国科技网(CSTNet)内. 从2008年 10 月到 2010 年 6 月,CoolFish 的访问用户数已超过 4 000 000,最近的日访问人数已经超过 7000,在中国境内的用户分布超过 28 个省市.

本文中提到的层次化搜索算法已经在 CoolFish 实现,系统用 C++编写,总代码量超过 80 000 行. 表 1 显示了 CoolFish 系统的 CCDT. 我们下一步将在 CoolFish 系统中对 Mediacoop 进行更深入的实验比较.

6 结论及下一步研究

节点搜索对于 P2P-VoD 系统是十分重要的,而最理想的搜索策略是既能满足内容匹配,又可以实现节点质量匹配. 本文中提出的层次化搜索模型

① http://www.cool-fish.org

Mediacoop 已经初步达到了上述目标. 它使用了层次化的双结构模型，在内容查找阶段可以避免传统方法中大量的网络开销，同时又可以查找到具有最低网络延迟的节点集合. 在理论分析上，证明了在小于 $O(\log N)$ 的情况下，就可以完成两个阶段的搜索过程. 从实验结果可以看到，Mediacoop 在用户体验以及系统扩展性上均好于其它方法. 同时，我们的方法在真实系统 CoolFish 中的实现也体现了 Mediacoop 的有效性. 下一步，我们将在 CoolFish 系统中进行更深入的对比分析.

参 考 文 献

[1] Huang Yan, Fu Tom Z J, Chiu Dah-Ming, Lui John C S, Huang Cheng. Challenges, design and analysis of a large-scale P2P-VoD system//Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication (SIGCOMM'08). ACM, New York, NY, USA, 2008: 375-388

[2] Pucha H, Andersen D G, Kaminsky M. Exploiting similarity for multi-source downloads using file handprints//Proceedings of the 4th USENIX Conference on Networked Systems Design Implementation (NSDI'07). Berkeley, CA, USA, 2007: 2-2

[3] Hefeeda M, Habib A, Botev B, Xu D, Bhargave B. PROMISE: Peer-to-peer media streaming using CollectCast//Proceedings of the 11th ACM International Conference on Multimedia (MULTIMEDIA'03). Berkeley, CA, USA, 2003: 45-54

[4] Zhou Xiao-Bo, Ippoliti Dennis, Zhang Li-Qiang. Fair bandwidth sharing and delay differentiation: Joint packet scheduling with buffer management. Computer Communications, 2008, 31(17): 4072-4080

[5] Guo Lei, Chen Song-Qing, Ren Shan-Si, Chen Xin, Jiang Song. PROP: A scalable and reliable P2P assisted proxy streaming system//Proceedings of the International Conference on Distributed Computing and Systems (ICDCS'04). Washington, DC, USA, 2004: 778-786

[6] Liao Chi-Shiang, Sun Wen-Hung, King Chung-Ta, Hsiao Hung-Chang. OBN: Peering for finding suppliers in P2P on-demand streaming systems//Proceedings of the 12th International Conference on Parallel and Distributed Systems—Volume 1 (ICPADS'06). Washington, DC, USA, 2006: 235-242

[7] Cheng Bin, Jin Hai, Liao Xiao-Fei. Supporting VCR functions in P2P VoD services using ring-assisted overlays//Proceedings of the IEEE International Conference on Communications (ICC'07). Glasgow, Scotland, 2007: 1698-1703

[8] Guo Yang, Suh Kyoungwon, Kurose Jim, Towsley Don. P2Cast: Peer-to-peer patching scheme for VoD service//Proceedings of the 12th International Conference on World Wide Web (WWW'03). New York, NY, USA, 2003: 301-309

[9] Do T T, Hua K A, Tantaoui M A. P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment//Proceedings of the 2004 IEEE International Conference on Communications. Paris, France, 2004, 3: 1467-1472

[10] Stoica I, Morris R, Karger D, Kaashoek M F, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications//Proceedings of the ACM SIGCOMM'01—Computer Communication Review. San Diego, California, USA, 2001: 149-160.

[11] Ren Shan-Si, Guo Lei, Zhang Xiao-Dong. ASAP: An AS-aware peer-relay protocol for high quality VoIP//Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06). Washington, DC, USA, 2006: 70

[12] One way transmission time. ITU-T Recommendation G. 114, May 2000

[13] Gummadi K P, Saroiu S, Gribble S D. King: Estimating latency between arbitrary internet end hosts//Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW'02). ACM, New York, NY, USA, 2002: 5-18

[14] Zegura E W, Calvert K L, Bhattacharjee S. How to model an internetwork//Proceedings of the 15th Annual Joint Conference of the IEEE Computer and Communications Societies Conference on The Conference on Computer Communications—Volume 2 (INFOCOM'96). Washington, DC, USA, 1996, 2: 594-602.

[15] Guo Lei, Chen Song-Qing, Xiao Zhen, Tan En-Hua, Ding Xiao-Ning, Zhang Xiao-Dong. Measurements, analysis, and modeling of BitTorrent-like systems//Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC'05). Berkeley, CA, USA, 2005: 4-4

ZHANG Tie-Ying, born in 1982, assistant professor. His research interests include computer networks, distributed computing, peer-to-peer systems, multimedia networking, and network security.

LIU Yue, born in 1971, Ph. D., associate professor. Her major research area is information retrieval and social network.

ZHONG Yun-Qin, born in 1984, Ph. D. candidate. His research interests focus on massive data management, spatial-temporal databases, and spatial cloud computing.

CHENG Xue-Qi, born in 1971, professor, Ph. D. supervisor. His research interests include network science, web search & data mining, and P2P & distributed computing.

Background

As Peer-to-peer (P2P) technologies have obtained enormous success in content delivery, more and more video streaming providers have paid attention to developing P2P streaming applications to reduce server costs and accelerate user downloading. In P2P architecture, clients' resources (bandwidth, CPU, storage) are used to power the P2P system while optimizing network resources utilization. As P2P networks do not require any special servers or routers, the cost of such solutions is appealing. P2P multicasting is an elegant alternative to CDN infrastructure which each end-host (peer) may act as a potential server for other peers. This avoids dedicated replication servers altogether. The approach is self-scaling, as the number of peer "servers" and peer clients increases at the same rate, hence it avoids the bottleneck of a central server (or dedicated replication server). The approach, in principle, would allow a highly dynamic support of changing multicast demand at very low cost.

P2P streaming focuses on real-time video streaming applications, which include both live and on-demand streaming. These systems are harder to deploy due to the real-time playback requirement at the receiver end. Many technologies exist for real-time video delivery. Broadcast video as used in TV is very good for delivering a limited number of streams to a very large audience. Point-to-point delivery is currently used for VoD and interactive video, as well as much of Internet video streaming, and can support a small audience with a large number of streams. The middle ground is covered by multicast delivery. Multicast delivery is very flexible and can enable a large number of senders to deliver content to any number of receivers.

IP multicast was the first solution to provide multicast functionality in the Internet. It put forth an ambitious vision to support all multicast functionality within the routers in the network and proposed a powerful abstraction to applications where a group address identifies a multicast group and any host can send a message to a group by simply sending to the group address. However, due to many technical and marketing reasons, it is still far from being widely deployed. Instead, P2P streaming not only provides the same function of IP multicast but also off loads the central servers.

This work is supported by the National High Technology Research and Development Program (863 Program) of China under Grant No. 2006AA010105-02, the National Basic Research Program (973 Program) of China (2004CB318109) and the National Natural Science Foundation of China under Grant No. 60933005.

面向互动式网络场景再现的流速控制系统与方法

褚伟波[1] 管晓宏[1),2)] 蔡忠闽[1] 陶 敬[1]
[1)](西安交通大学智能网络与网络安全教育部重点实验室 西安 710049)
[2)](清华大学自动化系 清华信息科学与技术国家实验室 北京 100084)

摘 要 互动式网络场景再现是一种重要的真实网络流量产生方法.然而,由于流量产生过程的复杂性,基于精确的数学模型对该过程产生流速进行控制是一个较为困难的新问题.文中设计实现了一个面向互动式网络场景再现的流速控制系统,并将网络场景再现过程中的流速控制问题转化为目标跟踪控制问题进行求解.该系统采用一种基于函数近似器的流速控制方法,利用函数近似器对系统的输入输出关系进行描述,通过动态调整系统输入流量来对回放过程输出流量进行跟踪.最后,利用真实网络流量实验考察了文中系统和方法在不同丢包、传输延迟以及会话阻断环境下的实际控制效果,并从收敛时间、产生输入输出、控制误差等角度对系统的控制性能进行了分析.

关键词 互动式网络场景再现;流量回放;流速控制;函数近似器
中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2012.01485

System and Method for Real-Time Volume Control in Reproducing Network Scenario

CHU Wei-Bo[1)] GUAN Xiao-Hong[1),2)] CAI Zhong-Min[1)] TAO Jing[1)]
[1)](*MOE Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an* 710049)
[2)](*Department of Automation, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Interactive network scenario reproduction is an important method for generating realistic and responsive traffic workloads. However, due to the complex traffic generation mechanisms, controlling traffic volume based on accurate mathematical models in interactive network scenario reproduction becomes a challenging problem. In this paper, we design and implement a volume control system for interactive network traffic replay, and formulate the volume control task as a target tracking problem where the output traffic volume is regulated through adjustment of input traffic volume. We then adopt a function approximator (FA) in our system to generate the desired input volumes. The FA characterizes the mapping that takes system measurements as input and directly produces controls as output. Meanwhile, simultaneous perturbation stochastic approximation (SPSA) algorithm is employed to adaptively estimate the parameters in control. To validate the system and the method we have implemented it and conducted experiments with actual traffic traces. Empirical studies show that our system can track target output volumes effectively under a wide range of network conditions. The performance of the control system is further investigated in terms of its convergence time, generated input/output traffic volume and target tracking error.

Keywords interactive network scenario reproduction; traffic replay; volume control; function approximator

---

收稿日期:2011-02-24;最终修改稿收到日期:2012-05-14.本课题得到国家自然科学基金(60921003,61175039,61103241)、中央高校基本科研业务费专项资金(xjj20100051)资助.**褚伟波**,男,1982 年生,博士研究生,主要研究方向为网络流量分析与建模、网络流量还原回放和网络设备测试评估.E-mail: wbchu@sei.xjtu.edu.cn.**管晓宏**,男,1955 年生,博士,教授,主要研究领域为计算机信息安全、系统优化与调度.**蔡忠闽**(通信作者),男,1975 年生,博士,副教授,主要研究方向为计算机网络安全和入侵检测.E-mail: zmcai@sei.xjtu.edu.cn.**陶 敬**,男,1978 年生,硕士,工程师,主要研究方向为网络场景仿真与再现.

1 引 言

网络场景再现[1-15]是近年来逐渐受到重视的一个新兴研究方向,其核心问题是再现符合真实网络场景的流量.网络场景再现研究对于网络安全、网络管理、网络设备测试评估等领域具有重要意义.当前网络流量再现方法主要有两种,即基于仿真的流量再现方法[1-6]和基于真实网络流量还原回放[7-15]的方法.基于仿真的流量再现方法能够通过调整模型参数来灵活产生需要的流量(如 Spirient's Smartbits①, IXIA's Traffic Generator②).然而,由于实际网络流量异常复杂,基于仿真方法产生的流量往往与实际流量相差较大(如数据包大小、IP 地址和端口分布、应用类型等),利用仿真流量对系统进行测评得到的结果并不足以代表系统在真实网络环境下的功能和性能.在实际情况中也经常发现系统在仿真流量环境下功能和性能均表现正常而在实际网络中却出现各种问题;相反地,网络流量回放方法则是将真实网络中的流量进行捕获记录[16]并还原回放到测试网络的一种流量产生方法.由于采用了实际网络中捕获得到的流量,流量回放方法可以在数据包层面(包大小、包内容、包时序等方面)还原实际网络场景、达到直接考察被测系统在真实网络中的功能和性能的目的.

根据流量回放机制的不同,流量回放方法又分为直接回放[7-11]和互动式回放[12-15]两种,分别对应直接的流量再现以及互动式流量再现.直接流量再现方法(如 TCPReplay③、TCPivo[7]、Monkey[8]、Tomahawk④)简单地将捕获到的数据包注入到测试网络来对网络场景进行还原,主要用于测试 IDS(入侵检测系统)、Sniffer(嗅探器)等各类被动型旁路设备;而互动式再现方法则在流量产生过程中仿真实现了 TCP/IP 协议栈并严格基于网络协议规范来产生数据报文,有效解决了直接流量再现方法在测试 Firewall(防火墙)、IPS(入侵阻断系统)等各类串接型安全设备过程中产生大量违反协议语义数据包的问题[12,14](如采用直接再现方法在测试 Firewall 的过程中,SYN_ACK 报文会在 SYN 报文被 Firewall 阻断的情况下仍然被回放到测试网络中,从而极大地影响到测试结果的准确性).由于在流量再现过程中考虑了数据包之间的协议语义并模拟会话两侧的行为来产生流量,因而互动式网络流量再现方法可以通过产生流量的变化反映被测设备的作用,达到考察被测设备串接在实际网络时的功能和性能的目的.当前,互动式网络流量再现方法已经成为了对 IPS、Firewall 等各类串接型网络设备进行测试评估的最新方法.

在网络场景再现过程中,对回放产生流量的流速进行控制具有重要意义.流速控制主要有两个方面,一方面是产生尽可能大的流速的流量,另外一方面是产生满足用户设定流速的流量.产生高速流量可用于考察被测系统在高速网络环境下的功能和性能,而产生用户指定流速的流量在测试过程中同样非常重要.在实际网络管理和设备测试过程中经常发现被测系统的一些深层次故障和问题往往是在流速突然发生变化的时候或在一些特定流量水平上才体现出来.所以,流速控制在流量再现过程中便成为了一个非常重要的环节.

在本文中,考虑在互动式网络场景再现过程中产生满足目标网络环境且具有指定流速的流量的问题.在直接网络场景再现过程中,由于数据包被直接注入到测试网络,在流量产生过程中对流速进行控制是一个相对较为简单的问题,而在互动式场景再现过程中对流速进行控制则较为复杂,其主要原因如下:

(1)作为系统输入的网络流量的不稳定性.大量研究表明真实网络流量是一个非常复杂的非平稳过程[17-19],各种特征诸如流量大小、数据包大小,IP 地址和端口分布等均是随时间动态变化的.特别是由于互动式流量回放过程是基于流来产生测试流量,流量中各种与流相关的特征诸如流的长短、流的持续时间和流的到达速率等均会影响到系统性能;

(2)测试环境的作用.流量产生过程中测试环境的丢包、会话阻断和数据包传输延迟等作用会影响到回放时流的状态,进而影响到回放系统的性能;

(3)回放系统自身的动态特性.互动式流量回放系统通常是由软件实现,因此操作系统中的进程调度、磁盘 IO、网络 IO 等随机因素也会影响到流量回放过程.

考虑到互动式场景再现过程的复杂性,对其建立精确的数学模型并基于模型进行流速控制便成为

① http://www.spirent.com/
② http://www.ixiacom.com/
③ http://tcpreplay.synfin.net/trac/
④ http://tomahawk.sourceforge.net/

了一个非常困难的过程.由于不知道系统的数学模型,一些传统基于模型的控制方法(如 H 无穷最优控制、模型参考控制等等)便无法适用.

在文中,我们设计实现了一个面向互动式网络场景再现的流速控制系统,并将网络场景再现过程中的流速控制问题转化为目标跟踪控制问题进行求解.所设计的系统采用一种基于函数近似器的流速控制方法,利用函数近似器对系统的输入输出关系进行描述,通过动态调整系统输入流量来对回放过程输出流量进行跟踪,并同时通过并行扰动随机估计理论[20]对近似器中的参数进行实时估计.该系统的最大优点在于可以在并不知道系统模型的情况下仍然对复杂过程进行有效控制.

我们采用真实网络流量对所提出的流量控制系统和方法在不同网络环境下的实际控制效果进行了分析与验证.实验结果表明,本文所设计实现的流速控制系统可以精确地对回放过程产生的流速进行控制,在测试网络丢包率达到 10%以及会话阻断率达到 30%等极端情况下,系统仍然能够达到很好的控制效果.此外,还从控制过程的收敛时间、产生输入输出以及控制误差等角度对该系统的控制性能进行了分析.

本文第 2 节介绍相关研究工作;第 3 节介绍互动式网络场景再现过程流速控制问题;第 4 节对影响互动式网络场景再现过程产生流速的因素进行分析;第 5 节介绍基于函数近似器和并行扰动随机估计理论的流速控制系统和方法;第 6 节给出用真实网络流量对所提系统和方法实际控制效果进行实验验证与分析的结果;第 7 节对全文工作进行小结并对未来进行展望.

2　相关研究工作

基于真实网络流量的场景再现研究工作始于 2001 年左右.最早研究人员在 sourceforge 上建立了 TCPDump① 和 Libpcap② 等开源项目,用于开发在网络中能够实时捕获数据报文的系统.在这之后很大一部分研究开始转向流量还原回放方面.文献[7]介绍了 TCPivo——一个 Linux 平台下的高性能数据包回放系统,并提出了一些用于提高数据报文回放效率的系统实现方法;Monkey[8] 是加州大学和 Google 共同开发出来用于测试服务器端性能的流量回放系统,分为 Monkey see 和 Monkey do 两个功能模块,Monkey see 用于在服务器端一侧捕获服务器与客户端交互的流量,并提取包括数据报文间隔、网络传输延迟、丢包率等参数信息;而 Monkey do 用于模拟客户端和传输网络行为,根据接收到的服务器端发送过来的数据包来产生(回放)客户端的流量,以此考察服务器端的参数变化(如不同的缓存大小)对于网络服务性能的影响.文献[9-11]考察了大规模网络流量并行回放中的流分割和回放质量评估问题.

上述研究工作主要定位于对捕获流量进行单向回放.在 TCPReplay 和 Tomahawk 中,研究人员开发出用于测试各类串接型设备的数据报文回放系统.该类系统采用了流量的双向收发机制,即利用两个测试接口来回放捕获的双向流量,每个测试接口用来回放一个方向的流量并接收来自另外一个方向的流量;被测设备串接在两个测试接口之间,回放系统通过对比接收的流量和回放的原始流量来考察串接设备对于流量施加的作用.

为了解决流量直接回放方法在测试 IPS、Firewall 等各类串接型安全设备过程中产生大量违反协议语义数据包的问题,在 2005 年加州大学的 Hong 和 Wu 首次提出了互动式的网络流量场景再现方法,并开发了原型系统 TCPOpera[12].在这种新型的流量回放方法中,除了采用流量的双向收发机制外,回放系统还模拟实现了 TCP/IP 协议栈,为每个被回放 TCP 流会话维护状态.在流量回放时,回放系统严格根据 TCP/IP 协议,采用基于状态的方法来回放 TCP 流量.由于在流量产生过程中考虑了数据包之间的协议语义并模拟会话两侧的行为来产生流量,互动式网络流量回放方法可以产生满足协议语义的流量,并通过产生流量的变化反映被测设备的作用,达到考察被测设备串接在实际网络时的功能和性能的目的.文献[13]对互动式流量回放测试系统的具体实现作了详细的介绍.

也有学者将仿真方法和流量回放技术相结合来产生目标流量,如文献[21-22].

本文作者所在研究小组自 2007 年开始对互动式网络场景再现方法进行了相关研究.在文献[14-15]中,作者基于 TCP 协议的确认机制和停等特性(在流量回放时体现为收发平衡现象),提出了一种基于收发平衡和状态判定相结合的新的 TCP 流量回放

① http://www.tcpdump.com/
② http://sourceforge.net/projects/libpcap/

方法.通过在回放 TCP 流量时将满足收发平衡条件的数据包优先发送出去,所提方法能够在保持 TCP 流量回放语义的前提下,有效减少数据包发送时的状态判定开销,从而提升回放性能.而本文的研究工作则是关注 TCP 流量回放过程在受到丢包、传输延迟和会话阻断等网络环境作用时的流速控制问题.因此,虽然两者的研究对象均为互动式流量回放过程,但是关注点不同.

与本文研究工作最为接近的是加州大学的 Vishwanash 和 Vahdat 的工作,他们开发的 Swing[23] 是一个将网络模型和真实流量数据相结合的流量产生系统.具体地,Vishwanash 和 Vahdat 认为产生真实网络流量需要一个层次化模型,从上到下分别对应用户层、会话层、链接层和网络层.他们将产生流量的模型参数分成两类,一类是流量特征参数,包括应用层协议分布、会话大小、会话时间间隔、链接数目、链接到达速率等;另外一类是网络特征参数,包括数据包丢包率、传输延迟和链路带宽.从捕获流量中提取出这些参数之后,Vishwanash 和 Vahdat 通过搭建模拟环境(包括两端的流量产生器和中间的网络仿真器)来产生流量.具体产生流量的方法是在流量产生器中设置链接产生时间、链接大小、链接持续时间等相关流量特征参数,而在网络仿真器中设置丢包率、传输延迟等网络参数.与 Swing 系统相比,本文工作虽然也是通过搭建模拟环境来产生流量,但存在如下不同之处:

(1)本文工作通过在模拟环境中回放捕获的真实流量来产生用户需要的流量,无需对真实流量中的特征参数进行提取.

(2)本文工作通过模拟 TCP/IP 协议栈来回放捕获的网络流量,产生的数据是真实的网络流量(数据包内容与原始网络流量相同).而 Swing 系统则是通过在流量产生器中建立真实 TCP 链接来传输一定大小人工仿真数据的方法来产生流量.因此,两者产生流量的机理不同.Swing 系统更多地关注产生流量的特征参数符合真实网络特性,对流量中传输内容并不关注.而本文工作则关注流量内容的真实性.

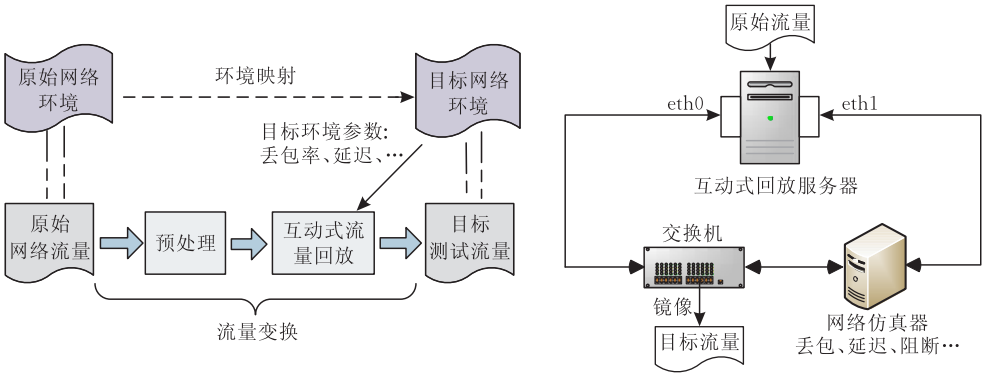
(3)本文主要研究回放过程在受到传输网络作用的条件下的流速控制问题,而 Swing 系统则主要解决产生满足网络环境特征的流量(产生流量的特征参数分布符合真实网络流量)的问题.

3 互动式网络场景再现流速控制

3.1 基于互动式场景再现的流量产生方法

除了用于设备测试,互动式网络场景再现还可用来当作真实网络流量的产生机制,用于产生目标网络环境下符合协议语义的真实流量.由于实际网络流量通常是在某个特定时间特定网络环境下捕获的,而对设备进行测试时往往又需要一些具有指定网络环境的流量(比如需要一定丢包率和网络延迟环境的真实流量,用来当作背景流量对 IDS 等设备进行功能和性能测试,而实际这样的网络环境却不容易获得).在这种情况下,就需要对从实际网络中捕获到的真实流量进行变换(从流量捕获时的原始网络环境映射到目标网络环境),使之产生目标网络环境下满足协议语义的流量.由于网络流量的复杂性,直接基于原始流量产生目标流量的仿真方法很容易破坏产生流量的协议语义,而互动式流量回放方法具有保持流量协议语义的特性,因此可用来产生指定网络环境下的真实流量.

图 1 是基于互动式流量回放过程产生目标流量的方法示意图.如图 1(a)所示,在具体应用该方法之前,一般需要先对原始流量进行处理(比如去除重



(a)过程示意图 (b)系统示意图

图 1 基于互动式流量回放过程产生目标流量的方法示意图

传的数据包、过滤掉不需要的流量等). 之后，将一台可配置丢包率、数据包传输延迟等环境参数的网络仿真设备当作中间设备串接在互动式流量回放系统中(该网络仿真设备用于模拟目标网络环境，见图 1(b))，通过控制互动式流量回放过程产生指定网络条件下的符合协议语义的目标流量.

3.2　互动式网络场景再现过程的流速控制

互动式网络场景再现的核心是流量的双向收发以及基于状态的流量产生机制. 在系统实现中涉及到磁盘 IO、网络 IO、会话状态维护和管理等多个方面. 基于流量在回放时的具体处理流程以及系统实现时的性能考虑(如磁盘 IO 和网络 IO 的并行化等)，互动式流量回放过程通常包含如下 5 个子过程(如图 2)：

① 数据包输入子过程. 该过程从磁盘文件中依序读入数据包，并至于系统的输入缓冲区中.

② 流量回放子过程. 该过程从系统输入缓冲区中获取数据包，并根据对应会话的状态，采用基于状态判定的方法尝试发送该数据包. 如果待发送数据包通过状态判定，则回放系统直接调用对应的网络测试接口将数据包发送出去并更新对应会话的状态；否则，该数据包将被缓存于系统中并等待会话状态更新后再次发送.

③ 数据包接收子过程. 该过程从回放系统的网络测试接口中提取到达的数据包，并至于系统的接收数据包缓冲区中.

④ 会话接收更新子过程. 该过程从回放系统的接收数据包缓冲区中取出到达系统的数据包，并根据数据包包含的协议状态信息更新对应会话的状态；在会话状态更新后，该过程还负责将缓存于系统中满足发送条件的数据包发送出去.

⑤ 超时重传子过程：该过程负责将当前系统已经发送出去但尚未被系统另外一个测试接口接收到的超时的数据报文依据网络协议规范进行重传.

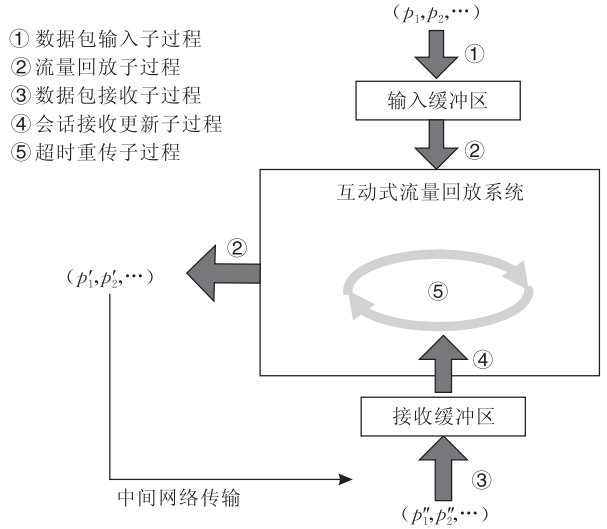


图 2　互动式网络流量回放系统的设计实现

本文考虑在互动式流量回放过程中，通过调整系统输入流量速率来控制回放过程输出流量. 由于不知道具体数学模型，因此考虑基于系统输入输出信息来对系统进行控制. 该问题建模成离散时间目标跟踪控制问题，具体描述如下：

将互动式流量回放过程视为一个动态的含有噪声的非线性过程，其中系统数学方程并不可知，但却可以通过测量得到一系列离散时间回放过程输出流量信息. 记回放控制的目标流量大小为 $t_1,t_2,\cdots$，测量得到的离散时间系统输出流量大小为 $y_1,y_2,\cdots$，相应的输入流量控制为 $u_0,u_1,\cdots$. 其中，当前时刻 k 的系统输入 u_k 影响 y_{k+1}，$y_{k+2},\cdots$. 基于到当前为止的历史流量输出信息 $y_1,y_2,\cdots,y_k$ 以及历史控制 $u_0,u_1,\cdots,u_{k-1}$，回放过程流速控制的目标是决定当前时刻 k 的输入 u_k，以便在该输入下回放过程产生的下一个时刻的输出 y_{k+1} 满足目标值 t_{k+1}.

4　影响网络场景再现过程产生流速的因素分析

本节将从网络环境行为以及网络流量特性两大方面对影响互动式网络场景再现过程产生流速的因素进行分析.

4.1　网络环境行为

互动式网络场景再现过程的本质是模拟 TCP/IP 协议栈来对 TCP 流量传输过程进行再现. 因此，所有影响 TCP 传输性能的网络行为都会影响到回放过程. 以下分别从网络丢包、传输延迟、会话阻断以及报文乱序等几方面进行分析，并对这些因素影响 TCP 流量回放过程的机理进行说明.

(1) 网络丢包. 网络传输过程中的数据包丢包行为会影响到 TCP 流量的传输性能. 根据 TCP/IP 协议规范，丢包是网络发生拥塞的表现. TCP 发送端在通过超时机制感知到丢包发生时，会减小数据发送窗口并进入拥塞避免阶段[24]，同时对丢失的数据报文进行重传. 由于发送窗口减小，TCP 数据传输速率也随之降低.

(2) 传输延迟. 数据包端到端传输延迟也会影响到 TCP 流量的传输性能. 根据文献[25]，TCP 会话的吞吐率与网络传输延迟近似成反比关系，即网络

传输延迟越大,则 TCP 流量的传输性能也就越低.

(3)会话阻断.在实际的流量传输过程中,若一个 TCP 会话被阻断(如接收到 Firewall 发送的 RST 报文),则该会话将会被终止,且其后续的流量将不再传输.

(4)报文乱序.报文乱序对 TCP 传输性能的影响体现在接收端的确认行为上.当发送方传输的数据报文在网络中发生乱序时,接收方将需要缓存乱序到达的数据报文[26].对于采用累计确认的 TCP 协议实现,报文乱序将明显降低 TCP 的数据传输效率.

4.2 输入流量特性

除了丢包、传输延迟等网络环境行为,输入流量特性同样会影响到回放过程.由于被回放流量通常是由多个 TCP 会话组成,因此,从单个 TCP 会话特性以及会话并发两个方面进行分析.

(1)单个 TCP 会话特性:从真实网络中采集的流量包含多种应用,产生的 TCP 会话特性也会根据应用的不同而不同.参考文献[1],我们将 TCP 会话大体分成两类,一类是交互式会话,对应的应用包括 Telnet、SSH、Rlogin、VoIP 等;另外一类是成块数据传输会话,如 FTP 文件传输、文件共享等.对于交互式会话,一般来说这类会话的流速较低,数据包之间的时间间隔相对较长,因此,在相同的输出流速下需要并发的会话个数较多;而成块数据传输的会话对应的流速较高,数据包之间的时间间隔较短,通常情况下同一时刻发送窗口中会有多个数据报文进行传输.因此,在相同的输出流速下需要的并发会话个数也较少.

(2)输入流量的会话并发:会话并发会显著改变流量回放过程输出速率.站在回放过程流速控制的角度来看,当网络丢包、传输延迟、会话阻断以及报文乱序等现象越严重时,单个 TCP 会话的传输性能也就越低.因此,为了产生相同速率的流量,回放系统需要在同一时刻回放尽可能多的并发会话,也即在系统控制时需要施加更大的流量输入速率.而当被回放流量中同一时刻并发会话数目达不到要求时(比如,输入流量中只包含少数几个会话),则无论施加多大输入流量,系统都不可能产生期望输出速率(在本文实验部分可以观察到).

5 基于 FA 和 SPSA 的互动式场景再现过程流速控制系统和方法

由于互动式流量产生机制异常复杂,很难通过内部机理分析建立精确的数学模型,因此考虑在系统的输入和输出层面来对系统进行建模并对产生流速进行控制.以下具体介绍基于函数近似器(Function Approximator,FA)和并行扰动随机估计理论(SPSA)的流速控制系统和方法.

5.1 互动式网络场景再现过程的 FA 控制方法

将流量回放系统当作一个黑盒(如图 3),采用函数近似器来产生控制输入并对系统进行控制.函数近似器用来描述目标流量和需要施加流量之间的映射,具体数学形式可以采用多层神经网络、代数多项式、三角型序列等.根据 Stone-Weierstrass 理论[27],这些近似器均可对任意一个数学函数进行精确逼近.

我们考虑采用结构固定的函数近似器(如多层神经网络中层数和节点数目一定、代数多项式的阶数一定),但却可以让近似器的参数随时间动态变化(比如权重可动态改变).函数近似器的输入是系统观测到的历史测量数据和历史控制信息(还可包含目标流量信息),而相应的输出则是系统当前时刻产生的控制.假设时刻 k 函数近似器的输入包含前 M 个历史测量数据和前 N 个历史控制信息,当函数近似器参数不变化时,近似器的输入可简单记为

$$I_k = \{y_k, y_{k-1}, \cdots, y_{k-M+1}; u_{k-1}, u_{k-2}, \cdots, u_{k-N}\} \tag{1}$$

当函数近似器的参数动态更新时(由扰动机制产生,在下面小节将会介绍),上述近似器的输入信息中还将包含一部分参数动态更新时得到的数据.

图 3 是基于函数近似器进行流速控制的互动式流量回放系统示意图.在该系统中,当前时刻 k 函数近似器的控制输入是目标流量信息 t_{k+1} 和前一个时刻的系统观测信息 $I_k=\{y_k\}$,而近似器的输出则直接对应当前时刻产生的控制 u_k.

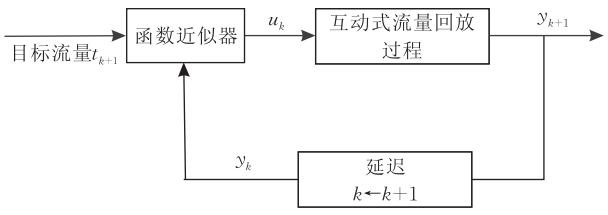


图 3 基于函数近似器的互动式流量回放过程流速控制系统框图(其中 $I_k=\{y_k\}$)

5.2 基于 SPSA 方法的参数估计

在基于函数近似器的控制系统和方法中,函数近似器被用来根据历史信息产生当前时刻需要的输入.然而,由于不知道近似器包含的参数,因此需要对近似器的参数 θ_k 进行实时估计.在假定函数近似

器结构固定的条件下，近似器的参数估计问题等同于求取使得代价函数 $L_k(\theta_k)$ 最小化的参数 $\theta_k \in R^p$（由于近似器结构固定，p 是与 k 无关的一个常量，代表参数个数）. 代价函数 $L_k(\theta_k)$ 可采用如下二次型：

$$L_k(\theta_k)=E[(\boldsymbol{y}_{k+1}-\boldsymbol{t}_{k+1})^{\mathrm{T}}\boldsymbol{A}_k(\boldsymbol{y}_{k+1}-\boldsymbol{t}_{k+1})+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{u}_k] \tag{2}$$

式中 $\boldsymbol{A}_k$ 和 $\boldsymbol{B}_k$ 是两个半正定矩阵，反映控制器对于输出流量和施加控制的不同权重；$\boldsymbol{y}_{k+1}$，$\boldsymbol{t}_{k+1}$，$\boldsymbol{u}_k$ 分别是由标量 y_{k+1}，t_{k+1} 和 u_k 组成的向量.

上述函数近似器的参数估计问题等价于在时刻 k 求取 θ_k^*，使之满足如下方程：

$$g_k(\theta_k)=\frac{\partial L_k}{\partial \theta_k}=\frac{\partial \boldsymbol{u}_k^{\mathrm{T}}}{\partial \theta_k}\cdot\frac{\partial L_k}{\partial \boldsymbol{u}_k}=0 \tag{3}$$

其中，$g_k(\theta_k)$ 即为代价函数 $L_k(\theta_k)$ 对参数 $\theta_k \in R^p$ 的梯度. 然而，由于系统数学模型并不可知，式(3)中 $\partial L_k/\partial \boldsymbol{u}_k$ 项便无法求得. 因此，传统基于目标函数梯度信息进行求解的方法或其它一些利用 $g_k(\theta_k)$ 进行参数求解的方法便不再适用.

为了获取系统参数 $\{\theta_k\}$，我们考虑采用随机估计的方法：

$$\hat{\theta}_k=\hat{\theta}_{k-1}-a_k\times(gradientapprox)_k \tag{4}$$

其中，$\hat{\theta}_k$ 是 k 时刻近似器参数 θ_k 的估计值，$\{a_k\}$ 是一个满足一定条件的标量序列. 此处，采用并行扰动随机估计方法(SPSA)[20] 进行参数估计，该方法具有如下形式：

$$\hat{\theta}_k=\hat{\theta}_{k-1}-a_k\,\hat{g}_k(\hat{\theta}_{k-1}) \tag{5}$$

式(5)中 $\hat{g}_k(\hat{\theta}_{k-1})$ 是扰动后 $g_k(\hat{\theta}_{k-1})$ 的估计值. $\hat{g}_k(\hat{\theta}_{k-1})$ 的第 l 个成员变量($l=1,2,\cdots,p$)可以由如下式子求得：

$$\hat{g}_{kl}(\hat{\theta}_{k-1})=\frac{\hat{L}_k^{(+)}-\hat{L}_k^{(-)}}{2c_k\Delta_{kl}} \tag{6}$$

在式(6)中：

(1) $\hat{L}_k^{(\pm)}$ 是利用 $y_{k+1}^{(\pm)}$ 和 $u_k^{(\pm)}$ 对 $L_k(\hat{\theta}_{k-1}\pm c_k\Delta_k)$ 进行计算得到的值. 对于形如式(2)的代价函数，$\hat{L}_k^{(\pm)}=(\boldsymbol{y}_{k+1}^{(\pm)}-\boldsymbol{t}_{k+1})^{\mathrm{T}}\boldsymbol{A}_k(\boldsymbol{y}_{k+1}^{(\pm)}-\boldsymbol{t}_{k+1})+\boldsymbol{u}_k^{(\pm)\mathrm{T}}\boldsymbol{B}_k\boldsymbol{u}_k^{(\pm)}$.

(2) $u_k^{(\pm)}$ 是基于扰动后的参数 $\theta_k=\hat{\theta}_{k-1}\pm c_k\Delta_k$ 时得到的函数近似器的控制输出，其中 $\Delta_k=(\Delta_{k1},\Delta_{k2},\cdots,\Delta_{kp})^{\mathrm{T}}$ 是一个随机序列. 通常情况下，$\{\Delta_{ki}\}$ 是独立同分布、有界、相对于 0 对称分布的随机量，并且 $\forall k,i$ 满足 $E(\Delta_{ki}^{-2})$ 一致有界的条件.

(3) $y_{k+1}^{(\pm)}$ 是当控制为 $u_k^{(\pm)}$ 时回放系统产生的流量测量值.

(4) $\{c_k\}$ 是满足一定条件的正实数序列. 通常情况下，根据被控系统是否稳定取 $c_k\to 0$ 或者 $c_k=c$.

基于并行扰动随机估计理论进行参数估计的算法如图 4 所示.

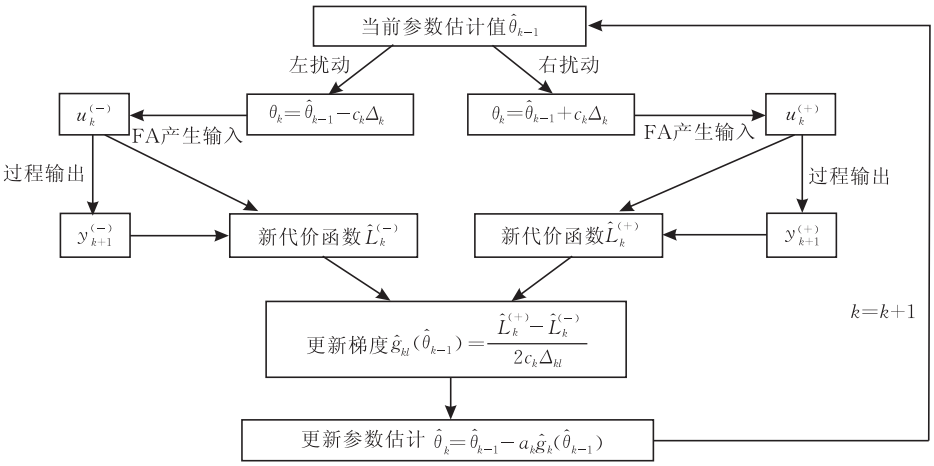


图 4　并行扰动随机估计方法的算法步骤

当扰动机制引入到控制过程时，函数近似器的输入将包含参数扰动后得到的系统测量值和控制输入. 此时，系统的测量值变为 $\{y_0,y_1^{(+)},y_1^{(-)},y_1,y_2^{(+)},y_2^{(-)},y_2,\cdots\}$，而对应的控制输入也将变为 $\{u_0,u_1^{(+)},u_1^{(-)},u_1,u_2^{(+)},u_2^{(-)},u_2,\cdots\}$. 由于每一个产生的控制均是基于前 M 个历史测量数据和前 N 个历史控制信息，对于 $u_k^{(+)}$、$u_k^{(-)}$ 和 u_k，相应的函数近似器的输入就变成了集合 $I_k^{(+)}$、$I_k^{(-)}$ 和 I_k. 比如，当 $M=N=2$ 且当前最新系统测量值为 $y_{k+1}^{(+)}$ 时，下一时刻系统的控制 $u_k^{(-)}$ 是基于参数 $\theta_k=\hat{\theta}_{k-1}-c_k\Delta_k$ 和集合 $I_k^{(-)}=\{y_{k+1}^{(+)},y_k,u_k^{(+)},u_{k-1}\}$ 来产生的.

5.3　流量回放过程流速控制流程

基于函数近似器的流速控制系统在采用 SPSA 算法进行参数估计后的具体流速控制过程如算法 1.

算法 1. 基于函数近似器和 SPSA 的流量回放过程流速控制流程.

1. 读入当前时刻 k 的参数估计值 $\hat{\theta}_{k-1}$ 和当前时刻近似器的输入 I_k;

2. 参数右扰动 $\theta_k=\hat{\theta}_{k-1}+c_k\Delta_k$,函数近似器根据扰动后的参数和当前输入信息产生需要施加的流量 $u_k^{(+)}$;

3. 测量流量回放过程的输出流量大小 $y_{k+1}^{(+)}$;

4. 计算参数右扰动后的代价函数 $\hat{L}_k^{(+)}$;

5. 更新当前函数近似器的输入信息;

6. 参数左扰动 $\theta_k=\hat{\theta}_{k-1}-c_k\Delta_k$,函数近似器根据扰动后的参数和当前输入信息产生需要施加的流量 $u_k^{(-)}$;

7. 测量流量回放过程的输出流量大小 $y_{k+1}^{(-)}$;

8. 计算参数左扰动后的代价函数 $\hat{L}_k^{(-)}$;

9. 更新当前函数近似器的输入信息;

10. 根据参数左右扰动后的代价函数变化信息更新当前参数 $\hat{\theta}_k=\hat{\theta}_{k-1}-a_k\hat{g}_k(\hat{\theta}_{k-1})$;

11. 当前时刻 $k=k+1$;

12. 返回步 1,直到流量回放过程结束.

采用函数近似器和 SPSA 方法进行系统控制的最大优点在于可以在并不知道系统模型的情况下仍然对动态系统进行精确控制. 然而,引入参数扰动机制也给该方法和对应的控制系统在具体应用时在输入输出和收敛时间等方面带来了一些副作用. 我们将在实验部分对此进行详细论述.

6 实验验证及结果分析

6.1 实验环境设置

为考察设计的系统和方法的实际控制效果,实现了一个高性能的互动式网络流量回放测试系统,如图 5 所示. 整个测试系统由组成测试回路的流量回放服务器、千兆交换机、FreeBSD 双穴主机(当作网络仿真器)和流量监测器组成. 其中,流量回放服务器由一台高性能 Linux 服务器充当(Redhat 9.0 系统,2.4.20 内核),配置 2×2.0 GHz Intel Xeon 处理器,4 GB RAM.

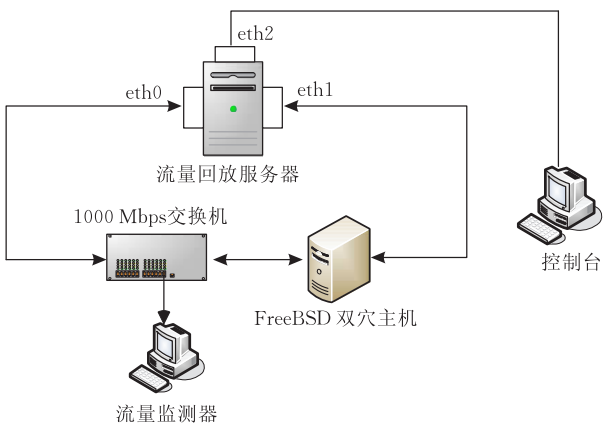


图 5 流量回放实验环境

由于 TCP 协议栈的实现在不同的操作系统平台以及不同系统的版本下存在差异,本文流量回放服务器上的 TCP 协议栈的模拟实现参考了 TCP-Opera,具体也是基于 BSD 4.4-Lite RELEASE 平台的 TCP 协议栈实现[24]. 目前实现的一些核心功能包括:6 个 TCP 计时器、传输过程中的数据包超时和重传、快速重传和快速恢复、拥塞控制以及 RTT 动态测量等.

网络仿真器的实现采用的是 FreeBSD 平台上的 Dummynet① 这一开源工具. Dummynet 可用于模拟多种复杂的网络环境,对包括传输延迟、网络带宽、丢包率以及排队策略等在内的多项参数进行选择设置,是当前网络研究领域最常用的网络仿真工具.

在我们的实验中,将网络仿真平台(安装 FreeBSD 8.1-RELEASE 系统)的两块网卡设置成网桥模式进行数据包转发,同时配合 Dummynet 对网桥传输流量的数据包延迟和丢包率进行控制;对回放过程会话阻断功能的实现则是通过预先在输入流量中对需要阻断会话的数据包的 MAC 地址进行标定,而后在网络仿真平台上通过 IP Firewall 设置相应的规则(根据 MAC 地址设置)进行阻断.

此外,流量回放服务器和 FreeBSD 双穴主机均配备 Intel e1000 千兆网卡,整个测试回路是一个纯千兆的环境. 在千兆交换机上将回放产生的流量镜像到流量监测器上进行观测.

实验中采用真实网络流量对本文方法进行考察. 所用流量在西安交通大学校园网出口处采集得到,共包含超过 20000000 个 TCP 数据包和 300000 个 TCP 会话. 实验中对串接在测试回路中的 FreeBSD 双穴主机设置不同的丢包率和传输延迟,同时在输入流量中以不同的概率对阻断会话进行标识,以此来模拟不同的目标网络环境.

6.2 实验结果及分析

考虑丢包、数据包传输延迟以及会话阻断 3 种不同的网络行为,实验过程中设置 FreeBSD 双穴主机分别以 0.1%,1%,5%和 10%的概率进行随机丢包,以 0.1 ms,1 ms,10 ms 进行数据包传输延迟. 另外,设置以 10%,20%和 30%的概率进行随机会话阻断. 这样设置 FreeBSD 双穴主机的丢包率、传输

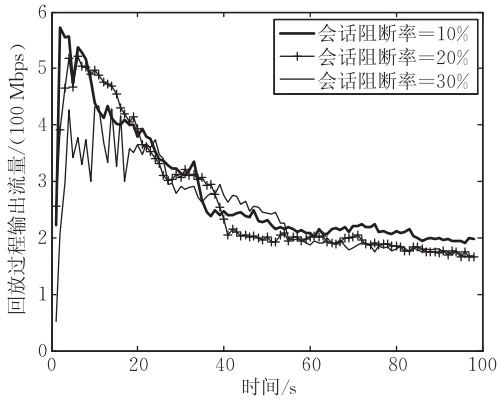
① http://info.iet.unipi.it/~luigi/ip_dummynet/

延迟以及回放过程会话阻断率的目的在于在一个较为宽广的目标网络环境下考察所提方法的实际控制性能.

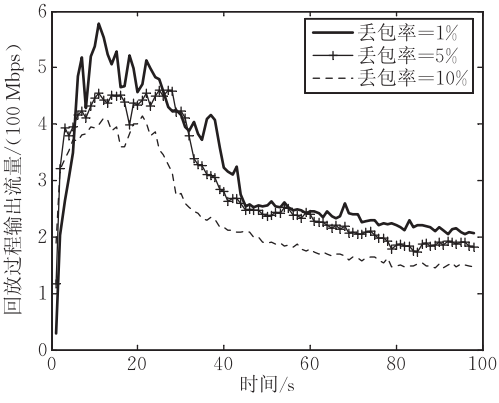
实验过程中将需要产生的目标流量大小设置为100 Mbps,离散采样时间取为0.5 s.对每个不同设置的实验均重复进行10次,最后取平均性能进行分析.

6.2.1　未施加控制时的回放性能

首先考察系统在不施加任何控制时的回放性能,实验结果如图6所示.在实验中观察到整个真实网络流量的互动式回放过程是一个非常不平稳的过程;受到丢包和会话阻断等网络因素的影响,系统性能发生显著变化,而且,两种因素对产生流量的影响较为复杂,难以用一个确切的数学方程进行描述.可以预见,在同时有丢包、会话阻断以及传输延迟等多种因素存在的网络环境下,系统性能的变化将变得更加复杂.



(a) 会话随机阻断情况



(b) 数据包随机丢包情况

图6　没有施加控制的情况下回放系统产生的流量大小

6.2.2　采用FA和SPSA方法的回放结果

(1) 方法参数选取

考察基于函数近似器和并行扰动随机估计理论的控制系统和方法的控制效果.实验中利用神经网络当作函数近似器,该神经网络的节点采用$1/(1+e^{-z})$的输入输出函数,每个节点的输入包括前一层节点的输出加权和再加上一个属于自身节点的偏移量.整个函数近似器的输入包括到当前时刻为止的前两个时刻回放系统产生流量值($M=2$)、上一个时刻的输入控制量($N=1$)和下一个时刻的目标流量大小,近似器的输出则是当前时刻需要施加的控制量.因此,作为函数近似器的神经网络具有4个输入节点,1个输出节点.假定包含两个隐层,每个隐层有5个节点,则该神经网络可记为$N_{4,5,5,1}$,共包含$(4\times5+5)+(5\times5+5)+(5\times1+1)=61$个参数需要同时估计.神经网络的具体结构如图7所示(实验中发现采用结构更为复杂的神经网络并没有显著提升控制性能,但却增加了计算量).

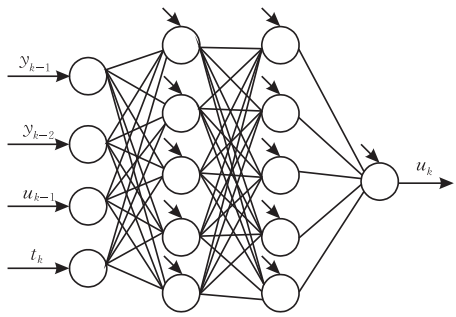


图7　作为函数近似器的神经网络

在SPSA算法中,参数$\{\Delta_{ki}\}$取为伯努利±1分布,$a_k=a=0.05$,$c_k=c=0.005$,作为函数近似器的神经网络的权值初始量取为标准正态分布$N(0,1)$;式(2)中代价函数的参数$\boldsymbol{A}_k=[1]$,$\boldsymbol{B}_k=[10]$.

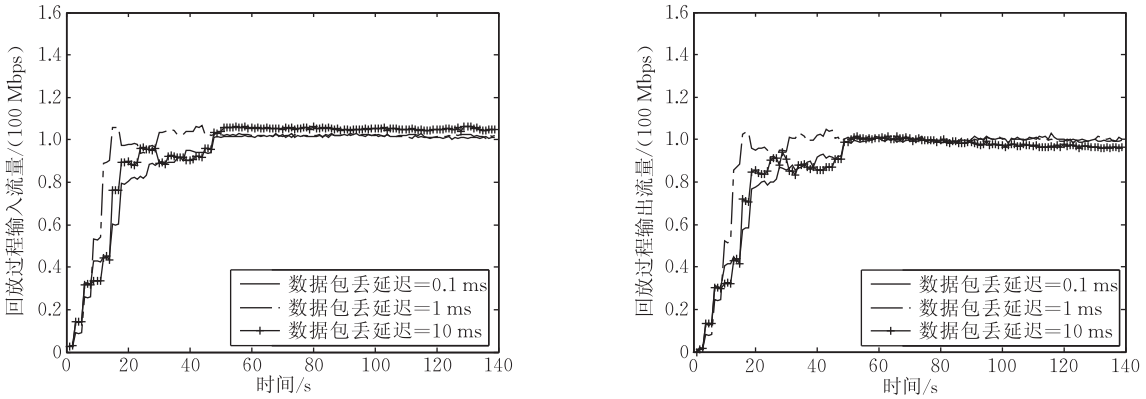
(2) 实际控制结果与分析

图8、图9以及图10分别是在不同数据包传输延迟、会话阻断和丢包情况下控制系统产生的输入流量大小和输出流量大小.

从实验结果中可以得到如下结论:

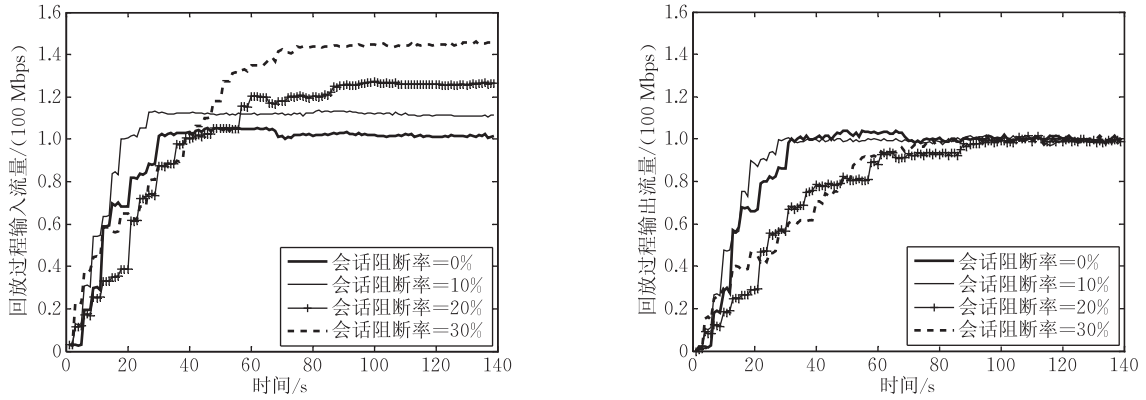
(1) 在互动式流量回放控制过程中,系统的输入流量和输出流量之间的关系确实是非线性的.比如从图9(a)中可以看出,为了追踪目标流量,系统在10%的会话阻断率下达到稳定时需要施加的输入流量约为110 Mbps,在20%的情况下为128 Mbps,而在30%的情况下需要施加的输入流量上升到145 Mbps.

(2) 在达到稳定状态时,基于FA和SPSA方法的控制系统在会话阻断、丢包和数据包传输延迟3种网络因素影响下均能够精确地对回放过程产生流速进行控制.在测试网络丢包率达到10%以及会话阻断率达到30%等极端情况下,系统仍然能够达到很好的控制效果.



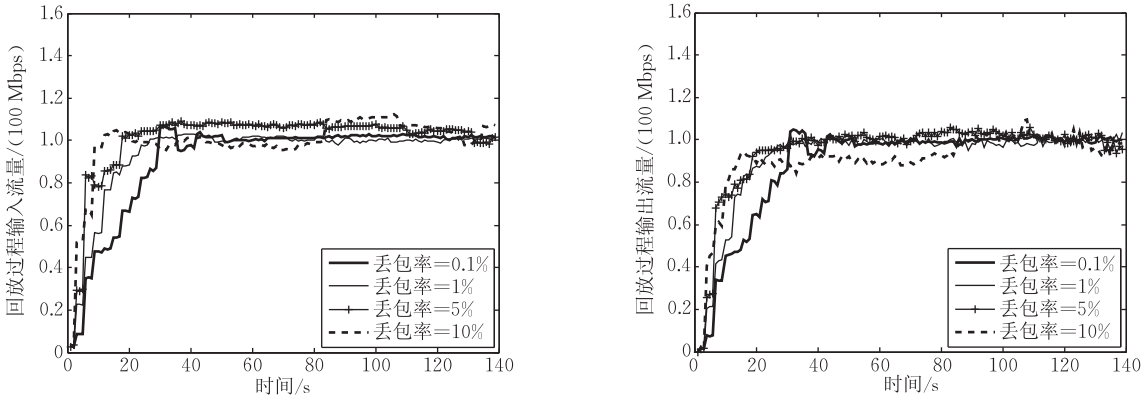
(a) 施加的输入流量大小 (b) 回放系统输出流量大小

图 8 基于函数近似器的控制系统在不同数据包传输延迟情况下产生输入和输出流量大小



(a) 施加的输入流量大小 (b) 回放系统输出流量大小

图 9 基于函数近似器的控制系统在不同会话阻断率的情况下产生输入和输出流量大小



(a) 施加的输入流量大小 (b) 回放系统输出流量大小

图 10 基于函数近似器的控制系统在不同数据包丢包率的情况下产生输入和输出流量大小

(3) 基于 FA 和 SPSA 方法的控制系统在不同网络环境因素影响下所需要的收敛时间不同. 总体上来说,在随机丢包情况下系统所需要的收敛时间最少,而在会话阻断情况下需要的收敛时间最长.

(4) 从稳定误差上面来看,在会话阻断、丢包和数据包传输延迟 3 种网络影响因素中,基于 FA 和 SPSA 方法的控制系统在会话随机阻断和数据包传输延迟情况下的控制效果要好于在随机丢包情况下(见图 8(b)、图 9(b)和图 10(b)). 我们认为引起该现象的主要原因在于不同网络事件引起的回放系统行为复杂性不同. 数据包传输延迟是一个相对较为简单的网络行为;当会话被阻断时(接收到 RST 报文),回放系统将简单地结束当前流的回放并且后续输入流量中所有属于该会话的数据包将不再被回放出去. 因此,会话阻断情况下系统的行为也相对简单. 而当数据包发生丢包时,回放系统将执行超时重

传行为.根据 TCP/IP 协议规范,TCP 数据报文的超时重传机制是一个非常复杂的过程[28-29].因此,随机丢包情况下的系统行为是最复杂的,相应地导致系统在随机丢包情况下的控制效果要低于在会话随机阻断以及数据包传输延迟时的情况.

(5)对于数据包传输延迟在 10 ms 的情况下,发现基于 FA 和 SPSA 方法的控制系统产生的输出流量在第 80 个采样周期后缓慢下降而跟踪不上目标流量(见图 8(b)).我们猜测引起该现象的原因在于输入流量的特性,即当数据包传输延迟在 10 ms 时,输入流量中的并发会话个数不足以使得回放过程跟踪上目标流量.通过监测流量回放过程发现此时并发会话个数约为 80.同时,在第 80 个时刻之后人为设置输入流量大小持续增长,同样发现系统的实际输出流量不再随着输入流量的增长而增长.因此,并发会话的个数会决定输出流量的大小,实验结果与我们的分析一致.

此外,我们将基于 FA 和 SPSA 的控制方法与单输入单输出系统的一些自适应控制方法进行了比较,发现有如下性质:

(1)稳定误差. 基于 FA 和 SPSA 的方法产生的误差要优于一般的自适应控制方法,尤其是在大丢包率的情况下.

(2)收敛时间. 基于 FA 和 SPSA 的方法需要的收敛时间要比一般的自适应控制方法长.而且,一般的自适应控制方法的收敛时间比较稳定,而基于 FA 和 SPSA 的方法在丢包、会话阻断以及不同数据包传输延迟情况下则相差较大.

(3)产生输入和输出.在系统稳定之前基于 FA 和 SPSA 的方法产生的输入和输出要比一般的自适应控制方法波动大.

对于基于 FA 和 SPSA 的方法与一般的自适应控制方法在收敛时间和产生输入输出方面的差异,我们认为主要原因是由引入的扰动机制造成的.为了对参数进行估计,基于 FA 和 SPSA 的方法通过在参数空间随机扰动来获得代价函数对参数的变化信息.而一般的自适应方法则是直接基于系统输出反馈信息产生需要的输入.因此,随机扰动增加了产生输入和输出的波动性,同时也使得方法和系统需要更多数据量和收敛时间来跟踪上目标流量大小.

下面,通过表 1 所示的数据对基于 FA 和 SPSA 方法的控制系统在稳定时的控制效果进行更为细致的分析和说明.从平均误差和误差方差上可以看出,总体上基于 FA 和 SPSA 方法的控制系统具有较好的稳定性.

表 1　基于函数近似器的控制系统在不同网络环境下稳定时的控制效果

性能指标		平均误差	误差方差
丢包率/%	0.1	0.0032	1.54E-4
	1.0	0.0068	2.19E-4
	10.0	0.0010	1.20E-3
会话阻断/%	10.0	0.0047	1.01E-4
	20.0	0.0071	8.33E-5
	30.0	0.0022	7.84E-5
数据包传输延迟/ms	0.1	0.0019	6.68E-5
	1.0	0.0029	1.24E-4
	10.0	0.0161	2.98E-4

利用其它网络流量对所提方法和系统进行了实验考察,得到的结果同上述结果基本一致,基于文章篇幅考虑此处便不再陈列.从实验结果中可以发现,本文所设计的系统与方法可以在一个较为宽广的网络环境下对回放系统产生的流量速率进行有效控制.

7　结论和未来展望

本文对互动式网络场景再现过程的流速控制问题进行了研究,设计实现了一个基于函数近似器的流速控制系统,并采用真实网络流量对系统的实际性能进行了实验验证与分析.

在未来,将重点进行如下工作:

(1)定量考察网络延迟、丢包和会话阻断等不同网络行为以及流量的各种特性如会话到达速率、会话大小分布等对回放过程性能的影响.

(2)在上述工作基础上,尝试对回放系统建立精确的数学模型.虽然采用无模型控制方法可以达到较为满意的控制效果,然而,精确的数学模型仍然是更好地对系统施加控制,并进行性能分析、最优化设计和稳定性分析等多项重要工作的前提.

参　考　文　献

[1] Danzig P B, Jamin S. Tcplib: A library of tcp internetwork traffic characteristics. Computer Science Department, University of Southern California, Technical Report USC-CS-91-495, 1991

[2] Terdik G, Gyires T. Lévy flights and fractal modeling of internet traffic. IEEE/ACM Transactions on Networking, 2009, 17(1): 120-129

[3] Dainotti A, Pescape A, Rossi P S, Palmieri F, Ventre G. Internet traffic modeling by means of hidden markov models. Computer networks, 2008, 52(14): 2645-2662

[4] Anand N C, Scoglio C, Natarajan B. GARCH-non-linear time series model for traffic modeling and prediction//

Proceedings of the 2008 IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services. Salvador, Bahia, Brazil, 2008: 694-697

[5] Chimeh J D, Hakkak M, Azmi P. Internet traffic modeling and capacity evaluation in UMTS. International Journal of Hybrid Information Technology, 2008, 1(2): 109-120

[6] Fras M, Mohorko J, Cucej Z. Packet size process modeling of measured self-similar network traffic with defragmentation method//Proceedings of the 15th International Conference on Systems, Signals and Image Processing. Bratislava, Slovakia, 2008: 253-256

[7] Wu C F, Goel A, Bezzaz A et al. TCPivo: A high performance packet replay engine. MoMeTools'03//Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research, 2003: 57-64

[8] Cheng Y, Hölzle U, Cardwell N, Savage S, Voelker G M. Monkey see, Monkey do: A tool for TCP tracing and replaying//USENIX Annual Technical Conference, General Track. Boston, MA, USA, 2004: 87-98

[9] Ye Tao, Veitch D, Iannaccone G, Bhattacharyya S. Divide and conquer: PC-based packet trace replay at OC-48 speeds//Proceedings of the 1st International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM 2005). Trento, Italy, 2005: 262-271

[10] Van P D, Zhanikeev M, Tanaka Y. Effective high speed traffic replay based on IP space//Proceedings of the 11th International Conference on Advanced Communication Technology. Gangwon-Do, 2009, 1: 151-156

[11] Chu Wei-Bo, Guan Xiao-Hong, Cai Zhong-Min, Chen Ming-Xu. A traffic splitting method of high-speed network using sub-network flow optimization. Journal of Xi'an Jiaotong University, 2011, 45(12): 22-27(in Chinese)
(褚伟波，管晓宏，蔡忠闽，陈明旭. 采用子网流量组合优化的网络流量分割方法. 西安交通大学学报，2011，45(12)：22-27)

[12] Hong S, Wu S F. On interactive internet traffic replay//Recent Advances in Intrusion Detection. Seattle, Washington, USA, 2005: 247-264

[13] Chen Zhong-Qiang, Delis Alex, Wei Peter. A pragmatic methodology for testing intrusion prevention system. Computer Journal, 2009, 52(4): 429-460

[14] Chu Wei-Bo, Guan Xiao-Hong, Cai Zhong-Min, Chen Ming-Xu. Balance based performance enhancement for interactive TCP traffic replay//Proceedings of the 2010 IEEE International Conference on Communications (ICC 2010). Cape Town, South Africa, 2010: 1-5

[15] Chu Wei-Bo, Guan Xiao-Hong, Cai Zhong-Min, Chen Ming-Xu. A new method for interactive TCP traffic replay based on balance-checking between transmitted and received packets. Chinese Journal of Computers, 2009, 32(4): 835-846(in Chinese)
(褚伟波，管晓宏，蔡忠闽，陈明旭. 基于收发平衡判定的TCP流量回放方法. 计算机学报，2009，32(4)：835-846)

[16] Kornexl S, Paxson V, Dreger H, Feldmann A, Sommer R. Building a time machine for efficient recording and retrieval of high-volume network traffic//Proceedings of the 5th ACM Internet Measurement Conference (IMC 2005). Berkeley, California, 2005: 267-272

[17] Qian F, Gerber A, Mao Z M et al. TCP revisited: A fresh look at TCP in the wild//Proceedings of the 9th ACM internet measurement conference (IMC 2009). Chicago, Illinois, 2009: 76-89

[18] Labovitz C, Iekel-Johnson S, McPherson D et al. Internet inter-domain traffic//Proceedings of the ACM SIGCOMM 2010 Conference. New York, USA, 2010: 75-86

[19] Zhang Yin, Breslau Lee, Paxson Vern, Shenker Scott. On the characteristics and origins of internet flow rates//Proceedings of the ACM SIGCOMM Computer Communication Review. Pittsburgh, PA, USA, 2002

[20] Spall J C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions Automatic Control, 1992, 37(3): 332-341

[21] Sommers J, Barford P. Self-configuring network traffic generation//Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC 2004). Taormina, Sicily, Italy, 2004: 68-81

[22] Antonatos S, Anagnostakis K G, Markatos E P. Generating realistic workloads for network intrusion detection systems//Proceedings of the 4th Workshop on Software and Performance (WOSP'04). Redwood Shores, CA, 2004: 207-215

[23] Vishwanash K V, Vahdat A. Swing: Realistic and responsive network traffic generation. IEEE/ACM Transactions on Networking (TON), 2009, 17(3): 712-725

[24] Stevens W R, Write G R. TCP/IP illustrated (vol. 2): The implementation. Addison-Wesley Longman Publishing Co. Inc., 1995

[25] Klein Thierry E, Leung Kin K, Parkinson R, Samuel Louis G. Avoiding spurious TCP timeouts in wireless networks by delay injection//Proceedings of IEEE Globecom 2004. Dallas, TX, 2004: 2754-2759

[26] Piratla N M, Jayasumana A P. Metrics for packet reordering—A comparative analysis. International Journal of Communication Systems, 2008, 21(1): 99-113

[27] Rudin W. Principles of Mathematical Analysis. New York: McGrawHill, 1964

[28] Padhye Jitendra, Firoiu Victor, Towsley Donald F, Kurose James F. Modeling TCP Reno performance: A simple model and its empirical validation. IEEE/ACM Transactions on Networking(TON), 2000, 8(2): 133-145

[29] Paxson V. Automated packet trace analysis of TCP implementations//Preceedings of the SIGCOMM'97. Cannes, France, 1997: 167-179

CHU Wei-Bo, born in 1982, Ph. D. candidate. His research interests include internet measurement and modeling, traffic analysis and performance evaluation.

GUAN Xiao-Hong, born in 1955, Ph. D., professor, Ph. D. supervisor. His research interests include network security, system optimization and scheduling.

CAI Zhong-Min, born in 1975, Ph. D., associate professor. His research interests include networked systems, data mining and computer security.

TAO Jing, born in 1978, M.S.. His research interests include network simulation and network scenario reproduction.

Background

Generating realistic and responsive traffic workloads is of particular importance in many areas such as network security, network management, network testing and evaluation, etc. Recently, a new traffic generation method called interactive network traffic replay is proposed. Besides as a completely new method for testing in-line networking devices, this method is also capable of transforming traffic workloads under a wide range of "what-if" scenario due to its novel ability to maintain consistency of protocol semantics in the generated traffic.

This paper considers the problem of controlling output traffic volume in the nonlinear interactive traffic replay process, which is a rather challenging new problem due to the complex traffic generation mechnisms (i. e., the instability of input traffic, the changing dynamics of the replay system, the impact of testing environment such as transmission delay, packet loss, connection blocking, etc). In this paper, we design and implement a volume control system for interactive network traffic replay, and formulate the volume control task as a target tracking problem where the output traffic volume is regulated through adjustment of input traffic volume. We then adopt a function approximator (FA) in our system as a controller to generate the desired input volumes. The FA characterizes the mapping that takes system measurements as input and directly produces controls as output. Meanwhile, simultaneous perturbation stochastic approximation (SPSA) algorithm is employed to adaptively estimate the parameters of the FA in control. To validate the system and the method we have implemented it and conducted experiments with actual traffic traces. Empirical studies show that our system can track target output volumes effectively under a wide range of network conditions. The performance of the control system is further investigated in terms of its convergence time, generated input/output traffic volume and target tracking error.

The work presented in this paper is supported in part by National Natural Science Foundation of China (grant Nos. 60921003, 61175039, 61103241) and Fundamental Research Funds for Central Universities (xjj20100051).

一种缩短下载时间优先的自适应 BitTorrent 激励协议

李治军 姜守旭

(哈尔滨工业大学计算机科学技术学院 哈尔滨 150001)

摘 要 BitTorrent 激励机制的目标是保证节点上传和下载之间的公平性,但相比公平性而言,实际应用中的节点更优先考虑的是文件下载时间,据此文中提出了一种缩短文件下载时间优先的自适应 BitTorrent 激励协议 AIPS. 文中首先基于 Markov 模型对 BitTorrent 现有激励机制的效果给出了定量分析,分析了激励机制下的文件传输结构,并用概率分析方法给出了该传输结构下最小化文件下载时间的条件. 应用分析结果文中定义了一个以缩短文件下载时间为效用的博弈,在该博弈达到 Nash 平衡时各节点采用的策略就是激励协议 AIPS. 模拟实验表明文中提出的 AIPS 较现有的 BitTorrent 激励协议能明显提高文件共享系统性能,提高文件下载效率.

关键词 BitTorrent;激励机制;自适应激励;Nash 平衡

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2012.01498

A Download Time First Self-Adaptive Incentive Protocol in BitTorrent

LI Zhi-Jun JIANG Shou-Xu

(*School of Computer Science and Technology in Harbin Institute of Technology, Harbin* 150001)

Abstract The goal of current incentive mechanism for BitTorent is to guarantee the node's fairness between upload bandwidth and download bandwidth. However, the file download time is more preferable than such fairness for the users in real file sharing environments. Therefore, a new self-adaptive incentive protocol denoted as AIPS to minimize the file download time is provided in this paper for BitTorrent. This paper firstly analyzes the influence of incentive mechanism on file sharing, the file transfer structures under incentive mechanisms based on Markov model. The conditions to minimize the file download time are inferred further based on the transfer structures and then a game with file download time as its utility is defined in this paper. The AIPS is the strategy adopted by nodes under the Nash equilibrium for that game. Simulations show that the AIPS can improve the efficiency for file sharing.

Keywords BitTorrent; incentive mechanism; self-adaptive incentive; Nash equilibrium

1 引 言

在现有的 P2P 文件共享系统,如 BitTorrent① 、eMule② 中,下载到部分文件的节点就能给其它节点上传,正是由于每个节点都能提供上传服务,所以系统总的服务能力会随着下载节点个数的增加而增加,使系统具有良好的可扩展性[1]. 这也是 P2P

收稿日期:2011-01-27;最终修改稿收到日期:2012-04-26. 本课题得到国家自然科学基金(60803148,60973124)、教育部高校博士点科研基金(20102302110036)、中央高校基本科研业务费专项资金(HIT.NSRIF.2010.047)资助.**李治军**,男,1977 年生,博士,副教授,主要研究方向为对等网络、无线网络、普适计算、操作系统. E-mail: lizhijun_os@hit.edu.cn. **姜守旭**,男,1968 年生,博士,教授,博士生导师,主要研究领域为传感器网络、普适计算、数据库.

① The BitTorrent Project. http://www.bittorrent.com

② The eMule Project. http://www.emule.org.cn/

网络和 P2P 文件共享系统得到广泛应用的根本原因，德国互联网调研机构 ipoque 的统计结果表明，2007 年互联网总流量中的 50%～90%都来自 P2P 程序，而在这些 P2P 程序里，BitTorrent 占据了 P2P 流量中的 50%～70%，在 2008 年的统计结果中，BitTorrent 仍然占据 P2P 流量的第一位①.

激励机制对提高 BitTorrent 的性能具有非常重要的作用，如果没有激励机制，BitTorrent 中的许多节点就会变成只下载文件而拒绝上传的 free riders（搭便车节点），系统不再是一个 P2P 系统，而退化为传统的 C/S 系统，系统的总服务能力会显著降低[2]. 目前 BitTorrent 采用的激励机制包括 tit-for-tat（以下简称 TFT）和 optimistic unchoking（以下简称 OU）两个策略. 在节点选择给哪些节点进行上传时，TFT 策略选择提供最大下载带宽的 n_t^u（默认 $n_t^u=4$）个节点上传，OU 策略随机选取 n_o^u（默认 $n_o^u=1$）个节点上传. TFT 策略可以保证提供上传带宽的节点才能交换到下载带宽，有效地抑制了搭便车行为，而 OU 策略的目的是试探那些未连接过的节点，找到更适合的带宽交换节点[2].

不难看出，TFT 激励机制会导致节点贡献的上传带宽接近于其获得的下载带宽，实现了公平性，但相比这个公平性而言，实际系统中的节点更关心文件下载时间：如果能缩短文件下载时间，节点愿意贡献比其下载带宽更多的上传带宽. 而现在的 BitTorrent 激励机制只是根据带宽这一个因素来制定激励策略，无法实现面向缩小文件下载时间的激励机制，据此本文针对 BitTorrent 设计了一种以最小化文件下载时间为目标的自适应激励协议 AIPS（self-Adaptive Incentive Protocols for file Swarming）. AIPS 的核心是将文件下载时间作为效用函数，将节点带宽分配作为策略进行相互博弈（传统的 BitTorrent 激励是一种以公平性为效用，带宽分配为策略的博弈[3]），而在这个博弈达到 Nash 均衡时各节点采用带宽分配策略就是本文给出的以最小化文件下载时间为目标的自适应激励协议 AIPS. 本文的主要贡献如下：

（1）本文用马尔可夫过程分析了 TFT 作用下进行文件块相互传输的节点集合具有的数量特征，用 fluid 模型分析了激励机制作用下的文件共享结构，揭示了文件共享结构对文件交换效率的影响，为 BitTorrent 激励机制的分析和改进奠定了量化基础.

（2）本文提出了以最小化文件下载时间为效用函数的节点博弈模型，分析了该博弈的 Nash 平衡点，给出了达到 Nash 平衡时各类节点采取的激励策略，完成了 BitTorrent 激励机制的改进.

（3）本文设计了一种易于实现的自适应激励协议 AIPS，并用模拟实验验证了 AIPS 对文件下载性能的影响，平均能提高文件下载效率 20%左右.

本文第 2 节介绍相关工作；第 3 节对现有 BitTorrent 激励机制进行理论建模和定量分析；第 4 节对激励下的 BitTorrent 文件传输结构进行建模分析，据此定义了以最小化文件下载时间为效用的博弈模型并得出其 Nash 平衡，最后设计了一个可直接用于 BitTorrent 的自适应激励协议 AIPS；第 5 节是对 AIPS 的模拟实验结果；第 6 节是本文的结论.

2　相关工作

针对文件传输和激励机制进行分析是 P2P 网络和 P2P 文件共享的一项重要研究，Qiu 等人[3]用 fluid 模型对 BitTorrent 的文件传输性能进行了详细的理论分析，针对激励机制，文献[3]得出的结论是节点在带宽分配博弈中存在 Nash 平衡点；具有相同上传带宽的节点会形成一个组相互传输，但文献[3]并没有定量分析激励机制对文件传输性能的具体影响. 其它针对 BitTorrent 的研究也都没有给出激励机制对文件共享性能影响的定量分析结果，如 Clevenot 等人[4]使用 fluid 模型对 P2PWeb 缓存系统的性能进行了理论分析，Clevenot 等人[5]使用多阶 fluid 模型对异构 P2P 文件共享系统进行建模分析，但给出的分析结果都没有考虑激励机制. Liao 等人[6]给出了激励机制下 BitTorrent 系统的分析，但该文得到的结果不够精确，如该文仍然没有量化地描述系统平稳时的状态，另外在文献[6]的分析中假设高、低带宽节点之间的比例为定值，由于不同带宽节点在系统中的停留时间不一样，所以这个比例应该不断变化，导致文献[6]仍然没有精确描述 P2P 文件共享的结构. Legout 等人[7]就激励机制对 BitTorrent 的影响给出详细的实验分析，由于只有实验结果，无法实现对 BitTorrent 激励机制的深入改进. 本文定量地分析了激励机制对 P2P 文件

① Ipoque. http://www.ipoque.com/resources/internet-studies/internet-study-2007, 2009

共享结构及其性能的具体影响，并以此为基础给出以提高文件共享性能为目标的 BitTorrent 激励机制.

通过改进激励机制来提高 BitTorrent 性能也是一类重要研究，Zhao 等人[8]给出了一个一般性的框架来分析具有自适应能力的激励协议，但文献[8]的分析针对了系统鲁棒性，而本文的目标是提高文件共享性能. Huang 等人[9]给出了一种称为动态配额分配的上传节点选择策略，其基本思想是动态调整节点上传带宽分配中 TFT 和 OU 所占的配额(BitTorrent 中这个配额固定为 4∶1)，用来解决该文中提出的供应和需求之间的 paradox 问题. 虽然本文对激励机制的改进采用了相似的思想，但本文的研究以严格的理论分析为基础，并以文件共享性能优化为目标，而不是解决供求 paradox 问题. Fan 等人[10]也给出通过控制 TFT 和 OU 之间的配额来实现文件平均下载时间和系统公平性之间的折中，但该研究假设文件下载时间和系统公平性之间相互独立，而实际系统中差的公平性会导致大量搭便车节点，降低共享效率(文献[10]没有考虑这种情况)，所以合适的激励协议应该以保证公平为基础，同时能将单纯的公平性扩展为对系统性能的提高，这正是本文的主要工作.

3 激励机制下的 BitTorrent 文件传输模型

3.1 BitTorrent 激励

定义 1. 集合 $D(p,t)$ 为 t 时刻向节点 p 提供下载的节点集合；集合 $U(p,t)$ 为 t 时刻从节点 p 处下载数据(即节点 p 提供上传)的节点集合.

节点 p 从 $D(p,t)$ 中的节点下载数据，给 $U(p,t)$ 中的节点上传数据，如图 1 所示. 由于 $D(p,t)$ 决定了节点 p 收到的下载速率，所以 $D(p,t)$ 直接决定 p 的文件下载时间，这就需要分析 $D(p,t)$ 的变化. 根据 BitTorrent 的激励策略 TFT 和 OU，p 从 $D(p,t)$ 中选择 n_t^u 个提供下载带宽最大的节点、随机选择 n_o^u 个节点形成 $U(p,t+1)$(记 $|U(p,t)|=m$，BitTorrent 中 $n_t^u=4, n_o^u=1, m=5$). 而 $U(p,t+1)$ 中的节点在时刻 $t+1$ 会根据同样的策略调整时刻 $t+2$ 给节点 p 的上传带宽，也就是会影响 $D(p,t+2)$，所以就有 $D(p,t)$ 影响 $U(p,t+1)$，而 $U(p,t+1)$ 又影响 $D(p,t+2)$，等等，该依赖关系表示为图 2.

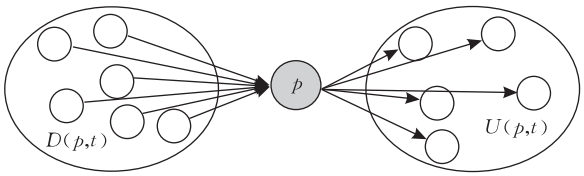


图 1 节点 p 的下载节点集合 $D(p,t)$ 和上传节点集合 $U(p,t)$

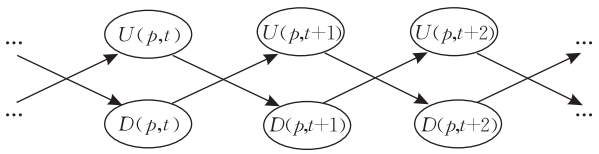


图 2 $U(p,t)$ 和 $D(p,t)$ 之间的依赖关系

类似于很多相关研究[6-10]，本文根据上传带宽将节点分为两类：高上传带宽节点集 P_h 和低上传带宽节点集 P_l，对应上传带宽 u_h 和 u_l，本文记 $N_h=|P_h|$，$N_l=|P_l|$，$N=N_h+N_l$. 由于激励策略表现为 n_t^u 和 n_o^u 之间的比值[8-10]，所以为了建模分析的方便，本文假定 n_o^u 等于 1，$n_t^u=m-1$，由于此时 $m\geq 2$，所以该假设会导致 $n_t^u:n_o^u\geq 1$，表示激励占据更大的比重，这对于无中心控制的开放 P2P 系统而言是合理的. 此时节点在执行 OU 时，找到 $n_o^u=1$ 个高(低)上传带宽节点的概率为 $\alpha=N_h/N(1-\alpha=N_l/N)$.

3.2 高带宽节点的上传节点集

定义 2. 定义 $U_p(n_h,n_l,t)$ 为节点 p 在 t 时刻的上传状态，其中 n_h 和 n_l 分别表示 $U(h,t)$ 中高上传带宽节点个数和低上传带宽节点个数.

由 BitTorrent 激励策略可知，节点 p 选择哪些节点上传文件只与 p 前一时刻的下载节点集 $D(p,t)$ 有关，因此以状态集合 $U_p(n_h,n_l\,|\,n_h+n_l=m)$(即 $\{U_p(0,m), U_p(1,m-1),\cdots,U_p(m,0)\}$)形成马尔可夫链，本文将该链记为 $\{U_p(t): t=0,1,2,\cdots\}$，其概率转移矩阵记为 $\boldsymbol{P}_u(p)$.

定理 1. 对于高带宽节点 h，其概率转移矩阵 $\boldsymbol{P}_u(h)$ 为

$$\begin{array}{c} \\ (0,m) \\ (1,m-1) \\ (2,m-2) \\ \cdots \\ (m-1,1) \\ (m,0) \end{array}\begin{array}{c} \begin{array}{ccccc} (0,m) & (1,m-1) & \cdots & (m-1,1) & (m,0) \end{array} \\ \begin{bmatrix} (1-a)^2 & 2(1-a)a & \cdots & 0 & 0 \\ 0 & (1-a)^2 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1-a & a \\ 0 & 0 & \cdots & 1-a & a \end{bmatrix} \end{array} \tag{1}$$

证明. 以 $\boldsymbol{P}_u(h)[(2,m-2),(2,m-2)]$ 为例，即当 t 时刻节点 h 的上传节点集 $U(h,t)$ 中高、低上传带宽节点的个数分别为 2 与 $m-2$ 时，$U(h,t+2)$

中高、低上传带宽节点个数仍为 2 与 $m-2$ 的概率. 事件“$U(h,t+2)$中高、低上传带宽节点个数分别为 2 与 $m-2$”按全概率公式可分为两种情况：(1) 节点 h 从 $D(h,t+1)$中根据 TFT 协议选择 2 个高上传带宽节点和 $m-3$ 个低上传带宽节点(该事件用 T_2 表示)，再用 OU 协议随机选择到 1 个低上传带宽节点(该事件用 O_0 表示)；(2) 节点 h 从 $D(h,t+1)$中根据 TFT 协议选择 1 个高上传带宽节点和 $m-2$ 个低上传带宽节点(该事件用 T_1 表示)，再用 OU 协议随机选择 1 个高上传带宽节点(该事件用 O_1 表示). 由于 TFT 和 OU 进行独立选择，所以

$$\boldsymbol{P}_u(h)[(2,m-2),(2,m-2)]=Pr(T_2)\times Pr(O_0)+Pr(T_1)\times Pr(O_1).$$

因为 $U(h,t)$中高、低上传带宽节点个数分别为 2 与 $m-2$，根据 TFT 协议 $U(h,t)$中的节点都会选择向高带宽节点(节点 h 是高带宽节点)上传，$U(h,t)\subseteq D(h,t+1)$，所以 $D(h,t+1)$中至少有 2 个高上传带宽节点. 用事件 D_2 和 D_{3+} 分别表示 $D(h,t+1)$中有 2 个高上传带宽节点的事件和有不少于 3 个高上传带宽节点的事件. 出现 D_{3+} 是因为有高上传带宽节点 OU 到了节点 h. 事件 D_2 的概率为 N_h 个高上传带宽节点都没有 OU 到节点 h 的概率，由于节点之间相互独立且节点 h 被另一节点 h' OU 到的概率为 $1/N$，所以 $Pr(D_2)=(1-1/N)^{Nh}=1-\alpha+o(\alpha)\approx 1-\alpha$；又由于 $D(p,t+1)$只能出现上述这两种情况，所以 $Pr(D_{3+}=\alpha)$. 根据 TFT 协议，$Pr(T_2\mid D_{3+})=0$，同理可知 $Pr(T_2\mid D_2)=1$. 因此

$$\begin{aligned}Pr(T_2)&=Pr(T_2\mid D_{3+})\times Pr(D_{3+})+Pr(T_2\mid D_2)\times Pr(D_2)\\&=0\times\alpha+1\times(1-\alpha).\end{aligned}$$

当 $m-1\geq 2$ 时，由于 $D(h,t+1)$中的高上传带宽节点个数至少为 2，根据 TFT 协议有 $Pr(T_1)=0$. 因此，当 $m\geq 4$ 时 $\boldsymbol{P}_u(h)[(2,m-2),(2,m-2)]=(1-\alpha)\times(1-\alpha)+0\times\alpha=(1-\alpha)^2$. 当 $m=2$ 时，节点 h 用 TFT 协议从 $D(h,t+1)$中选出 1 个节点，$U(h,t)\subseteq D(h,t+1)$导致该节点一定是高上传带宽节点，所以 $Pr(T_2)=0$，$Pr(T_1)=1$，所以 $\boldsymbol{P}_u(h)[(2,m-2),(2,m-2)]=\alpha$. 当 $m=3$ 时，节点 h 会用 TFT 协议从 $D(h,t+1)$中选出 2 个节点，$U(h,t)\subseteq D(h,t+1)$导致这 2 个节点一定都是高上传带宽的节点，$Pr(T_2)=1$，$Pr(T_1)=0$，有 $\boldsymbol{P}_u(h)[(2,m-2),(2,m-2)]=1-\alpha$.

由于概率转移矩阵与 m 有关，因此可用 $\boldsymbol{P}_{u,m}(h)$来表示不同 m 取值下的概率转移矩阵，有

$$\boldsymbol{P}_{u,3}(h)=\begin{pmatrix}(1-\alpha)^2 & 2(1-\alpha)\alpha & \alpha^2 & 0\\ 0 & (1-\alpha)^2 & 2(1-\alpha)\alpha & \alpha^2\\ 0 & 0 & 1-\alpha & \alpha\\ 0 & 0 & 1-\alpha & \alpha\end{pmatrix}.$$

找到这些矩阵的共同规律即得出式(1). 证毕.

对于 BitTorrent，$\{U_h(t):t=0,1,2,\cdots\}$上的状态转移，如图 3 所示.

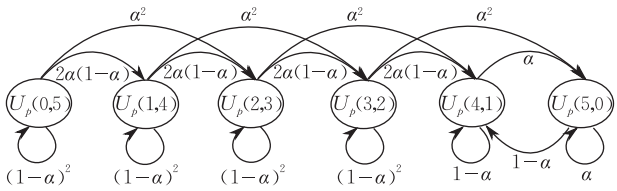


图 3　BitTorrent 中$\{U_h(t):t=0,1,2,\cdots\}$上的状态转移

3.3　高带宽节点 $U_h(t)$：$t=0,1,2,\cdots$的极限分布

定理 2.　令 $\boldsymbol{P}_u^{\infty}(h)=\lim\limits_{n\to\infty}\boldsymbol{P}_u^n(h)$，则 $\boldsymbol{P}_u^{\infty}(h)$为

$$\begin{array}{c}\\(0,m)\\(1,m-1)\\\cdots\\(m-1,1)\\(m,0)\end{array}\begin{array}{c}\begin{array}{ccccc}(0,m) & (1,m-1) & \cdots & (m-1,1) & (m,0)\end{array}\\\begin{pmatrix}0 & 0 & \cdots & 1-\alpha & \alpha\\ 0 & 0 & \cdots & 1-\alpha & \alpha\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & \cdots & 1-\alpha & \alpha\\ 0 & 0 & \cdots & 1-\alpha & \alpha\end{pmatrix}\end{array}\tag{2}$$

证明.　令 $\boldsymbol{P}_u^k(h)[ij]=Pr(U_h(l+k)=j\mid U_h(l)=i)$，$\boldsymbol{P}_u^{\infty}(h)[ij]=\lim\limits_{k\to\infty}\boldsymbol{P}_u^k(h)[ij]$，$f^k(i,j)$为时刻 0 从状态 i 出发在时刻 k 首次到达状态 j 的概率.

以 $m=5$(即 BitTorrent)为例来分析，为了描述的方便，将状态 $U_h(0,5)$，$U_h(1,4)$，$U_h(2,3)$，$U_h(3,2)$记为 0,1,2,3，将状态 $U_h(4,1)$，$U_h(5,0)$记为 4,5，将 $\boldsymbol{P}_u^k(h)[ij]$简记为 $\boldsymbol{P}_{ij}^k$. 根据图 3 显然有，状态 0,1,2,3 为非常返态，状态 4,5 为常返态，因此 $\boldsymbol{P}_{ij}^{\infty}=0\mid_{i\in\{0,1,2,3,4,5\};j\in\{0,1,2,3\}}$. 接下来需要计算 $\boldsymbol{P}_{ij}^{\infty}\mid_{i\in\{0,1,2,3,4,5\};j\in\{4,5\}}$.

$\boldsymbol{P}_{44}^2=\sum\limits_{x=0}^{5}\boldsymbol{P}_{4x}\times\boldsymbol{P}_{x4}=0\times 0+0\times 0+0\times\alpha^2+0\times 2(1-\alpha)\alpha+(1-\alpha)\times(1-\alpha)+(1-\alpha)\times\alpha=1-\alpha$，同样的方法得出，$\boldsymbol{P}_{45}^2=1-\alpha$，$\boldsymbol{P}_{54}^2=\alpha$，$\boldsymbol{P}_{55}^2=\alpha$，应用数学归纳法假设 $\boldsymbol{P}_{44}^k=\boldsymbol{P}_{45}^k=1-\alpha$，$\boldsymbol{P}_{54}^k=\boldsymbol{P}_{55}^k=\alpha$，则根据 $\boldsymbol{P}_{44}^{k+1}=\sum\limits_{x=0}^{5}\boldsymbol{P}_{4x}^k\times\boldsymbol{P}_{x4}$ 有 $\boldsymbol{P}_{44}^{k+1}=1-\alpha$，从而有 $\boldsymbol{P}_{44}^{\infty}=1-\alpha$. 同样归纳出 $\boldsymbol{P}_{45}^{\infty}=1-\alpha$，$\boldsymbol{P}_{54}^{\infty}=\boldsymbol{P}_{55}^{\infty}=\alpha$.

现在求 $\boldsymbol{P}_{14}^{\infty}$. 由于 $\lim\limits_{k\to\infty}\boldsymbol{P}_{14}^k=\lim\limits_{k\to\infty}\sum\limits_{t=1}^{k}f_{14}^t\times\boldsymbol{P}_{44}^{k-t}=\lim\limits_{k\to\infty}\sum\limits_{t=1}^{K}f_{14}^t\times\boldsymbol{P}_{44}^{k-t}+\lim\limits_{k\to\infty}\sum\limits_{t=K+1}^{k}f_{14}^t\times\boldsymbol{P}_{44}^{k-t}$. 固定 K 后：

$\lim_{k\to\infty}\sum_{t=1}^{K} f_{14}^{t}\times \boldsymbol{P}_{44}^{k-t}+\lim_{k\to\infty}\sum_{t=K+1}^{k} f_{14}^{t}\times \boldsymbol{P}_{44}^{k-t}=\sum_{t=1}^{K} f_{14}^{t}\times \boldsymbol{P}_{44}^{\infty}+\lim_{k\to\infty}\sum_{t=K+1}^{k} f_{14}^{t}\times \boldsymbol{P}_{44}^{k-t}$，再令$K\to\infty$有$\lim_{K\to\infty}\sum_{t=1}^{K} f_{14}^{t}\times \boldsymbol{P}_{44}^{\infty}=\sum_{t=1}^{\infty} f_{14}^{t}\times \boldsymbol{P}_{44}^{\infty}$以及$\lim_{K\to\infty}\lim_{k\to\infty}\sum_{t=K+1}^{k} f_{14}^{t}\times \boldsymbol{P}_{44}^{k-t}=0$.

显然，$\sum_{t=1}^{\infty} f_{14}^{t}$为从状态1出发到达状态4(此处该这一事件记为$A$)的概率，定义$B_1$为从状态1出发且在到达状态5之前先到达状态4的事件；而B_2为从状态1出发且在到达状态4之前先到达状态5事件. 由状态迁移图3可知，从状态1出发最终必定到达状态$\{4,5\}$，所以$Pr(B_1\cup B_2)=1$，而显然有B_1和B_2不可能同时发生，即$Pr(B_1\cap B_2)=0$. 因此可用全概率公式得出$\sum_{t=1}^{\infty} f_{14}^{t}=Pr(A)=Pr(A\mid B_1)Pr(B_1)+Pr(A\mid B_2)Pr(B_2)$，而其中$Pr(A\mid B_1)=1$；$Pr(A\mid B_2)=\sum_{t=1}^{\infty} f_{54}^{t}=(1-\alpha)+(1-\alpha)\alpha+(1-\alpha)\alpha^2+\cdots=(1-\alpha)/(1-\alpha)=1$. 因此就有$\sum_{t=1}^{\infty} f_{14}^{t}=Pr(B_1)+Pr(B_2)=1$，也就有$\boldsymbol{P}_{14}^{\infty}=\boldsymbol{P}_{44}^{\infty}=1-\alpha$. 同样的方法可以求出$\boldsymbol{P}_{u,5}^{\infty}(h)$中的其它项.

对于其它m取值，由于满足$\boldsymbol{P}_{(m-1)(m-1)}^{2}=\sum_{x=0}^{m}\boldsymbol{P}_{(m-1)x}\times\boldsymbol{P}_{x(m-1)}=0+0+\cdots+0+(1-\alpha)\times(1-\alpha)+(1-\alpha)\times\alpha=1-\alpha$，其中的0项是由于$\boldsymbol{P}_{(m-1)x}=0$或$\boldsymbol{P}_{x(m-1)}=0$，所以求出的$\boldsymbol{P}_{u,m}^{\infty}(h)[ij]$和$m=5$时有类似的结果，即式(2). 证毕.

3.4 高上传带宽节点的聚簇

定理3. 高上传带宽节点h其$\{U_h(t):t=0,1,2,\cdots\}$的极限分布$\pi_u(h)=(0,\cdots,1-\alpha,\alpha)$，并且与初始分布无关.

证明. 设高上传带宽节点h的初始上传状态的分布为ω，则其极限分布：

$$\pi_u(h)=\omega\times\boldsymbol{P}_u^{\infty}(h)=(0,\cdots,1-\alpha,\alpha)\qquad(3)$$

定理3的结果表明，随着TFT和OU对上传节点的逐步选择，具有高上传带宽的节点最终将选择给同样具有高上传带宽的节点进行数据上传. 定量地说，一个高上传带宽节点平均会将其$(1-\alpha)\times(m-1)/m+\alpha\times m/m=(m-1+\alpha)/m$的上传带宽分配给其它高带宽节点.

3.5 节点聚簇后的宏观带宽分配特征

由于BitTorrent文件共享系统中的总上传带宽等于总下载带宽，在这样的平衡条件下，可以分析出节点聚簇以后高带宽节点集合P_h和低带宽节点集合P_l上带宽分配的宏观特征，该特征如图4所示，其中的U_{hl}表示P_h提供给P_l的总上传带宽. 根据定理3聚簇以后的高带宽节点会以$(1-\alpha)$的概率将其带宽的$1/m$分配给P_l节点，以α的概率不分配带宽给P_l中的节点，所以

$$U_{hh}=(m-1+\alpha)u_hN_h/m\qquad(4)$$

$$U_{hl}=(1-\alpha)u_hN_h/m\qquad(5)$$

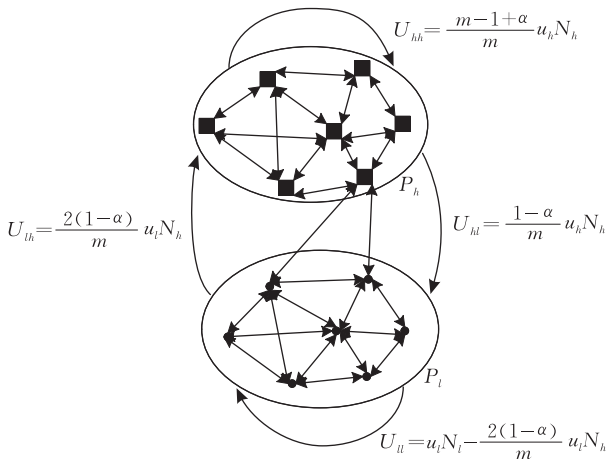


图4 高、低上传带宽节点簇在文件共享系统中的传输结构

现通过对$D(h)$的分析来求U_{ll}和U_{lh}：由于节点h是高上传带宽节点，所以$U(h)$中的节点给h提供下载带宽(TFT的结果)，所以$D(h)$中将至少有m个节点形成$\{n_1,n_2,\cdots,n_{m-1},n_m\}$，其中$\{n_1,n_2,\cdots,n_{m-1}\}$是高带宽节点(这是定理3的结果). 系统中其它$N-1-m$(节点$h$和$n_1,n_2,\cdots,n_{m-1},n_m$除外)个节点也可能给节点$h$上传，因为这些节点OU到节点$h$，而任一节点OU到节点$h$的概率为$1/N$. 对于节点$n_m$，该节点是低上传带宽节点的概率为$1-\alpha$，所以该节点对$U_{lh}$的贡献的期望为$(1-\alpha)u_l/m$. 对于剩下的节点$n_i\mid_{i=m+1,\cdots,N-1}$，这些节点是低带宽节点的概率为$1-\alpha$，都以$1/N$的概率给节点$h$提供上传，这些节点对$U_{lh}$的贡献为$\sum_{i=m+1,\cdots,N-1}(1-\alpha)/N\times u_l/m=(1-\alpha)(N-m-1)/N\times u_l/m\approx(1-\alpha)u_l/m$(假定$m\ll N$)，所以

$$U_{lh}=2(1-\alpha)u_lN_h/m\qquad(6)$$

$$U_{ll}=u_lN_l-2(1-\alpha)u_lN_h/m\qquad(7)$$

分析结果表明P_h和P_l之间的数据传输相比集合内部的传输要少很多，且这种聚类会随着m的增大而加剧. 式(4)～式(7)定量地说明了激励协议对BitTorrent文件传输的影响，而这些数量结果正是

本文的研究基础.

3.6 激励机制下的文件传输结构

对于 file swarming 系统而言，影响文件传输性能的关键是下载同一文件的节点集合（记为 Ψ_F）内各节点间文件块交换的效率. 由于节点会动态加入，会在下载完成后离开，所以 Ψ_F 是一个随时间变化的动态系统，记 $\Psi_F(t)$ 为 t 时刻的 Ψ_F. 在 BitTorrent 激励机制下，$\Psi_F(t)$ 根据节点上传带宽形成若干个紧密连续的部分：高（低）上传带宽节点更多的是和高（低）上传带宽节点进行传输.

不失一般性，此处仍假定 $\Psi_F(t)$ 中包含高上传带宽节点和低上传带宽节点两类，并令 t 时刻 $\Psi_F(t)$ 中高上传带宽节点的个数为 $h(t)$，低上传带宽节点的个数为 $l(t)$. 由于一个新的下载者会以参数为 λ 的泊松过程进入 $\Psi_F(t)$. 所以 $h(t)(l(t))$ 会以参数为 $\lambda_h(\lambda_l)$ 的泊松过程变成 $h(t)+1(l(t)+1)$，其中 $\lambda_h+\lambda_l=\lambda$. 本文假定节点在下载完 F 后马上离开系统，且不在下载中途离开，seed 节点只对文件交换初始化起作用，并令高、低上传带宽节点的下载带宽分别为 c_h 和 c_l，高、低上传带宽节点的文件共享效率为 η_h 和 η_l，借助文献[3-4,11]采用的 fluid 模型，可以构造：

$$\mathrm{d}h(t)/\mathrm{d}t=\lambda_h-\min(c_h h(t),\eta_h U_{hh}+\eta_l U_{lh}),$$

$$\mathrm{d}l(t)/\mathrm{d}t=\lambda_l-\min(c_l l(t),\ \eta_l U_{ll}+\eta_h U_{hl}),$$

其中 $U_{ij}|_{\{i,j\}\in\{h,l\}}$ 就是式(4)～(7)给出的结果，只是其中的 N_h 和 N_l 在这里相应地变成 $h(t)$ 和 $l(t)$，而 $\alpha=h(t)/(h(t)+l(t))$. 对于通常的网络，上传带宽是网络的瓶颈，如 ADSL 的上行带宽就和下行带宽不对称，且大多数 P2P 系统和研究都做同样假定[3-4,11-12]，所以有 $c_h h(t)\geq\eta_h U_{hh}+\eta_l U_{lh}$. 此时有

$$(h(t)+l(t))\begin{pmatrix}h'(t)\\ l'(t)\end{pmatrix}=\begin{pmatrix}-\mu_h\eta_h-\mu_h\eta_l/m & 0\\ \mu_h\eta_l/m & -\mu_l\eta_l\end{pmatrix}\begin{pmatrix}-(m-1)\mu_h\eta_h/m-\mu_l\eta_l/m & \lambda_h & \lambda_h\\ -(m-2)\mu_l\eta_l/m-\mu_h\eta_h/m & \lambda_l & \lambda_l\end{pmatrix}\begin{pmatrix}h^2(t)\\ l^2(t)\\ h(t)l(t)\\ h(t)\\ l(t)\end{pmatrix}\quad(8)$$

4 缩短下载时间优先的 BitTorrent 激励协议

4.1 BitTorrent 文件下载时间

定义 3.　对于任意节点 $p\in\Psi_F$，p 的文件下载时间 $t_F(p)$ 就是指通过 Ψ_F 中节点间的文件块交换使节点 p 下载完成 $F=\{c_1,c_2,\cdots,c_k\}$ 的时间.

定义 4.　从 P2P 文件共享系统角度，文件传输性能体现为平均文件下载时间 T_F：

$$T_F=\sum_{p\in\Psi F}t_F(p)/|\Psi_F|\quad(9)$$

节点 p 以参数为 λ 的泊松过程在 t_1 时刻进入系统，会在 t_2 时刻下载完文件 F 后马上离开 Ψ_F，显然 $t_F(p)=t_2-t_1$，T_F 是系统平稳后节点在系统 $\Psi_F(t)$ 中停留的平均时间.

4.2 BitTorrent 激励对文件下载时间的影响

在激励机制下，$\Psi_F(t)$ 会根据其带宽而汇聚成多个节点类，而节点类内部的文件块交换就成为 BitTorrent 文件块交换的绝对主体. 由于交换对方没有的文件块是文件块交换的前提，而交换节点集合的规模又会随着节点聚簇而缩小，所以节点找到提供感兴趣文件块的节点的机会就会减少，导致有些下载周期会因找不到有效下载节点而浪费，使文件下载时间增长.

首先需对这一影响进行定量分析，其定量特征就体现为式(8)中的 η，就是概率值：$Pr\{$节点 p 能向其连接节点提供感兴趣文件块$\}=1-\prod_{q\in N(p)}Pr(C(p)\subseteq C(q))=1-Pr(C(p)\subseteq C(q))^{|N(p)|}$，其中 $N(p)$ 是节点 p 连向的邻居节点，$C(p)$ 是节点 p 已经下载获得的文件块集合，上式的推导中假定各节点的文件下载是独立的. 文献[3]得出 η 值接近 1，表明在 BitTorrent 中的 rarest first 块选择策略下，文件交换效率很高，但在文献[3]的分析中假定所有文件块都能交换的事实，当出现节点聚簇时，文件块集合也会聚簇，导致一个节点类集合内并不一定总能交换所有的文件块，更可能的情况是当一个节点簇中的文件块交换完成时，其它簇向该簇引入的文件块数量可能不满足簇内带宽的消费，此时就会出现下载带宽浪费.

对于由高、低两类带宽组成的共享系统中，由于高带宽节点簇能很快将簇中的文件块交换完，所以高带宽节点集合会出现上述饥饿现象. 为保证文件块交换的效率，要求低带宽节点集合向高带宽节点集合引入的“新文件块”数量大于高带宽节点集合的总下载能力，即

$$Y\times S\times|P_h|\geq S/u_l\times u_h\times|P_h|\Leftrightarrow Y\geq u_h/u_l \tag{10}$$

其中,Y 是低带宽给高带宽节点引入的新文件块数量;S 是每个文件块的大小;$Y\times S\times|P_h|$ 表示低带宽节点集合向高带宽节点集合引入的可供传输的新数据量;S/u_l 是 Y 个文件块由低带宽节点并行上传给高带宽节点的时间,乘以 u_h 和 $|P_h|$ 是这段时间高带宽节点集合所能下载的总数据量,也就是下载能力.

式(10)中的 Y 与高、低带宽节点集合之间的连接数量、节点集合已经下载的文件块数量、文件块在节点上的分布等因素有关. 令 $C(p)$ 表示节点 p 已经下载到的文件块集合,高、低带宽节点集合之间的连接一个割 $L[P_h,P_l]$. 通过割 L 中的边将 $C(P_l)$ 中的不在 $C(P)$ 中的新块加入 $C(P)$ 中,这个新块数量是一个随机变量,本文将分析其期望(即 Y). 图 5 给出 L、"新块"、$C(P)$、$C(P^c)$、$c(p)$ 等对象之间的关系(其中的 P 就是 P_h).

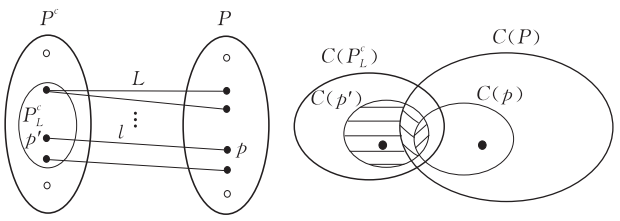


图 5 $C(P)$,$C(P^c)$,割 L 等集合之间的关系

对于割 L 中的连接 $l=(p',p)$,定义随机变量 X_l 为

$$X_l=\begin{cases}1, & \text{如果通过 } l \text{ 引入了新块}\\ 0, & \text{其它}\end{cases}.$$

为完成 L 共引入新块个数的计算,不失一般性假定 l 引入新块的过程是有序的,规定这个连接顺序为 $1,2,\cdots,l,\cdots$. 再将上面的 $X_l=1$ 修改为在 $1,2,\cdots,l-1$ 完成加入后,l 又引入一个"新块". 定义 $Y_{l-1}=X_1+X_2+\cdots+X_{l-1}$ 后有 $Pr(X_l=1|Y_{l-1}=0)\geq(|C(p')-C(P)|)/(|C(p')-C(p)|)$,相应就有 $Pr(X_l=1|Y_{l-1}=i)\geq(|C(p')-C(P)|-i|C(p')|/|C(P^c_L)|)/(|C(p')-C(p)|)$. 其中 $|C(p')|/|C(P^c_L)|$ 表示已经被连接 $1,2,\cdots,l-1$ 引入到 $C(P)$ 的新块再落入 $C(p')$ 中的概率,这是由于(文献[3]表明采用了)rarest first 块选择策略后,每个节点下载到的文件块可认为是 F 上的均匀分布,即任一块等概率的出现在 $C(p')\cup C(p'')$ 中(均匀分布的并仍然是均匀分布),所以每个块将等概率地属于 $C(P^c_L)$. 而被连接 $1,2,\cdots,l-1$ 引入的某块 b 落入 $C(p')$ 中的概率就是

$$\begin{aligned}&Pr(b\in C(p')|b\in C(P^c_L))\\&=Pr(b\in C(p')\cap C(P^c_L))/Pr(b\in C(P^c_L))\\&=Pr(b\in C(p'))/Pr(b\in C(P^c_L))\\&=|C(p')|/|C(P^c_L)|.\end{aligned}$$

据此可以求出 Y_l 的数学期望(详细推导见附录 I):

$$\begin{aligned}&E[Y_l]\geq c_1(1-(1-c_2)^l)/c_2,\\&c_1=E[(|C(p')-C(P)|)/(|C(p')-C(p)|)]\\&c_2=E[|C(p')|/(|C(P^c_L)||C(p')-C(p)|)]\end{aligned} \tag{11}$$

因此在满足下面条件时,高、低带宽节点聚簇不会对文件下载时间产生大的影响.

$$c_1(1-(1-c_2)^{|L|})/c_2\geq u_h/u_l \tag{12}$$

由于只有高带宽节点交换完簇内文件块以后才需要从低带宽节点引入新文件块(此时 $C(p)=C(P)$),代入后有 $c_1=1$,此时式(12)的左边随 c_2 单调递减:

(1) 当 $c_2=o(1/|L|)$ 时,条件(12)变为

$$|L|\geq u_h/u_l \tag{13}$$

$c_2=o(1/|L|)$ 意味着 $|C(P)|$ 较小,割 L 对应的低带宽节点下载到的文件块集合不相交,OU 和 rarest first 机制下此时的 $|L|$ 应该很小,在系统中的节点规模较大时,该条件发生的可能性很低.

(2) 当 c_2 是一个属于 $(1/|L|,u_h/u_l)$ 的常数时,条件(12)变为

$$|L|\geq\log_{1-c_2}1-c_2u_h/u_l \tag{14}$$

(3) 而当 $c_2\geq u_h/u_l$ 时,条件(12)不满足,此时高带宽节点内部没有可供交换的文件块,尽量和低带宽节点相连.

4.3 以文件下载时间为效用的博弈

BitTorrent 的文件下载是一个典型的博弈过程:不同种类的节点通过不断调整其带宽分配策略来达到收益的最大化,虽然一些相关工作也定义和分析了这个博弈[3],但文献[3]的博弈效用定义为节点付出带宽与接收带宽的差值(即公平性),相比这一公平性,在实际的 BitTorrent 文件共享中,用户更关心的节点收益应该是文件下载时间,因此本文将分析以文件下载时间为效用、以上传节点选择为策略的文件块交换博弈.

定理 4. 在节点理性假设下,P2P 文件交换系统中的高带宽节点 h 采用如下上传节点选择策略时节点 h 达到 Nash 平衡:

(1) 当 $|C(h)|<YP$ 时,h 选取上传节点时 m 值

应满足

$$\rho\lambda/((1-1/m)(u_h-u_l))\geq \log_{1-1/(k-|C(h)|)}1-\beta/(k-|C(h)|) \quad (15)$$

(2) 当 $|C(h)|>YP$ 时，h 全选择低带宽节点进行上传.

其中 YP 是一个临界值 $YP=k-\beta$，k 是文件 $F=\{c_1,c_2,\cdots,c_k\}$ 的块数，$\beta=u_h/u_l$.

证明.　根据式(12)和(14)，$c_2=1/\beta$ 就是临界条件. 在 rarest first 机制下可以认为 $|C(p')|/|C(P_L^c)|\approx 1/2$，而对于文件块 c_i，$Pr(c_i\in C(p'))\times Pr(c_i\notin C(p))=(k-|C(p)|)/2k$，根据数学期望的加和特性，$|C(p')-C(p)|$ 的均值为 $(k-|C(p)|)/2$. 联立以后就有临界条件 $C(h)=k-\beta$. 此时 $c_2\approx 1/(k-|C(h)|)$，将该值代入式(15)有 $|L|\geq \log_{1-1/(k-|C(h)|)}1-\beta/(k-|C(h)|)$.

现分析 $|L|$，应该有 $|L|=U_{lh}/(u_l/m)$，但不能用式(6)的 U_{lh} 代入计算，这是由于在以文件下载时间为效用的博弈下，低带宽节点会采取不同于以往的节点选择策略. 在以下载时间为效用的博弈中，高带宽节点之所以 unchoking 低带宽节点原因是内部已无文件可传，而不是由于低带宽节点 OU 到了高带宽节点，因此在该博弈的 Nash 平衡时低带宽选择的策略只 OU 低带宽节点，只有在高带宽节点 OU 到低带宽节点时才 TFT 到高带宽节点. 根据低带宽节点在博弈中选择的策略，式(6)变成 $U_{lh}=(1-\alpha)u_lN_h/m$，将其代入后可得条件 $h(t)(1-\alpha)\geq \log_{1-1/(k-|C(h)|)}1-\beta/(k-|C(h)|)$. 再综合式(8)的结果，在系统稳定时就有条件(15)(详细证明见附录 II).　　证毕.

当 $1/(k-|C(h)|)$ 很小时，$(1-1/(k-|C(h)|))^{\beta}\approx 1-\beta/(k-|C(h)|)$，此时 $\log_{1-1/(k-|C(h)|)}1-\beta/(k-|C(h)|)\approx\beta$，$1/(k-|C(h)|)$ 很小是由于 k 通常很大，如 1000(文件块大小通常为 256 KB，一个数百 MB 的文件有 1000 块左右). 此时式(15)可以近似为

$$1/m\geq 1-\rho\lambda/(\beta(\beta-1)u_l) \quad (16)$$

应用式(20)，$h(t)=m\rho\lambda/((m-1)u_h)$，$l(t)=((m-1-m\rho)\lambda)/((m-1)u_l)$ 和 $N=h(t)+l(t)$，式(16)变为

$$Nm\geq(\beta-1)(1+\beta/\rho-\beta)m-\beta(\beta-1)/\rho \quad (17)$$

表 1 给出了两类典型场景下 m 的取值(其中 ρ 设为 0.2). 从表 1 的数值可以看出：

(1) 对于一定的 β，如果 N 足够大(大于表 1 中给出的“N 的临界值”)，m 可以任意取值，这是由于此时很大的 N 使得 $h(t)$ 也大到无需从 $l(t)$ 那里获得新数据块来维持下载效率；

(2) 随着 β 的增大，对 N 的要求迅速增大(随 $O(\beta^2)$ 增大)，实际上当 β 很大时，如 $\beta\geq 100$ 时，OU 的比率必需很大，否则会导致高带宽节点无文件可传，而 OU 比率很大时高节点处的公平性会很差，但在实际文件系统中还是应该选择加大 OU 比率，因为可以缩短下载时间，这也是本文工作的出发点；

(3) 在 β 适中时，如 $10\leq\beta\leq 30$ 时(是有线 P2P 网络中的典型情况)，适当控制 swarm 的规模和适当控制 m 的大小都能显著提高文件下载效率，文献[13]的工作从侧面反映了这一结果的正确性，另一个有趣的现象是 β 和 N 适中时，此处的 m 和 5(这是实际 BitTorrent 协议中采用的数值)相差不大，在 $\beta=10$，$N=280$ 时算出的 m 正好是 5.

表 1　几类典型场景下的 m 取值

β	N(节点规模)	N 的临界值	m 取值
5	80	84	$m\leq 25$
10	300	369	$m\leq 6.5$
100	10 000	39 699	$m\leq 1.67$
100	1000	39 699	$m\leq 1.28$

定理 5.　在节点理性的假设下，对于 P2P 文件交换系统中的低带宽节点 l 而言，当 l 采取的上传节点选择策略为：节点 l 只选择低带宽节点进行 OU，并采用和 TFT 一样的 choking 策略时，节点 l 将在本文提出的 BitTorrent 文件交换博弈模型下达到 Nash 平衡.

证明.　在定理 4 的证明中已给出，此处略去.　　证毕.

4.4　缩短下载时间优先的激励协议

综合上面给出的理论分析结果，本文针对 BitTorrent 设计了一个缩短下载时间优先的自适应激励协议 AIPS(self-Adaptive Incentive Protocols for file Swarming)，如图 6 所示，其核心是高带宽节

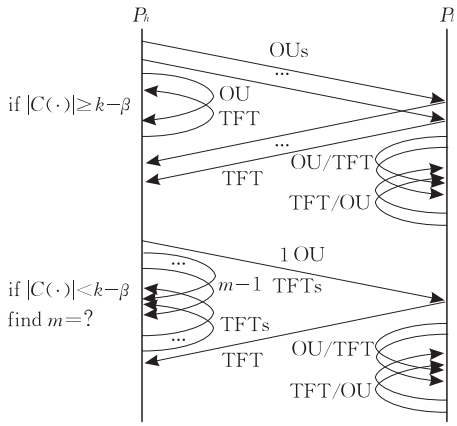


图 6　缩短下载时间优先激励协议 AIPS 的会话图

点要根据$|C(p)|$、$k-\beta$的关系和式(17)对应的等式来决定上传连接中 OU 的数量,低带宽节点只向低带宽节点发起 OU,两类节点的 TFT 原则和现在的 BitTorrent 一样.

5 实验验证及结果分析

5.1 模拟实验平台及参数

本文采用 NetLogo 模拟工具对 PLXU 进行实验验证. NetLogo 是一个基于多 Agent 的 AI 群落模拟工具①,由于 Agent 对象间的通信轻便,特别合适于模拟大规模系统在应用层上的性能表现,TFT 和 OU 机制对 BitTorrent 文件交换性能的影响都是典型的应用层表现. 用 NetLogo 中的一个 Agent 来模拟 BitTorrent 中的一个 leecher,并给每个进入系统的节点初始化随机分配若干文件块. 实验中假定低带宽类节点的带宽为 256 kbps,假定文件块数为 1000,文件块大小为 256 kb. 模拟时将时间分为周期,每个模拟周期所有节点执行一遍 TFT 算法,每隔一定数量的模拟周期后(实验中设为 3)所有节点执行一遍 OU 算法. 实验中用到的符号如表 2 所示.

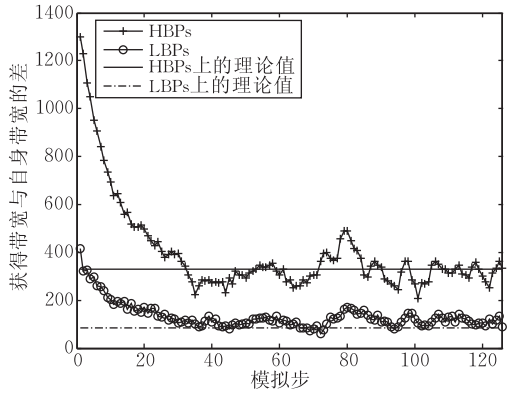
表 2 模拟实验中用到的符号及参数

符号	描述	缺省值
N	系统中的节点个数	—
β	高低带宽节点的带宽之比	20
ρ	进入系统中高带宽节点所占的比率	0.2
m	上传节点选择中 TFT 和 OU 之比($m-1:1$)	5

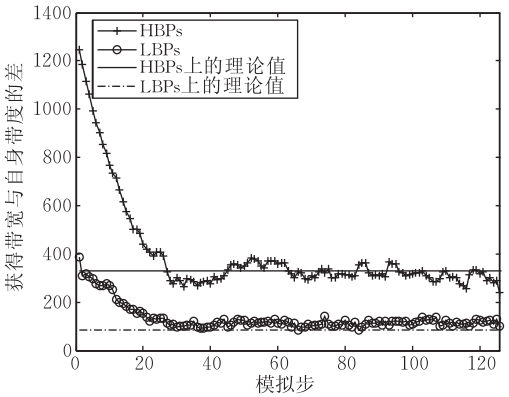
5.2 实验结果与分析

5.2.1 节点按带宽聚类的效果

当节点按带宽聚类时,和节点交互节点的带宽(表现为节点收到的带宽)会接近节点自身的带宽,所以图 7 用节点获得带宽和自身带宽之间的差说明节点聚类特性,由于这个插值随时间下降且停到一个较小的数值上,说明 BitTorrent 激励机制下的节点会出现聚类现象. 另一方面,该差值并不收敛到 0,这是由于高、低带宽节点之间发生的少量数据传输造成的,高带宽节点上述差值的理论计算结果为 $|U_{hh}+U_{lh}-N_h u_h|/N_h=(1-\alpha)(\beta-2)u_l/m$;对低带宽节点该理论值为 $|U_{ll}+U_{hl}-N_l u_l|/N_l=\alpha(\beta-2)u_l/m$. 图 7 的实验参数设定为 $\beta=10$,$\alpha=0.2$,$m=5$,$u_l=256$,可算出 $|U_{hh}+U_{lh}-N_h u_h|/N_h=327.68$,$|U_{ll}+U_{hl}-N_l u_l|/N_l=81.92$(图 7 中的两条横线),侧面验证了本文给出的理论分析结果,另外图 7 的结果说明低带宽节点收敛结果平稳,N 越大收敛结果越平稳,这是随机过程的必然结果.



(a) $N=300$, $\beta=10$, $\alpha=0.2$, $m=5$, $u_l=256$



(b) $N=1000$, $\beta=10$, $\alpha=0.2$, $m=5$, $u_l=256$

图 7 节点在 BitTorrent 文件传输中按带宽聚类的现象

5.2.2 AIPS 对高带宽节点文件下载时间的影响

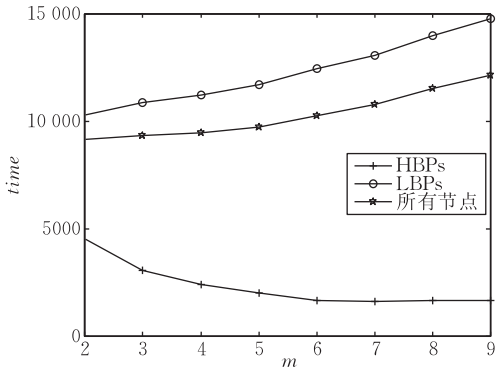
AIPS 和 BitTorrent 激励机制的本质区别在于 BitTorrent 中的 TFT 和 OU 所占的配额是固定的,而 AIPS 是自适应变化的,是随着环境(如网络规模,高低带宽节点所占比例等)的变化而自适应地发生变化. 所以分析 AIPS 对下载效率的影响集中在下载效率与 m 关系上. 表 3 给出了几类典型环境(不同的 β,N,ρ)下 BitTorrent 和 AIPS 对高带宽节点文件下载时间的对比,由于 AIPS 的核心思想是用带宽换取文件块,所以 AIPS 对高带宽节点下载的效率提高是显著的. 结果表明在这个环境下,AIPS 对高带宽节点文件下载效率的提高接近 20%.

表 3 若干环境下 BitTorrent 和 AIPS 的对比

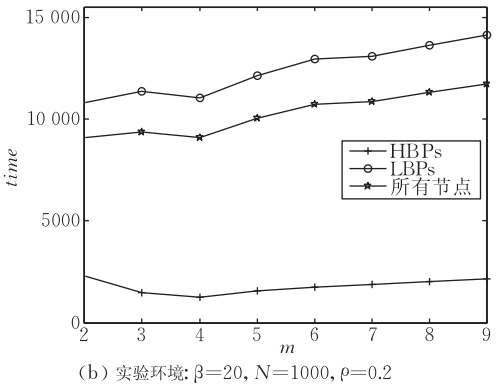
环境参数设置	高带宽节点平均下载时间/s		AIPS 的提高/%
	BitTorrent	AIPS	
$\beta=10$, $N=300$, $\rho=0.2$	1988	1597	19.6
$\beta=10$, $N=300$, $\rho=0.1$	2233	1844	17.4
$\beta=10$, $N=400$, $\rho=0.1$	2184	1791	18
$\beta=20$, $N=300$, $\rho=0.2$	1565	1236	21
$\beta=20$, $N=1000$, $\rho=0.2$	1559	1257	19.4

① Wilensky U. Netlogo. http://ccl.northwestern.edu/netlogo

图 8 给出了 $\beta=10, N=300, \rho=0.2$ 和 $\beta=20, N=1000, \rho=0.2$ 两类环境中文件下载时间($time$)随 m 的变化关系. 式(17)的计算结果表明在下载时间优先的激励机制下两类条件下 m 的取值分别为 6.5 和 3.5. 图 8 的实验结果表明当 m 小于理论计算阈值(式(17)的计算结果)时，会使过多的高带宽上传连接被 unchoking 到低带宽节点那里，当然同时也从低带宽节点获得了大量的新文件块，因此此时高带宽节点的下载时间会因为用过多的高上传带宽换取了低下载带宽而增加，当然低带宽节点的文件下载时间会减小，此时由于高低带宽节点之间的文件块交换频繁，不会由于文件块缺乏而导致带宽浪费，系统平均下载时间会减小. 另一方面，当 m 大于计算阈值时，尤其是 m 很大时，高带宽节点从低带宽节点那里换取的新文件块不足以支持满负荷下载，所以文件的下载时间要增加，而低带宽节点由于很少能从高带宽节点那里获取高下载带宽，其文件下载时间也要增加，当然平均文件下载时间也增加. 因此合适的 m 取值对系统中各类节点文件下载效率的提高与折中具有重要影响.



(a) 实验环境: β=10, N=300, ρ=0.2

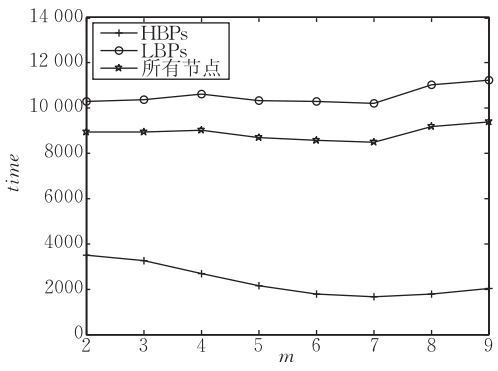


(b) 实验环境: β=20, N=1000, ρ=0.2

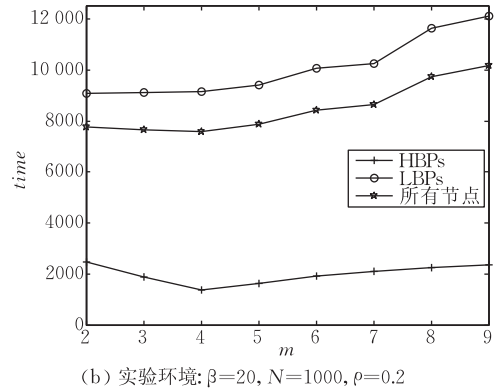
图 8　不同 m 取值对文件下载时间的影响

5.2.3　非诚实环境下 AIPS 的效果分析

如果是在诚实环境下，单从系统平均文件下载时间出发，应该是让 m 越小越好，此时文件块可在系统中的所有节点之间频繁交换，因没有文件块交换而导致下载效率降低的可能性很低. 但另一方面，很小的 m 导致大量的 OU，而 OU 正是导致 free riders 的根本原因[2,14]，所以在存在大量搭便车节点的非诚实环境下，m 值的选取又是越大越好，而 AIPS 激励协议是折中上述两种情况的结果，从缩短文件下载时间角度，要求 m 满足定理 4；从公平性角度(抵抗搭便车行为)，要求 m 尽量大，所以本文的 AIPS 就采用满足定理 4 的那个临界值，虽然文献[9-10]也讨论了 OU 和 TFT 配额对共享效率的影响，但都没有在非诚实环境下面向系统效率针对这一配额给出定量结果. 图 9 给出了搭便车环境下 AIPS 和其它配额调整(包括了文献[9-10]的调整方法)对系统效率影响，选择 $\beta=10, N=300, \rho=0.2$；$\beta=20, N=1000, \rho=0.2$ 两类参数进行实验，并假定其中随机选取 20% 节点作为搭便车节点. 结果表明存在搭便车节点时 AIPS 能显著缩短平均下载时间.



(a) 实验环境: β=10, N=300, ρ=0.2



(b) 实验环境: β=20, N=1000, ρ=0.2

图 9　搭便车环境下不同 m 取值对文件下载时间的影响

6　结束语

由于上传节点的选择会直接导致节点接收到的下载带宽，所以选择哪些节点进行上传将直接影响

BitTorrent 中的节点下载时间以及文件共享系统的效率. BitTorrent 系统中的节点采用 TFT 协议,用大的上传带宽去换取大的下载带宽,提高公平性,但从节点的 Quality of Experience 角度出发,用"大的上传带宽换取到大的下载带宽"目标是为了提高文件下载效率,此时必需满足条件:从换取到下载带宽的节点那里得到想要的文件块. 因此相比上述抽象的公平性而言,实际使用中更加合理的选择策略是用带宽去换取有用的数据块下载. 为了完成这一问题的形式化定义与求解,本文扩展了传统的以公平性为效用的 BitTorrent 节点选择博弈模型,提出了以缩短文件下载时间为效用的博弈模型,综合地刻画出了带宽换带宽以及带宽换文件块的更符合实际环境的节点间博弈结构. 综合随机过程与博弈论等数学手段得出节点在以其缩短文件下载时间优先的博弈中应该采取的策略:(1) 高带宽节点 h 在满足 $k-|C(h)|<\beta$ 条件时,选择满足条件 $Nm\geq(\beta-1)(1+\beta/\rho-\beta)m-\beta(\beta-1)/\rho$ 的参数 m 来用其高带宽换取一定量的高带宽及一定量的新文件块;在满足条件 $k-|C(h)|>\beta$ 时,选择完全用其高带宽来换取有用文件块. (2)低带宽节点总采取用其文件块换取高带宽的策略. 其中的 N,ρ,β 都是能从实际系统中测量出来的参数. 最后的模拟实验结果可反映出本文给出的理论结果及相应策略的正确性和有效性.

参 考 文 献

[1] Yang Xiangying, de Veciana Gustavo. Service capacity of peer to peer networks//Proceedings of the 23rd Conference of the IEEE Computer and Communications Societies. Hong Kong, China, 2004: 2242-2252

[2] Piatek Michael, Isdal Tomas, Anderson Thomas, Krishnamurthy Arvind, Venkataramani Arun. Do incentives build robustness in bittorrent?//Proceedings of the 4th USENIX Symposium on Networked Systems Design & Implementation, Cambridge, MA, USA, 2007: 1-14

[3] Qiu D, Srikant R. Modeling and performance analysis of BitTorrent-Like peer-to-peer networks//Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. Portland, Oregon, USA, 2004: 367-378

[4] Clevenot F, Nain P. A simple fluid model for the analysis of the squirrel peer-to-peer caching system//Proceedings of the 23rd Conference of the IEEE Computer and Communications Societies. Hong Kong, China, 2004: 86-95

[5] Clevenot F, Nain P, Ross K. Multiclass P2P networks: Static resource allocation for service differentiation and bandwidth diversity. Performance Evaluation, 2005, 62(1-4): 32-49

[6] Liao W C, Papadopoulos F, Psounis K. Performance analysis of BitTorrent-like systems with heterogeneous users. Performance Evaluation, 2007, 64(9-12): 876-891

[7] Legout P, Liogkas N, Kohler E, Zhang L. Clustering and sharing incentives in BitTorrent systems. ACM SIGMETRICS Performance Evaluation Review, 2007, 35(1): 301-312

[8] Zhao B Q, Lui J C S, Chiu D M. Analysis of adaptive incentive protocols for P2P networks//Proceedings of the 28th Conference of the IEEE Computer and Communications Societies. Rio de Janeiro, Brazil, 2009: 325-333

[9] Huang K, Wang L, Zhang D, Liu Y. Optimizing the BitTorrent performance using an adaptive peer selection strategy. Future Generation Computer Systems, 2008, 24(7): 621-630

[10] Fan B, Lui J C, Chiu D M. The design trade-offs of BitTorrent-like file sharing protocols. IEEE/ACM Transactions on Networking, 2009, 17(2): 365-376

[11] Massoulie Laurent, Vojnovic Milan. Coupon replication systems. IEEE Transactions on Networking, 2008, 16(3): 603-616

[12] Tian Y, Wu D, Wing K N. Modeling, analysis and improvement for BitTorrent-like file sharing networks//Proceedings of the 25th Conference of IEEE Computer and Communications Societies. Barcelona, Catalunya, Spain, 2006: 1-11

[13] Gyorgy Dan, Niklas Carlsson. Dynamic swarm management for improved BitTorrent performance//Proceedings of the 8th International Workshop on Peer-to-Peer Systems. Boston, MA, USA, 2009: 55-60

[14] Sirivianos M, Park J H, Chen R, Yang X. Free-riding in BitTorrent networks with the large view exploit//Proceedings of the 6th International Workshop on Peer-to-Peer Systems. Bellevue, WA, USA, 2007: 19-24

附录 I. $E[Y_l]$的推导.

$$E[X_l|Y_{l-1}=i]\geq(|C(p')-C(P)|-i|C(p')|/|C(P_L^c)|)/(|C(p')-C(p)|)$$

$$E[E[X_l|Y_{l-1}]]=E[X_l]$$

$$E[X_l]\geq\sum_{i=0}^{l-1}(|C(p')-C(P)|-i|C(p')|/|C(P_L^c)|)/(|C(p')-C(p)|)\times Pr(Y_{l-1}=i)$$

$$\Rightarrow E[X_l]\geq\sum_{i=0}^{l-1}(|C(p')-C(P)|)/(|C(p')-C(p)|)\times Pr(Y_{l-1}=i)-\sum_{i=0}^{l-1}i|C(p')|/(|C(P_L^c)||C(p')-C(p)|)\times Pr(Y_{l-1}=i)$$

$$\Rightarrow E[X_l]\geq(|C(p')-C(P)|)/(|C(p')-C(p)|)-$$

$|C(p')|/(|C(P_L^c)||C(p')-C(p)|)\times E[Y_{l-1}]$

又由于 $E[Y_l]=E[X_l]+E[Y_{l-1}]$

$\Rightarrow E[Y_l]-E[Y_{l-1}]\geq c_1-c_2E[Y_{l-1}]$

其中，

$c_1=(|C(p')-C(P)|)/(|C(p')-C(p)|)$，

$c_2=|C(p')|/(|C(P_L^c)||C(p')-C(p)|)$

$\Rightarrow E[Y_l]\geq(1-c_2)E[Y_{l-1}]+c_1$

$\Rightarrow E[Y_l]\geq(1-c_2)^{l-1}E[Y_{l-1}]+c_1(1-(1-c_2)^{l-1})/c_2$

$E[Y_1]=E[X_1]=Pr(X_1=1)\geq(|C(p')-C(P)|)/(|C(p')-C(p)|)=c_1$

$\Rightarrow E[Y_l]\geq c_1(1-(1-c_2)^l)/c_2$

其中 $E[E[X_l|Y_{l-1}]]=E[X_l]$ 是条件数学期望的直接结果，$E[Y_l]-E[Y_{l-1}]\geq c_1-c_2E[Y_{l-1}]\Rightarrow E[Y_l]-E[Y_{l-1}]\geq E[c_1]-E[c_2]E[Y_{l-1}]$. 因此上式中的 c_1，c_2 是指 $E[c_1]$，$E[c_2]$.

附录 II.　定理 4 证明的补充.

在 $\eta_l=\eta_h=1$ 的假定下，当系统稳定时有

$$\lambda_h=U_{hh}+U_{lh}=(m-1+\alpha)u_hh(t)/m+(1-\alpha)u_lh(t)/m \tag{18}$$

$$\lambda_l=U_{ll}+U_{hl}=(1-\alpha)u_hh(t)/m+u_ll(t)-(1-\alpha)u_lh(t)/m \tag{19}$$

将上面两个式子相加后有

$$\lambda=u_hh(t)+u_ll(t) \tag{20}$$

用式(18)可计算 $h(t)(1-\alpha)$ 为

$$h(t)(1-\alpha)=(mu_hh(t)-m\rho\lambda)/(u_h-u_l) \tag{21}$$

其中 $\rho=\lambda_h/\lambda$，表示进入系统中的高带宽节点的比率，这是一个能从 BitTorrent 测量中得到的一个统计值. 联立式(18)、(20)、(21)以及 α、λ 的定义可得下面的方程：

$$u_h(m-1)(u_h-u_l)h^2(t)-\lambda[(m-1)u_h-u_l+m\rho(u_h-u_l)]h(t)+m\rho\lambda^2=0 \tag{22}$$

为方便求解，可以用下面的一元二次方程来近似式(22)，这是由于上式中的 u_l 较 $(m-1)u_h+m\rho(u_h-u_l)$ 在 β 很大时会相差很多，而在实际系统中，高带宽节点和低带宽节点的带宽之比超过 10 是很常见的现象，再加上 u_h 前面的系数 m，舍去 u_l 导致的误差通常不超过 1%：

$$u_h(m-1)(u_h-u_l)h^2(t)-\lambda[(m-1)u_h+m\rho(u_h-u_l)]h(t)+m\rho\lambda^2=0 \tag{23}$$

求解式(23)的方程可得两个根：$r_1=m\rho\lambda/((m-1)u_h)$ 和 $r_2=\lambda/(u_h-u_l)$，将根 r_2 代入到式(20)会发现求出的 $l(t)<0$，因此这个根是无意义的. 因此将根 r_1 代入到式(21)即有式(15).　　证毕.

LI Zhi-Jun, born in 1977, Ph. D., associate professor. His research interests include peer-to-peer network, wireless network, wireless sensor networks, pervasive computing and operating system.

JIANG Shou-Xu, born in 1968, Ph. D., professor, Ph. D. supervisor. His research interests include wireless sensor networks, pervasive computing and database.

Background

This work was supported by the National Natural Science Foundation of China under Grant No. 60803148 and No. 60973124, by the Doctoral Fund of Ministry of Education of China under Grant No. 20102302110036, and by the Fundamental Research Funds for the Central Universities under Grant No. HIT. NSRIF. 2010. 047.

Peer-to-Peer(or P2P) techniques have influenced the improvement of many research subjects such as Internet, network, operating system, computing communications and have influenced the peoples' daily-life, especially under the prevalence of BitTorrent and P2P stream system. However, the selfness of P2P will hinder the performance of P2P and the incentive mechanisms have become a focus of P2P researches. Nowadays, the incentive mechanisms for P2P are all fairness driven, i. e. aiming to implement the balance between the supply and consumption. But comparing the pure fairness, the nodes in real P2P systems are prefer to the efficiency of the download. Therefore, a download time first self-adaptive incentive protocol is provided and implemented in this paper for BitTorrent. Specifically, a game with file download efficiency as its utility is designed to analyze the incentive strategy taken by nodes, and the download time first self-adaptive incentive strategies are depicted by the Nash equilibrium of the game. With best of our knowledge, this paper is the first work to design the incentive protocol aiming at the improvement of the file sharing efficiency in BitTorrent. Sufficient simulation results show that the incentive protocol provided in this paper can improve the file download efficiency by 20%.

基于 Vague 集相似度量的图像隐写系统安全性测度

欧阳春娟[1),2),3)] 李 斌[1),2)] 李 霞[1),2)] 王 娜[1),2)]

[1)](深圳大学信息工程学院 广东 深圳 518060)

[2)](深圳市现代通信与信息处理重点实验室 广东 深圳 518060)

[3)](井冈山大学电子信息工程学院 江西 吉安 343009)

摘 要 由于图像隐写所引起的各种统计特征变化是不确定的,文中将 Vague 集相似度量引入隐写系统的安全性评价中.从图像的一阶统计特征和二阶统计特征两方面,定义了基于载体数据及载密数据相关 Vague 集相似度量的隐写系统一阶和二阶安全性测度,证明了该安全性测度的有界性,对称性和一致性.在假设图像满足独立同分布的条件下,证明了所提出的两种安全性测度是等价的.结果表明,所提出的二阶安全性比一阶安全性能更好地反映隐写引起的载体统计分布变化.与确定模式下的安全性测度相比,当嵌入率低于 0.5 比特/像素,新的测度可更好地度量隐写系统安全性,因此该测度对小容量的隐写及隐写分析算法设计具有更好的指导作用.

关键词 隐写;安全性;Vague 集;相似度量;有界性

中图法分类号 TP391 **DOI 号**: 10.3724/SP.J.1016.2012.01510

A New Security Evaluation for Steganographic System Based on Vague Set Similarity Measure

OUYANG Chun-Juan[1),2),3)] LI Bin[1),2)] LI Xia[1),2)] WANG Na[1),2)]

[1)](*College of Information and Engineering, Shenzhen University, Shenzhen, Guangdong* 518060)

[2)](*Shenzhen Key Laboratory of Modern Communications and Information Processing, Shenzhen, Guangdong* 518060)

[3)](*College of Electronics and Information Engineering, Jinggangshan University, Ji'an, Jiangxi* 343009)

Abstract Since variations of statistical features caused by image steganography are indeterministic, vague set similarity measure is introduced to steganographic system for security evaluation in this paper. The first-order and the second-order security measure based on vague set similarity measure are defined to measure the similarity between cover images and stego images in the sense of the first-order and the second-order image statistical feature, respectively. The newly defined security measures are then proved to have the properties of boundedness, symmetry and unity. Finally, under the assumption that the pixels' statistical distribution is identically independent distributed, the first-order and the second-order security measure are proved to be equivalent. Simulation results show that the second-order security measure can always capture more obvious statistical changes than the first-order security measure. In addition, compared with the security measure defined under a deterministic image statistical distribution model, the newly defined security measure is more sensitive when the embedding rate is lower than 0.5 bit per pixel, thus providing better guidance for the design of steganography and steganalysis with a small embedding rate.

Keywords steganography; security; vague set; similarity measure; boundedness

收稿日期:2011-01-26;最终修改稿收到日期:2011-05-26.本课题得到国家自然科学基金(61171124,61103174,60902069,61005049)、广东省科技计划项目(2011B010200045)、广东省高校优秀青年创新人才基金(LYM10116)、深圳市重点实验室提升项目(CXB201105060068A)资助.**欧阳春娟**,女,1974年生,博士研究生,副教授,主要研究方向为信息隐藏及分析.E-mail:oycj001@163.com.**李 斌**,男,1982年生,博士,讲师,主要研究方向为信息隐藏及模式识别.**李 霞**(通信作者),女,1968年生,博士,教授,博士生导师,主要研究领域为智能优化、智能计算及应用.E-mail:lixia@szu.edu.cn.**王 娜**,女,1977年生,博士,教授,主要研究领域为智能优化和模式识别.

1 引　言

隐写和隐写分析是信息隐藏研究领域的一个重要方面[1]，作为一种新的信息安全手段，其安全性研究必然是最为重要的课题. 在研究初期，大多数文献都集中在研究新的隐写方法上，导致算法的安全性往往依赖于算法的保密，这违背了 Kerckhoffs 准则[2]. 还有一些算法盲目追求大容量，而忽略了安全性考虑. 目前，完整的隐写技术理论体系尚未建立，对隐写系统安全性的讨论仍集中于具体的隐写和隐写分析方法，通用的安全性定义或模型方面的研究成果比较少. 然而建立完善的安全性测度模型是隐写系统最基本的问题，其对于隐写及隐写分析技术的研究起着重要的指导意义.

目前，隐写系统使用较广泛的安全性评价是 Cachin[3] 在假设获得了载体数据和载密数据确定的概率分布函数，提出的相关熵的 ε-secure 指标. Wang 等人[4] 将 Cachin 的结果扩展到多元高斯变量，对 Cachin 理论中的载体数据进行数学建模，该模型只考虑了由高斯平稳过程生成的载体，对复杂的载体图像，没有给出安全性定义. Sullivan 等人[5] 考虑图像像素相关性，将图像建模为马尔可夫链(Markov Chain，MC)模型，根据载体数据和载密数据对应 MC 经验矩阵的散度距离，定义了隐写系统的安全性. 文献[6] 陈丹等人采用图像可逆的去相关变换后的相对熵，定义了隐写系统的安全性. Ambalavanan 等人[7] 根据 Markov 随机场(Markov Random Feild，MRF)模型下的条件相对熵，提出了数字图像的隐写安全性定义. MRF 与 MC 相比，包含的图像相关性信息更全面，但模型复杂度及计算量都极大. Pevný 等人[8] 采用高维特征集对图像进行建模，定义了隐写系统最大平均差异(Maximum Mean Discrepancy，MMD)安全性测度. 张湛等人[9-10] 将图像建模为 n 阶马尔可夫链(n-MC)模型，采用 Hilbert 扫描方式构建图像的 n 阶马尔可夫链，利用其经验矩阵的散度距离定义隐写系统的安全性，对阶数 n 进行调节，可调整 n-MC 模型的计算复杂度.

上述文献都是在假设获得了确定的载体数据与载密数据的某种统计分布模式下，借鉴 Shannon 信息论中的通信模型思想描述隐写系统安全性. 然而，概率分布估计问题仍是目前尚未很好解决的统计学难题. 统计学描述的是观察空间整体的概率分布，其对于单个样本的概率分布描述是没有意义的. 在隐写及隐写分析的实际应用中，无法获得无限多的载体数据样本或载密数据样本，所以也无法获得载体数据和载密数据精确的概率分布. 此外，自然图像是一个非平稳过程，隐写将引起图像各局部细节变化，图像的各种统计特征会发生不确定性变化. 因此，上述文献采用确定的信息处理方法进行隐写系统安全性评价具有一定的局限性. 本文将图像隐写过程建模为一个模糊不确定性模型，利用载体数据与载密数据相关的 Vague 集相似度量更合理地定义了隐写系统的一阶及二阶安全性测度，并证明了该安全性测度的有界性、对称性和一致性. 在假设图像满足独立同分布条件下，证明了所提出的一阶和二阶安全性测度的等价性. 仿真实验验证了本文提出的基于 Vague 集相似度量的隐写系统安全性测度对不同隐写算法的通用性，并且该测度可区分不同隐写算法在同一嵌入率下的安全性. 测量同一算法在不同嵌入率下的安全性取值表明，二阶安全性测度比一阶安全性测度能更好地反映隐写引起的载体统计特征分布变化. 此外，与确定模式下安全性测度相比较，当嵌入率较低时，Vague 集相似度量的隐写安全性测度取得了更高的反应灵敏度值，说明该测度对小嵌入量的隐写及隐写分析算法设计具有更好的指导意义. 隐写算法设计实验也表明，Vague 集相似度量的安全性测度指导设计的隐写算法抗隐写分析能力更强，因此，该测度可以更好地指导设计高安全性的隐写算法.

2 确定模式下图像隐写系统一阶及二阶统计分布安全性测度

2.1 图像隐写系统一阶统计分布安全性测度

文献[3]的隐写安全性测度定义为：载体数据构成集合 C，假设载体图像样本集合 c 和载密图像样本集合 s 分别为 C 上的一个随机变量，且具有确定的概率分布函数，分别为 P_c，P_s. 采用 P_c 和 P_s 的相对熵(relative entropy)定义图像隐写系统的安全性. 从图像一阶统计特征角度，该定义可以理解如下：

载体图像样本集合为

$$\begin{bmatrix} x \\ P_c(x) \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ p_c(x_1) & p_c(x_2) & \cdots & p_c(x_n) \end{bmatrix};$$

载密图像样本集合为

$$\begin{bmatrix} x \\ P_s(x) \end{bmatrix}=\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ p_s(x_1) & p_s(x_2) & \cdots & p_s(x_n) \end{bmatrix},$$

其中，$x_1,x_2,\cdots,x_n$ 为图像像素值，所有像素值构成集合 X；$p_c(x_1)$，$p_c(x_2)$，$\cdots$，$p_c(x_n)$ 及 $p_s(x_1)$，$p_s(x_2)$，$\cdots$，$p_s(x_n)$ 为对应的载体图像与载密图像像素的概率分布. 定义 $P_c(x)$ 和 $P_s(x)$ 两个集合的相对熵（也称为 Kullback-Leibler 距离）为隐写系统的安全性测度.

$$D(P_c \parallel P_s)=\sum_{x\in X}P_c(x)\log\frac{P_c(x)}{P_s(x)} \tag{1}$$

可证明 $D(P_c \parallel P_s)\geq 0$. 当 $D(P_c \parallel P_s)=0$，称此隐写系统是绝对安全的；当 $0\leq D(P_c \parallel P_s)\leq \varepsilon$，称此隐写系统是 ε-安全的. 由于很难估计高维数据的概率分布函数，该安全性测度在实际应用中通常采用独立同分布等简单模型来估计概率分布函数.

2.2 图像隐写系统二阶统计分布安全性测度

为了更好地捕捉图像像素之间的相关性，文献[5]通过计算图像马尔可夫链模型二阶统计分布的散度距离，定义了图像隐写系统的二阶统计分布安全性. 设 C 和 S 分别为按某种给定的扫描方式（如行、列、zig-zag、Hilbert 等方式）扫描载体图像和载密图像得到的一阶马尔可夫链，$\boldsymbol{M}_c$、$\boldsymbol{M}_s$ 分别为 C 和 S 的经验矩阵. 定义 $\boldsymbol{M}_c$、$\boldsymbol{M}_s$ 的散度距离为图像隐写前后的二阶统计分布的改变，即二阶统计分布安全性测度.

$$D(\boldsymbol{M}_c,\boldsymbol{M}_s)=\sum_{i,j\in R}M^c_{ij}\log\left(\frac{M^c_{ij}}{\sum\limits_j M^c_{ij}}\frac{\sum\limits_j M^s_{ij}}{M^s_{ij}}\right) \tag{2}$$

其中 $\dfrac{M^c_{ij}}{\sum\limits_j M^c_{ij}}$，$\dfrac{M^s_{ij}}{\sum\limits_j M^s_{ij}}$ 分别代表载体图像和载密图像中从像素 i 到像素 j 的转移概率. 像素 $i,j\in R$，$[0,R]$ 为图像像素的取值范围. 可证明 $D(\boldsymbol{M}_c,\boldsymbol{M}_s)\geq 0$. 当且仅当 $\boldsymbol{M}_c=\boldsymbol{M}_s$ 时，$D(\boldsymbol{M}_c,\boldsymbol{M}_s)=0$，此时隐写系统是绝对安全的.

上述图像隐写系统一阶统计分布和二阶统计分布的安全性测度，分别从图像的一阶统计特征（如一维直方图特征），二阶统计特征（如差分直方图、二维直方图特征）两方面，对隐写系统的安全性给出了定义. 其对隐写系统的安全性评价具有一定的指导作用. 但这些安全性定义必须基于一个假设：假设无限的载体数据与载密数据的概率分布是可以明确得到的，其概率转移矩阵是确定的. 然而，图像表示的客观世界中自然景物的视觉信息非常丰富，包括灰度、颜色、纹理、形状等. 对图像数据处理形式也是多种多样的. 所以，图像隐写操作会引起图像各方面变化，同时存在很多不确定的因素（如在传输过程中受到噪声影响）. 隐写后的图像统计特征会产生动态变化，针对某一具体的图像像素值进行统计，它们隶属于某区域的隶属程度往往都是不确定的. 因此，本文将图像隐写过程建模为一个模糊不确定性模型，采用载体数据和载密数据相关 Vague 集相似度量可更合理地定义图像隐写系统的安全性.

3 Vague 集相关知识

模糊集方法只能处理确定的隶属和非隶属信息，对未确定的隶属信息无效[11]. Gau 和 Buehrer[12] 于 1993 年提出了 Vague 集理论，它将模糊集理论中的隶属函数单值扩充为一个区间. Vague 集理论采用真假隶属函数定义了元素对模糊概念的属于与不属于的程度和证据，与模糊集相比，具有更强的表达不确定性能力.

定义 1[12-13]. 设 $X=\{x_1,x_2,\cdots,x_n\}$ 是一个论域，$V(x)$ 表示 X 上的所有 Vague 集的集合，$V(x)$ 中任意的 Vague 集 A 可用一个真隶属函数 t_A 和一个假隶属函数 f_A 表示，$t_A:X\to[0,1]$，$f_A:X\to[0,1]$，其中 $t_A(x_i)$ 为由支持 x_i 的证据所导出的肯定隶属度的下界，$f_A(x_i)$ 则是由反对 x_i 的证据所导出的否定隶属度的下界，且 $t_A(x_i)+f_A(x_i)\leq 1$. 元素 x_i 在 Vague 集 A 中的隶属度被区间 $[0,1]$ 的一子区间 $[t_A(x),1-f_A(x)]$ 所界定，称该区间为 x_i 在 A 中的 Vague 值，记为 $V_A(x_i)$.

对于 Vague 集 A，当 X 离散时，记为

$$A=\sum_{i=1}^{n}[t_A(x_i),1-f_A(x_i)]/x_i,\ x_i\in X.$$

定义 2[14-15]. 假定 $X=\{x_1,x_2,\cdots,x_n\}$，A,B 为 X 上的两个 Vague 集，

$$E(A)=-\frac{1}{n\ln 2}\sum_{i=1}^{n}[t_A(x_i)\ln t_A(x_i)+f_A(x_i)\ln f_A(x_i)] \tag{3}$$

为 Vague 集 A 的熵（entropy）.

$$E_B(A)=-\sum_{i=1}^{n}[t_B(x_i)\ln t_A(x_i)+f_B(x_i)\ln f_A(x_i)] \tag{4}$$

为 Vague 集 A 关于 Vague 集 B 的偏熵（partial entropy）.

Vague 集合之间的相似度量可以很好地表示两个集合的匹配性[14-16]. 在实际的图像隐写系统安全性研究中,根本无法得到无限载体图像数据样本和载密图像数据样本精确的概率分布. 由于图像样本的有限性,图像内部不同景物之间存在明显的差异性,同一景物内部灰度、颜色、纹理具有的同一性,图像隐写引起这些性质改变的不确定性,导致载体图像数据样本集合和载密图像数据样本集合的概率分布等统计数据是不确定的. 因此,对于载体数据样本集合和载密数据样本集合的概率分布等统计数据采用离散的 Vague 集合来描述更合理.

4　基于 Vague 集相似度量的图像隐写系统安全性测度

对于隐写系统,安全性是其首要指标,包括不可检测性和不可感知性两方面要求. 在保证载体样本与载密样本之间满足一定的感知失真限制下,一个隐写系统可描述如下:

称系统 $\Sigma=(F,G,M,C,K,S)$ 为隐写系统,其中 C 为载体数据样本集合,M 为消息源,S 为载密数据样本集合,K 为密钥源,满足

(1) $F: C\times M\times K\to S$ 为隐写映射;

(2) $G: S\times K\to M$ 为提取映射;

(3) 对任取 $c\in C, m\in M, k\in K$, $G(F(c,m,k),k)=m$.

4.1　Vague 集相似度量的图像隐写系统一阶安全性测度

定义 3.　对隐写系统 $\Sigma=(F,G,M,C,K,S)$,其中 C 为载体信号集合,M 为消息源,S 为载密信号集合,K 为密钥源,定义 PC 和 PS 分别为载体图像数据样本集合 C 和载密图像数据样本集合 S 的概率分布集合. 所有样本的概率分布集合组成论域 $X=\{x_1,x_2,\cdots,x_n\}$,$V(x)$ 表示 X 上所有 Vague 集的集合,则 PC 和 PS 为 X 上的两个 Vague 集,

$$PC=\sum_{i=1}^{n}[t_{PC}(x_i),1-f_{PC}(x_i)]/x_i,\ x_i\in X,$$

$$PS=\sum_{i=1}^{n}[t_{PS}(x_i),1-f_{PS}(x_i)]/x_i,$$

$$x_i\in X,\ i\in[1,n],$$

$[1,n]$ 为图像像素的取值范围. 将 Vague 集 PC 和 PS 的相似度量 $T(PC,PS)$ 定义为图像隐写系统的一阶安全性测度:

$$T(PC,PS)=\frac{n\ln 2(E(PC)+E(PS))}{E_{PC}(PS)+E_{PS}(PC)}\tag{5}$$

其中 $E(PC)$,$E(PS)$ 分别为 Vague 集 PC 和 PS 的熵,$E_{PC}(PS)$ 为 Vague 集 PS 关于 PC 的偏熵,$E_{PS}(PC)$ 为 Vague 集 PC 关于 PS 的偏熵. 规定 $0\times\ln 0=0$.

当 $T(PC,PS)=1$,隐写系统是绝对安全的. 当 $T(PC,PS)=T$, $T\in(0,1)$, $\varepsilon=1-T$,则隐写系统是 ε-安全的. 当 PC 完全隶属于 X,而 PS 完全不隶属于 X,即当且仅当 $V_{PC}(x_i)=[1,1,\cdots,1]$, $V_{PS}(x_i)=[0,0,\cdots,0]$ 时,$T(PC,PS)$ 取最小值 0,此时隐写系统是绝对不安全的.

定理 1.　设 $T(PC,PS)$ 为图像隐写系统基于 Vague 集相似度量的一阶安全性测度. 则 $T(PC,PS)$ 有如下性质:

(1) 有界性. $0\le T(PC,PS)\le 1$;

(2) 对称性. $T(PC,PS)=T(PS,PC)$;

(3) 一致性. $T(PC,PS)=1\Leftrightarrow PC=PS$.

证明.　(1)

$$\begin{aligned}&E_{PC}(PS)+E_{PS}(PC)-n\ln 2(E(PC)+E(PS))\\&=-\sum_{i=1}^{n}[t_{PC}(x_i)\ln t_{PS}(x_i)+f_{PC}(x_i)\ln f_{PS}(x_i)+\\&\quad t_{PS}(x_i)\ln t_{PC}(x_i)+f_{PS}(x_i)\ln f_{PC}(x_i)]+\\&\quad\frac{n\ln 2}{n\ln 2}\sum_{i=1}^{n}[t_{PC}(x_i)\ln t_{PC}(x_i)+f_{PC}(x_i)\ln f_{PC}(x_i)+\\&\quad t_{PS}(x_i)\ln t_{PS}(x_i)+f_{PS}(x_i)\ln f_{PS}(x_i)]\\&=\sum_{i=1}^{n}\left[t_{PC}(x_i)\ln\frac{t_{PC}(x_i)}{t_{PS}(x_i)}+f_{PC}(x_i)\ln\frac{f_{PC}(x_i)}{f_{PS}(x_i)}+\right.\\&\quad\left.t_{PS}(x_i)\ln\frac{t_{PS}(x_i)}{t_{PC}(x_i)}+f_{PS}(x_i)\ln\frac{f_{PS}(x_i)}{f_{PC}(x_i)}\right].\end{aligned}$$

根据不等式: $\ln x\ge 1-\frac{1}{x}$,上式有

$$\begin{aligned}&\ge\sum_{i=1}^{n}\left[t_{PC}(x_i)\left(1-\frac{t_{PS}(x_i)}{t_{PC}(x_i)}\right)+f_{PC}(x_i)\left(1-\frac{f_{PS}(x_i)}{f_{PC}(x_i)}\right)+\right.\\&\quad\left.t_{PS}(x_i)\left(1-\frac{t_{PC}(x_i)}{t_{PS}(x_i)}\right)+f_{PS}(x_i)\left(1-\frac{f_{PC}(x_i)}{f_{PS}(x_i)}\right)\right]\\&=\sum_{i=1}^{n}[t_{PC}(x_i)-t_{PS}(x_i)+f_{PC}(x_i)-f_{PS}(x_i)+\\&\quad t_{PS}(x_i)-t_{PC}(x_i)+f_{PS}(x_i)-f_{PC}(x_i)]\\&=0,\end{aligned}$$

所以

$$E_{PC}(PS)+E_{PS}(PC)-n\ln 2(E(PC)+E(PS))\ge 0,$$

所以

$$E_{PC}(PS)+E_{PS}(PC)\ge n\ln 2(E(PC)+E(PS)),$$

又因为 $t_{PC}(x_i)$, $t_{PS}(x_i)$, $f_{PC}(x_i)$, $f_{PS}(x_i)$ 都属于

$[0,1]$,且规定 $0\times\ln 0=0$,

所以 $E(PC),E(PS),E_{PC}(PS),E_{PS}(PC)$都是非负,

所以 $0\leq T(PC,PS)\leq 1$.

(2) $T(PC,PS)=\dfrac{n\ln 2(E(PC)+E(PS))}{E_{PC}(PS)+E_{PS}(PC)}$,而

$$T(PS,PC)=\frac{n\ln 2(E(PS)+E(PC))}{E_{PS}(PC)+E_{PC}(PS)}.$$

所以 $T(PC,PS)=T(PS,PC)$.

(3) 在证明(1)中

$$E_{PC}(PS)+E_{PS}(PC)-n\ln 2(E(PC)+E(PS))$$
$$=\sum_{i=1}^{n}\left[t_{PC}(x_i)\ln\frac{t_{PC}(x_i)}{t_{PS}(x_i)}+f_{PC}(x_i)\ln\frac{f_{PC}(x_i)}{f_{PS}(x_i)}+\right.$$
$$\left.t_{PS}(x_i)\ln\frac{t_{PS}(x_i)}{t_{PC}(x_i)}+f_{PS}(x_i)\ln\frac{f_{PS}(x_i)}{f_{PC}(x_i)}\right].$$

上式等于 0 的充要条件是

$$t_{PC}(x_i)=t_{PS}(x_i),f_{PC}(x_i)=f_{PS}(x_i),$$

所以 $T(PC,PS)=1\Leftrightarrow PC=PS$. 证毕.

4.2 Vague 集相似度量的图像隐写系统二阶安全性测度

基于 Vague 集相似度量的图像隐写系统一阶安全性测度定义受到假设图像像素点是独立同分布的限制.事实上,图像各像素之间存在很强的相关性.本文进一步考虑图像像素相关性,利用图像马尔可夫链模型二阶统计分布的 Vague 集相似度量定义隐写系统的二阶安全性测度.首先,对载体图像和载密图像,按某种给定的扫描方式(如行、列、zig-zag、Hilbert 等方式)分别得到马尔可夫链序列 C 和 S,$\boldsymbol{MC}$、$\boldsymbol{MS}$ 分别为 C 和 S 的经验矩阵.同理,将经验矩阵 $\boldsymbol{MC}$、$\boldsymbol{MS}$ 定义为两个 Vague 集,可更准确地描述图像像素转移的不确定性.

定义 4. 设 C 和 S 分别为按给定扫描方式得到的载体图像和载密图像的马尔可夫链,$\boldsymbol{MC}$、$\boldsymbol{MS}$ 分别为 C 和 S 的经验矩阵.m_{ij} 为经验矩阵的元素,表示从像素 i 到像素 j 的转移概率.所有经验矩阵的元素组成论域 M_{ij},$V(m)$ 表示 M_{ij} 上的所有 Vague 集的集合,则 $\boldsymbol{MC},\boldsymbol{MS}$ 为论域 M_{ij} 上的两个 Vague 集,

$$\boldsymbol{MC}=\sum_{i,j=1}^{n}[t_{\boldsymbol{MC}}(m_{ij}),1-f_{\boldsymbol{MC}}(m_{ij})]/m_{ij},\ m_{ij}\in M_{ij},$$

$$\boldsymbol{MS}=\sum_{i,j=1}^{n}[t_{\boldsymbol{MS}}(m_{ij}),1-f_{\boldsymbol{MS}}(m_{ij})]/m_{ij},\ m_{ij}\in M_{ij}.$$

将 Vague 集 $\boldsymbol{MC}$ 和 $\boldsymbol{MS}$ 的相似度量 $T(\boldsymbol{MC},\boldsymbol{MS})$ 定义为图像隐写系统的二阶安全性测度:

$$T(\boldsymbol{MC},\boldsymbol{MS})=\frac{n\ln 2(E(\boldsymbol{MC})+E(\boldsymbol{MS}))}{E_{\boldsymbol{MC}}(\boldsymbol{MS})+E_{\boldsymbol{MS}}(\boldsymbol{MC})}\tag{6}$$

其中

$$E(\boldsymbol{MC})=-\frac{1}{n\ln 2}\sum_{i,j=1}^{n}[t_{\boldsymbol{MC}}(m_{ij})\ln t_{\boldsymbol{MC}}(m_{ij})+f_{\boldsymbol{MC}}(m_{ij})\ln f_{\boldsymbol{MC}}(m_{ij})]\tag{7}$$

$$E_{\boldsymbol{MS}}(\boldsymbol{MC})=-\sum_{i,j=1}^{n}[t_{\boldsymbol{MS}}(m_{ij})\ln t_{\boldsymbol{MC}}(m_{ij})+f_{\boldsymbol{MS}}(m_{ij})\ln f_{\boldsymbol{MC}}(m_{ij})]\tag{8}$$

$E(\boldsymbol{MC})$ 为 Vague 集 $\boldsymbol{MC}$ 的熵,$E_{\boldsymbol{MS}}(\boldsymbol{MC})$ 为 Vague 集 $\boldsymbol{MC}$ 关于 Vague 集 $\boldsymbol{MS}$ 的偏熵 $i,j\in[1,n]$,$[1,n]$为图像像素的取值范围.同理可定义 $E(\boldsymbol{MS}),E_{\boldsymbol{MC}}(\boldsymbol{MS})$.规定 $0\times\ln 0=0$.

当 $T(\boldsymbol{MC},\boldsymbol{MS})=1$,隐写系统是绝对安全的.当 $T(\boldsymbol{MC},\boldsymbol{MS})=T,T\in(0,1),\varepsilon=1-T$,系统是 ε-安全的.当 $\boldsymbol{MC}$ 完全隶属于 M_{ij},而 $\boldsymbol{MS}$ 完全不隶属于 M_{ij},即当且仅当 $V_{\boldsymbol{MC}}(m_{ij})=[1,1,\cdots,1]$,$V_{\boldsymbol{MS}}(m_{ij})=[0,0,\cdots,0]$时,$T(\boldsymbol{MC},\boldsymbol{MS})$取得最小值 0,此时,隐写系统是绝对不安全的.

定理 2. 设 $T(\boldsymbol{MC},\boldsymbol{MS})$为图像隐写系统基于 Vague 集相似度量的二阶安全性测度.则 $T(\boldsymbol{MC},\boldsymbol{MS})$有如下性质:

(1) 有界性. $0\leq T(\boldsymbol{MC},\boldsymbol{MS})\leq 1$;

(2) 对称性. $T(\boldsymbol{MC},\boldsymbol{MS})=T(\boldsymbol{MS},\boldsymbol{MC})$;

(3) 一致性. $T(\boldsymbol{MC},\boldsymbol{MS})=1\Leftrightarrow\boldsymbol{MC}=\boldsymbol{MS}$.

定理 2 的证明可参照定理 1 证明(略).

定理 3. 假设载体图像与载密图像的像素分布满足独立同分布条件,则图像隐写系统基于 Vague 集相似度量的一阶安全性测度 $T(PC,PS)$ 与二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$等价.

证明. 设载体图像与载密图像的像素分布满足独立同分布条件,m_{ij} 为马尔可夫链 MC 模型经验矩阵中的元素,i,j 为图像像素值,x_i,x_j 分别为像素值 i,j 的概率分布.则

$$t_{\boldsymbol{MC}}(m_{ij})=t_{PC}(x_ix_j)=t_{PC}(x_i)t_{PC}(x_j),$$
$$f_{\boldsymbol{MC}}(m_{ij})=f_{PC}(x_ix_j)=f_{PC}(x_i)f_{PC}(x_j),$$
$$t_{\boldsymbol{MS}}(m_{ij})=t_{PS}(x_ix_j)=t_{PS}(x_i)t_{PS}(x_j),$$
$$f_{\boldsymbol{MS}}(m_{ij})=f_{PS}(x_ix_j)=f_{PS}(x_i)f_{PS}(x_j).$$

且
$$\sum_{j=1}^{n}t_{PC}(x_j)=1,\ \sum_{j=1}^{n}f_{PC}(x_j)=1,$$
$$\sum_{j=1}^{n}t_{PS}(x_j)=1,\ \sum_{j=1}^{n}f_{PS}(x_j)=1.$$

由此可得

$$
\begin{aligned}
T(\boldsymbol{MC},\boldsymbol{MS}) &= \frac{n\ln 2(E(\boldsymbol{MC})+E(\boldsymbol{MS}))}{E_{\boldsymbol{MC}}(\boldsymbol{MS})+E_{\boldsymbol{MS}}(\boldsymbol{MC})} \\
&= \frac{\sum_{i,j=1}^{n}\left[t_{\boldsymbol{MC}}(m_{ij})\ln t_{\boldsymbol{MC}}(m_{ij})+f_{\boldsymbol{MC}}(m_{ij})\ln f_{\boldsymbol{MC}}(m_{ij})+\cdots+t_{\boldsymbol{MS}}(m_{ij})\ln t_{\boldsymbol{MS}}(m_{ij})+f_{\boldsymbol{MS}}(m_{ij})\ln f_{\boldsymbol{MS}}(m_{ij})\right]}{\sum_{i,j=1}^{n}\left[t_{\boldsymbol{MC}}(m_{ij})\ln t_{\boldsymbol{MS}}(m_{ij})+f_{\boldsymbol{MC}}(m_{ij})\ln f_{\boldsymbol{MS}}(m_{ij})+\cdots+t_{\boldsymbol{MS}}(m_{ij})\ln t_{\boldsymbol{MC}}(m_{ij})+f_{\boldsymbol{MS}}(m_{ij})\ln f_{\boldsymbol{MC}}(m_{ij})\right]} \\
&= \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[t_{PC}(x_ix_j)\ln t_{PC}(x_ix_j)+f_{PC}(x_ix_j)\ln f_{PC}(x_ix_j)+\cdots+t_{PS}(x_ix_j)\ln t_{PS}(x_ix_j)+f_{PS}(x_ix_j)\ln f_{PS}(x_ix_j)\right]}{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[t_{PC}(x_ix_j)\ln t_{PS}(x_ix_j)+f_{PC}(x_ix_j)\ln t_{PS}(x_ix_j)+\cdots+t_{PS}(x_ix_j)\ln t_{PC}(x_ix_j)+f_{PS}(x_ix_j)\ln f_{PC}(x_ix_j)\right]} \\
&= \Big(\sum_{i=1}^{n}\Big[t_{PC}(x_i)\sum_{j=1}^{n}t_{PC}(x_j)\ln\Big(t_{PC}(x_i)\sum_{j=1}^{n}t_{PC}(x_j)\Big)+\cdots+f_{PC}(x_i)\sum_{j=1}^{n}f_{PC}(x_j)\ln\Big(f_{PC}(x_i)\sum_{j=1}^{n}f_{PC}(x_j)\Big)+\cdots+ \\
&\quad t_{PS}(x_i)\sum_{j=1}^{n}t_{PS}(x_j)\ln\Big(t_{PS}(x_i)\sum_{j=1}^{n}t_{PS}(x_j)\Big)+\cdots+f_{PS}(x_i)\sum_{j=1}^{n}f_{PS}(x_j)\ln\Big(f_{PS}(x_i)\sum_{j=1}^{n}f_{PS}(x_j)\Big)\Big]\Big)\Big/ \\
&\quad \Big(\sum_{i=1}^{n}\Big[t_{PC}(x_i)\sum_{j=1}^{n}t_{PC}(x_j)\ln\Big(t_{PS}(x_i)\sum_{j=1}^{n}t_{PS}(x_j)\Big)+\cdots+f_{PC}(x_i)\sum_{j=1}^{n}f_{PC}(x_j)\ln\Big(f_{PS}(x_i)\sum_{j=1}^{n}f_{PS}(x_j)\Big)+\cdots+ \\
&\quad t_{PS}(x_i)\sum_{j=1}^{n}t_{PS}(x_j)\ln\Big(t_{PC}(x_i)\sum_{j=1}^{n}t_{PC}(x_j)\Big)+\cdots+f_{PS}(x_i)\sum_{j=1}^{n}f_{PS}(x_j)\ln\Big(f_{PC}(x_i)\sum_{j=1}^{n}f_{PC}(x_j)\Big)\Big]\Big) \\
&= \frac{\sum_{i=1}^{n}\left[t_{PC}(x_i)\ln t_{PC}(x_i)+f_{PC}(x_i)\ln f_{PC}(x_i)+\cdots+t_{PS}(x_i)\ln t_{PS}(x_i)+f_{PS}(x_i)\ln f_{PS}(x_i)\right]}{\sum_{i=1}^{n}\left[t_{PC}(x_i)\ln t_{PS}(x_i)+f_{PC}(x_i)\ln f_{PS}(x_i)+\cdots+t_{PS}(x_i)\ln t_{PC}(x_i)+f_{PS}(x_i)\ln f_{PC}(x_i)\right]} \\
&= \frac{n\ln 2(E(PC)+E(PS))}{E_{PC}(PS)+E_{PS}(PC)} \\
&= T(PC,PS).
\end{aligned}
$$

证毕.

本文从 Vague 集的相似度量出发，定义了图像隐写系统的一阶和二阶安全性测度，并证明了该安全性测度满足有界性、对称性和一致性. 该安全性测度的有界性克服了图像在确定模式下文献[3]及文献[5]提出的图像隐写安全性测度取无穷大的缺陷，使隐写系统安全性测度对隐写引起载体的统计分布改变限定在一定范围内，从而对统计分布改变的反映更为明显. 当载体数据和载密数据相关的两个 Vague 集完全相同时，安全性测度取得最大值 1，此时隐写系统是绝对安全的. 当载体数据相关的 Vague 集完全地隶属于所对应的论域，而载密数据相关的 Vague 集完全不属于所对应的论域时，安全性测度取得最小值 0，此时隐写系统是绝对不安全的. 定理 3 证明了当图像分布满足独立同分布时，基于 Vague 集相似度量的图像隐写系统的一阶安全性与二阶安全性是等价的，从理论上统一了基于 Vague 集相似度量的图像隐写安全性测度.

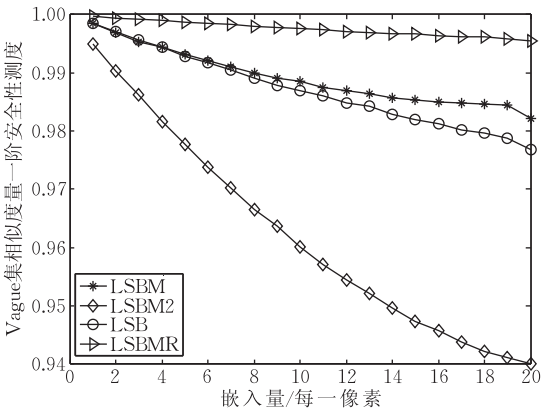
5　实验结果与讨论

由以上的理论分析及证明可知，基于 Vague 集相似度量的图像隐写系统安全性测度同确定模式下的图像隐写系统安全性测度相比更具合理性. 为了说明该测度在处理实际隐写系统安全性问题时具有以上优势，实验首先采用 Vague 集相似度量的图像隐写系统一阶和二阶安全性测度度量不同隐写算法在不同嵌入率下的安全性，验证该测度的有效性和通用性. 其次，采用本文提出的一阶和二阶安全性测度分别与文献[3]及文献[5]提出的确定模式下的安全性测度相比较，通过度量同一隐写算法在不同嵌入率下的安全性，说明 Vague 集相似度量的隐写系统安全性测度的性能优势. 最后，用两种模式的安全性测度指导设计隐写算法，通过比较隐写算法的抗分析能力，说明 Vague 集相似度量的隐写系统安全性测度具有更好的指导设计隐写算法能力.

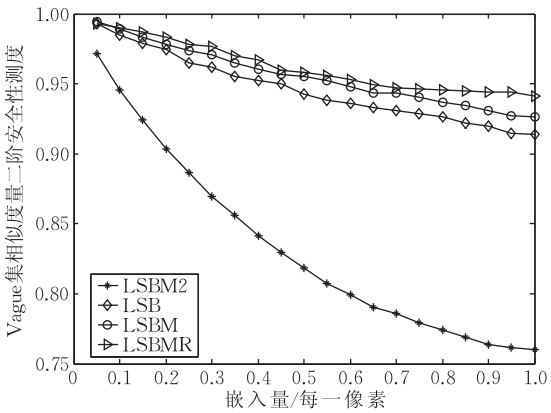
5.1　验证 Vague 集相似度量的图像隐写系统安全性测度的通用性

实验采用两组不同隐写算法，对图像进行隐写操作，比较不同隐写算法在同一嵌入率下的 Vague 集相似度量的图像隐写系统安全性. 第 1 组为空域隐写算法，包括 LSB 替换[17]、LSB±1[18]、LSB±2 和 LSB Matching Revisited[19] 4 种隐写算法（分别

记作 LSB、LSBM、LSBM2 和 LSBMR). 第 2 组隐写算法为 DCT 域隐写算法，包括 JSteg①、F3、F4 和 F5[20] 4 种隐写算法. 在第 1 组隐写算法中，采用 NRCS 图像库② 中的 1542 幅图像，将其裁剪为 1024×1024，并转化为灰度图像. 在每幅图像中嵌入相同的秘密信息，嵌入率从 0.05 bpp 增至 1 bpp，步长为 0.05 bpp，对图像进行隐写得到载密图像. 针对每种隐写算法，计算 1542 幅图像的 Vague 集相似度量的一阶和二阶安全性测度 $T(PC,PS)$ 和 $T(\boldsymbol{MC},\boldsymbol{MS})$ 在同一嵌率下的均值，如图 1 所示. 在第 2 组隐写算法中，4 种算法都是在图像分块的不为零 DCT 系数上嵌入秘密信息，数据嵌入率较低. 为了比较不同算法的安全性，选取 USC-SIPI 标准图像库③ “girl.bmp”、“lake.bmp”、“lena.bmp” 和 “baboon.bmp”4 个标准图像，在不同的图像中嵌入相同容量的秘密信息，计算这 4 幅图像分别针对不同 DCT 域隐写算法的 Vague 集相似度量隐写系统的一阶和二阶安全性测度，如表 1 所示.



(a) 一阶安全性 $T(PC,PS)$ 随嵌入率的改变图



(b) 二阶安全性 $T(\boldsymbol{MC},\boldsymbol{MS})$ 随嵌入率的改变图

图 1 不同隐写算法在不同嵌入率下的 Vague 集相似度量安全性测度

表 1 不同 DCT 域隐写算法在相同嵌入容量下的 Vague 度集相似度量的安全性测度

图像	一阶安全性测度 $T(PC,PS)$				二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$			
	Jsteg	F3	F4	F5	Jsteg	F3	F4	F5
girl.bmp	0.98219	0.9837	0.98533	0.98610	0.92327	0.92573	0.92791	0.92933
lake.bmp	0.83355	0.83427	0.83838	0.83918	0.81342	0.81475	0.81635	0.81894
lena.bmp	0.96555	0.97155	0.97239	0.97673	0.93456	0.93657	0.93678	0.93825
baboon.bmp	0.93130	0.93470	0.93673	0.93856	0.91146	0.91388	0.91659	0.91713

图 1(a)为针对第 1 组隐写算法的一阶安全性测度 $T(PC,PS)$ 均值随嵌入率增加时的变化曲线图. 由图 1(a)可知，所有的曲线都满足随着嵌入率增加，$T(PC,PS)$ 下降的趋势，即符合隐写系统随着嵌入率增加，安全性下降的规律. 从图 1(a)可得，对于空域中的 4 种隐写算法，在同样的嵌入率下，不同隐写算法的安全性是不同的. LSBMR 的 $T(PC,PS)$ 取值最大，安全性最高，其次是 LSBM、LSB，最低是 LSBM2. 其中，LSBM2 隐写的安全性与其它 3 种相比，安全性下降较大. 该实验结果与 LSBMR 的改变率为 0.375/像素，LSBM、LSB 改变率为 0.5/像素，LSBM2 改变率为 1.0/像素的理论分析得出 4 种算法的安全性排序完全相符. 图 1(b)为针对第 1 组空域隐写算法的 Vague 集相似度量的二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$ 均值随嵌入率增加时的变化曲线图，可得到类似的分析结果. 表 1 为在相同嵌入量下，DCT 域中 4 种隐写算法的 Vague 集相似度量的隐写系统一阶安全性 $T(PC,PS)$ 和二阶安全性 $T(\boldsymbol{MC},\boldsymbol{MS})$ 取值. 由表 1 可知，对于同一图像在相同的嵌入量下，F5 算法的 $T(PC,PS)$ 和 $T(\boldsymbol{MC},\boldsymbol{MS})$ 取值与其它 3 种隐写算法的安全性相比，取值最大，安全性最高，其次是 F4 算法、F3 算法，最低是 JSteg 隐写. 该实验结果与这 4 种隐写算法的理论安全性分析排序相符合. 由此可见，本文提出的基于 Vague 集相似度量的图像隐写系统安全性测度具有通用性和有效性.

5.2 本文提出的隐写安全性测度与确定模式下安全性测度相比的优越性

本实验采用Vague集相似度量的隐写系统一

① Upham D. Jsteg. http://zooid.org/~paul/crypto/jsteg/

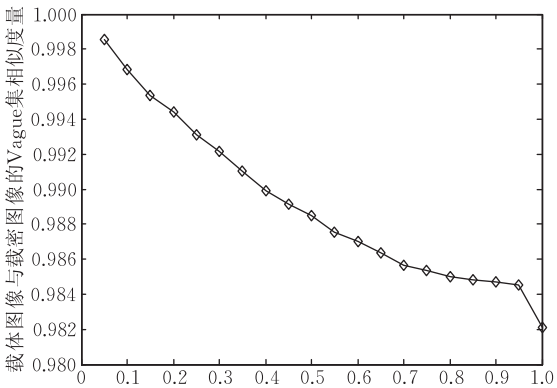
② Natural resources conservation service photo gallery[EB/OL], available: http://photogallery.nrcs.usda.gov/

③ USC-SIPI Image Database [DB/OL]. Available: http://sipi.usc.edu/services/database/index.html

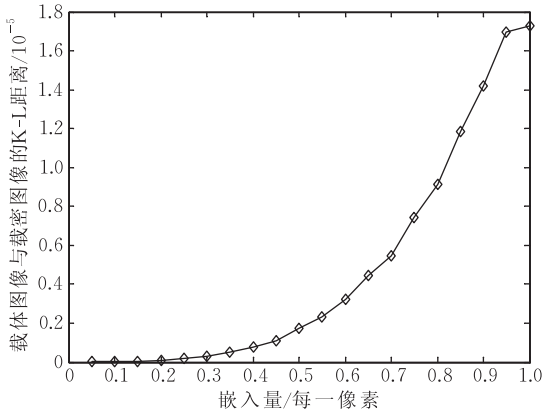
阶和二阶安全性测度分别与文献[3]在假设图像分布为独立同分布时的相对熵安全性测度，及文献[5]在确定模式下的安全性测度进行比较. 通过比较两种模式下的测度对同一隐写算法在不同嵌入率下引起的图像统计分布变化的反应灵敏度，从数学角度定量地说明模糊 Vague 集隐写系统一阶和二阶安全性测度的优越性. 实验同样采用 NRCS 图像库中的 1542 幅图像，将其转化为灰度图像，裁剪大小为 1024×1024. 嵌入率从 0.05 bpp 增至 1 bpp，步长为 0.05 bpp，采用 LSB±1 分别进行隐写嵌入. 计算在同一嵌入率下，针对 1542 幅图像的 Vague 集相似度量的一阶安全性 $T(PC,PS)$ 均值和确定模式下文献[3]所提的一阶统计分布的安全性 $D(P_c \| P_s)$ 均值. 其中图 2(a)为 $T(PC,PS)$ 均值随嵌入率增加时的变化曲线图，图 2(b)为 $D(P_c \| P_s)$ 均值随嵌入率增加时的变化曲线图. 定义 $\delta=\Delta y/y$ 为安全性测度对隐写引起变化的反应灵敏度，其中 Δy 为在一定的嵌入率变化下安全性测度值的变化量，y 为嵌入率 0～1 变化时安全性测度的总变化量. 对于图 2(a)，$\delta_T=(\Delta|T(PC,PS)|)/(|T(PC,PS)|)$，对于图 2(b)，$\delta_D=(\Delta|D(P_c \| P_s)|)/(|D(P_c \| P_s)|)$. δ 取值越大，说明该安全性测度在嵌入率变化下，对隐写所引起的统计分布反映更灵敏，从而能更精确地对隐写安全性进行度量. 由图 2 可得出以下结论：

(1) 由图 2(a)可知，$T(PC,PS)$ 的取值范围为有限区间[0,1]，随着嵌入率的增加，$T(PC,PS)$ 取值变小，隐写系统的安全性逐渐下降. 由图 2(b)可知，$D(P_c \| P_s)$ 的取值范围为无限区间$[0,+\infty)$随着嵌入率的增加，$D(P_c \| P_s)$ 取值增大，安全性降低. 两种描述相比，采用 $T(PC,PS)$ 的取值范围在有限区间中，取值范围更好控制，描述隐写系统安全性符合随着嵌入率增加，安全性下降，安全测度取值变小的单调下降规律.

(2) 由图 2(b)可知，当嵌入率从 0～0.5 变化时，$\delta_D\approx(0.2\times10^{-5})/(2\times10^{-5})=0.1$，说明当嵌入率小于 0.5 时，其反应灵敏度较低. 而由图 2(a)可知，在嵌入率从 0～0.5 的变化过程中，$\delta_T\approx(1-0.988)/(1-0.982)\approx0.67$，$\delta_T$ 约为 δ_D 的 7 倍. 且在整个嵌入率变化过程中，δ_T 反应灵敏度取值都比较一致，说明 Vague 集一阶安全性测度可以清晰地度量不同嵌入率下的隐写安全性. 而嵌入率较低时，确定模式下的隐写安全性测度 $D(P_c \| P_s)$ 不能清晰地反映隐写引起的各种统计分布变化.



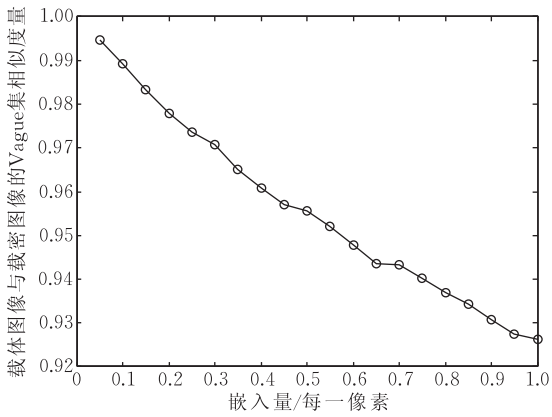
(a) $T(PC,PS)$随嵌入率增加的变化图



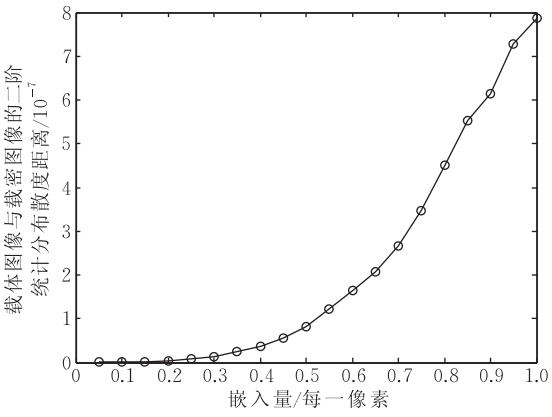
(b) $D(P_c \| P_s)$随嵌入率增加的变化图

图 2　Vague 集一阶安全性与文献[3]安全性随嵌入量增加的变化图

图 3(a)、(b)是在嵌入量从 0.05 bpp 增至 1 bpp，步长为 0.05 bpp 下，Vague 集相似度量二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$ 均值和文献[5]确定模式下的安全性测度 $D(\boldsymbol{M}_c,\boldsymbol{M}_s)$ 均值随嵌入率增加的变化曲线图. 可得出与图 2 相类似的结论. 图 2(a)和图 3(a)分别为采用 Vague 集相似度量的一阶安全性测度 $T(PC,PS)$ 和二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$，对同一种隐写算法 LSBM 在嵌入率从 0～1 变化时的安全性测度取值. 由两图可得，一阶安全性测度 $T(PC,PS)$ 取值在[0.98211,0.99855]之间，测度变化总量为 0.01644. 而二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$ 取值在[0.92617,0.99465]之间，测度变化总量为 0.06848，约为 $T(PC,PS)$ 变化总量的 4 倍. 可见，$T(\boldsymbol{MC},\boldsymbol{MS})$ 能更好地反映隐写引起的载体数据统计特征变化. 因此，二阶安全性测度 $T(\boldsymbol{MC},\boldsymbol{MS})$ 与一阶安全性测度 $T(PC,PS)$ 相比，二阶安全性测度对隐写及隐写分析算法的设计具有更好的指导作用.



(a) $T(\boldsymbol{MC}, \boldsymbol{MS})$随嵌入率增加的变化图



(b) $D(\boldsymbol{M}_c, \boldsymbol{M}_s)$随嵌入率增加的变化图

图 3 Vague 集二阶安全性与文献[5]安全性随嵌入量增加的变化图

5.3 两种模式下的安全性测度指导设计隐写算法比较实验

本实验采用 Vague 集相似度量的隐写系统一阶和二阶安全性测度分别与文献[3]及文献[5]在确定模式下的安全性测度来指导设计隐写算法. 通过比较两种模式的安全性测度指导设计的隐写算法的抗隐写分析能力,来说明本文提出的模糊 Vague 集隐写系统安全性测度对隐写算法设计具有更好的指导作用. 在 LSB±1 隐写(记作 LSBM)算法中,当秘密信息与载体像素的最低有效位相同时,不更改载体图像像素,当两者不相同时,对载体像素进行随机加减 1 得到载密图像. 本实验采用粒子群算法[21]优化隐写嵌入过程中需要修改像素的加减 1 序列. 通过对图像不同的加减 1,尽量保持图像的统计特征,提高隐写算法的安全性. 粒子的初解为 0,1 序列,其中 1 代表加 1,0 代表减 1,序列的长度为隐写过程中需进行加减 1 的像素个数. 在优化过程中,首先以文献[3]中确定模式下的安全性测度 $D(P_c \parallel P_s)$为优化目标,该目标函数取值越小,隐写图像的安全性越好,得到以确定模式下的一阶统计分布的安全性测度为指导的改进 LSBM 隐写算法(记作 PSO_LSBMD1). 再以 Vague 集相似度量的一阶安全测度 $T(PC, PS)$为目标函数进行优化,该目标函数取值越大,隐写图像的安全性越好,得到以 Vague 集相似度量的一阶安全性测度为指导的改进 LSBM 隐写算法(记作 PSO_LSBMT1). 针对 NRCS 图像库中的 1542 幅图像,全部转化为灰度图像,裁剪大小为 512×512,粒子数为 30,迭代次数为 20,用以上 3 种算法在嵌入率为 1.0 bpp 下得到加密图库. 采用文献[22]中(该文献以一维质心和二维质心特征进行隐写分析)的隐写分析算法,对以上 3 种隐写算法得到的加密图库及原图库提取特征进行隐写分析. 用 Fisher 作为分类器,其中 500 幅图像为训练集,其余的图像为测试集. 实验得到的 ROC 曲线(Receiver Operating Characteristic Curve)图如图 4 所示. 由图可知,PSO_LSBMT1 和 PSO_LSBMD1 两种隐写算法与 LSBM 算法相比,具有更低的 AUC(Area under ROC Curve)值. 说明采用两种安全性测度均可指导设计安全性更高的隐写算法. 同时,PSO_LSBMT1 获得了比 PSO_LSBMD1 更小的 AUC 值为 0.6634,可见,PSO_LSBMT1 具有更强的抵抗隐写分析能力. 因此,采用本文提出的 Vague 集相似度量的隐写安全性测度与确定模式下的安全性测度相比,指导设计高安全性的隐写算法能力更强. 分析其原因,是由于采用 Vague 集相似度量的隐写安全性测度,充分考虑了图像隐写引起的不确定性,用模糊 Vague 集相似度量来计算载体数据与载密数据相关统计量之间的差距. Vague 集相似度量实际上是用分段的方式进行度量两者的差距,所以比用确定方式度量两者的差距更精确.

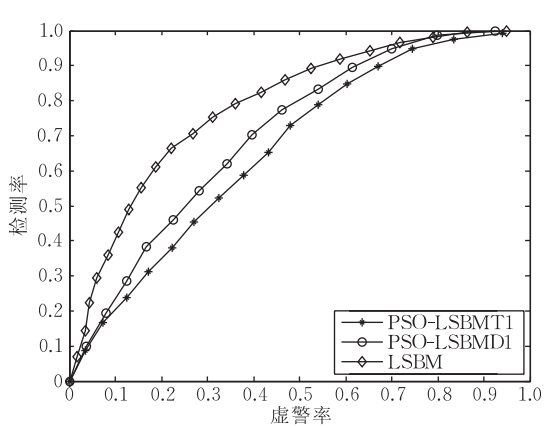


图 4 Vague 集相似度量的一阶安全性与文献[3]的安全性测度指导设计的隐写算法 ROC 曲线图

采用文献[5]中确定模式下的安全性测度 $D(\boldsymbol{M}_c, \boldsymbol{M}_s)$和 Vague 集相似度量的二阶安全性测度

$T(\boldsymbol{MC},\boldsymbol{MS})$按以上方式指导设计改进的 LSBM 算法，分别记为 PSO_LSBMD2 和 PSO_LSBMT2. 图 5 为这 3 种算法按以上相同的实验设计得到的隐写分析 ROC 曲线图，可得到与图 4 相似的实验结论.

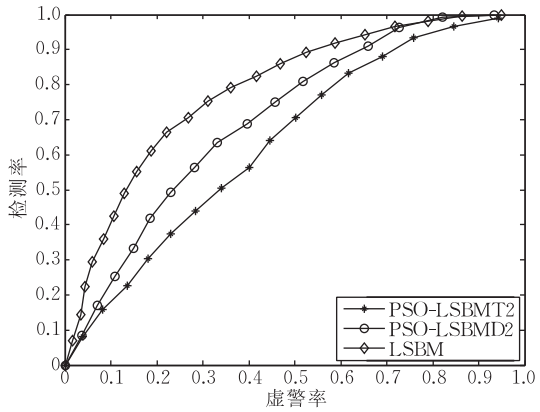


图 5　Vague 集相似度量的二阶安全性与文献[5]的安全性测度指导设计的隐写算法 ROC 曲线图

6　结束语

Vague 集理论作为一种新的不确定性智能信息处理方法，具有比 Fuzzy 集更强的数据表达能力. Vague 集之间的相似度量表示了两集合的匹配程度. 本文根据图像数据的复杂性与隐写引起图像统计特征变化的不确定性，在假设图像为独立同分布和马尔可夫链模型下，分别定义了基于 Vague 集相似度量的隐写系统一阶和二阶安全性，并证明了该安全性测度的有界性、对称性和一致性. 在假设图像为独立同分布条件下，证明了 Vague 集相似度量的隐写系统一阶和二阶安全性测度的等价性. 通过仿真实验，验证了基于 Vague 集相似度量的隐写系统安全性测度对不同的隐写算法均可得到随着嵌入率增大，隐写系统安全性下降的结论，该测度同时可区分不同隐写算法在同一嵌入率下的安全性. 通过度量同一隐写算法在不同嵌入率下的安全性可得，二阶安全性测度比一阶安全性测度能更好地反映隐写引起的载体数据统计特征变化. 此外，与同条件确定模式下的安全性测度相比，在嵌入率低于 0.5 bpp 时，Vague 集相似度量的隐写系统安全性测度反应灵敏度远高于确定模式下的反应灵敏度，说明该测度对小嵌入量的隐写及隐写分析算法具有更好的指导作用. 隐写算法设计实验进一步表明，Vague 集相似度量的隐写安全性测度指导的隐写算法抗分析能力更强，说明了该测度对隐写算法设计的指导能力更强.

一阶马尔可夫链模型中任一像素只受前一像素的影响，考虑了图像像素间的相关性信息，但由于自然图像每一像素点至少与其邻域内像素有着较强的相关性. 因此，使用一阶马尔可夫链模型仍将丢失一些像素相关信息，当分析者利用更复杂的模型提取统计特征或分析计算能力特别强时，采用 Vague 集相似度量的图像隐写系统一阶和二阶安全性测度将会具有一定的局限性. n 阶马尔可夫链模型可较全面地描述数字图像相关性信息，所包含图像相邻像素间的相关性信息量更全面. 如何定义 n-MC 模型下的 Vague 集相似度量的隐写系统安全性测度，以及利用 Vague 集相似度量的隐写系统安全性测度指导设计高安全性的隐写算法及高性能的隐写分析算法，均可作为下一步研究方向.

参　考　文　献

[1] Wang Shuo-Zhong, Zhang Xin-Peng, Zhang Wei-Ming. Recent advances in image-based steganalysis research. Chinese Journal of Computers, 2009, 32(7): 1247-1263(in Chinese)
(王朔中，张新鹏，张卫明. 以数字图像为载体的隐写分析研究进展. 计算机学报，2009，32(7)：1247-1263)

[2] Wang Yu-Min, Liu Jian-Wei. The Security of Communication Network——Theory and Technology. Xi'an: Xidian University Press, 1999(in Chinese)
(王育民，刘建伟. 通信网的安全——理论与技术. 西安：西安电子科技大学出版社，1999)

[3] Cachin C. An information-theoretic model for steganography. Information and Computation, 2004, 192(1): 41-56

[4] Wang Y, Moulin P. Steganalysis of block-structured stego text//Proceedings of the SPIE: Security, Steganography and Watermarking of Multimedia Contents VI. Bellingham, WA, USA, 2004, 5306: 477-488

[5] Sullivan K, Madhow U, Chandrasekaran S et al. Steganalysis for Markov cover data with applications to images. IEEE Transactions on Information Forensics and Security, 2006, 1(2): 275-287

[6] Chen Dan, Wang Yu-Min. The steganography security definition based on image model. Journal of Harbin Institute of Technology, 2006, 38(Supplement): 901-904(in Chinese)
(陈丹，王育民. 基于图像模型的掩密安全性定义. 哈尔滨工业大学学报，2006，38(增刊)：901-904)

[7] Ambalavanan A, Chandramouli R. A Bayesian image steganalysis approach to estimate the embedded secret message length//Proceedings of the 7th Workshop on Multimedia and Security. New York, USA, 2005: 33-38

[8] Pevný T, Fridrich J. Benchmarking for steganography//Proceedings of the 10th Information Hiding International Workshop. Santa Barbara, CA, USA, 2008: 251-267

[9] Zhang Zhan, Qu Fang, Liu Guang-Jie et al. A novel security evaluation method for digital image steganography based on high-order Markov Chain model. Information and Control, 2010, 39(4): 455-461(in Chinese)
(张湛, 瞿芳, 刘光杰等. 基于高阶 Markov 链模型的数字图像隐写安全性评估方法. 信息与控制, 2010, 39(4): 455-461)

[10] Zhang Zhan, Liu Guang-Jie, Wang Jun-Wen et al. Steganalysis of spread spectrum image steganography based on high-order Markov chain mode. Acta Electronica Sinica, 2010, 38(11): 2578-2584(in Chinese)
(张湛, 刘光杰, 王俊文等. 基于图像高阶 Markov 链模型的扩频隐写分析. 电子学报, 2010, 38(11): 2578-2584)

[11] Cao Wen-Ming, Wang Rui. The Study of Sensor Networks Coverage by Fuzzy Information Processing. Beijing: Electronic Industry Press, 2010(in Chinese)
(曹文明, 王瑞. 传感器网络覆盖定位模糊信息处理方法. 北京: 电子工业出版社, 2010)

[12] Gau W L, Buehrer D J. Vague sets. IEEE Transactions on Systems, Man and Cybernetics Society, 1993, 23(2): 610-614

[13] Zhang Cheng-Yi, Dang Ping-An, Li Dong-Ya. Note on "fuzzy entropy of Vague sets and its construction method". Computer Applications and Software, 2004, 21(5): 27-28

[14] Li Fan, Xu Zhang-Yan. Measures of similarity between Vague sets. Journal of Software, 2001, 12(6): 922-927(in Chinese)
(李凡, 徐章艳. Vague 集之间的相似度量. 软件学报, 2001, 12(6): 922-927)

[15] Quan Shuang-Yan. Similarity measures between Vague sets based on information. Computer Engineering and Applications, 2007, 43(25): 87-89(in Chinese)
(权双燕. 信息意义下的 Vague 集的相似度量. 计算机工程与应用, 2007, 43(25): 87-89)

[16] Li Yan-Hong, Olson David L, Qin Zheng. Similarity measures between intuitionistic fuzzy (Vague) sets: A comparative analysis. Pattern Recognition Letters, 2007, 28(2): 278-285

[17] Petitcolas F A P, Anderson R J, Kuhn M G et al. Information hiding — A survey. Proceedings of the IEEE, 1999, 87(7): 1062-1078

[18] Sharp T. An implementation of key-based digital signal steganography//Proceedings of the 4th Information Hiding Workshop. Pittsburgh, PA, USA, 2001, 2137: 13-26

[19] Mielikainen J. LSB matching revisited. Signal Processing Letters, 2006, 13(5): 285-287

[20] Westfeld A. F5—A steganographic algorithm: High capacity despite better steganalysis//Proceedings of the 4th Information Hiding Workshop. Pittsburgh, PA, USA, 2001, 2137: 289-302

[21] Shi Y, Eberhart R. Empirical study of particle swarm optimization//Proceedings of the International Conference on Evolutionary Computation. Washington, USA, 1999: 1945-1950

[22] Ker A D. Steganalysis of LSB matching in grayscale images. IEEE Signal Processing Letters, 2005, 12(6): 441-444

OUYANG Chun-Juan, born in 1974, Ph. D. candidate, associate professor. Her research interests focus on image steganography and steganalysis.

LI Bin, born in 1982, Ph. D., lecturer. His research interests focus on image steganography and pattern recognition.

LI Xia, born in 1968, Ph. D., professor, Ph. D. supervisor. Her research interests include intelligence optimization and intelligence computing and application.

WANG Na, born in 1977, Ph. D., professor. Her research interests include intelligence optimization and pattern recognition.

Background

Steganography is the art and science of secret communication in such a way that the presence of a message cannot be detected, while steganalysis tries to reveal the presence of the hidden messages. The security is a basic issue for the steganographic system. Therefore the study of the security measure becomes one of the hotspots in the research field of information security. So far, most of existing evaluation systems for steganographic security are founded upon the assumption that image statistical distribution is deterministic. However, it is impractical to obtain infinite data samples, which means an exact probability distribution of the cover and stego images cannot be derived. As a result, we propose to consider the steganography as a fuzzy process. The vague set is more effective in revealing the indeterminacy when compared to the

fuzzy set. Thus an evaluation system for steganographic security is introduced in this paper to measure the similarity between cover images and stego images. Firstly, the first-order and the second-order security measure were defined in terms of the vague set similarity measure between the distributions of the cover images and the stego images. Secondly, the newly defined security measures are proved to have the properties of boundedness, symmetry and unity. Particularly, the property of boundedness limits the value of the security measure in the range of [0,1], which is an absent characteristic in other evaluation systems with a deterministic image data statistical distribution model. Besides, when a steganographic system satisfies the property of unity, it is perfectly secure. Lastly, under the assumption that the pixels statistical distribution is independent identical distributed, the first-order and the second-order security measure are proved to be equivalent. Simulation results show the effectiveness of the proposed security evaluation system for evaluating different steganographic algorithms. When measuring the security under the same embedding rate, the second-order security measure can always reflect more obvious statistical changes than the first-order security measure. The new security measure is more sensitive than other measures with the deterministic image statistical model when the embedding rate is low. Experimental results also indicate that the proposed security measure may shed a light on designing steganography with high security level.

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61171124, 61103174, 60902069, 61005049), Science Technology Planning Project of Guangdong (Grant No. 2011B010200045). Foundation for Distinguished Young Talents in Higher Education of Guangdong (Grant No. LYM10116), and the Upgrade Projects of Shenzhen Key Laboratory (Grant No. CXB201105060068A).

覆盖表生成的遗传算法配置参数优化

梁亚澜 聂长海
(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 覆盖表生成是组合测试的关键问题,很多数学方法、贪心算法以及演化搜索方法等被应用于生成各种覆盖表.针对演化搜索方法的性能受到方法本身配置参数影响很大这一实际问题,文中以二维覆盖表生成为实例,系统地对典型的演化搜索方法——遗传算法的种群规模、进化代数、交叉概率、变异概率以及遗传算法的变种算法等因素进行探索,设计了 pair-wise 法、Base choice 法和爬山法 3 条实验路线探索遗传算法的这些配置参数及其相互作用对算法生成二维覆盖表效果的影响,并回答两个问题:对于特定二维覆盖表生成问题,是否存在遗传算法的最优参数配置;对于一般的二维覆盖表生成问题,是否存在通用的遗传算法最优参数配置.

关键词 二维覆盖表;遗传算法;配置参数优化;组合测试;测试用例生成
中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2012.01522

The Optimization of Configurable Genetic Algorithm for Covering Arrays Generation

LIANG Ya-Lan NIE Chang-Hai
(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093)

Abstract Covering array generation is one of the key issues in combinatorial testing. Many mathematical methods, greedy algorithms and evolutionary search methods have been applied in this field. Since the performance of evolutionary search methods is significantly impacted by their configurable parameters, we take genetic algorithm, one of the typical evolutionary search methods, as an example to discuss the different influences of its five configurable parameters (population size, evolution generation, crossover probability, mutation probability, variants of the algorithm) on the performance of 2-way covering array generation. Meanwhile we design three classes of experiments to systemically analyze the influences of each of the configurable parameters and the interactions among them. Our contributions are to answer the following questions: whether there exists an optimal configuration of genetic algorithm for a particular 2-way covering array generation and whether there exists a common optimal configuration for all 2-way covering arrays generation.

Keywords 2-way covering array; genetic algorithm; optimal configuration; combinatorial testing; test case generation

1 引 言

软件作为一个复杂的逻辑体,其最终的运行效果受到软件自身及外部许多因素的影响.如何以尽可能小的代价检测出软件系统中各种因素及其相互作用所引发的各类故障是一个非常重要的问题.

组合测试方法通过生成少量测试用例[1],以较

收稿日期:2011-01-06;最终修改稿收到日期:2011-06-14.本课题得到国家自然科学基金(60773104,61021062)、国家"八六三"高技术研究发展计划项目基金(2008AA01Z143)、江苏省自然科学基金(BK2010372)资助.**梁亚澜**,男,1988 年生,硕士研究生,主要从事软件测试领域的科研工作.**聂长海**(通信作者),男,1971 年生,教授,博士生导师,主要从事软件测试领域的科研和教学工作. E-mail: changhainie@nju.edu.cn.

小代价最大限度地对软件系统各因素间的相互作用关系进行覆盖，可以有效检测系统中各种因素相互作用所触发的软件故障. 根据 Kuhn 等人[2]的研究发现，测试二维交互关系能够找出 70%的错误，因此全盘考虑测试的成本、覆盖表的生成难度以及各种组合覆盖程度对覆盖表规模的影响等因素. 两两组合测试是人们最常使用的方法[1].

目前国内外对于二维覆盖表生成方法的研究已有不少成果，人们提出了很多数学方法[3]、贪心算法[4-6]以及演化搜索算法[7-8]等. 遗传算法是一种典型的演化搜索方法，该方法被广泛地应用于解决复杂的优化问题. 人们也已将其应用于覆盖表生成[9-11]，但现有的文献未能考虑到演化搜索方法的性能受到方法本身配置参数影响很大这一实际问题. 本文针对该问题，以二维覆盖表生成为实例，系统地探索遗传算法各配置参数及其相互作用对算法性能的影响.

本文重点关注于遗传算法生成覆盖表的效果与遗传算法本身配置参数变化的关系，从而更好地发掘遗传算法生成覆盖表的潜力. 我们通过实验对遗传算法的种群规模、进化机制、交叉概率、变异概率及其变种算法这 5 个因素进行取值组合，设计了 pair-wise 法、Base choice 法[12-13]和爬山法 3 条不同的实验路线，系统地探索遗传算法的 5 个因素及其相互作用对遗传算法覆盖表生成效果的影响程度和影响性质，并以生成覆盖表规模和消耗时间为分析依据寻找出最佳配置，为今后进一步研究覆盖表生成的遗传算法打下理论和实践基础.

本文第 2 节介绍相关研究，并指出存在的问题；第 3 节给出覆盖表的定义及相关概念；第 4 节对遗传算法 4 个参数配置及其变种算法的概念进行简要介绍；第 5 节是具体的实验设计；第 6 节分析实验数据并得出结论；第 7 节是总结和展望.

2　相关工作及存在的问题

Ghazi 等人[9]在 2003 年最先将遗传算法应用于覆盖表生成，他们通过实验得出在参数个数为 4，每个参数取值个数都为 3 的待测系统中，利用遗传算法可以得到包含 11 条测试用例的二维覆盖表. 文章只表明使用遗传算法生成覆盖表的可行性，实验只用了一个例子，实验设计过于简单且没有进行深入的研究.

Shiba 等人[10]在 2005 年介绍覆盖表生成的人工智能技术的文章中，对遗传算法如何确定染色体结构、适应度函数、选择交叉变异策略等进行了讨论，并对其中一组配置进行了实验，文章中并没有对遗传算法生成覆盖表的性能进行分析.

McCaffrey[11]在 2010 年提出遗传算法生成覆盖表的性能会受到其自身配置参数的影响，并提取出种群规模、染色体结构、进化选择策略等因素进行了讨论. 但是 McCaffrey 只通过实验验证了在其中一种参数配置下，遗传算法生成覆盖表的效果比较理想，并没有系统地探索遗传算法自身的这些因素及其相互作用对算法性能的影响特点和影响程度.

针对已有工作未能系统探索遗传算法生成覆盖表的性能，本文对遗传算法的配置参数进行了更为深入的探索，重点关注种群规模、进化代数、交叉概率、变异概率及遗传算法的变种算法这 5 个重要配置参数及其相互关系对遗传算法生成覆盖表性能的影响程度和影响性质，通过科学的实验设计，对这 5 个配置参数的取值组合进行优化以找出覆盖表生成遗传算法的最优参数配置，从而进一步发掘遗传算法在覆盖表生成方面的潜力.

3　覆盖表

软件质量受到其自身及外部许多因素的影响，针对这一实际问题，要求一个完备的检测方法必须能够检测出这些因素及其相互作用关系所引发的各类故障. 覆盖表，特别是二维覆盖表可以用最少的测试代价，系统地覆盖待测试系统中的各种参数组合. 以下我们先给出测试用例的定义，再通过一个例子介绍什么是覆盖表，然后给出覆盖表的一般定义.

定义 1.　设待测系统 SUT 有 n 个参数，每个参数有 $v_i(1\leq i\leq n)$ 个取值，形成集合 V_i. n 元组 $\langle a_1,a_2,\cdots,a_n\rangle$ 称为 SUT 的一个测试用例($a_i\in V_i$).

如表 1 所示，一个待测系统 SUT 存在 4 个可能影响其性能的参数分别为 C_1、C_2、C_3、C_4. 每个参数各有 3 个不同的取值{0、1、2}. 如果以穷举法对 4 个参数的所有取值组合进行检测，我们需要 $3^4=81$ 个测试用例才可以检测出各因素相互之间的关系所引发的故障，测试代价相对较大. 这个缺点在系统因素及其相互作用更复杂的待测软件中更为明显. 而且测试数据表明，并不是所有参数取值之间都存在相互作用，因此穷举法会产生大量冗余的测试用例.

表 1 SUT

C_1	C_2	C_3	C_4
0	0	0	0
1	1	1	1
2	2	2	2

组合测试方法所生成的测试用例集——覆盖表[1]，通过对软件系统各因素间的相互作用关系进行有选择的针对性覆盖，大大减少了冗余的测试用例，同时在保证测试效果的前提下有效地缩减了测试用例集的规模. 在表 1 的例子中，如需要覆盖任意两个参数间的相互关系，则只需要表 2 中的 9 条测试用例，从该表中可以检查它覆盖了任意两个参数之间的 9 个组合，在实际检测中可以检测出 SUT 中所有因为两两参数之间的相互作用而引发的软件故障.

表 2 SUT 的覆盖表

C_1	C_2	C_3	C_4
0	0	0	0
1	0	2	1
2	1	2	0
2	0	1	2
1	1	0	2
0	1	1	1
2	2	0	1
1	2	1	0
0	2	2	2

覆盖表作为组合测试的测试用例集，其定义如下.

定义 2. 假设影响一个待测系统的参数共 n 个，每个参数有 v 种可能取值. 则覆盖表 $\mathrm{CA}(N;t,n,v)$ 是一个 $N\times n$ 的数组，每行对应着一条测试用例，每列对应一个参数的所有取值. 任意 t 个参数形成的子数组 $N\times t$ 覆盖了该 t 个参数所有取值的组合，即每个 t 元组至少出现 1 次. 其中 t 为组合覆盖测试的强度，本文中提到的两两组合测试指的是 $t=2$ 的情况，生成的即为二维覆盖表[1].

在真实的软件系统中，影响系统的参数取值数目可能不尽相同. 于是人们将覆盖表扩展，形成了混合覆盖表 $\mathrm{MCA}(N;t,n,(v_1,v_2,\cdots,v_n))$. 表中第 i 列对应第 i 个参数，该参数取值数目记为 v_i，任意 t 个参数形成的 $N\times t$ 的子数组中包含了该 t 个参数的所有 t 元组. 当 $v_1=v_2=\cdots=v_n=v$ 时，$\mathrm{MCA}(N;t,n,(v_1,v_2,\cdots,v_n))$ 即为 $\mathrm{CA}(N;t,n,v)$[1].

覆盖表的生成是组合测试的关键环节，目前国内外对组合测试的很多研究主要集中在该领域. 除了数学方法[3]、启发式贪心算法[4-6]等，演化搜索算法是其中一类重要的方法，这一类方法利用模拟退火[7]、遗传算法[9-11]、蚁群算法和粒子群算法等演化计算技术生成覆盖表，是已有贪心算法和数学方法的重要补充，具有很强的灵活性和有效性，但其性能受其自身配置参数影响很大，本文针对这一实际问题，以遗传算法这一典型的演化搜索算法为研究对象，系统地探索算法中各配置参数对算法生成覆盖表的最终效果的影响性质和影响程度.

4 覆盖表生成的遗传算法

遗传算法[9-11,14]是一种通过模拟自然世界生物进化过程探索问题最优解的演化搜索算法，已被广泛地用于复杂的优化问题. 在遗传算法中，问题的一组候选解以染色体的形式构成算法的初始种群，算法对每一代种群中的染色体进行交叉变异，然后通过适应度函数值这一评价标准淘汰适应值较差的染色体，保留适应值较好染色体进入下一代，直到某一代种群中出现适应值满足问题要求的染色体，该染色体即为问题的最优解.

在本文中，我们采用待测系统 SUT 中参数 C_1，C_2，…，C_n 的一个取值组合 $\langle a_1,a_2,\cdots,a_n\rangle$ 作为染色体，即候选测试用例，以逐条生成测试用例的方式构造二维覆盖表. 算法流程图如图 1 所示，其具体步骤如下：

1. 初始化测试用例集 TG（TG 可设空集，也可放入所需的初始用例）；初始化待测系统 SUT 中需要被覆盖的二元组集合 S；随机生成初始的种群，种群规模为 m；初始化进化代数 T；初始化种群个体的适应值数组 fitness.

2. 计算种群中各染色体的适应度函数值，适应值即为该染色体所包含的未被 TG 覆盖的二元组合对个数.

3. 对种群中的染色体进行选择、交叉（交叉概率为 Pc）、变异（变异概率为 Pm）以产生新的种群. 对新种群中的个体进行适应值评价，若新的种群中存在最优染色体 c，则将其作为覆盖表中的一条新的用例，跳至步 5；否则重复该步过程，直到迭代次数达到 T.

4. 在当前种群中选择适应值最高的染色体 c，作为覆盖表中的一条新的用例.

5. 将 c 加入 TG 中，并从 S 中删去 c 所包含的二元组合对，若 S 为空集，即当前 TG 中的用例包含了所有二元组合对，则算法结束，TG 即为所求覆盖表；否则，跳至步 3.

由于遗传算法的效果受其自身配置参数的影响，为了进一步提高算法生成覆盖表的性能，以下我们从中提取出种群规模 m、进化代数 T、交叉概率 Pc、变异概率 Pm 以及遗传算法变种算法这 5 个对其性能影响较大的重要因素进行进一步讨论.

```
/*第1步*/
Initialize (TG)                                    //初始化测试用例TG(TG可以设空集，也可放入一些指定的初始实例)
Initialize S;                                      //初始化未被TG覆盖的二元组集合S
Initialize (group[m][n]);                          //初始化种群，种群规模为m，每条染色体为一条测试用例，n为测试用例参数个数
Initialize (T)                                     //初始化最终进化次数T
Initialize (fitnesss [m])                          //初始化种群个体的适应值数组
/*第2步*/
Evaluate (group [m][n], fitness [m]);              //对种群适应值进行评价
while (There exist uncovered pairs in S)
    {
    /*第3步*/
    t=0;                                           //重置当前进化次数t
    while (t<T)                                    //当前进化代数t没有达到T
    {
        Select (temp [m][n], group [m][n]);        //利用轮盘赌算法从group中选取染色体进入temp
        Crossover (temp[m][n]);                    //对temp中的染色体进行交叉操作
        Mutate (temp [m][n]);                      //对temp中的染色体进行变异操作
        Transit (group [m][n], temp[m][n]);        //将temp中的染色体放回group中
        Evaluate (group[m][n], fitness[m]);        //对种群适应值进行评价
        if (There exists a best chromosome "c" in group [m][n])
        {
            break;
        }
        t++;
    }
    /*第4步*/
    if (There exists no best chromosome "c" in group [m][n])
    {
            Choose a chromosome "c" in group [m][n] possessing the highest fitness;
    }
    /*第5步*/
    TG=TGAc                                        //将最优解放入测试用例集TG中
    S=S-{c};                                       //从S中删除c所包含的二元组合对
}
```

图 1　覆盖表生成的遗传算法流程图

4.1　种群规模 m

遗传算法中种群是由一定数量的染色体组成，种群中染色体的数量称为种群规模 m. 通常种群规模太小则不能提供足够的采样点，造成算法过早收敛；种群太大虽然可以增加优化信息，阻止早熟收敛的发生，但无疑会增加计算量，造成收敛时间太长.

为了准确找到最合适的种群规模取值，在正式实验前，我们对不同种群规模下遗传算法生成覆盖表的性能进行了初步的测试工作，结果显示：种群规模在6000以上时，算法生成覆盖表的规模下降程度不明显，算法速度却大大降低；种群规模在100以下时，算法生成覆盖表的规模有增加趋势且性能很不稳定.

综合考虑实验的准确度和复杂度等因素，本文种群规模 m 的取值集合为{100，2100，4100，6100}，该取值集合既保证实验能够击中最合适的种群规模取值，又保证实验复杂度在可操作范围内.

4.2　进化代数 T

通过进化代数 T 限定算法每轮的迭代次数，迭代次数达到进化代数 T 则算法终止. 与种群规模取值相似，终止进化代数过小，则生成的解无法接近最优解，过大，则运行消耗代价过大. 因此，在实际问题解决时，需要权衡各方面因素，力图找到一个平衡点.

在正式实验前，根据对不同进化代数取值下遗传算法生成覆盖表性能的初步测试我们发现：进化代数在100以下时，算法生成覆盖表的规模有增大趋势；进化代数在1000以上时，算法生成覆盖表的规模没有明显减小趋势且算法的速度大大降低. 综合考虑实验的准确度和复杂度等因素，本文 T 的取值集合为{100，600，1100}.

4.3　交叉概率 Pc

对种群中的染色体进行交叉操作以产生新的染色体，实质上是对问题解空间的广度搜索. 交叉概率太大，则种群中个体更新很快，容易破坏已有的高适应值的个体；概率太小，则交叉操作很少进行，会使搜索停滞不前，造成算法的不收敛.

通过测试工作，我们发现交叉概率 Pc 对遗传算法性能的影响没有明显的趋向性，为了保证实验结果的准确，对其采取在合法取值空间中均匀取值的方法，Pc 的取值集合为{1，0.8，0.6，0.4，0.2}.

4.4　变异概率 Pm

变异操作是对种群模式的扰动，有利于增加种群的多样性，实质上是对问题解空间的深度搜索. 同样，变异概率太小则很难产生新模式；变异概率太大则会使遗传算法成为随机搜索算法，失去遗传算法的特性和优点.

与交叉概率 Pc 相同，变异概率 Pm 对算法性能的影响也没有明显的趋向性，为了保证能够击中最合适的 Pm 取值，本文 Pm 的取值集合为{1，0.8，0.6，0.4，0.2}.

4.5　遗传算法的变种算法

本文在对传统遗传算法4个配置参数进行探索的同时，对遗传算法进行一定程度的改动，以发掘遗

传算法生成覆盖表的潜力.本文所探索的算法分别为:

GA:即遗传算法.GA 采用逐条生成测试用例的方法;在种群进化过程中,父代的最优个体 c 直接进入子代,其余染色体的选择策略 select 为轮盘赌选择法;适应值 f 为该条染色体所包含的未被覆盖对个数;交叉策略为多点交叉;变异策略是单点变异.

GA-:在种群进化过程中,GA-将 GA 中适应值 f 改为该条染色体所包含的已被覆盖对的个数,使 GA 的优胜劣汰制变为优汰劣胜制(注:优汰劣胜仅为种群进化中 select 的标准,而个体进入覆盖表的标准仍是选取包含未被覆盖对个数最多的个体).

GAr:在种群进化过程中,GAr 将 GA 染色体选择策略 select 改为随机选择,即父代中的个体随机选择进入子代.

GA climb:在 GA 的基础上采用爬山法对每一代种群中最优个体进行处理:GA climb 算法中每一代的最优个体 c 只可能被当前适应值更高的个体替换,否则直接进入下一代而不参加其它操作.算法保证了最优个体的优良特性能够在不被交叉变异破坏的基础上得到不断优化.

GA- climb:在 GA-的基础上改变对每一代种群中最优个体的处理,处理过程同 GA climb.

GAr climb:在 GAr 的基础上改变对每一代种群中最优个体的处理,处理过程同 GA climb.

5 实验设计

实验表明,覆盖表生成的遗传算法的效果会受到其自身 5 个因素(配置参数)的影响.如何确定相应的配置参数取值以求算法达到最好效果,即如何对有序集〈算法,种群规模,进化次数,交叉概率,变异概率〉进行取值,这是本文需要解决的问题.本文通过对形如〈GA-,2100,600,0.21,0.41〉的不同参数配置下的遗传算法的性能进行测试和比较,主要回答以下问题:(1)遗传算法这 5 个因素对于算法的性能的影响程度和性质如何.(2)是否存在一组最优参数配置,使得二维覆盖表的遗传算法对于某个特定问题,总是能发挥较优的性能.(3)该组最优参数配置是否具有通用性,即对于其它覆盖表生成问题也能产生近优解.

本文对不同参数配置下遗传算法生成覆盖表性能的评判标准为:首先考虑算法生成覆盖表规模,生成覆盖表规模最小的为最优参数配置;若不同参数配置下生成覆盖表规模相同,则选取生成覆盖表过程消耗时间最少的为最优参数配置.

遗传算法 5 个配置参数在相应取值范围内产生的值组合(即参数配置)规模十分庞大,在表 3 中的参数取值情况下,共产生 $6\times4\times3\times5\times5=1800$ 种不同的参数配置,利用每一种配置对应的遗传算法进行覆盖表生成并选优是不可行的.

表 3 遗传算法 5 个配置参数的取值集合

算法	种群规模	进化代数	交叉概率	变异概率
GA	100	100	1	1
GA-	2100	600	0.8	0.8
GAr	4100	1100	0.6	0.6
GA climb	6100		0.4	0.4
GA- climb			0.2	0.2
GAr climb				

为了更高效地解决问题,本实验采用 3 条实验路线对最优参数配置进行搜索,利用 3 个不同的参数配置集,在确保击中最优参数配置的前提下大大减少了所需测试的配置个数(为了在配置集中方便记录各参数的取值,本文将其对应为有序的自然数,如表 3 中"种群规模"中有 4 个不同取值,我们将其与 0,1,2,3 这 4 个自然数相对应):

(1) pair-wise 参数配置集

假设表 3 中各参数两两之间存在相互作用,我们对其进行二维组合覆盖,生成的二维覆盖表作为 pair-wise 配置集.为了概率更高地包含较优配置,本文没有采用规模最小的覆盖表,而是加入了适量冗余.如表 4 所示,该覆盖表共有 34 组不同的参数配置,这 34 条参数配置覆盖了所有参数之间的二维组合.比较该组参数配置集下遗传算法的待测实例覆盖表生成效果,从中得到最优参数配置 P_x($1\leq x\leq34$).

表 4 pair-wise 配置集

测试号	a_1	a_2	a_3	a_4	a_5	测试号	a_1	a_2	a_3	a_4	a_5
P_1	5	3	1	0	1	P_{18}	3	1	0	0	4
P_2	4	1	2	0	3	P_{19}	2	3	1	2	2
P_3	2	2	0	4	4	P_{20}	5	2	2	4	3
P_4	3	0	1	1	0	P_{21}	2	2	1	0	0
P_5	0	0	0	2	1	P_{22}	3	3	0	3	1
P_6	5	1	0	3	0	P_{23}	1	0	0	3	4
P_7	3	2	2	3	2	P_{24}	4	2	1	4	1
P_8	1	3	0	1	3	P_{25}	0	1	1	2	3
P_9	0	3	2	4	0	P_{26}	0	1	1	1	2
P_{10}	1	1	1	4	2	P_{27}	3	0	2	4	3
P_{11}	4	3	1	2	4	P_{28}	4	3	1	3	0
P_{12}	2	0	1	3	3	P_{29}	0	2	0	0	4
P_{13}	**5**	**0**	**2**	**1**	**4**	P_{30}	5	3	0	2	2
P_{14}	4	0	0	0	2	P_{31}	1	0	2	0	1
P_{15}	1	2	2	2	0	P_{32}	0	2	1	3	1
P_{16}	0	2	1	1	1	P_{33}	4	3	2	1	1
P_{17}	2	1	2	1	1	P_{34}	3	3	1	2	0

(2) Base choice 参数配置集

Base choice 是一种组合设计方法[12-13]. 该方法先选取一个参数配置作为基准参数配置 B(在本实验中该基准配置 B 即为 pair-wise 法所得最优参数配置 P_x),在此基础上改变其中某一个因素的取值并保持其余因素取值不变,从而产生一组参数配置集. 采用上述方法对遗传算法所有因素取值进行覆盖,得到的参数配置集即为 Base choice 参数配置集. 例如表 5 是以表 4 中的 P_{13} 为基准参数配置所对应的 Base choice 配置集.

表 5　Base choice 配置集

测试号	a_1	a_2	a_3	a_4	a_5	测试号	a_1	a_2	a_3	a_4	a_5
B_1	**5**	**0**	**2**	**1**	**4**	B_{11}	5	0	**1**	1	4
B_2	**0**	0	2	1	4	B_{12}	5	0	2	**0**	4
B_3	**1**	0	2	1	4	B_{13}	5	0	2	**2**	4
B_4	**2**	0	2	1	4	B_{14}	5	0	2	**3**	4
B_5	**3**	0	2	1	4	B_{15}	5	0	2	**4**	4
B_6	**4**	0	2	1	4	B_{16}	5	0	2	1	**0**
B_7	5	**1**	2	1	4	B_{17}	5	0	2	1	**1**
B_8	5	**2**	2	1	4	B_{18}	5	0	2	1	**2**
B_9	5	**3**	2	1	4	B_{19}	5	0	2	1	**3**
B_{10}	5	0	**0**	1	4						

(3) 爬山法参数配置集

爬山法首先选取一个参数配置作为基准参数配置 C_0,例如假设 C_0 为〈5,0,2,1,4〉(在本实验中该基准配置 C_0 即为 pair-wise 法所得最优参数配置 P_x),从遗传算法第 1 个配置参数变种算法开始,依次改变其在 C_0 中的取值并保持其余参数取值不变,从而产生 6 个参数配置集. 比较该组参数配置下遗传算法的覆盖表生成效果,选取生成效果最好的参数配置 C_1 作为新的选定配置,例如变种算法取值为 2 时效果最好,则 C_1=〈2,0,2,1,4〉. 然后在 C_1 的基础上改变遗传算法第 2 个配置参数种群规模的取值,得到 4 个参数配置集并产生 C_2,例如种群规模取值为 1 时生成效果最好,则 C_2=〈2,1,2,1,4〉. 依此类推,直到改变第 5 个配置参数的取值得到 C_5,C_5 即为爬山法所得的最优参数配置. 上述过程说明爬山法参数配置集是在实验中一步步动态生成的.

为满足实验需求,我们设计了可配置的覆盖表生成的遗传算法工具 COGA(Configurable and Optimized Genetic Algorithm). COGA 可自动以上文所描述的 pair-wise 法、Base choice 法和爬山法这 3 条实验路线对遗传算法的 5 个因素进行取值集合空间的搜索和取值组合的优化,得到待测实例的覆盖表生成的最优参数配置. 其输入为:(1) 待测实例描述,如 3^{13}(13 个参数,每个参数都有 3 个取值);(2) 遗传算法 5 个配置参数各自的参数取值集合. 输出为:(1) 运行 3 条实验路线过程中覆盖表的生成规模和消耗时间等数据记录;(2) 能为待测实例生成最优覆盖表的遗传算法配置.

6　实验结果及分析

根据实验设计我们对不同规模的待测实例进行覆盖表生成,得到各待测实例在不同配置集下遗传算法生成的覆盖表规模和消耗时间. 考虑到每一个待测实例的参数配置优化过程都需要对 34+23+23=80 个不同参数配置下的待测实例进行覆盖表生成实验,且为了保证数据的准确性,本文对 Base choice 配置集和爬山配置集中的每一个配置都进行 5 次实验并取最优解为生成结果,消耗时间较多,所以只选取 15 个具有代表性的待测实例进行实验(在 24 GB 内存,4 核 CPU(Intel(R)Xeon(R) E7450,主频 2.40 GHz)的刀片机远程终端上运行,实验周期为 1 个月),其中待测实例的选取兼顾下列几点:(1) 选取 4^{10}、3^{13} 和 6^4 等较为常见的待测实例. (2) 有选择地加入覆盖表生成规模和消耗时间相差较大的待测实例,以增加实验用例选择空间的广度. (3) 包含参数取值不一致的待测实例,以保证实验结论更加准确可靠.

下面我们分别对 3 个配置集下的实验数据进行整理分析.

6.1　pair-wise 实验

实验所得覆盖表的生成规模和消耗时间分别如表 6、表 7 所示,表中第 1 列列出 pair-wise 配置集中 34 个不同配置{$P1,P2,\cdots,P34$};第 1 行是各个待测实例,例如 4^{10} 表示该待测实例有 10 个参数,每个参数取值个数均为 4;$7^6\,6^7\,5^6$ 表示该待测实例有 19 个参数,其中 6 个参数有 7 个取值,7 个参数有 6 个取值,还有 6 个参数有 5 个取值.

通过不同配置下的覆盖表生成结果对比可以发现,利用遗传算法生成覆盖表的最终效果受算法自身配置参数的影响很大,例如在实例 6^{20} 中,不同参数配置下的覆盖表生成规模最多时相差 42 个,差距十分明显. 综合覆盖表的生成规模和消耗时间这两个标准,我们得到 pair-wise 配置集中各待测实例的最优参数配置 P_x 并在表中用加粗标记.

表 6 pair-wise 配置集生成覆盖表的规模

	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
P_1	30	19	62	39	106	23	87	46	106	173	92	**71**	39	45	23
P_2	30	20	61	37	106	22	84	45	102	167	89	73	39	44	21
P_3	31	20	70	44	125	24	101	53	121	207	103	80	41	**42**	22
P_4	31	19	66	41	116	24	96	50	113	190	100	76	39	43	**20**
P_5	36	21	79	49	151	28	120	60	139	245	121	95	45	45	25
P_6	30	19	64	42	113	23	93	48	112	184	96	75	39	44	21
P_7	31	19	60	38	104	21	86	45	104	170	91	74	40	44	22
P_8	32	19	71	44	124	24	102	52	121	205	105	79	40	45	22
P_9	31	20	70	44	125	24	101	52	122	205	104	78	44	43	25
P_{10}	32	20	72	45	128	25	105	54	126	214	108	81	42	44	22
P_{11}	30	19	62	37	102	22	79	**41**	96	162	85	74	40	43	24
P_{12}	37	23	82	51	148	27	121	61	141	246	126	92	47	47	25
P_{13}	30	**18**	64	**35**	**100**	23	**77**	42	**90**	**157**	**82**	75	39	45	21
P_{14}	31	20	70	43	120	24	99	51	118	198	100	77	39	45	23
P_{15}	32	20	72	44	125	23	101	52	123	207	107	79	42	44	23
P_{16}	30	20	70	44	124	24	100	53	122	204	106	79	41	42	21
P_{17}	32	19	72	45	130	25	105	55	124	213	110	81	41	43	24
P_{18}	30	19	60	38	103	22	84	45	101	167	90	75	41	43	23
P_{19}	33	20	69	43	123	25	101	53	121	205	104	79	40	45	23
P_{20}	29	19	60	37	102	22	84	44	103	165	87	74	**37**	42	23
P_{21}	31	20	70	44	126	24	102	52	121	208	105	80	42	43	22
P_{22}	30	19	63	39	107	22	90	48	109	178	93	75	38	45	22
P_{23}	37	22	86	49	150	28	119	58	140	250	122	94	44	47	24
P_{24}	29	19	62	39	105	22	88	47	107	174	93	74	38	42	22
P_{25}	31	20	68	44	122	24	100	51	120	202	102	80	42	43	22
P_{26}	31	21	71	44	127	24	102	52	122	206	105	79	42	44	22
P_{27}	**28**	19	**59**	37	101	**21**	82	44	99	162	86	74	39	44	22
P_{28}	31	19	62	40	111	22	86	48	105	172	91	72	40	44	23
P_{29}	29	19	64	42	112	24	93	50	114	188	98	77	39	43	22
P_{30}	29	21	64	41	109	22	91	47	109	178	93	73	39	43	21
P_{31}	38	23	82	52	153	28	120	60	142	252	122	95	48	48	26
P_{32}	32	21	71	44	123	24	100	52	121	205	104	80	41	44	24
P_{33}	30	20	63	38	105	22	86	47	105	173	91	73	39	42	23
P_{34}	30	19	61	40	111	22	88	47	106	174	89	75	40	44	22

表 7 pair-wise 配置集生成覆盖表消耗时间

	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
P_1	124.7	126.9	255.11	576.346	395.384	343.17	1303.92	1836.19	4256.82	785.918	3756	**218.42**	1224.05	38.657	151.883
P_2	196.3	159.4	388.58	431.14	682.645	251.43	1014.19	965.365	2247.35	1147.47	2533	796.65	876.616	139.028	201.132
P_3	12.3	12.45	27.285	59.717	45.849	34.663	139.309	206.733	479.157	89.622	460.4	36.036	140.635	**2.917**	12.636
P_4	1.96	1.716	4.477	7.113	7.472	4.04	17.191	16.942	39.359	13.713	18.89	4.244	7.02	0.811	**1.825**
P_5	0.406	0.328	0.936	1.467	1.732	0.827	3.697	3.447	8.175	3.213	3.93	0.811	1.358	0.14	0.344
P_6	6.598	5.928	13.962	27.846	22.885	15.554	63.758	74.584	171.835	44.024	186.8	17.113	58.781	2.091	6.131
P_7	696.1	485.6	1327.9	1341.4	2192.5	716.63	3033.81	2817.99	6692.29	3760.31	6569	1884.06	1808.05	575.207	703.798
P_8	146.1	97.53	322.81	292.798	548.734	163.957	692.926	558.983	1311.05	949.984	1231	675.906	307.042	124.91	145.128
P_9	1489	1123	3422.26	3282.93	6018.41	1818.74	7675.65	6090.19	14289.4	10288.7	13278	7073.29	3733.32	1218.49	1799.47
P_{10}	121.4	88.69	266.72	305.04	465.10	170.07	713.938	657.934	1525.18	840.705	1699	539.062	350.222	119.232	118.296
P_{11}	801.1	563.4	1586.3	1462.6	2565.8	852.28	3087.67	**2516.31**	6030.95	4446.86	5770	3602.51	1762.69	734.421	927.551
P_{12}	2.137	1.856	4.852	8.268	8.3	4.259	20.047	19.547	46.27	17.332	21.92	6.864	7.098	0.687	2.199
P_{13}	2.855	**2.43**	6.208	**9.579**	**8.93**	5.99	**22.4**	23.26	**53.414**	**17.77**	**23.45**	8.253	10.046	1.154	3.261
P_{14}	0.343	0.312	0.827	1.295	1.342	0.718	3.042	2.995	6.927	2.714	3.34	1.466	1.077	0.234	0.608
P_{15}	695.6	511.7	1564.6	1604.0	2714.6	800.39	3662.59	3314.43	7847.8	9516.42	7005	3282.45	1834.4	652.615	740.755
P_{16}	370.7	281.5	819.29	881.27	1491.9	480.359	1993.18	1850.17	4284.3	5367.04	3526	1768.58	982.853	230.741	366.743
P_{17}	79.61	68.48	178.12	340.50	303.51	178.309	802.344	927.036	2177.24	1590.6	1714	285.169	369.083	21.996	88.094
P_{18}	18.17	13.38	35.522	40.763	59.202	22.183	89.763	87.719	198.823	223.908	180.9	77.673	57.346	14.602	19.687
P_{19}	140.7	133.1	278.42	671.83	465.35	384.95	1505.13	2115.12	4855.7	4060.78	3386	532.322	815.682	45.302	150.432
P_{20}	136.6	133.0	268.63	578.47	426.79	338.01	1307.88	1914.07	4636.33	2481.77	3510	519.515	**620.95**	37.846	156.952
P_{21}	84.396	80.48	182.52	396.96	308.69	220.68	904.743	1219.8	2986.44	2316.97	2333	336.588	501.044	21.201	86.394
P_{22}	136.4	93.88	273.09	266.43	463.05	147.88	607.062	515.365	1184.25	1263.51	951	578.545	293.594	134.925	147.421
P_{23}	0.375	0.328	0.92	1.389	1.513	0.78	3.339	3.089	7.707	3.089	3.59	0.749	1.124	0.094	0.483
P_{24}	338.9	257.0	721.11	774.65	1197.3	421.40	1712.39	1671.07	3744.48	3971.43	3159	939.906	906.459	284.624	417.49
P_{25}	116.5	85.50	246.84	292.05	437.94	149.59	665.375	587.781	1403.49	1406.24	1265	301.191	346.977	95.909	127.531
P_{26}	118.2	92.22	264.41	299.68	464.32	158.80	693.456	604.192	1503.99	1471.95	1303	305.512	343.109	125.409	133.911
P_{27}	**3.09**	2.995	**6.93**	11.23	11.31	**6.02**	26.146	26.723	61.963	22.776	28.1	6.287	11.388	1.872	4.82
P_{28}	849.1	557.5	1618.7	1628.9	2895.02	884.35	3437.12	3066.03	6772.35	7634.66	5555	2023.26	1846.54	838.849	966.489
P_{29}	57.77	42.63	124.94	134.85	213.54	77.064	292.767	281.145	664.408	620.151	651.9	165.283	153.255	53.492	66.347
P_{30}	20.12	22.68	42.557	107	68.313	58.329	223.83	309.459	732.799	357.086	648.4	48.298	133.319	6.584	24.305
P_{31}	4.79	3.963	10.609	17.176	18.86	9.079	40.201	38.158	91.058	38.158	44.1	9.017	14.742	1.794	5.476
P_{32}	382.9	291.2	850.99	882.16	1454.7	466.21	1954.01	1816.52	4260.86	4007.92	3378	1066.63	991.37	355.323	446.693
P_{33}	1496.	1094.	3023.1	2786.1	4958.5	1579.9	6372.31	5476.74	12436.2	12170	9921	3844.5	3310.33	1256.71	1701.78
P_{34}	816.2	562.9	1621.6	1632.3	2932.8	870.30	3540.49	2962.76	6801.13	6821.83	5463	2147.67	1859.83	834.793	866.039

表中数据显示，所有待测实例除了 6^4，其余最优参数配置均选择 climb 算法，由此得出初步结论：利用遗传算法生成覆盖表时，算法选择以 climb 算法为优.

Pair-wise 实验对有限的 34 个不同参数配置下的待测实例进行覆盖表生成，可以得出不同的参数配置对覆盖表生成效果影响很大这一结论. 而这些影响具体是由哪一组参数的两两组合所引起，需进行更有针对性的实验才能确定. 目前对于如何确定参数两两交互作用对算法性能的影响这一问题，仅有 Bryce 等[6]基于贪心算法 6 个决策点的两两交互作用的相关研究，其实验过程和文章篇幅都较长. 所以要验证本文中遗传算法的参数交互作用，需以 Bryce 的实验设计为参考，另文详细讨论.

Pair-wise 实验主要考虑了遗传算法各配置参数之间的二维组合关系对算法效果的影响，并从中得到各待测实例的覆盖表生成的最优参数配置 P_x，但实验没有对遗传算法的 5 个配置参数进行单独地深入探索，因此无法准确评价这 5 个配置参数各自对算法效果的影响性质和影响程度，这在很大程度上影响了实验结论的完备性. 为了解决这一问题，我们在 pair-wise 实验的基础上设计了 Base choice 实验. Base choice 实验以 pair-wise 实验所得的配置 P_x 为基准参数配置，通过改变 P_x 中的某一个参数取值得到参数配置集并进行实验，更注重于发掘各配置参数自身对算法效果的影响.

6.2　Base choice 实验

实验以 pair-wise 法所得的最优参数配置 P_x 作为基准参数配置. 由于在 pair-wise 实验中得到的不同待测实例的覆盖表生成的最优参数配置不一致（如表 6、表 7 所示），相应的 Base choice 实验配置集也不相同. 考虑到演化搜索算法的不确定性，本文对每一个配置都进行了 5 次测试并取最优解为生成结果. 下面我们分别探索各个集合中每个配置参数对遗传算法生成覆盖表效果的影响性质和影响程度，找出该配置参数的最优取值.

6.2.1　遗传算法的变种算法

本文中遗传算法共有 6 个不同的变种算法，即每个待测实例对应 6 个不同的参数配置. 这 6 个配置下各覆盖表生成规模和消耗时间如表 8、表 9 所示（5 次测试取最优结果），由于基准参数配置 P_x 不一致，本文以算法选择为标准对生成结果进行统一排序，其中 P_x 用加粗标记.

表 8　Base choice/爬山实验中选择不同算法时覆盖表的生成规模

算法	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
0	33#	19#	76#	44#	126#	26#	104#	48#	126#	212#	107#	78	38#	44	24
1	36	22	81	50	147	27	115	52	139	249	121	77	40	45	24
2	35	21	82	49	147	27	116	52	136	245	121	76#	41	**42**#	24#
3	**28**	19	**59***	36	99*	**21**	74*	41	89*	154*	82	74	38	42	**20**
4	28	19	60	35	100	21*	77	**41**	90	160	83	70*	37	43	20
5	28*	**17***	61	**35***	**100**	21	**77**	41*	**90**	**157**	**82***	**71**	**36***	42*	20*

表 9　Base choice/爬山实验中选择不同算法时覆盖表的消耗时间

算法	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
0	3.9#	2.9#	9.19#	13.2#	13.76#	7.94#	32.18#	2995#	76.3#	27#	32.7#	2225	1588.6#	56.1	2.93
1	4.24	3.50	9.68	15.09	16.19	8.22	35.21	3322	82.46	33.20	37.47	2204	1698.8	54.7	2.18
2	3.68	2.9	8.83	13.79	14.15	7.83	33.38	1965	79.2	28.74	34.88	225.3#	869.1	**2.9**#	1.69#
3	**3.09**	2.8	**6.93***	10.42	10.25*	**6.02**	22.93*	2547	53.3*	19.8*	25.02	2141	1619.8	44.9	**1.83**
4	3.14	2.7	7.01	10.16	10.39	5.67*	23.29	**2516**	54.87	19.97	25.11	2034*	1562.2	45.4	1.93
5	2.9*	**2.3***	6.48	**9.58***	**8.93**	5.79	**22.4**	1524*	**52.87**	**17.8**	**23.5***	**218.42**	**627.7***	2.7*	1.78*

为了准确分析表中的数据，本文用“*”标记出各待测实例的覆盖表生成的最优算法选择，并将算法进行如下划分：$A=\{$GA, GA-, GAr$\}$；$B=\{$GA climb, GA- climb, GAr climb$\}$. 通过比较 A，B 集合的数据可以发现表 8 和表 9 中“*”都集中于 B 集合，即无论是覆盖表的生成规模还是消耗时间，B 集合中算法的整体性能明显优于 A 集合. 以待测实例 10^{11} 为例，B 中覆盖表最大生成规模（90 个测试用例）相比 A 中覆盖表最小生成规模（126 个测试用例）仍少 36 个测试用例，且消耗时间也少于 A，这表明，对遗传算法的种群最优个体进行爬山优化能够在加快算法收敛速度的同时有效提高算法的性能. 通过进一步的分析，我们发现在 A 集合中 GA 的性能总体优于 GA-和 GAr，但在 B 中这种性能上的差距却基本消失，这表明选择机制的不同对算法性能也有影响（优胜劣汰的轮盘赌选择机制要优于优汰

劣胜的轮盘赌选择机制和随机选择机制),但这种影响会在种群最优个体的爬山优化过程中被削弱.

6.2.2 种群规模

本文中种群规模的参数取值集合为{100, 2100, 4100, 6100},对应4个不同的参数配置,这4个参数配置下各待测实例的覆盖表生成规模和消耗时间如表10、表11所示(5次测试取最优解).加粗部分是基准参数配置 P_x,"*"标记的是Base choice实验所得各待测实例覆盖表生成的遗传算法种群规模最优取值.

表10 Base choice实验中选择不同种群规模时覆盖表的生成规模

种群规模	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^{6}6^{7}5^{6}$	$8^{2}7^{2}6^{2}5^{2}$	$6^{1}5^{1}4^{6}3^{8}2^{3}$	6^{4}	$5^{1}3^{8}2^{2}$
100	**28***	**18***	**59***	**35***	**100**	**21***	**77***	41*	**90***	**157***	**82***	75	39	46	**20***
2100	29	19	59	36	100	22	77	42	91	159	85	72	39	42*	21
4100	28	18	59	35	99*	21	77	41	91	159	83	73	**37***	**42**	21
6100	29	18	59	35	100	22	77	**41**	91	157	83	**71***	38	43	21

表11 Base choice实验中选择不同种群规模时覆盖表的消耗时间

种群规模	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^{6}6^{7}5^{6}$	$8^{2}7^{2}6^{2}5^{2}$	$6^{1}5^{1}4^{6}3^{8}2^{3}$	6^{4}	$5^{1}3^{8}2^{2}$
100	**3.09***	**2.43***	**6.93***	**9.58***	**8.94**	**6.02***	**22.4***	13.0*	**53***	**17.8***	**23.5***	3.21	10.733	0.05	**1.83***
2100	195.1	58.52	381.1	240.8	197.4	249.2	544.1	476.1	1451	631	1146	93.5	307.2	1.1*	113.8
4100	622.5	118.3	1239	506.4	391*	696.3	1177.	1371	3928	1657	3133	231	**620.95***	**2.92**	361.5
6100	1363	191.7	2842	861.4	599.7	1503	2002	**2516**	6354	3408	4914	**218***	1079.3	5.16	799.6

表10、表11中数据显示"*"大多集中在第一行,即种群规模取值为100时算法性能最优.通过进一步比较我们发现,种群规模取值不同时同一待测实例的覆盖表生成规模相互之间差距并不大,但消耗时间会随着种群规模的增加而明显增加.如待测实例 6^{20} 中4个配置下的覆盖表生成规模均为77,消耗时间的差距最大时却达到100倍之多.这表明,在本实验中种群规模对算法的影响主要体现在时间上,即种群规模的增加会增加时间负担,但不会为算法性能的提升提供较大帮助.不过除了待测实例 6^{4},上述4个配置中的算法选择都是climb算法,而没有涉及非climb算法,为了保证结论的准确性,我们利用各待测实例另做了一组验证实验,以探索种群规模取值对GA,GA-,GAr这3个非climb算法的性能影响.

验证实验步骤如下:先从表8和表9中的非climb算法中选出对应各待测实例的覆盖表生成效果最优的算法,用"#"标记(观察可知,算法集中于GA),然后用其代替各待测实例原基准配置 P_x 中的算法.在此基础上重复6.2.2节实验.实验数据如表10′和表11′所示.

表10′ Base choice验证实验中选择不同种群规模时覆盖表的生成规模

种群规模	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^{6}6^{7}5^{6}$	$8^{2}7^{2}6^{2}5^{2}$	$6^{1}5^{1}4^{6}3^{8}2^{3}$	6^{4}	$5^{1}3^{8}2^{2}$
100	33	19	76	44	126	26	104	53	126	212	107	93	43	46	24
2100	30	19	67	41	114	23	94	49	115	189	97	79	39	42	22
4100	29	19	67	41	112	23	93	49	113	185	96	79	38	42	22
6100	30	18	65	41	111	24	93	48	112	183	95	76	38	43	22

表11′ Base choice验证实验中选择不同种群规模时覆盖表的消耗时间

种群规模	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^{6}6^{7}5^{6}$	$8^{2}7^{2}6^{2}5^{2}$	$6^{1}5^{1}4^{6}3^{8}2^{3}$	6^{4}	$5^{1}3^{8}2^{2}$
100	3.9	2.9	9.19	13.2	13.76	7.94	32.18	17.14	76.3	27	32.7	3.94	12.32	0.047	1.69
2100	192.7	133.8	430.6	466.1	707.7	261.7	1086	521.1	2339	1337	1162	69.59	442.2	1.12	30.56
4100	595.7	416.3	1410	1300	2242	706.3	3020	1570	6951	4314	3383	136.2	1588.6	2.9	62.33
6100	1342	872.9	3080	2653	5026	1499	6269	2995	13553	9421	7072	225.3	2680	5.164	102

数据显示,虽然非climb算法的整体性能不如climb算法,但在非climb算法中,一定范围内增加种群规模能够减小覆盖表的生成规模.

因此我们得出结论,种群规模对非climb遗传算法的性能具有一定影响,在一定范围内种群规模越大,算法性能越好,但这种影响会被climb遗传算法中种群最优个体的爬山优化过程削弱.

6.2.3 进化代数

本文中进化代数的取值集合为{100, 600, 1100},对应3个不同的配置,这3个参数配置下各待测实例的覆盖表生成规模和消耗时间如表12、表13所示(5次测试取最优解).

表 12　Base choice 实验中选择不同进化代数时覆盖表的生成规模

进化代数	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	28*	18*	62	39	108	22	87	45	103	178	90	73	37*	**42***	20*
600	29	19	60	36	100*	21*	79	**41***	94	160	85	**71***	38	42	**20**
1100	**28**	**18**	**59***	**35***	**100**	**21**	**77***	41	**90***	**157***	**82***	72	**37**	43	21

表 13　Base choice 实验中选择不同进化代数时覆盖表的消耗时间

进化代数	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	0.30*	0.23*	0.67	1.0	0.94	0.61	2.31	479.3	5.6	1.92	2.43	69.1	52.28*	**2.9***	0.44*
600	1.78	1.40	3.82	5.40	5.04*	3.43*	12.46	**2516***	30.1	9.77	13.4	**218.4***	330.07	17.3	**1.83**
1100	**3.09**	**2.43**	**6.93***	**9.58***	**8.93**	**6.02**	**22.4***	4463	**53***	**17.8***	**23.5***	602.9	**620.95**	32.1	3.78

根据以往的经验来看，进化代数的取值会对算法的性能产生较大影响，但本实验的数据显示，对于不同大小待测实例的覆盖表生成，进化代数的影响程度并不相同. 例如在较大的待测实例 10^{11} 和 6^{20} 中，随着进化代数的增加，覆盖表的规模明显减小；但在较小的待测实例 4^{10} 和 3^{13} 中，进化代数取值不同时覆盖表的规模变化不大. 上述结论表明，进化代数的取值改变不会对较小规模待测实例的覆盖表生成产生很大影响，但对于较大规模待测实例的覆盖表生成，进化代数在一定范围内取值越大，生成规模越小. 为了验证这种现象是否是由算法种群最优个体的爬山优化所引起，我们将 3 个配置中的 climb 算法全部替换为非 climb 算法并再次测试，方法同 6.2.2 节的验证实验，实验数据如表 12′和表 13′所示.

表 12′　Base choice 验证实验中选择不同进化代数时覆盖表的生成规模

进化代数	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	33	20	76	44	127	25	103	49	126	210	108	78	39	42	24
600	33	21	77	44	128	25	104	48	125	210	108	76	40	42	24
1100	33	19	76	44	126	26	104	48	126	212	107	76	38	43	24

表 13′　Base choice 验证实验中选择不同进化代数时覆盖表的消耗时间

进化代数	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	0.34	0.266	0.8	1.22	1.29	0.70	2.95	508.2	7.22	2.56	2.86	34.58	119.4	2.9	0.25
600	2.06	1.685	4.93	7.25	7.69	4.19	17.6	2995	42.9	14.95	17.1	225.3	731.97	17.3	1.69
1100	3.9	2.9	9.19	13.2	13.8	7.94	32.2	5512	76.3	27	32.7	352.2	1588.6	32.1	2.81

数据显示：在非 climb 算法中，随着进化代数的增加，算法性能未出现明显变化，即 climb 遗传算法中种群最优个体的爬山优化过程会削弱进化代数对算法性能的影响.

6.2.4　交叉概率

本文中交叉概率的取值集合为{1，0.8，0.6，0.4，0.2}，对应 5 个不同的配置，这 5 个配置下各待测实例的覆盖表生成规模和消耗时间如表 14、表 15 所示(5 次测试取最优解).

表 14　Base choice 实验中选择不同交叉概率时覆盖表的生成规模

交叉概率	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	29	19	59	36	100	21	77	42	91	158	82	**71***	36	42	20
0.8	28	**18**	59*	**35**	**100**	21	**77**	42	**90**	**157**	**82***	71	36*	43	**20***
0.6	28	18	59	35	100	22	76	**41**	90	157	84	73	36	41*	21
0.4	29	18*	60	36	99*	21	76	41*	89	158	84	72	37	42	21
0.2	**28***	19	**59**	35*	100	**21***	76*	42	88*	156*	84	72	**37**	**42**	20

表 15　Base choice 实验中选择不同交叉概率时覆盖表的消耗时间

交叉概率	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	3.42	2.56	7.254	9.89	9.13	6.08	22.29	2567	53.6	17.58	23.67	**218***	627.67	2.86	1.97
0.8	3.28	**2.43**	6.83*	**9.58**	**8.93**	6.07	**22.4**	2551	**53**	**17.8**	**23.5***	219.9	568.05*	2.87	**1.83***
0.6	3.26	2.45	6.99	9.58	9.03	6.35	21.97	**2516**	52.8	17.58	23.84	219.7	569.9	2.5*	1.86
0.4	3.26	2.42*	7.08	9.83	8.83*	6.13	21.86	2510*	51.6	17.63	23.70	225	620.79	2.82	1.95
0.2	**3.09***	2.54	**6.93**	9.55*	8.97	**6.02***	21.84*	2574	51.4*	17.3*	23.80	214.1	**620.95**	**2.92**	1.86

表 14、表 15 中“*”分布十分分散，且比较每一列数据我们发现，交叉概率取不同值时，同一待测实例的覆盖表生成规模最大差距仅为 2 个测试用例. 由于算法性能十分接近，我们将其看做是遗传算法的不确定性所引起的随机扰动，即交叉概率在一定范围内的改变不会对覆盖表生成的遗传算法性能造成太大影响. 为了验证这一现象是否是由 climb 算法中种群最优个体的爬山优化所引起，同 6.2.2 节，我们进行了验证实验，实验数据如表 14′和表 15′所示. 数据显示，交叉概率的改变对算法性能影响仍然很小，即交叉概率对算法性能的影响程度与算法是否是 climb 算法无关.

表 14′ Base choice 验证实验中选择不同交叉概率时覆盖表的生成规模

交叉概率	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	34	21	77	45	127	25	104	48	125	210	107	76	39	42	24
0.8	33	19	76	44	126	25	104	48	126	212	107	75	39	42	24
0.6	33	20	77	44	124	25	104	48	126	211	105	78	39	41	24
0.4	33	20	76	44	124	25	103	47	125	209	106	77	39	42	23
0.2	33	20	76	44	123	26	103	49	125	209	104	79	38	43	25

表 15′ Base choice 验证实验中选择不同交叉概率时覆盖表的消耗时间

交叉概率	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	3.93	3.09	9.08	13.59	13.93	7.74	32.12	3008	78.66	27.86	30.78	225.3	1287	2.86	1.53
0.8	3.79	2.9	8.92	13.2	13.76	7.72	32.18	2986	76.3	27	32.7	196.6	1288	2.9	1.69
0.6	3.65	2.93	9.0	13.39	13.57	7.74	32.11	2995	79.03	27.59	30.30	202.1	1284	2.53	1.53
0.4	3.78	2.93	9.03	13.23	13.57	7.75	31.78	2974	78.08	26.83	30.75	196.7	1344	2.82	1.47
0.2	3.9	2.92	9.19	13.23	13.37	7.94	31.68	3070	77.86	27.09	30.03	199.2	1588.6	2.78	1.59

6.2.5 变异概率

本文中变异概率的取值集合为{1，0.8，0.6，0.4，0.2}，对应 5 个不同的配置，这 5 个配置下各待测实例的覆盖表生成规模和消耗时间如表 16、表 17 所示(5 次测试取最优解).

表 16 Base choice 实验中选择不同变异概率时覆盖表的生成规模

变异概率	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	29	18	63	41	113	22	92	47	112	184	97	73	40	42	**20***
0.8	29	19	62	40	112	22	92	46	112	185	95	**71**	40	42	21
0.6	28	18	61	39	107	21	91	46	110	178	93	71*	39	42	20
0.4	**28**	18	**59**	37	103	**21**	83	45	102	164	87	72	**37***	42	20
0.2	28*	**18***	59*	**35***	**100***	21*	**77***	**41***	**90***	**157***	**82***	73	38	**42***	21

表 17 Base choice 实验中选择不同变异概率时覆盖表的消耗时间

变异概率	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	3.39	2.70	7.61	12.45	11.78	6.8	29.23	2973	69.7	23.68	30.3	219.9	689.38	3.24	**1.83***
0.8	3.53	2.9	7.41	12.4	12.06	6.93	29.31	2880	70.7	24.28	30.3	**218.4**	695.59	3.51	2.01
0.6	3.23	2.75	7.44	12.15	11.69	6.62	29.03	2883	69.1	23.56	30.2	206*	671.1	3.49	1.90
0.4	**3.09**	2.58	**6.93**	11.03	10.23	**6.02**	25.49	2809	62.7	20.47	27.5	214.8	**620.9***	3.43	2.11
0.2	2.9*	**2.43***	6.44*	**9.58***	**8.93***	5.77*	**22.4***	**2516***	**53***	**17.8***	**23.5***	197.2	588.11	**2.9***	2.06

表 16、表 17 中“*”集中在最后一行，即本实验中绝大多数待测实例的变异概率最优取值都为 0.2. 通过对每一列数据的进一步比较可以发现，在多数覆盖表中，其生成规模和消耗时间都随着变异概率的减小总体呈递减趋势. 如在待测实例 8^{10} 中，随着变异概率的逐渐减小，生成的覆盖表规模减小了 13 个测试用例，同时算法的消耗时间从 11.78 s 逐步减到了 8.93 s. 上述结果表明，变异概率的取值对覆盖表生成的遗传算法性能具有较大影响，多数情况下，变异概率越小，算法性能越好.

同样，为了验证这一现象是否是由 climb 算法中种群最优个体的爬山优化所引起，我们进行了验证实验，实验步骤同 6.2.2 节，即以表 8 和表 9 中的覆盖表生成效果为标准，将各配置中的 climb 算法改为非 climb 算法. 实验数据如表 16′和表 17′所示. 数据显示，随着变异概率的减小，多数覆盖表的规模和消耗时间仍总体呈减少趋势，即变异概率对算法性能的影响程度与算法是否是 climb 算法无关.

表 16′　Base choice 验证实验中选择不同变异概率时覆盖表的生成规模

变异概率	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	36	22	84	51	151	27	119	51	140	249	123	78	42	42	24
0.8	35	22	81	49	147	27	116	51	139	245	121	76	41	42	24
0.6	35	21	79	48	143	26	113	51	136	239	119	77	40	41	24
0.4	33	21	76	47	139	26	111	50	132	230	114	78	38	41	24
0.2	31	19	70	44	126	24	104	48	126	212	107	77	40	41	25

表 17′　Base choice 验证实验中选择不同变异概率时覆盖表的消耗时间

变异概率	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	4.17	3.53	9.83	16.29	17.93	8.82	38.92	3319	91.35	35.48	38.03	198.0	1485.2	3.25	1.69
0.8	4.18	3.62	9.95	16.22	18.32	8.61	38.95	3362	92.59	35.13	38.31	225.3	1462.5	2.9	1.58
0.6	4.01	3.43	9.52	15.65	17.71	8.35	37.81	3262	90.75	34.71	37.22	196.1	1446.6	2.49	1.56
0.4	3.9	3.29	9.19	14.87	16.37	7.94	35.88	3191	86.21	31.92	34.71	190.9	1588.6	3.32	1.48
0.2	3.18	2.9	7.58	13.2	13.76	6.89	32.18	2995	76.3	27	32.7	173.0	1382.2	2.49	1.43

综合上述实验数据可以得到遗传算法 5 个配置参数在不同待测实例中的覆盖表生成的最优参数取值.我们将这些最优参数取值进行组合,即得到不同待测实例的遗传算法覆盖表生成的最优参数配置.如表 18 所示,待测实例不同,其最优参数配置也不同.下面我们利用这些最优参数配置生成 15 个待测实例的覆盖表,为了保证结果的准确性,我们从 10 次生成结果中取最优解.

表 18　base choice 实验所得各待测实例的最优参数配置及覆盖表生成结果

实例	算法	m	T	Pc	Pm	$Size$	$Time$/s	实例	算法	m	T	Pc	Pm	$Size$	$Time$/s
4^{10}	GAr climb	100	100	0.2	0.2	**28**	**0.234**	6^{30}	GA climb	100	1100	0.2	0.2	**87**	**71.99**
3^{13}	GAr climb	100	100	0.4	0.2	**18**	**0.187**	10^{11}	GA climb	100	1100	0.2	0.2	**154**	**22.09**
6^{10}	GA climb	100	1100	0.8	0.2	**59**	**6.724**	$7^6 6^7 5^6$	GAr climb	100	1100	0.8	0.2	**82**	**29.39**
4^{20}	GAr climb	100	1100	0.2	0.2	**35**	**11.75**	$8^2 7^2 6^2 5^2$	GA- climb	6100	600	1.0	0.6	**70**	**2413**
8^{10}	GAr climb	4100	600	0.4	0.2	**101**	**252.6**	$6^1 5^1 4^6 3^8 2^3$	GAr climb	4100	100	0.8	0.4	**39**	**69.51**
3^{20}	GA- climb	100	600	0.2	0.2	**21**	**3.57**	6^4	GAr climb	2100	100	0.6	0.2	**42**	**1.123**
6^{20}	GA climb	100	1100	0.2	0.2	**75**	**30.9**	$5^1 3^8 2^2$	GAr climb	100	100	0.8	1.0	**21**	**0.234**
4^{30}	GAr climb	100	600	0.4	0.2	**41**	**16.97**								

Base choice 实验能够准确评价遗传算法 5 个配置参数各自对算法性能的影响性质和影响程度,并找出每一个配置参数在不同待测实例的覆盖表生成问题中的最优取值,但实验只注重算法配置参数的局部优化,没有考虑算法整体性能是否得到提升.实验数据显示,在待测实例 8^{10} 中,覆盖表最小生成规模为 99(见实验过程),而 Base choice 实验得出最优参数配置的覆盖表生成规模为 101,因此算法各配置参数达到局部最优时,算法整体性能不一定最优.为了更好地探索覆盖表生成的遗传算法最优配置,我们设计了另一条实验路线——爬山实验,爬山实验能够较好地处理算法局部配置参数优化和算法整体性能提升之间的平衡关系,实验对参数配置的每一轮优化都是在上一轮优化的基础上进行,算法的整体性能会随着配置参数的逐步优化而一步步提升,从而得到能使算法整体性能达到最优的参数配置.

6.3　爬山实验

实验以 pair-wise 实验所得最优参数配置 P_x 作为基准参数配置.由于爬山实验是在实验过程中一步步对遗传算法的配置参数进行优化,所以配置集是动态生成的,下面我们将按照算法配置参数的优化顺序对实验进行逐步分析并找出该配置参数的最优取值.考虑到演化搜索算法结果的不确定性和运行时间较长这一实际问题,实验从 5 次运行结果中选取最优解作为测试结果.

6.3.1　遗传算法的变种算法

首先改变遗传算法的变种算法这一参数取值,由于基准参数配置 P_x 不一致,每个待测实例各自产生 5 个参数配置,对应算法的不同参数取值.实验结果见表 8、表 9.综合覆盖表的生成规模和消耗时间这两个标准,我们得到算法选择的最优取值,其所对应的参数配置 C_1 如表 19 所示.

表 19　爬山实验各待测实例中算法优化所对应的参数配置C_1

实例	a_1	a_2	a_3	a_4	a_5	实例	a_1	a_2	a_3	a_4	a_5
4^{10}	**5**	0	2	4	3	6^{30}	**3**	0	2	1	4
3^{13}	**5**	0	2	1	4	10^{11}	**3**	0	2	1	4
6^{10}	**3**	0	2	4	3	$7^6 6^7 5^6$	**5**	0	2	1	4
4^{20}	**5**	0	2	1	4	$8^2 7^2 6^2 5^2$	**4**	3	1	0	1
8^{10}	**3**	0	2	1	4	$6^1 5^1 4^6 3^8 2^3$	**5**	2	2	4	3
3^{20}	**4**	0	2	4	3	6^4	**5**	2	0	4	4
6^{20}	**3**	0	2	1	4	$5^1 3^8 2^2$	**5**	0	1	1	0
4^{30}	**5**	3	1	2	4						

6.3.2 种群规模

在 C_1 的基础上我们对种群规模的取值进行改变，每个待测实例各自对应 4 个不同的参数配置，测试结果如表 20、表 21，表中“*”标记种群规模的最优参数取值. 由此我们得到各待测实例的参数配置 C_2，如表 22 所示. 通过观察表 20、表 21 中数据可知，种群规模的改变对算法性能的影响主要体现在消耗时间上，对覆盖表的生成规模没有产生较大影响，这进一步验证了 Base choice 实验中的结论.

表 20 爬山实验中选择不同种群规模时覆盖表的生成规模

种群规模	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	29	17*	59	35*	99	21*	74*	40*	89*	154*	82*	75	39	42*	20*
2100	28*	18	60	35	98*	22	77	41	89	155	85	70*	39	42	21
4100	29	18	60	36	99	21	77	41	90	156	83	72	36*	42	20
6100	28	18	58*	36	98	21	76	41	91	155	83	70	38	43	21

表 21 爬山实验中选择不同种群规模时覆盖表的消耗时间

种群规模	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	2.99	2.28*	6.16	10.1*	10.25	5.67*	22.9*	12.7*	53.3*	19.8*	23.5*	3.68	10.73	0.13*	1.78*
2100	64.4*	56.1	135	394.5	614*	246	897	459	1898	1529	1146	283.2*	37.21	1.31	38.75
4100	131	118.9	269	1222	1972	679	2612	1354	5591	5150	3133	899.7	627.7*	2.92	77.84
6100	202	190.3	413*	2431	4379	1465	5272	2474	10500	9267	4914	2034	1079.3	4.56	132.4

表 22 爬山实验各待测实例中种群规模优化所对应的参数配置 C_2

实例	a_1	a_2	a_3	a_4	a_5	实例	a_1	a_2	a_3	a_4	a_5
4^{10}	**5**	**1**	2	4	3	6^{30}	**3**	**0**	2	1	4
3^{13}	**5**	**0**	2	1	4	10^{11}	**3**	**0**	2	1	4
6^{10}	**3**	**3**	2	4	3	$7^6 6^7 5^6$	**5**	**0**	2	1	4
4^{20}	**5**	**0**	2	1	4	$8^2 7^2 6^2 5^2$	**4**	**1**	1	0	1
8^{10}	**3**	**1**	2	1	4	$6^1 5^1 4^6 3^8 2^3$	**5**	**2**	2	4	3
3^{20}	**4**	**0**	2	4	3	6^4	**5**	**0**	0	4	4
6^{20}	**3**	**0**	2	1	4	$5^1 3^8 2^2$	**5**	**0**	1	1	0
4^{30}	**5**	**0**	1	2	4						

6.3.3 进化代数

在 C_2 的基础上改变进化代数的参数取值，对应各待测实例的 3 个不同参数配置，测试结果如表 23、表 24 所示，其中“*”标记进化代数的最优取值. 比较表中的数据，我们发现在 $7^6 6^7 5^6$、10^{11} 和 8^{10} 这些规模较大的待测实例中，进化代数取值越大时覆盖表生成规模越小；而在 3^{20}、4^{10} 和 $5^1 3^8 2^2$ 这些规模较小的待测实例中覆盖表生成规模变化却没有明显规律，这验证了 Base choice 实验中的结论. 实验最终得到各待测实例的参数配置 C_3，见表 25.

表 23 爬山实验中选择不同进化代数时覆盖表的生成规模

进化代数	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	29	19	61	37	103	22	83	43	101	166	90	72	37	42*	20*
600	28*	19	60	36	100	21*	77	40*	91	158	85	70*	38	43	20
1100	28	17*	58*	35*	98*	21	74*	40	89*	154*	82*	71	36*	43	21

表 24 爬山实验中选择不同进化代数时覆盖表的消耗时间

进化代数	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
100	6.11	0.172	39.1	1.03	58.9	0.61	2.34	2.36	5.59	2.18	2.43	45.24	52.28	0.13*	0.44*
600	34.8*	1.7	224	5.71	339	3.34*	12.83	12.7*	30.0	11.6	13.4	283*	330.07	0.33	1.78
1100	64.4	2.28*	413*	10.1*	614*	5.67	22.9*	22.2	53.3*	19.8*	23.5*	489.1	627.7*	0.70	3.26

表 25 爬山实验各待测实例中进化代数优化所对应的参数配置 C_3

实例	a_1	a_2	a_3	a_4	a_5	实例	a_1	a_2	a_3	a_4	a_5
4^{10}	**5**	**1**	**1**	4	3	6^{30}	**3**	**0**	**2**	1	4
3^{13}	**5**	**0**	**2**	1	4	10^{11}	**3**	**0**	**2**	1	4
6^{10}	**3**	**3**	**2**	4	3	$7^6 6^7 5^6$	**5**	**0**	**2**	1	4
4^{20}	**5**	**0**	**2**	1	4	$8^2 7^2 6^2 5^2$	**4**	**1**	**1**	0	1
8^{10}	**3**	**1**	**2**	1	4	$6^1 5^1 4^6 3^8 2^3$	**5**	**2**	**2**	4	3
3^{20}	**4**	**0**	**1**	4	3	6^4	**5**	**0**	**0**	4	4
6^{20}	**3**	**0**	**2**	1	4	$5^1 3^8 2^2$	**5**	**0**	**0**	1	0
4^{30}	**5**	**0**	**1**	2	4						

6.3.4 交叉概率

在 C_3 的基础上改变交叉概率的取值，对应各待测实例的 5 个不同参数配置，测试结果如表 26、表 27 所示，其中“*”标记交叉概率的最优取值. 通过对表中数据进行观察可知，交叉概率的改变对覆盖表的生成规模及消耗时间没有太大影响，不同交叉概率取值下覆盖表的生成规模之间差距很小，且没有明显的规律可循. 结论与 Base choice 实验一致. 在对交叉概率进行优化后我们得到各待测实例的参数配置 C_4，见表 28.

表 26　爬山实验中选择不同交叉概率时覆盖表的生成规模

交叉概率	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	29	19	61	37	99	21	76	41	89	155	82	70	36	42	21
0.8	28	17*	60	35*	98	21	74*	41	89	154*	82*	70*	36*	42	20*
0.6	29	18	61	36	98*	22	75	40	88	155	84	71	36	41*	21
0.4	28*	19	59	36	99	21	75	41	88	156	84	72	37	43	22
0.2	28	19	58*	36	98	21*	76	40*	87*	155	84	71	36	42	21

表 27　爬山实验中选择不同交叉概率时覆盖表的消耗时间

交叉概率	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	37.44	2.76	416.4	11.2	607.7	3.37	23.1	12.41	54.2	20.8	23.67	283	626.2	0.047	0.45
0.8	35.68	2.28*	407.2	10.1*	614	3.4	22.9*	12.24	53.3	19.8*	23.5*	281*	568.1*	0.19	0.44*
0.6	35.37	3.06	418.1	11.3	604*	3.45	22.95	12.71	53.3	20.3	23.84	282	569.9	0.03*	0.47
0.4	34.3*	2.67	411.2	10.6	611	3.45	22.74	12.25	53.0	20.3	23.70	291	620.8	0.047	0.47
0.2	34.8	2.78	413*	10.3	614.	3.34*	22.93	12.4*	52.6*	20.15	23.80	263	627.7	0.13	0.45

表 28　爬山实验各待测实例中交叉概率优化所对应的参数配置 C_4

实例	a_1	a_2	a_3	a_4	a_5	实例	a_1	a_2	a_3	a_4	a_5
4^{10}	**5**	**1**	**1**	**3**	3	6^{30}	**3**	**0**	**2**	**4**	4
3^{13}	**5**	**0**	**2**	**1**	4	10^{11}	**3**	**0**	**2**	**1**	4
6^{10}	**3**	**3**	**2**	**4**	3	$7^6 6^7 5^6$	**5**	**0**	**2**	**1**	4
4^{20}	**5**	**0**	**2**	**1**	4	$8^2 7^2 6^2 5^2$	**4**	**1**	**1**	**1**	1
8^{10}	**3**	**1**	**2**	**2**	4	$6^1 5^1 4^6 3^8 2^3$	**5**	**2**	**2**	**1**	3
3^{20}	**4**	**0**	**1**	**4**	3	6^4	**5**	**0**	**0**	**2**	4
6^{20}	**3**	**0**	**2**	**1**	4	$5^1 3^8 2^2$	**5**	**0**	**0**	**1**	0
4^{30}	**5**	**0**	**1**	**4**	4						

6.3.5　变异概率

在 C_4 的基础上对变异概率的取值进行改变，对应各待测实例的 5 个不同参数配置，测试结果见表 29、表 30，其中“*”标记交叉概率的最优取值. 通过观察我们发现，随着变异概率的减小，覆盖表的生成规模和消耗时间总体呈减小趋势. 这表明在一定范围内变异概率的取值越小，覆盖表生成的遗传算法性能越好，与 Base choice 实验结论一致. 实验最终得到各待测实例的参数配置 C_5，见表 31.

表 29　爬山实验中选择不同变异概率时覆盖表的生成规模

变异概率	覆盖表规模														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	29	19	61	41	109	23	93	49	112	185	97	71	40	43	20
0.8	29	18	61	39	104	22	91	48	110	183	95	70	40	43	20
0.6	28	19	59	39	102	22	88	47	107	172	93	70*	39	42	20
0.4	28	19	58	36	99	21	81	44	98	161	87	71	36*	42	21
0.2	28*	17*	58*	35*	98*	21*	74*	40*	87*	154*	82*	70	38	41*	20*

表 30　爬山实验中选择不同变异概率时覆盖表的消耗时间

变异概率	消耗时间/s														
	4^{10}	3^{13}	6^{10}	4^{20}	8^{10}	3^{20}	6^{20}	4^{30}	6^{30}	10^{11}	$7^6 6^7 5^6$	$8^2 7^2 6^2 5^2$	$6^1 5^1 4^6 3^8 2^3$	6^4	$5^1 3^8 2^2$
1.0	37.71	2.87	431.2	12.1	716.9	3.84	30.22	16.47	71.1	27.5	30.3	290.2	699.38	0.047	0.44
0.8	38.36	3.70	425.4	12.2	680.3	3.74	30.14	16.54	71.5	27.6	30.3	281	695.59	0.047	0.45
0.6	37.04	3.15	422.2	11.5	666.5	3.74	29.16	16.21	69.0	25.8	30.2	277*	671.1	0.047	0.45
0.4	34.3	3.01	413.7	10.8	635.4	3.34	26.04	14.52	61.7	22.7	27.5	273.4	568.1*	0.047	0.442
0.2	32.1*	2.3*	402*	10.1*	604*	3.31*	22.9*	12.4*	52.6*	19.8*	23.5*	279.5	588.1	0.03*	0.43*

表 31　爬山实验各待测实例中变异概率优化所对应的参数配置 C_5

实例	a_1	a_2	a_3	a_4	a_5	实例	a_1	a_2	a_3	a_4	a_5
4^{10}	**5**	**1**	**1**	**3**	**4**	6^{30}	**3**	**0**	**2**	**4**	**4**
3^{13}	**5**	**0**	**2**	**1**	**4**	10^{11}	**3**	**0**	**2**	**1**	**4**
6^{10}	**3**	**3**	**2**	**4**	**4**	$7^6 6^7 5^6$	**5**	**0**	**2**	**1**	**4**
4^{20}	**5**	**0**	**2**	**1**	**4**	$8^2 7^2 6^2 5^2$	**4**	**1**	**1**	**1**	**2**
8^{10}	**3**	**1**	**2**	**2**	**4**	$6^1 5^1 4^6 3^8 2^3$	**5**	**2**	**2**	**1**	**3**
3^{20}	**4**	**0**	**1**	**4**	**4**	6^4	**5**	**0**	**0**	**2**	**4**
6^{20}	**3**	**0**	**2**	**1**	**4**	$5^1 3^8 2^2$	**5**	**0**	**0**	**1**	**4**
4^{30}	**5**	**0**	**1**	**4**	**4**						

通过这 5 轮实验，我们得到各待测实例覆盖表生成的最优参数配置. 如表 32 所示，不同覆盖表的最优参数配置各不相同. 下面我们利用这些最优参数配置生成各待测实例的覆盖表，为了保证结果的准确性，我们从 10 次生成结果中取最优解.

表 32 爬山实验所得各待测实例的最优参数配置及覆盖表生成结果

实例	算法	m	T	Pc	Pm	$Size$	$Time$/s	实例	算法	m	T	Pc	Pm	$Size$	$Time$/s
4^{10}	GAr climb	2100	600	0.4	0.2	**28**	**32.1**	6^{30}	GA climb	100	1100	0.2	0.2	**87**	**52.6**
3^{13}	GAr climb	100	1100	0.8	0.2	**17**	**2.28**	10^{11}	GA climb	100	1100	0.8	0.2	**154**	**19.8**
6^{10}	GA climb	6100	1100	0.2	0.2	**58**	**402**	$7^6 6^7 5^6$	GAr climb	100	1100	0.8	0.2	**82**	**23.5**
4^{20}	GAr climb	100	1100	0.8	0.2	**35**	**10.1**	$8^2 7^2 6^2 5^2$	GA- climb	2100	600	0.8	0.6	**70**	**277**
8^{10}	GA climb	2100	600	0.6	0.2	**98**	**604**	$6^1 5^1 4^6 3^8 2^3$	GAr climb	4100	1100	0.8	0.4	**36**	**568.1**
3^{20}	GA- climb	100	600	0.2	0.2	**21**	**3.31**	6^4	GAr climb	100	100	0.6	0.2	**41**	**0.03**
6^{20}	GA climb	100	1100	0.8	0.2	**74**	**22.9**	$5^1 3^8 2^2$	GAr climb	100	100	0.8	0.2	**20**	**0.43**
4^{30}	GAr climb	100	600	0.2	0.2	**40**	**12.4**								

6.4 实验结论

本实验通过 pair-wise 实验，Base choice 实验以及爬山实验这 3 条实验路线系统地探索了覆盖表生成的遗传算法中各配置参数的取值及其相互作用对算法性能的影响. 3 条实验路线探索算法性能的角度和侧重点各不相同，pair-wise 实验主要探索遗传算法各配置参数之间的二维组合关系对算法性能的影响，实验的参数配置集是遗传算法 5 个配置参数的二元组合覆盖表. Base choice 实验更侧重于发掘各配置参数自身对算法效果的影响，实验对遗传算法的 5 个配置参数进行了单独的深入探索. 爬山实验则是对参数配置进行逐步优化，参数配置的整体性能会随着每一轮参数优化的完成而一步步提高.

通过实验我们得出结论，算法选择、进化代数和变异概率这 3 个配置参数对覆盖表生成的遗传算法整体性能影响较大，种群规模的影响只体现在算法消耗时间上，而交叉概率的影响相对较小. 其中算法选择以 GA climb 和 GAr climb 算法较优，且进化代数取值较大，交叉概率取值较小时算法性能会有明显提高. 结论表明，对于某个特定问题的二维覆盖表生成，遗传算法中存在一组最优参数配置，使其总是能发挥较优的性能. 表 33 中是比较三组实验后所得 15 个待测实例的最优参数配置和覆盖表生成结果. 观察可知，不同待测实例中，覆盖表生成的最优参数配置之间差异很大，即不存在一组适用于所有二维覆盖表生成的最优参数配置.

表 33 各待测实例的最终最优配置和覆盖表生成结果

实例	算法	m	T	Pc	Pm	$Size$	$Time$/s	实例	算法	m	T	Pc	Pm	$Size$	$Time$/s
4^{10}	GAr climb	100	100	0.2	0.2	**28**	**0.234**	6^{30}	GA climb	100	1100	0.2	0.2	**87**	**52.6**
3^{13}	GAr climb	100	1100	0.8	0.2	**17**	**2.28**	10^{11}	GA climb	100	1100	0.8	0.2	**154**	**19.8**
6^{10}	GA climb	6100	1100	0.2	0.2	**58**	**402**	$7^6 6^7 5^6$	GAr climb	100	1100	0.8	0.2	**82**	**23.5**
4^{20}	GAr climb	100	1100	0.8	0.2	**35**	**10.1**	$8^2 7^2 6^2 5^2$	GA- climb	2100	600	0.8	0.6	**70**	**277**
8^{10}	GA climb	2100	600	0.6	0.2	**98**	**604**	$6^1 5^1 4^6 3^8 2^3$	GAr climb	4100	1100	0.8	0.4	**36**	**568.1**
3^{20}	GA- climb	100	600	0.2	0.2	**21**	**3.31**	6^4	GAr climb	100	100	0.6	0.2	**41**	**0.03**
6^{20}	GA climb	100	1100	0.8	0.2	**74**	**22.9**	$5^1 3^8 2^2$	GAr climb	100	100	0.8	0.2	**20**	**0.43**
4^{30}	GAr climb	100	600	0.2	0.2	**40**	**12.4**								

7 总结和展望

本文系统地研究了遗传算法这一演化搜索算法在进行覆盖表生成时的性能优化问题，通过 3 条实验路线探索遗传算法的变种算法、种群规模、进化代数、交叉概率和变异概率这 5 个因素及其相互作用对算法性能的影响，得出以下结论：

(1) 覆盖表生成的遗传算法性能受算法的参数配置影响很大，不同参数配置下的覆盖表生成结果之间具有明显差异. 例如参数配置中算法选择为 GA climb 或 GAr climb 时算法生成覆盖表的规模普遍较小，而算法选择为 GA-时，则会生成较大规模的覆盖表.

(2) 遗传算法的各配置参数对算法生成覆盖表性能存在不同程度的影响，其中算法选择、进化代数和变异概率这 3 个配置参数对覆盖表生成的遗传算法整体性能影响较大，种群规模的影响只体现在算法消耗时间上，而交叉概率的影响相对较小.

(3) 对于特定的二维覆盖表生成问题，存在一组最优参数配置，使得遗传算法总能发挥较优的性能. 对于一般的二维覆盖表生成问题，则不存在这样一组通用的最优参数配置. 不同的覆盖表生成问题，其最优参数配置之间存在差异，但也具有一定的共性，实验结论显示，在本文涉及的 15 个待测实例的最优参数配置中，算法选择集中于 GA climb 和 GAr climb 算法，进化代数取值总体较大，变异概率取值总体较小.

在以后的工作中，我们将在已有的工作基础上对实验中所得到的遗传算法最优参数配置进行验证

和进一步优化，例如，以实验结论为基础扩大配置参数取值集合，进一步探索 climb 算法下更大进化代数，更小变异概率的遗传算法性能；对相同参数配置下的待测实例进行 10 次甚至 20 次的覆盖表生成实验以使实验结果更加精确等. 另一方面，由于实验规模和复杂度会进一步增加，我们计划将实验迁移到云计算环境中，利用云计算平台优越的计算能力更好更快地设计和实施实验.

针对覆盖表生成过程中，遗传算法的通用最优配置不存在的情况，我们将研究利用本文实验方法构建一种两阶段多配置并行遗传算法，该方法首先使用 pair-wise 参数覆盖表配置一组遗传算法先行进行计算，然后选择其中性能好的配置，以这几个配置为基础分别作爬山和 Base choice 配置试验，用它们中产生的最好的结果作为并行遗传算法的结果. 关于这个方法的效果验证我们将另文讨论.

针对特定的覆盖表生成实例，另外一个改进遗传算法性能的方向是我们拟采用配置演化的方法，演化过程中首先随机生成一组遗传算法的配置作为初始种群，运行这组配置对应的遗传算法，生成该特定实例的覆盖表，以每个算法生成的该特定实例的覆盖表规模作为适应值度量，覆盖表规模越小，适应值越高，从而对这组配置进行适应值排序，并对其进行选择、交叉和变异，形成第二代种群. 循环该过程，直到性能不再提高或达到某种收敛标准，系统研究这种方法的性能和成本.

参 考 文 献

[1] Nie Changhai, Leung Hareton. A survey of combinatorial testing. ACM Computing Survey, 2011, 43(2), Article 11: 1-29

[2] Kuhn D, Reilly M. An investigation of the applicability of design of experiments to software testing//Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop. Los Alamitos, CA, 2002: 91

[3] Williams Alan W, Prober Robert L. A practical strategy for testing pair-wise coverage of network interfaces//Proceedings of the 7th International Symposium on Software Reliability Engineering (ISSRE1996). White Plaints, NY, USA, 1997: 246-254

[4] Cohen D M, Dalal S R, Fredman M L, Patton G C. The AETG system: An approach to testing based on combinatorial design. IEEE Transactions on Software Engineering, 1997, 23(7): 437-444

[5] Colbourn C J, Cohen M B, Turban R C. A deterministic density algorithm for pairwise interaction coverage//Proceedings of the IASTED International Conference on Software Engineering. Innsbruck, Austria, 2004: 242-252

[6] Bryce Renée C, Colbourn Charles J, Cohen Myra B. A framework of greedy methods for constructing interaction test suites//Proceedings of the 27th International Conference on Software Engineering (ICSE2005). St. Louis, Missouri, USA, 2005: 146-155

[7] Cohen Myra B, Gibbons Peter B, Mugridge Warwick B, Colbourn Charles J. Constructing test suites for interaction testing//Proceedings of the 25th International Conference on Software Engineering (ICSE2003). Portland, Oregon, USA, 2003: 38-48

[8] Nurmela Kari J. Upper bounds for covering arrays by tabu search. Discrete Applied Mathematics, 2004, 138 (1-2): 143-152

[9] Ghazi S A, Ahmed M A. Pair-wise test coverage using genetic algorithms//Proceedings of the 2003 Congress on Evolutionary Computation. Canberra, Australia, 2003, 2: 1420-1424

[10] Shiba Toshiaki, Tsuchiya Tatsuhiro, Kikuno Tohru. Using artificial life techniques to generate test cases for combinatorial testing//Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC2004). Hong Kong, China, 2004: 72-78.

[11] McCaffrey James D. An empirical study of pairwise test set generation using a genetic algorithm//Proceedings of the 7th International Conference on Information Technology. Las Vegas, NV, 2010: 992-997

[12] Grindal Mats, Lindstrom Birgitta, Jefferson Offutt A, Andler Sten F. An evaluation of combination strategies for test case selection. Department of Computer Science, University of Skovde: Technical Report HS-IDA-TR-03-001, 2004

[13] Grindal Mats, Lindstrom Birgitta, Offutt Jefferson A, Andler Sten F. An evaluation of combination strategies for test case selection. Empirical Software Engineering, 2006, 11 (4): 583-611

[14] Frenzel James F. Genetic algorithms, a new breed of optimization. IEEE Potentials, 1993: 21-24

LIANG Ya-Lan, born in 1988, master candidate. His research interests is software testing, especially on combinatorial testing.

NIE Chang-Hai, born in 1971, professor, Ph. D. supervisor. His research interests are software testing techniques, including test suite reduction and optimization, metamorphic testing, evolutionary testing, and especially combinatorial testing.

Background

This work was supported by the National Natural Science Foundation of China (60773104, 61021062, The research on the key issues of combinatorial testing), the National High Technology Research and Development Program (863 Program) of China (2008AA01Z143, Combinatorial testing technology and its support tools) and the National Natural Science Foundation of Jiangsu province, China (BK2010372, The theory, method and application of combinatorial testing).

Combinatorial testing is an effective software testing method which can detect the failures triggered by various factors and their interactions in the software system. To date, the existing research in the world are mainly focused on test case generation, the application of combinatorial testing et al. we will strive to put forward the existing research to a new height. Our research has got some achievements involving the following aspects: combinatorial testing model; mining on the parameters, their values and interactions; various test case generation and prioritization algorithm; the design, evolution and application of combinatorial testing procedure and strategy; software quality evaluation method and testing automation. Our research can not only make combinatorial testing find faults effectively with low cost, but also can diagnosis and reveal faults further around the exposed ones, provide reasonable evaluation method et al. our work in this area will enhance combinatorial testing and provide theory, method and tool for its application and extension.

This paper focused on the study of test case generation and prioritization algorithm. We take genetic algorithm, one of the typical evolutionary search methods, as an example to systemically explore the different influences of its five configurable parameters on the performance of 2-way covering array generation, trying to find the optimal configuration of genetic algorithm and further optimize the performance of the algorithm.

基于类型预测的甚块预测器

苟鹏飞 喻明艳 杨 兵 李清波 王诗博
(哈尔滨工业大学微电子中心 哈尔滨 150001)

摘 要 高性能的甚块预测器是保证EDGE体系结构性能的关键手段.为研究性能更好的甚块预测器,文中通过仿真实验发现甚块的出口类型独立于甚块的出口个数和甚块的动态执行结果而存在.以此为据,提出了基于类型预测的甚块预测器.该预测器摈弃了甚块出口号,直接对甚块出口类型进行预测.随后,根据对甚块出口类型可预测性的分析,通过实验证明甚块出口类型与历史和路径信息相关.仿真结果显示,与经典的基于出口预测的甚块预测器相比,文中提出的基于类型预测的甚块预测器能够将每千条指令误预测次数平均降低约10%.

关键词 甚块预测器;分支预测器;EDGE体系结构;出口类型预测;可预测性
中图法分类号 TP338 **DOI号**: 10.3724/SP.J.1016.2012.01539

Type-Only Hyperblock Predictor

GOU Peng-Fei YU Ming-Yan YANG Bing LI Qing-Bo WANG Shi-Bo
(*Microelectronic Center, Harbin Institute of Technology, Harbin* 150001)

Abstract Since EDGE architecture is hyperblock-based, high performance hyperblock predictors are crucial to guarantee promising performance of EDGE. Based on the analysis of hyperblocks' exit types, we find that the exit type of a hyperblock is independent on the exit ID, motivating us to propose a type-only hyperblock predictor that predicts exit types without exit ID. Analysis of the predictability of exit types for hyperblocks proves that exit types are correlated to histories and/or paths, permitting us to incorporate aggressive prediction techniques into type-only hyperblock predictors to harvest better performance. Compared with conventional exit-based hyperblock predictors, experiments show that MPKI (Mispredicts Per Kilo Instructions) of our proposal is able to outperform by 10%.

Keywords hyperblock predictor; branch predictor; EDGE architecture; exit type predcition; predictability

1 引 言

近年来,诸多文献都已表明,多核/众核技术实际上是微处理器设计者在面临处理器发展瓶颈后的一种无奈之举[1-2],其是否代表未来的方向还有待时间的检验[3-5].根据Amdahl定律[3],处理器的整体性能将仍然在很大程度上受制于其单线程处理能力.因此,进一步探索单处理器设计空间,依然是推动微处理器发展的动力之一.EDGE(Explicit Data Graph Execution)体系结构[6-7]是一种近年来学术界出现的能够改善单处理器能耗、设计复杂度和

收稿日期:2011-07-12;最终修改稿收到日期:2012-02-12.**苟鹏飞**,男,1983年生,博士研究生,研究方向为高性能计算机体系结构.E-mail: pengfeidaxia@gmail.com.**喻明艳**,男,1967年生,教授,中国计算机学会(CCF)会员,研究领域为超大规模集成电路设计、高性能计算机体系结构、模拟集成电路设计等.**杨 兵**,男,1976年生,博士,研究方向为高性能计算机体系结构、高性能编译器等.**李清波**,男,1985年生,硕士研究生,研究方向为高性能计算机关键路径分析.**王诗博**,女,1989年生,本科生,研究方向为指令块预测器.

性能的下一代体系结构. EDGE 体系结构通过块执行(Block-atomic Execution)和指令间显式通信(Direct Instruction Communication)的方式,为下一代体系结构提供了不同于传统 RISC/CISC 体系结构的思考角度. EDGE 体系结构的基本概念[8]、微结构细节[9]、实现方式[10]和整体性能评估[11]并不是本文的研究范围,因此不作详细介绍.

EDGE 体系结构以甚块(Hyperblock)而不是单个指令为基本执行单元[6],所有改变机器状态(寄存器和存储器)的行为都以甚块为单位发生. 甚块[12]是一种单入口、多出口(Single Entry, Multiple Exits)并且包含多个基本块的指令集合. 甚块使用谓词化(Predication)技术,将甚块内基本块之间的控制流转化为数据流,从而保证甚块中能够容纳尽可能多的指令. 甚块执行完毕时,通过出口(Exit)处的跳转指令,跳转到下一个甚块. 而甚块内部的指令则按照纯粹的数据流相关性执行. 为提高性能,EDGE 体系结构通过甚块控制流推测技术,为执行引擎同时提供多个推测的(Speculative)甚块,实现指令窗口的充分填充,从而最大限度地保证对指令级并行的发掘. 甚块控制流推测技术则是通过甚块预测器对下一甚块地址的预测来完成的.

在之前的研究成果中,EDGE 处理器使用基于出口预测的甚块预测器[13]. 该方法在预测时,首先使用出口预测器预测当前甚块的出口号(Exit ID),随后在该出口号基础上完成目标(Target)预测. 这种基于出口预测的方法同样被 Multiscalar[14] 等处理器使用,是目前学术界对指令块进行预测时的通行方法. 该方法中的出口预测与传统分支预测技术中的跳转方向预测相类似,因此,当前的研究人员将绝大部分精力投入到了设计出口预测器中. 然而,由于甚块具备单入口、多出口特性,使得出口预测器需要解决"多选一"问题,而不是传统分支预测器所面临的"二选一"问题,因此,诸多应用于传统预测器的二值预测技术需要经过不同程度的修改,才能够满足甚块出口预测的需要[13,15-16]. 从实验结果来看,尽管使用了诸多激进的预测技术,出口预测器仍然导致了甚块预测器中约 50%的误预测[13,16],是甚块预测器的性能瓶颈.

为设计性能更佳的甚块预测器,本文在对甚块行为及其出口类型进行统计分析后,提出了一种不使用出口预测器,直接对甚块出口类型进行预测的方案,称之为基于类型预测的甚块预测器(Type-only Hyperblock Predictor). 由于不使用出口预测器,简化了甚块预测步骤,从而能够提供更好的甚块预测性能. 总体来说,本文有如下 3 点贡献:

(1) 针对 EDGE 体系结构中的甚块预测问题,根据实验分析和统计结果,提出了一种不使用出口预测器,直接对甚块出口类型进行预测的方案.

(2) 对甚块出口类型的可预测性进行了研究,发现甚块出口类型与历史和路径信息相关.

(3) 构建基于类型预测的甚块预测器,并分析了其性能. 实验结果表明,在针对甚块出口类型的特点使用 TAGE 预测技术后,本文提出的基于类型预测的甚块预测器相对于基于出口预测的甚块预测器,将每千条指令误预测数(Mispredicts Per Kilo Instructions, MPKI)平均降低了约 10%.

本文第 2 节将对应用于 EDGE 体系结构的甚块预测器基本概念及研究现状进行简单介绍;第 3 节将描述甚块出口和出口类型的关系,分析甚块出口类型的统计特性,并对甚块类型预测的合理性进行阐述;第 4 节将给出基于类型预测的甚块预测器结构;第 5 节将介绍本文所使用的仿真方法和基本实验环境;在第 6 节中,将对甚块类型的可预测性进行分析,并比较基于类型预测的甚块预测器与基于出口预测的甚块预测器的性能.

2 背景知识和相关工作

分支预测器(Branch Predictor)是一种为高性能处理器提供推测执行能力的部件. 分支预测器通过预测分支指令的跳转方向,并配合相应的分支目标预测技术,在分支指令执行前获取其跳转结果,为处理器提供了无停顿的推测指令流,增大了处理器的取指能力、指令窗口填充能力,并相应地保证了性能.

与传统的、工作于指令粒度上的分支预测器一样,指令块预测器(Next-block Predictor)是一种为基于块的处理器(Block-atomic Processors)提供控制流推测能力的部件. 虽然指令块预测器有不同的研究背景[13-14,17-20],但其基本思想是一致的,即为每个指令块提供跳转目标预测,使得基于块的处理器可以同时取入多个推测的指令块. 通常来说,在基于块的处理器中,指令块由编译器根据特定的约束将多个基本块组合而成[6,21]. 类似的,EDGE 体系结构为了尽可能地增大指令块大小,使用了甚块技术. 甚块是编译器生成的具有谓词化指令(Predication)的"单入口、多出口"指令块. 甚块将多个传统指令集中

的基本块(Basic Block)集合到一起，因此其内部通常具有多条执行路径，不同的执行路径对应不同的出口.执行开始时，控制流从甚块唯一的入口进入，触发甚块的执行.执行结束时，甚块从其多个出口中，选择且仅选择一个出口作为控制流的转移点，开始下一甚块的执行.为了区分甚块中不同的出口，编译器为甚块的每个出口都分配唯一的出口号(Exit ID).出口号在本质上是编译器对甚块内部不同执行路径的近似[22]，用以表征构建甚块时被条件转换(If-Conversion)技术湮没的相关性.因此，从这个角度来看，用出口号能够从理论上还原某条执行路径的历史，从而使得对该路径的跳转结果进行预测变得可能.由于在对甚块进行预测时，最终的执行路径还未知，因此需要首先对出口号进行预测.甚块“多出口”的特性则对出口号预测提出了与传统分支预测器不一样的要求：

(1)由于具有多个出口，因此传统分支预测器的“2 选 1”预测技术需要被修改为“多选 1”预测技术.

(2)由于具有多种出口类型，因此甚块预测器需要对出口类型进行预测.

目前应用于 EDGE 体系结构中的甚块预测器使用了如图 1 所示的结构，该结构在 TRIPS 原型芯片中得到了应用[11].图 1 中，预测过程由两个步骤构成：出口预测和目标地址预测.首先，使用出口预测器预测甚块在执行完成后跳转出口的出口号.随后，该出口号与指令块地址一起，被用于预测该出口的类型.紧接着，根据出口的类型访问相应的分支目标缓存，产生最终的跳转地址.图 1 中的出口预测器使用了能效比较高的全局/本地锦标赛预测器，而文献[16]中则评估了多种二值预测技术的出口预测潜力，给出了多种不同的预测器结构.对于目标地址预测部分，除基于粘滞位表(Hysteresis)的出口类型预测

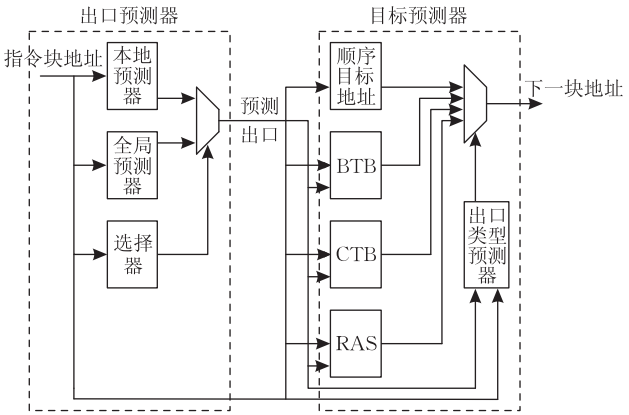


图 1　TRIPS 原型芯片中使用的指令块预测器结构[13]

器外，还有 3 种针对不同出口类型的跳转目标缓存，分别为：分支目标缓存(BTB)、函数调用目标缓存(CTB)和函数返回地址缓存(RAS).此外，如果跳转地址是当前甚块的下一个甚块(In Program Order)，则是顺序跳转目标，可以直接产生跳转地址.

这种先预测出口，再针对特定出口预测跳转目标的策略，是甚块预测中应用最广泛，也是最直观的结构之一[13-14,20].也正因为如此，当前针对甚块预测器的研究几乎都将目光聚焦到了出口预测上，而仅仅为出口类型预测提供非常有限的资源和极为简单的结构.本文将尝试打破这种格局，将目光转移到甚块出口类型预测上，并证明这种方法相对于基于出口预测的方法更加有效.

3　甚块出口类型预测技术

出口预测是当前甚块预测器的重要组成部分.甚块预测器将出口预测得到的出口号与指令块地址相结合，预测该出口的类型，并进一步选择相应的分支类型缓存，输出甚块的跳转地址.由于出口号本质上是编译器对甚块内部执行路径的近似，因此对出口号的使用实际上是利用甚块内部的执行历史来预测跳转类型和跳转目标.但是，由于出口号本身需要经过预测得到，因此一旦这种“推测(Speculative)”的出口号(或者说推测的甚块内部执行历史)出现误预测，对最终跳转目标的影响将呈现错误叠加的效果.从已发表的文献可以看出，约 50%左右的甚块误预测均由出口误预测导致[13,15-16]，体现了出口预测器的低效性.本节对出口预测的缺点进行了分析，并在此基础上，提出了一种摈弃出口预测器，直接进行类型预测的甚块预测方案.

3.1　出口预测器所面临的问题

在甚块预测器中，甚块跳转目标最终由针对不同类型的分支目标缓冲产生.与传统的二值分支预测不同，出口预测器预测得到的出口号并不能直接给出该甚块跳转目标的信息.其原因在于，传统的二值分支预测(主要是针对条件分支指令的预测)如果预测结果为“不跳转(Not-Taken)”，则跳转目标即是该分支指令的下一条指令，无需访问分支目标缓存.因此，传统的二值分支预测器在解决“2 选 1”问题后得到的答案中，有可能直接包含分支跳转目标信息.而对于甚块预测器的出口预测而言，解决“多选 1”问题后，仅仅得到了“推测”的出口号信息，除非进一步通过出口号预测该出口的类型，否则无法

从出口号本身获得甚块的跳转目标.正因如此,出口预测的作用体现在如下两点中:(1)使用预测得到的出口号区分同一甚块中不同出口的分支类型;(2)使用预测得到的出口号区分同一甚块中不同出口的跳转目标.结合图1中甚块预测器的典型结构,出口预测器的功能体现在下述的甚块预测器工作步骤中:

首先,出口预测器使用甚块地址、历史信息、地址路径信息等,预测某一甚块的出口号.

其次,将预测得到的出口号与甚块地址相结合,访问出口类型预测器,得到该出口的类型.

最后,根据出口的类型,选择相应的分支目标缓冲,并使用出口号和甚块地址对其进行索引,得到最终的跳转地址.

从上述步骤中可以看到,出口预测虽然无法直接产生最终的甚块跳转地址,但却影响着出口类型预测和目标预测的结果.一旦出口预测器性能低下,将极大地损害甚块预测器的整体性能.不幸的是,由于出口预测器需要完成"多选1"的使命,其性能相对于传统分支预测器的下降,是显而易见的.其原因在于传统的二值分支预测技术需要经过一定程度的修改才能被应用于出口预测器.这些修改包括:将每次预测产生的历史信息从1位变为多位[13-14,17]、改变预测结果的编码方式(增加计数器位数[13-14,17]、使用PPE预测方法[16])、修改预测器的决策方案(例如将OGEHL预测器的加法树更改为多数投票[16])等.最为重要的是,由于出口预测器的相关信息(执行历史、地址路径等)需要在甚块粒度上获取,相对于基于基本块的分支预测器,其历史信息有可能会丢失.这些因素使得出口预测器的性能变得并不尽如人意,从而使得甚块预测器的整体性能受到影响.

值得注意的是,虽然"多选1"的出口预测器存在上述弊端,但之前的研究人员几乎都倾向于在甚块预测器中保留出口预测器,并尝试使用各种激进的预测技术来改善其性能.这些尝试建立在这样的事实基础之上:甚块中的每个出口分别属于其内部不同的执行路径,由不同的基本块构成,出口号表征了这种湮没在甚块中的路径历史信息.对甚块出口的预测与传统分支预测器对基本块控制流的预测,在本质上是一致的,因此通过改善出口预测的性能来提高预测器整体性能,是十分合理的.

结合甚块预测器本身的工作流程可以发现,即使有上述事实的存在,出口预测器仍然需要额外的条件才能高效地工作:甚块中每个出口都拥有不同的出口类型和不同的跳转目标,通过出口号(或者说甚块内部的执行路径)对其进行区分是必要的.当前的研究人员通常都默认该条件成立,但事实上,由于编译器、指令集、体系结构等诸多因素的影响,情况并非如此.本文将说明在EDGE体系结构中甚块的出口类型与出口号并无直接联系,并且能够在不需要出口预测器的情况下直接对出口类型进行预测.

3.2 甚块出口类型特征

本节将对甚块的出口类型进行分析,以所使用的TRIPS指令集为基础对甚块进行分类,其细节以及相应的仿真方法和测试程序集在第5节中有详细描述.由于使用了TRIPS指令集,因此每个甚块最多拥有4种出口类型,分别为:顺序目标(Sequential Target)、普通分支目标(Branch Target)、函数调用目标(Function Call Target)和函数返回目标(Function Return Target)[10].某个甚块可能具有多个出口,且每个出口的类型可能不同.但由于在执行时,对该甚块的每一次调用(动态实例)都只能使用一个出口,并产生一种特定的出口类型,因此,为分析EDGE体系结构中甚块的出口类型特征,本文首先将甚块按照其动态实例所使用的出口类型分为6大类.分别为

类型1~4.甚块的所有动态实例(Dynamic Instances,即执行时的多次调用)都只产生同一种类型的出口.根据TRIPS指令集[10],这将定义4类甚块,分别为只产生顺序目标的甚块、只产生普通分支目标的甚块、只产生函数调用目标的甚块、只产生函数返回目标的甚块.

类型5.甚块的动态实例既可能产生顺序目标,也可能产生普通分支目标.

类型6.其它的情况.

图2展示了11个SPEC CPU2000程序中,对于动态实例个数排名前100的甚块,按照上述分类方法分类后不同类型甚块所占的比例.从图2中,可以得到

(1)平均约有近80%的甚块在执行过程中只产生某种特定的出口类型.也就是说,无论这些甚块从哪个出口跳转,其出口类型都将始终保持一致.

(2)平均约有近20%的甚块在执行过程中,即可能产生顺序目标,亦可能产生普通分支目标.这些甚块的行为与普通的条件分支指令非常类似.即如果这些甚块产生顺序目标,那么其跳转地址将可以直接得到,无须访问任何分支目标缓存,这与普通条件分支指令的"不跳转"情况类似;如果这些甚块产

生普通分支目标，那么最终的跳转目标地址将通过访问 BTB 得到，这同样与普通条件分支指令的“跳转”情况类似.

(3) 其它类型的甚块数目微乎其微，小于 1%.

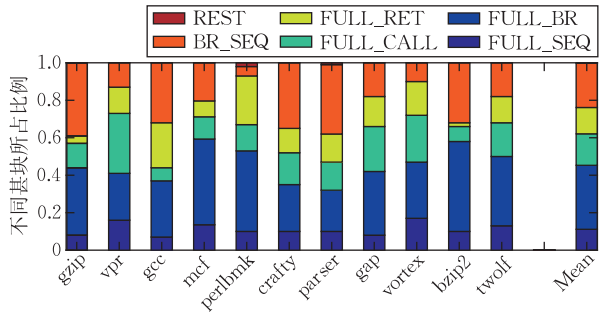


图 2　甚块的出口类型特征统计(其中，FULL_SEQ、FULL_BR、FULL_CALL、FULL_RET 分别表示动态实例只产生某种特定类型出口的甚块；BR_SEQ 表示动态实例中既产生顺序目标又产生普通分支目标的甚块；REST 为其它所有情况)

从上述现象能够得出这样的结论：尽管文献[16]中的数据说明甚块的出口号分布较为均匀①，但其出口类型具有较强的特征，且该特征自成体系. 进一步观察可以看到，其中大部分甚块(约 80%)的偏向性极强. 由于这种极强的偏向性，这些甚块的出口类型在不需要出口预测器的情况下，通过简单的、不使用相关信息的粘滞位表就能够很好地预测. 本文设计了实验来初步验证上述结论. 实验中将 TRIPS 原型芯片甚块预测器中的出口预测器删除，直接使用简单粘滞位表(仅使用甚块地址来索引该表，不使用历史信息)来预测甚块出口类型，与原有的 TRIPS 原型芯片预测器进行性能比较. 原有的 TRIPS 原型芯片预测器配置与文献[13]中相同. 两种方案均为出口预测器和类型预测器分配 32 KB 的资源，并使用每千条指令误预测数(Mispredictions Per Kilo Instructions，MPKI)作为量化标准，这也是评估预测器性能的常用指标[13,23]. 比较结果如图 3 所示，图中给出了对 11 个 SPEC CPU2000 整型程序的 MPKI 剖析，将 MPKI 归结为不同的来源，分别为：出口预测器(EXIT)、出口类型预测器(TYPE)、普通分支缓存(BTB)、函数调用目标缓存(CTB)和函数返回缓存(RAS).

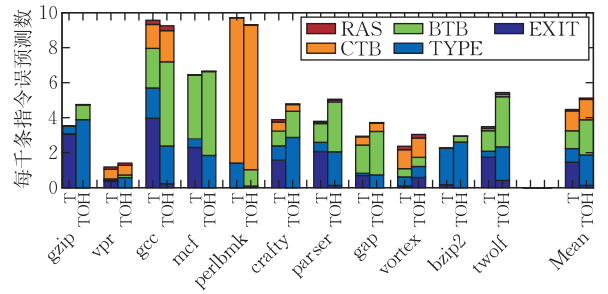


图 3　使用简单粘滞位表对甚块进行预测的结果(其中 T 为 TRIPS 原型芯片甚块预测器；TOH 为无出口预测器且仅使用简单粘滞位表进行出口类型预测的甚块预测器)

从图 3 中可以看到，相对于原有的 TRIPS 原型芯片预测器而言，删除了出口预测器且只用了简单粘滞位表进行甚块出口类型预测的方案，其 MPKI 平均仅仅上升了 8% 左右. 更为重要的是，原有的 TRIPS 原型芯片预测器中由出口预测器和类型预测器导致的 MPKI，在无出口预测器的方案中，转换为了几乎等量的由类型预测器导致的 MPKI，且有约 10% 的下降. 该现象说明，甚块的出口类型在无出口预测器的情况下就能够进行预测，并且凭借简单的仅用甚块地址进行索引的粘滞位表就能够获得较好的预测效果. 这与前述 EDGE 体系结构中约 80% 甚块的出口类型具有极强的偏向性是相符的. 另外约 20% 的甚块由于其行为与条件分支指令行为类似，因此，可以大胆假设这些甚块的出口类型与甚块“间”历史或地址路径信息(而不是出口号)相关，使用更为复杂的相关性预测技术能够对出口类型进行较好的预测. 本文将在第 6 节中通过实验结果来证明这个假设.

总而言之，本节的实验数据说明，甚块的出口类型呈现出独立于出口号的特征，这样的特征使其可在不使用出口号时，直接被出口类型预测器所预测. 因此，前述甚块中每个出口都拥有不同的出口类型，通过出口号对其进行区分是必要的这个假设，实际上在 EDGE 体系结构中并不成立. 进而可以认为，至少对于甚块的类型预测来说，出口预测器并不是必要的.

值得注意的是，图 3 中，虽然删除出口预测器后甚块出口和出口类型的预测性能提高了，但总体 MPKI 还是有约 8% 的上升，这是由 BTB 所导致的. 由于在无出口预测器的方案中，仅使用了甚块地址来索引 BTB，因此 BTB 无法区分同一个甚块产生的不同跳转目标. 这种现象说明，甚块中不同的出口确实会产生不同的跳转目标；甚至同一出口，同样的出口类型，也会由于间接跳转指令的存在[10,16]，产生不同的跳转目标. 虽然这个现象部分说明前述甚块中每个出口都拥有不同的跳转目标，通过出口号对其进行区分是必要的这个假设成立，但该问题实际可以通过使用相关性分支目标缓冲技术来解决

① 实验数据也说明出口号的分布较为均匀.

(Correlative BTB)[14,24-25],即将甚块地址和全局历史或地址路径信息结合到一起来索引 BTB. 由于这并不是本文的研究目标,因此并不对这个问题作详细阐述. 尽管如此,本文随后的实验将证明即使不使用相关性分支目标缓冲,本文所提出的方案依然具有较好的性能.

3.3 如何看待甚块类型预测的合理性

为更形象和深入地理解无出口预测的甚块类型预测,需要进一步分析甚块的跳转行为. 实际上,可以将甚块的跳转行为看作具有间接跳转目标的条件分支指令,如图 4 所示. 从图中可以看到,无论甚块拥有多少出口,从哪个出口跳转,最终呈现的结果无外乎两种情况:(1) 跳转到该甚块在程序上的下一甚块;(2) 跳转到其它甚块. 前一种情况,即是出口类型预测中预测到顺序分支目标的情况;而后一种情况,可以进一步细分为 3 种不同的分支类型,并通过不同类型的分支目标缓冲(在 TRIPS 指令集中,分别是 BTB、CTB、RAS)来得到跳转目标. 因此,从甚块是否为顺序分支目标这个角度来看,可以将其看作具有"跳转"或"不跳转"行为的条件分支指令. 而从甚块跳转时需要根据不同的执行上下文来确定跳转目标的角度来看,则可以将其看作间接跳转指令. 以这样的角度为基础,直接使用甚块出口类型构成的历史则与条件分支指令的跳转历史相似. 该历史表征了甚块的跳转行为,可被出口类型预测器用于更准确地预测甚块出口类型,从而在不需要出口号的情况下完成甚块跳转地址的预测任务. 这就能够避免传统甚块预测器中低效的出口预测器对出口类型预测性能的影响,直接将重点聚焦到甚块类型预测上,实现问题的简化,提供更多的性能提升可能.

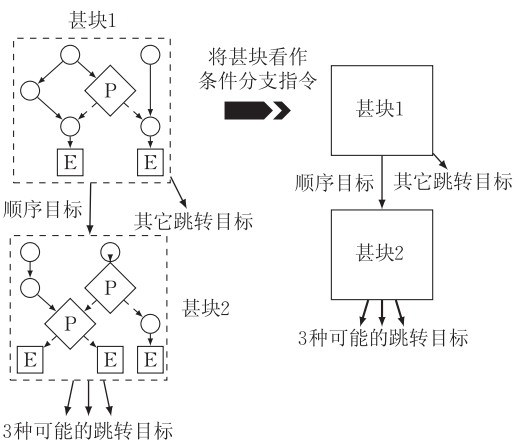


图 4 将甚块看做间接跳转的条件分支指令

从分支历史相关性角度来看,取消出口预测事实上是在预测过程中忽略了甚块"内"的相关信息,这种相关信息对被预测的甚块而言是"推测"的,不确定的;而直接使用甚块的出口类型构成历史,则是使用甚块"间"的相关信息来指导预测器的工作,这种相关信息是既有的、确定的. 本文随后的实验将证明,使用甚块"间"的相关信息能够获得更好的甚块预测性能.

本节通过上述分析得到了如下 3 点结论. 首先,传统甚块预测器中的出口预测器是整体性能的瓶颈,并且仅仅产生预测过程的中间结果,不包含直接的甚块跳转目标信息;其次,甚块的出口类型具有独立于甚块出口号的特性,并具备被出口类型预测器直接预测的潜力;再次,甚块的出口类型表示了甚块"间"的相关性,能够在没有出口号的情况下完备地描述甚块的跳转行为. 综合这 3 点因素,可以认为,在没有出口预测器的情况下直接对甚块的出口类型进行预测,是一种可行的、并具有性能提升潜力的方案. 鉴于此,本文将在该思路的指导下,提出基于类型预测的甚块预测器结构.

4 基于类型预测的甚块预测器结构

4.1 总体结构

本节根据上述分析结果,提出一种基于类型预测的甚块预测器结构. 由于甚块出口类型也有多种可能(TRIPS 指令集定义了 4 种),因此需要使用与出口预测器类似的多值预测器. 具体的做法是,使用"4 选 1"类型预测器从顺序目标、普通分支目标、函数调用目标和函数返回目标中确定当前的预测结果,并根据预测结果访问相应的分支目标缓存,得到最终的跳转目标. 这种类型预测器所使用的历史信息与传统分支预测器中所使用的二值历史信息类似,由每个甚块产生的出口类型组成,表征了甚块"间"的执行历史. 与传统甚块预测器所使用的出口号历史信息不同,这种历史信息仅仅保留甚块间的相关性,忽略了甚块内的执行路径信息. 由于一共需要预测 4 种类型,因此每一个甚块会产生 2 位历史. 地址路径则是由执行过程中甚块的地址构成,表征了执行轨迹上已执行的甚块地址.

该方案依然使用多值预测结构,可能会由于多值预测本身的缺点造成性能损失,但由于直接使用了甚块"间"相关信息对出口类型进行预测,因此并不会受到出口预测器的干扰. 另一方面可以将有限

的资源全部投入到出口类型预测器中，提高改善性能的可能性. 具体的预测器结构如图 5 所示. 图中使用与出口预测器类似的多值预测器，在历史或地址路径信息的帮助下直接对出口类型进行预测. 如果甚块类型预测器预测结果为顺序目标，则直接产生预测的跳转目标，如果是其它 3 种类型，则根据相应的分支目标缓冲产生预测结果.

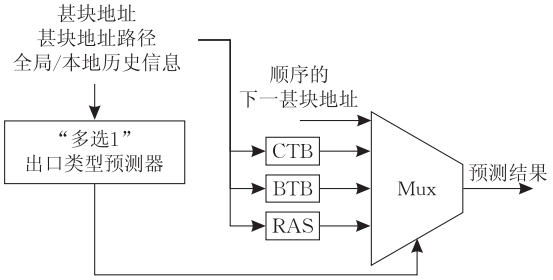
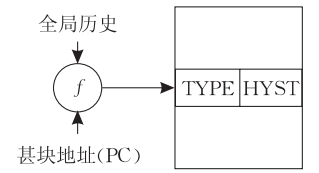


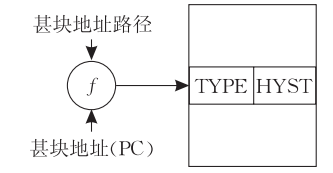
图 5　基于类型预测的甚块预测器结构

4.2　使用历史信息和甚块地址路径的甚块类型预测器

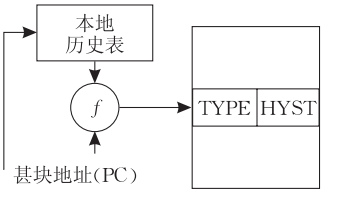
图 5 中的多值预测器可用类似经典 gshare[26] 预测器的结构实现，如图 6 所示. 该结构将甚块地址和全局/本地历史信息（或者甚块地址路径信息）相结合来索引相关性表，是一种利用历史和地址路径相关性进行分支预测的经典结构. 图 6(a)～(c)分别为全局历史、全局路径和本地历史预测器，分别使用相应的历史和地址路径信息与甚块地址按照一定的方法（通常是异或）结合后，对样本历史表（Pattern History Table）进行索引①. 表中每一项由两部分组成，第一部分存储出口类型（TYPE），即当前访问该项能够得到的预测结果；另一部分则是使用饱和计数器实现的粘滞位（HYST），用于更新预测结果. 值得注意的是，该结构中历史信息由每一个甚块产生的出口类型构成，而地址路径信息则由执行轨迹上的甚块地址构成. 与传统分支预测器或出口预测器一样，图中的 3 种预测器方案可以两两组合为锦标赛预测器. 另外，在对甚块出口类型的历史和路径相关性进行研究时，这 3 种预测器也是最具代表性的结构.



(a) 全局历史预测器



(b) 全局路径预测器



(c) 本地历史预测器

图 6　全局历史预测器、全局路径预测器和本地历史预测器

4.3　使用 TAGE 结构的甚块类型预测器

在传统的二值分支预测器中，TAGE（TAgged GEometric history）预测器[23] 是硬件可实现的性能最好的预测器之一. 文献[13]和文献[16]中就利用 TAGE 预测技术，实现了基于出口预测的甚块预测器. 对出口类型预测器来说，TAGE 预测技术同样能够提供新的设计思路.

TAGE 预测技术使用数个不同的历史样本表，分别由呈几何级数增长的历史长度索引，并通过精巧的更新机制和标签（TAG）匹配机制，使得与不同历史长度相关的分支指令可以被相应长度的历史所预测，从而避免不同历史样本之间的别名冲突. 根据 TAGE 预测技术的特点，针对 3.2 节中描述的甚块出口类型特性，本文使用 TAGE 预测技术设计了一种 TAGE 甚块出口类型预测器，可用于图 5 中的多值预测，其结构如图 7 所示. 该 TAGE 甚块类型预测器由简单粘滞位表 T_0（仅使用甚块地址进行索引，历史长度为 0）以及一组历史长度呈几何级数增长的历史样本表 $T_1 \cdots T_n$ 组成，该几何级数历史长度由 $L(j)=\alpha^{(j-1)}L(1)$[23] 计算得到. 与经典 TAGE 预测技术类似，表 $T_1 \cdots T_n$ 中每一项由 3 部分组成：预测结果（TYPE）、粘滞位（HYST）和标签（TAG），T_0 表中则没有标签位. 更新机制和标签匹配机制与文献[23]中相同. 根据 TAGE 预测技术的特点，结合图 2 中给出的甚块出口类型特性，TAGE 甚块类型预测器可以达到如下效果：(1) 甚块中 80% 偏向性极强、跟历史信息无关的甚块，将由简单粘滞位表 T_0 来预测；(2) 剩余 20% 与历史相关的甚块，将分别由 $T_1 \cdots T_n$ 中相应的表来预测. 这就使得不同特性甚块之间发生别名冲突的可能性大大减小，从而提高预测性能. 随后的章节中，将对本节所提出的甚块类型预测器进行性能评估.

① 虽然不同预测器结构有所不同，但通常都有类似样本历史表（PHT）的结构.

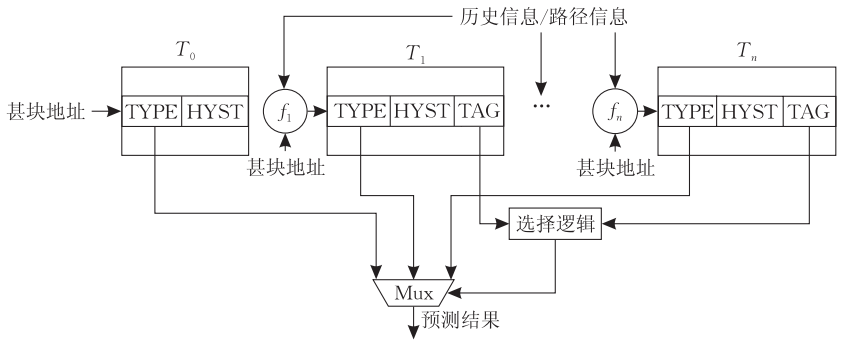


图 7 基于 TAGE 技术的甚块出口类型预测器

5 仿真环境和评估方法

由于本文的诸多分析都基于实验方法得到,因此,在进行详细的性能评估前,本节将首先描述所采用的仿真环境和性能评估方法. 本文所有的实验数据都将在本节所描述的仿真工具、基本配置参数和测试程序集基础上得到.

5.1 仿真工具

本文使用 M5_EDGE[27]模拟器作为仿真工具. M5_EDGE 是基于 M5[28]模拟器,使用 C++/Python 构建的 EDGE 体系结构仿真工具,支持 TRIPS 指令集[10],具有高层次的 4 级时序模型,能够在微结构层面对 EDGE 体系结构进行快速的设计空间探索. 得益于 M5_EDGE 良好的面向对象和模块化特性,本文能够方便地对甚块预测器进行建模,并在统一的仿真框架和基准配置参数下进行比较. 在仿真时,所使用的基准配置如表 1 所示,文中所涉及到的预测器都将在本框架下实现. 为了消除处理器其它部分对预测器的影响,操作数网络(Operand Network)、谓词化指令(Predicated Instruction)、Cache 和访存顺序(Memory Access Order)都被配置为理想状态,即仿真时操作数网络延迟为 0、甚块中的谓词化指令在执行前就已经获取了结果、Cache 总是命中、访存顺序在执行前就已经获得.

表 1 基准处理器配置参数

取指	每周期 16 条指令(一个 Cache Line)
指令映射策略	静态映射,并有 1 个周期的映射延迟
指令分派开销	1 周期
指令窗口大小	1K 指令(8 个 TRIPS 的 Hyperblock)
操作数网络延迟	完美
执行引擎	每周期最多 16 条指令能被同时执行
谓词化	完美
提交延迟	1 周期
Cache	完美
Memory 访问顺序	完美

5.2 指令集、编译策略和测试程序

本文选择 TRIPS 指令集的原因在于:其一,TRIPS 指令集是开发较为完善的 EDGE 指令集,工具链较为完善;其二,TRIPS 指令集在 M5_EDGE 中有较好的支持. 在编译时,本文使用了 TRIPS 工具链中 tcc 的-Omax 优化选项[29]. 对于有出口预测的仿真,编译器将为每个甚块的出口按照正常的策略分配出口号;对于只有类型预测的仿真,编译器将为每个甚块的出口都分配出口号 0.

本文使用 SPEC CPU2000 中能够被 TRIPS 工具链正确编译的 11 个整形测试程序作为测试基准. 之所以这样选择是因为整形程序对于控制流推测的要求较高,能够更好地体现甚块预测器特性,而浮点程序对于甚块预测器的性能则不是那么敏感. 在仿真时,选择了 ref 输入集,并使用 Simpoint[30]仿真方法,以便将仿真时间缩短到可接受的范围之内.

6 实验结果分析与性能评估

本节将在第 5 节所示的实验方法基础上,使用不同方案来实现甚块出口类型预测器,并观察、分析和比较其性能. 首先,本节将使用无冲突(Interference-free)的全局预测器、地址路径预测器和本地预测器对甚块出口类型的可预测性及其与历史/地址路径信息的相关性进行分析. 随后,将比较本文提出的基于类型预测的甚块预测器和传统的基于出口预测的甚块预测器的性能. 由于本文主要关注二者性能的差异,因此本节实验所使用的分支目标缓冲大小均固定,分别为 BTB 拥有 4096 个入口、CTB 拥有 128 个入口、RAS 拥有 64 个入口,每个入口的具体实现与文献[13]中保持一致.

6.1 甚块出口类型的可预测性分析

使用无冲突(Interference-free)表对分支指令的可预测性进行研究是一种由来已久的方法[31-32].

文献[16]中，作者使用类似的方法对 EDGE 体系结构中甚块出口的可预测性进行了研究.通过使用无冲突表，预测器可以在表中为每一个样本(pattern)分配独立的入口(Entry)，从而使得不同的样本之间不存在别名(Aliases)冲突，将历史或地址路径对预测能力的影响最纯粹地反映出来.

图 8 中给出了使用图 6 中 3 种预测器时，11 个 SPEC CPU2000 整型程序在历史/地址路径信息的长度从 0 比特变化到 63 比特时的平均 MPKI，其中每种预测器都使用了无冲突表.对于全局历史预测器和地址路径预测器，历史/路径和甚块地址使用了两种方式进行组合.一种为使用 gshare 中给出的异或(XOR)方式，另一种为拼接(Concatenated)方式，即从甚块地址和历史/路径信息中各取一部分，互不干扰地组合起来形成表的索引.本地预测器中，历史和路径按照异或方式结合.地址路径信息使用执行轨迹上甚块地址的低 4 位组合而成，而历史信息中每个甚块产生 2 比特历史，对应着甚块出口类型的 2 位编码(TRIPS 指令集中共 4 种出口类型).

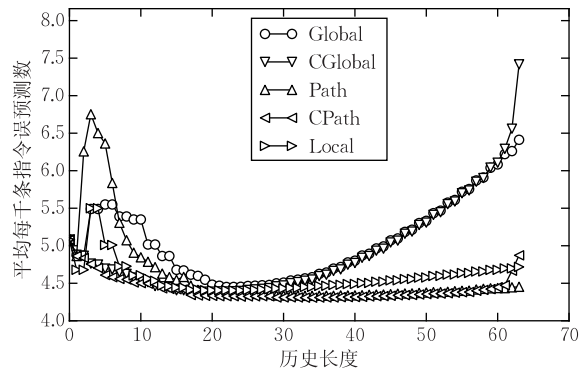


图 8　对于基于类型预测的甚块预测器，使用无冲突样本表时全局历史、全局路径和本地预测器的 MPKI 变化情况(其中，Global/CGlobal 分别为使用异或策略和拼接策略的全局历史预测器；Path/CPath 分别为使用异或策略和拼接策略的地址路径预测器；Local 为本地预测器)

图 8 中的结果显示：当历史长度为 0 时，即仅使用甚块地址对表进行索引时，所有预测器的 MPKI 相等，约为 5.07.由于本实验中的预测器使用无冲突表，因此这也是使用简单粘滞位表进行甚块出口类型预测的极限值.从第 3.2 节中对甚块出口类型分布特征的分析可知，当历史长度为 0 时预测器主要对约 80%具有极强偏向性的甚块提供准确预测.而与图 3 中使用简单粘滞位表(32 KB)进行甚块出口类型预测的结果(平均 MPKI 为 5.15)相比，该极限值仅仅下降了约 1.6%.这说明使用大小为 32 KB 的粘滞位表已经足够对约 80%具有极强偏向性的甚块出口类型进行预测.如希望进一步提升预测率，只能寄望于改善另外约 20%甚块的预测性能.

随着历史长度的增长，有两种不同的趋势.其一，对于使用拼接方式进行索引的预测器，MPKI 随着历史长度的增长持续而稳定地下降.这说明两层含义：(1)甚块出口类型与全局历史/地址路径信息相关；(2)该相关性可以被出口类型预测器捕捉.其二，对于使用异或方式进行索引的预测器，MPKI 在短暂的下降(历史长度小于约 2 bit)后，突然上升，直到历史长度继续增加到一定程度(全局历史预测器为大于 13 bit，地址路径预测器为大于 9 bit，本地历史预测器为大于 5 bit)，MPKI 才重新下降到比不使用历史/路径相关信息(0 bit 历史长度)时更小的程度.这说明，使用异或索引方式时，在历史/地址路径长度较短的情况下，预测器能够捕获部分甚块的历史/地址路径相关性.当历史/地址路径长度增加时，在某个区间段内，由甚块出口类型自身特点和异或索引方式共同作用导致了较多的样本重复(即虽然历史不同且甚块地址不同，但异或之后的索引却相同)，使得 MPKI 出现了上升趋势.这样的上升趋势在历史/路径长度增大到一定程度时得到了抑制，这说明当历史长度足够时，出口类型预测器依然有能力利用历史/地址路径相关性获得性能提升.

当历史长度继续增加时，所有预测器都在历史长度 15 bit 到 25 bit 之间达到性能极限，MPKI 的最小值为 4.32(使用拼接索引的全局路径预测器，历史长度 20)，相对于历史长度为 0 时有约 14%的下降.之后，所有预测器的 MPKI 都出现不同程度的上升，且全局历史预测器相对于地址路径预测器上升更为明显.这说明甚块出口类型在本文的实验条件下，与距离太远的甚块(对于本地预测器，则是相距太久的动态实例)之间的相关性并不明显，且地址路径信息相对于历史信息更能反映较远距离甚块的相关性.另外，使用拼接方式进行索引的预测器在历史长度增加到 60 之后，MPKI 有较为明显的突然增加.这主要是由于最大索引长度固定为 64，从而使得甚块地址信息在历史和地址路径信息太长后造成了丢失，导致索引样本的差异化减弱，增加了误预测次数.这也说明，甚块类型不能单纯使用历史或地址路径信息进行预测.

总体来看，对全局预测器而言，使用地址路径信息比使用历史信息性能更优.地址路径预测器的 MPKI 最小为 4.32(使用拼接索引的全局路径预测

器),而全局历史预测器的 MPKI 最小为 4.39,上升了约 1.5%.整体而言,地址路径预测器的 MPKI 亦小于全局历史预测器.另外,本地预测器性能则介于地址路径预测器和全局历史预测器之间.综上所述,本节通过分析使用无冲突表的全局历史/地址路径预测器和本地历史预测器在不同历史长度下的性能,能够得出如下结论:

(1)甚块出口类型与甚块"间"的历史/地址路径信息(全局/本地)相关,在忽略甚块"内"的相关信息(出口号)后,利用这些甚块"间"的相关信息能够获得预测性能的提升.这证明了第 3.2 节的猜测:对于行为与条件分支指令类似的甚块(约占 20%),其出口类型与甚块"间"的历史/地址路径信息相关.

(2)在预测甚块出口类型时,使用地址路径信息比使用全局历史信息更有效,且本地历史预测器也能够获得可观的预测准确率.

6.2 基于类型预测的甚块预测器与基于出口预测的甚块预测器性能比较

上节的分析证实了甚块出口类型与甚块"间"的历史/地址路径信息相关,因而可以利用这些信息来提升其预测性能.本节将从性能的角度比较传统的基于出口预测的甚块预测器和本文所提出的基于类型预测的甚块预测器.比较时,对于基于出口预测的甚块预测器,将使用文献[13]中所提出的全局/本地锦标赛出口预测方案,其具体结构如图 1 所示.这也是基于出口预测的甚块预测器在相对简单的结构下能达到较好性能的方案.对于基于类型预测的甚块预测器,将在图 5 所示结构的基础上,使用全局/本地锦标赛预测器实现其中的"多选 1"类型预测器,直接对 4 种甚块出口类型进行预测.比较时为出口预测器和类型预测器分配的资源将从 1KB 变化到 512KB,资源的大小根据每个表的入口数和每个入口的比特数来确定,具体的配置信息如表 2 所示.

表 2 中,G/L EXIT 为基于出口预测的全局/本地锦标赛甚块预测器,其中 $G/L/C$ 分别为其中的全局历史长度、本地历史长度和选择器历史长度,相应的样本表入口数为 $2^{G/L/C}$,H 则为该预测器中用于在出口预测后进行类型预测的粘滞位表入口数;H TYPE为基于类型预测的仅使用简单粘滞位表的甚块预测器,其中 H 表示其粘滞位表的入口数;P/L TYPE 为基于类型预测的全局路径/本地锦标赛甚块预测器,其中 $L/G/C$ 分别为本地历史长度、地址路径长度和选择器历史长度,相应的样本表入口数为 $2^{L/G/C}$.另外,对于预测出口的样本表来说,其每个入口由 3 比特出口号信息和 1 比特粘滞位构成;对于预测出口类型的样本表来说,其每个入口由 2 比特出口类型信息和 1 比特粘滞位构成;对于选择器样本表来说,每个入口由 3 比特粘滞位构成.

表 2 性能比较配置表

	G/L EXIT				H TYPE	P/L TYPE		
	G	*L*	*C*	*H*	*H*	*L*	*G*	*C*
1KB	10	7	10	512	2KB	8	10	10
2KB	11	8	11	1K	4KB	9	11	11
4KB	12	9	12	1K	8KB	10	12	12
8KB	13	10	13	2K	16KB	11	13	13
16KB	14	11	14	2K	32KB	12	14	14
32KB	15	12	15	4K	64KB	13	15	15
64KB	16	13	16	4K	128KB	14	16	16
128KB	17	14	17	4K	256KB	15	17	17
256KB	18	15	18	4K	512KB	16	18	18
512KB	19	16	19	4K	1MB	17	19	19

图 9 中给出了针对 11 种 SPECCPU2000 测试程序的平均 MPKI.作为对比,基于类型预测的甚块预测器使用了两种方案:(1)地址路径(使用拼接索引方式)/本地预测器(PATH/LOCAL_TYPE)锦标赛方案,这也是在对甚块出口类型的可预测性进行分析后,得出的最优组合;(2)使用简单粘滞位表(即历史长度为 0,仅使用甚块地址进行索引)进行甚块出口类型预测的方案(HYST_TYPE),该方案用于展示无相关信息时,甚块出口类型预测能力的变化.另外,对于基于类型预测的地址路径/本地锦标赛甚块预测器,还引入了使用无冲突表(IF_PL_TYPE)时的情况(该方案中历史长度与图 9 中普通方案相同,但使用无冲突表),以便观察使用锦标赛预测器时甚块出口类型预测的极限.

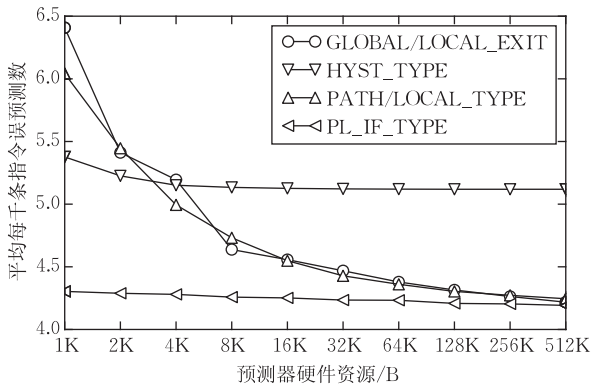


图 9 基于类型预测的甚块预测器和基于出口预测的甚块预测器性能比较(其中,GLOBAL/LOCAL_EXIT 为基于出口预测的全局路径/本地锦标赛甚块预测器;HYST_TYPE 为基于类型预测的仅使用简单粘滞位表的甚块预测器;PATH/LOCAL_TYPE 为基于类型预测的地址路径/本地锦标赛甚块预测器;PL_IF_TYPE 为使用无冲突表的基于类型预测的地址路径/本地锦标赛甚块预测器)

图 9 中的结果说明：首先，当资源从 1 KB 变化到 512 KB 时，对基于类型预测的地址路径/本地锦标赛甚块预测器而言，其 MPKI 和基于出口预测的全局历史/本地锦标赛甚块预测器几乎相等. 除在 1 KB 时，基于类型预测的方案能够将 MPKI 降低约 8%外，其余情况下，两者性能持平. 其次，在资源最小(1 KB)时，基于类型预测的简单粘滞位表甚块预测器性能最好，相对于基于出口预测的全局历史/本地锦标赛甚块预测器，MPKI 降低了约 15%，相对于基于类型预测的地址路径/本地锦标赛甚块预测器，MPKI 降低了约 10%. 其原因在于，当资源受限时，由于甚块出口类型本身较强的偏向性(约 80%的甚块只产生一种特定的出口类型)，无历史信息、仅使用甚块地址对甚块出口类型进行预测的简单粘滞位表依然能够获得可观的预测准确率. 而此时，由于资源较少，历史样本表入口数受限，因此如果使用地址路径/本地锦标赛甚块预测器进行出口类型预测，会因为相关信息的加入而导致太多不必要的别名冲突，从而增加了误预测次数. 对于基于出口预测的甚块预测器而言，资源受限使得出口误预测次数增多，并因此导致更加低效的出口类型预测和跳转目标预测，使得其总体性能在资源受限时最差. 但是，使用简单粘滞位表的方案由于不使用相关信息，因此其虽然在资源很小时能够提供较好的性能，但随着资源的增大很快就陷入饱和(资源大于 4 KB 后)，从而无法提供持续的性能增长. 最后，基于类型预测的地址路径/本地锦标赛甚块预测器与使用无冲突表的方案相比，其 MPKI 还有较大差距，该现象在资源小于 128 KB 时极为突出. 这一方面说明在资源受限时，地址路径/本地锦标赛方案确实由于别名冲突的增多而导致性能下降；另一方面，该现象也说明，如果能够寻找到更高效的结构，则可能将 MPKI 进一步降低.

6.3　使用 TAGE 预测器提升甚块类型预测器性能

正如前述，对于甚块出口类型的预测而言，如果能够找到更优秀的方案，那么其性能还有较大的上升空间. 图 7 中给出的 TAGE 甚块类型预测器正是这样一种潜在的方案. 为分析 TAGE 甚块出口类型预测器的性能，本节在图 5 中所示的基于类型预测的甚块预测器结构基础上，用 TAGE 甚块出口类型预测器实现其中的“多选 1”预测器. 另外，传统的 TAGE 预测器通常使用全局历史/地址路径进行索引，而为了利用本地历史信息，本节仿照锦标赛预测器的方式，将 TAGE 类型预测器和本地类型预测器结合到一起. 作为对比，引入了在文献[13]中使用的基于出口预测的 TAGE/本地甚块预测器(在具体实现时，该预测器同样将 TAGE 出口预测器和本地出口预测器以锦标赛预测的方式结合). 这两种预测器的具体配置如表 3 所示. 其中，所有的配置都使用了典型的有 5 个表的 TAGE 结构[23]. 表中，T/L TYPE 是基于类型预测的 TAGE/本地甚块预测器，其中 L/C 分别为本地历史长度和选择器历史长度，相应的样本表入口数为 $2^{L/C}$，$\alpha/l(1)$ 分别为 TAGE 预测器历史长度计算参数，相应样本表 $T_1 \cdots T_n$ 的大小由计算得到的历史长度按照与本地历史预测器和选择器相同的方式得到，D 为 TAGE 预测器中默认表 T_0 的入口数；T/L EXIT 是基于出口预测的 TAGE/本地甚块预测器，其中 $L/C/\alpha/l(1)/D$ 参数之含义与 T/L TYPE 中一致，样本表的入口数计算方式亦相同，H 为在出口预测之后，用于类型预测的简单粘滞位表入口数. 在本配置中，本地历史预测器和选择器的样本表入口配置与表 2 中一致. TAGE 预测器的样本表中，如果是进行出口类型预测，则每个入口由 2 比特出口类型信息、2 比特粘滞位和 9 比特标签位构成，其中表 T_0 无标签位；如果进行出口预测，则每个入口由 3 比特出口号信息、2 比特粘滞位和 9 比特标签位构成.

表 3　性能比较配置表

	T/L TYPE					T/L EXIT					
	L	C	α	$l(1)$	D	L	C	α	$l(1)$	D	H
1 KB	9	10	1.2599	3	512	7	8	1.205	4	512	512
2 KB	10	11	1.2599	4	512	8	9	1.4938	3	512	512
4 KB	11	12	1.2164	5	1 K	9	10	1.542	3	1 K	1 K
8 KB	12	13	1.542	3	1 K	10	11	1.4422	4	1 K	1 K
16 KB	13	14	1.5874	3	2 K	11	12	1.375	5	2 K	2 K
32 KB	14	15	1.4812	4	2 K	12	13	1.2599	7	2 K	2 K
64 KB	15	16	1.671	3	4 K	13	14	1.2331	8	4 K	4 K
128 KB	16	17	1.4422	5	4 K	14	15	1.2114	9	4 K	4 K
256 KB	17	18	1.4736	5	8 K	15	16	1.1934	10	8 K	8 K
512 KB	18	19	1.415	6	8 K	16	17	1.1784	11	8 K	8 K

性能比较结果如图 10 所示. 作为参照，图中引入了上节中基于出口预测的全局历史/本地锦标赛甚块预测器和基于类型预测的地址路径/本地锦标赛甚块预测器. 从图 10 中可以看到，基于类型预测的 TAGE/本地甚块预测器对性能的提升非常明显. 首先，相对于基于类型预测的地址路径/本地锦标赛预测器，该方案将 MPKI 在资源为 1 KB 时降低了约 18%. 虽然该差距随着资源的增多而逐渐缩小，但使用 TAGE 技术的方案一直保持着优势，直到 512 KB 时两者的 MPKI 才趋于相等. 其次，相对于基于出口预测的全局历史/本地锦标赛甚块预测

器,总体趋势相同.即在资源为1KB时该方案能将MPKI降低约23%,直到512KB时,两者MPKI才趋于相等.最后,相对于基于出口预测的TAGE/本地甚块预测器,总体趋势亦相同.在资源大于256KB后,基于出口预测的TAGE/本地甚块预测器略占优势,MPKI低4%左右.

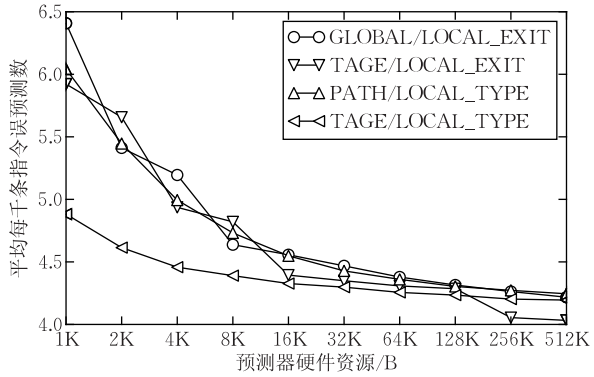


图10 使用TAGE技术的预测器比较(其中,GLOBAL/LOCAL_EXIT为基于出口预测的全局历史/本地锦标赛甚块预测器;TAGE/LOCAL_EXIT为基于出口预测的TAGE/本地甚块预测器;PATH/LOCAL_TYPE为基于类型预测的地址路径/本地锦标赛甚块预测器;TAGE/LOCAL_TYPE为基于类型预测的TAGE/本地甚块预测器)

总体来说,基于类型预测的TAGE/本地预测器在资源较小时效果较好,平均将MPKI降低了约10%.随着资源的增大,基于类型预测的TAGE/本地预测器一直保持优势.基于出口预测的TAGE/本地甚块预测器仅在资源大于256KB后才能达到更低的MPKI.而1KB到64KB之间通常是处理器能够为甚块预测器分配硬件资源的合理范围,因此,基于类型预测的TAGE/本地预测器在较低资源时的性能领先具有较强的现实意义.

总而言之,上述试验结果说明TAGE技术能够提供更好的甚块出口类型预测性能,从而是一种较好的、适用于直接预测甚块出口类型的技术.这是因为TAGE技术能够很好地与甚块出口类型的特性相契合,能根据甚块出口类型特性的不同选择相应的条件进行预测.上述试验结果也说明,相对于基于出口预测的甚块预测器,基于类型预测的甚块预测器能够提供更低的MPKI,从而是一种更好的保证处理器甚块控制流推测能力的方案.

7 结 论

简而言之,本文所提出的直接对甚块出口类型进行预测的方案,简化了甚块预测步骤,提高了甚块的预测能力.值得注意的是,虽然本文以EDGE体系结构作为研究,但本文对甚块出口类型的分析和结论对其它甚块体系结构中的甚块预测器设计来说,同样具有指导和借鉴意义.

本文后续的研究工作包括:(1)进一步探索甚块出口类型的特性,开发性能更优的甚块预测器;(2)将基于类型预测的甚块预测器应用到实际的EDGE处理器设计中,提升处理器的总体性能;(3)将本文的研究成果扩展到其它甚块体系结构中,为设计下一代处理器提供新的思路.

参 考 文 献

[1] Shalf J, Asanovic K, Patterson D et al. The manycore revolution: Will HPC community lead or folow? SciDAC Review, 2009, 1(14): 40-49

[2] Asanovic K, Bodik R, Demmel J et al. A view of the parallel computing landscape. Communications of the ACM, 2009, 52(10): 56-67

[3] Hill M D, Marty M R. Amdahl's law in the multicore era. IEEE Computer, 2008, 41(7): 33-38

[4] Eyerman S, Eeckhout L. Modeling critical sections in Amdahl's law and its implications for multicore design//Proceedings of the 37th Annual International Symposium on Computer Architecture. Saint-Malo, France, 2010: 362-370

[5] Esmaeilzadeh H, Blem E, Amant R St et al. Dark silicon and the end of multicore scaling//Proceedings of the 38th Annual International Symposium on Computer Architecture. San Jose, CA, USA, 2011: 365-376

[6] Burger D, Keckler S W, McKinley K S et al. Scaling to the end of silicon with edge architectures. IEEE Computer, 2004, 37(7): 44-55

[7] Nagarajan R, Kushwaha S K, Burger D et al. Static placement, dynamic issue (SPDI) scheduling for edge architectures//Proceedings of the 13th International Conference on Parallel Architectures and Compilation Techniques (PACT'04). Antibes Juan-les-Pins, France, 2004: 74-84

[8] Sankaralingam K, Nagarajan R, Liu H et al. Exploiting ILP, TLP, and DLP with the polymorphous trips architecture//Proceedings of the 30th Annual International Symposium on Computer Architecture (ISCA'03). San Diego, CA, USA, 2003: 422-433

[9] Sankaralingam K, Nagarajan R, McDonald R G et al. Distributed microarchitectural protocols in the trips prototype processor//Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture. Orlando, Florida, USA, 2006: 480-491

[10] McDonald R, Burger D, Keckler S W et al. TRIPS processor reference manual. Department of Computer Science, University of Texas at Austin: Technical Report TR-05-19, 2005

[11] Gebhart M, Maher B A, Coons K E et al. An evaluation of the trips computer system//Proceedings of the 14th Interna-

tional Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'09). Washington, DC, USA, 2009: 1-12

[12] Mahlke S, Lin D, Chen W et al. Effective compiler support for predicated execution using the hyperblock//Proceedings of the 25th Annual International Symposium on Microarchitecture (MICRO 25). Portland, Oregon, 1992: 45-54

[13] Ranganathan N, Burger D, Keckler S. Analysis of the trips prototype block predictor//Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2009). Boston, MA, USA, 2009: 195-206

[14] Jacobson Q, Bennett S, Sharma N et al. Control flow speculation in multiscalar processors//Proceedings of the 3rd International Symposium on High-Performance Computer Architecture. San Antonio, Texas, USA, 1997: 218-229

[15] Ranganathan N, Nagarajan R, Jiménez D et al. Combining hyperblocks and exit prediction to increase front-end bandwidth and performance. Department of Computer Science, University of Texas at Austin: Technical Report TR-02-41, 2002

[16] Ranganathan N. Control flow speculation for distributed architectures[Ph. D. dissertation]. The University of Texas at Austin, United States, Texas, 2009

[17] Hao E, Chang P Y, Evers M et al. Increasing the instruction fetch rate via block-structured instruction set architectures//Proceedings of the 29th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-29). Paris, France, 1996: 191-200

[18] Jacobson Q, Rotenberg E, Smith J. Path-based next trace prediction//Proceedings of the Thirtieth Annual IEEE/ACM International Symposium on Microarchitecture. Research Triangle Park, North Carolina, USA, 1997: 14-23

[19] Patel S, Lumetta S. Replay: A hardware framework for dynamic optimization. IEEE Transactions on Computers, 2001, 50(6): 590-608

[20] Zmily A, Kozyrakis C. Block-aware instruction set architecture. ACM Transactions on Architecture and Code Optimization, 2006, 3(3): 327-357

[21] Sohi G, Breach S, Vijaykumar T. Multiscalar processors//Proceedings of the 22nd Annual International Symposium on Computer Architecture. Santa Margherita Ligure, Italy, 1995: 414-425

[22] Esmaeilzadeh H, Burger D. Hierarchical control prediction: Support for aggressive predication//Proceedings of the 2nd Workshop on Parallel Execution of Sequential Programs on Multi-Core Architectures (In Conjunction with ISCA 2009). Austin, Texas, USA, 2009: 71-80

[23] Seznec A, Michaud P. A case for (partially)-tagged geometric history length predictors. Journal of Instruction Level Parallelism, 2006, 8(1): 1-23

[24] Chang P Y, Hao E, Patt Y. Target prediction for indirect jumps//Proceedings of the 24th Annual International Symposium on Computer Architecture. Denver, Colorado, USA, 1997: 274-283

[25] Driesen K, Holzle U. The cascaded predictor: Economical and adaptive branch target prediction//Proceedings of the 31st Annual ACM/IEEE International Symposium on Microarchitecture (MICRO-31). Dallas, Texas, USA, 1998: 249-258

[26] McFarling S. Combining branch predictors. Digital Western Research Laboratory, Palo Alto, CA, USA: Technical Report TN-36, 1993

[27] Guo Pengfei, Li Qingbo, Jin Yinghan et al. M5 based edge architecture modeling//Proceedings of the 2010 IEEE International Conference on Computer Design (ICCD). Amsterdam, Netherlands, 2010: 289-296

[28] Binkert N L, Dreslinski R G, Hsu L R et al. The M5 simulator: Modeling networked systems. IEEE Micro, 2006, 26(4): 52-60

[29] Yoder B, Burrill J, Mcdonald R et al. Software infrastructure and tools for the trips prototype//Proceedings of the 3rd Annual Workshop on Modeling, Benchmarking and Simulation. San Diego, CA, USA, 2007: 1-10

[30] Van Biesbrouck M, Calder B, Eeckhout L. Efficient sampling startup for simpoint. IEEE Micro, 2006, 26(4): 32-42

[31] Evers M, Patel S, Patt Y. An analysis of correlation and predictability: What makes two- level branch predictors work//Proceedings of the 25th Annual International Symposium on Computer Architecture. Barcelona, Spain, 1998: 52-61

[32] Loh G. A simple divide-and-conquer approach for neural-class branch prediction//Proceedings of the 14th International Conference on Parallel Architectures and Compilation Techniques (PACT 2005). Saint Louis, MO, USA, 2005: 243-254

GOU Peng-Fei, born in 1983, Ph. D. candidate. His research interests include high performance computer architecture research.

YU Ming-Yan, born in 1967, professor. His research interests include VLSI design, high performance computer architecture, analogy circuit design, etc.

YANG Bing, born in 1976, post-doctor. His research interests include high performance computer architecture and high performance compiler techniques.

LI Qing-Bo, born in 1985, M. S. candidate. His research interest is in critical path analysis of high performance computers.

WANG Shi-Bo, born in 1989, undergraduate. Her research interest is in next-block predictor design.

Background

EDGE (Explicit Data Graph Execution) architectures, with features of block-atomic fetch/execute/commit and explicit instruction communication, are one of the most promising alternatives for future processors. Contemporary EDGE processors, such as TRIPS and TFlex, usually maintain block-atomicity at the granularity of hyperblocks. Hyperblocks, formed at compile time, are single-entry, multiple-exit blocks with possibly predicated instructions. To guarantee efficient execution models, EDGE processors adopt control-flow speculation techniques to speculatively fetch multiple hyperblocks onto execution substrates. Since the unit of fetch/execute/commit is hyperblock, control-flow misspeculation resolving and recovery should be done at hyperblock boundaries. Therefore, EDGE architectures rise challenges to conventional instruction-centric control-flow speculation techniques, especially to the branch prediction, which is at the heart of control-flow speculation.

To face these challenges, the next-block predictor was introduced by Ranganathan et al to the research community, enabling high performance control-flow speculation for EDGE architectures. Basic idea of the next-block predictor is to employ the compiler-approximate intra-block paths such as exit IDs to predict branch targets for each hyperblock. At prediction time, the next-block predictor predicts exit ID of a hyperblock by an exit predictor. To incorporate conventional branch predictor techniques into exit predictors, researchers have to make modifications, due to inherent differences between the hyperblock exit prediction and the conventional branch prediction. After producing the predicted exit ID, which is inherently the speculative intra-block path, the next-block predictor predicts the branch type of this exit, then accesses different target buffers in accordance with the predicted branch type. Target buffers tuned for different branch types finally produce the predicted address of the next hyperblock. Published achievements reveal that though a bunch of aggressive techniques has been evaluated, more than 50% mispredicts are still induced by exit ID mispredicts, indicating exit predictors are the performance bottleneck, leaving spaces for future optimizations.

In this paper, our studies show that without exit IDs (intra-block path informations), inter-block informations such as branch type histories are effective enough in predicting hyperblock branch types. As a result, based on these findings, we propose a type-only next-block predictor, which eliminates the exit predictor and directly predicts branch types using non-speculative inter-block informations. By employing TAGE technique in branch type predictors, our proposal outperforms published exit-based next-block predictors by 10% in MPKI (Mispredicts Per Kilo Instructions) across a wide resource budget range for SPEC CPU2000 integer benchmarks.

直接匿名证言协议的性能估算新方法

谭 良[1),2)] 孟伟明[1)] 周明天[3)]

[1)](四川师范大学计算机学院,四川省可视化计算与虚拟现实重点实验室 成都 610066)

[2)](中国科学院计算技术研究所 北京 100190)

[3)](电子科技大学计算机学院 成都 610054)

摘 要 性能问题是阻碍DAA推广和应用的首要问题.为了进一步优化该协议的性能,找出性能瓶颈,定量地分析和测量DAA中各个实体的性能负荷分布是一个十分重要且必须的工作.文中详细分析了DAA的协议流程,提出了以机器周期为基本性能单位的性能负荷分布测量方法——归一化统计法(Normalized Statistics,NS).该方法需要首先分析DAA协议中的各种复杂运算,针对不同的运算选用当前性能较好的算法,然后统计各个算法中大整数单精度乘法、单精度加法、读内存、写内存等基本运算的数目,最后通过汇总并转换得出DAA协议中各实体以机器周期为单位的性能负荷分布和总性能负荷.比较分析表明,该方法不仅能相对准确、精细、有效地定量计算出DAA协议中各实体的性能负荷和总的性能负荷,而且测出的性能负荷具有平台无关性.最后为了说明该方法的有效性,将NS方法应用于有关可信计算匿名证明的一个典型方案的性能负荷估算.

关键词 可信计算;直接匿名证言;Camenisch-Lysyanskaya签名;知识证明;性能负荷

中图法分类号 TP311 **DOI号**: 10.3724/SP.J.1016.2012.01553

A New Method of Performance Estimate of Direct Anonymous Attestation Scheme in TCG

TAN Liang[1),2)] MENG Wei-Ming[1)] ZHOU Ming-Tian[3)]

[1)](*College of Computer, Sichuan Normal University, Key Laboratory of Visualization in Scientific Computing and Virtual Reality of Sichuan, Chengdu* 610066)

[2)](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[3)](*School of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu* 610054)

Abstract Performance is a most important problem to Direct Anonymous Attestation Scheme in TCG. It is very necessary and important to analyze and measure performance to every entity quantitatively for optimizing DAA. In this paper, DAA protocol is first analysed detailedly, and then a new performance measurement method, called Normalized Statistics method, which takes the machine period as the basic performance unit, is put forward. When using this method, all complex calculates in DAA protocol must be found out and statistic, and better algorithms to every complex calculate are chosen, moreover, to each algorithms, we need to compute the sum for each basic operation, such as multiplication of single big integer, addition of single big integer, reading and writing memory, and so on. Finally, the every entity performance and the whole performance burden in DAA, whose unit is the machine period, are summed. The theoretical analysis results show that the performance estimate is exact, meticulous and effective by this method which is independent of actual platform. For proving availability of the method, we apply it to estimate performance of other one DAA scheme.

Keywords trusted computing; direct anonymous attestation; Camenisch-Lysyanskaya sign; knowledge proof; performance burden

收稿日期:2011-01-24;最终修改稿收到日期:2011-05-03.本课题得到国家自然科学基金(60970113)、国家自然科学基金青年基金(60903073)资助.**谭 良**,男,1972年生,博士,教授,主要研究领域为可信计算、网络安全.E-mail:tanliang2008cn@126.com.**孟伟明**,男,1985年生,硕士研究生,研究方向为可信计算.**周明天**,男,1937年生,教授,主要研究领域为网络计算、信息安全.

1 引 言

可信计算组织 TCG 在 TPM 规范 1.2 版中采用了直接匿名证言(Direct Anonymous Attestation, DAA)协议[1],该协议主要基础包括 Camenisch-Lysyanskaya 签名方案[2]、基于离散对数的知识证明和 Fiat-Shamir 启发式方法[3],既解决了隐私 CA 的瓶颈问题,又实现对 TPM 芯片的认证和匿名,是当前可信计算平台身份证明最好的理论解决方案之一. 但是该协议非常复杂,实现过程中不仅涉及到多个实体,而且涉及大量的耗时运算. 性能问题突出制约了该协议的广泛应用.

为了进一步优化该协议的性能,找出性能瓶颈,定量地分析和测量 DAA 中各个实体的性能负荷是一个十分重要且必须的工作. 对于这一问题,最好的方法是开发出实际的 DAA 系统,在真实的环境中实际测量. 但实际上,目前产业界生产的可信计算产品如 TPM 芯片、可信计算机(台式机和笔记本)等采用 DAA 协议进行平台身份认证的产品还比较少见,再加上完整的 DAA 协议流程包括 TPM、平台 Host、发布方 Issuer、验证方 Verifier 和可信第三方 TTP 等共 5 个实体,在真实的运行环境测试性能负荷分布不仅需要 TPM 支持,而且还需要开发和建设其它 4 个实体. 显然,这是一个复杂的系统工程,需要较大的成本和时间开销. 因此,通常采用以下两种方法对 DAA 的性能进行定量分析和测试.

(1)仿真环境实测法. 文献[1]是在 IBM ThinkPad T41(1.7GHz Intel Pentium M CPU,1GB RAM,Linux)上运行 IBM 开发的 DAA 原型系统,测量的结果是,TPM 向发布者申请 DAA 证书需要 2.4 s,其中 TPM 约占用 25%,主机约占用 25%,发布者约占用 50%的时间;TPM 每次向验证者认证自己需要 4.4 s,其中 TPM 约占用 8%,主机约占用 47%,验证者约占用 45%的时间. 文献[4]是在 Intel dual-core 3.2GHz、1GB RAM、Windows 平台上开发 DAA 原型系统,设计了 tpm-module, host-module, server-module 3 个模块. 测量的结果是,在 Join 阶段,TPM 需要 15.2 s, Host 需要 12 s;在 Sign 阶段,TPM 需要 13.8 s,Host 需要 20.6 s;在 Verify 阶段,TPM 需要 0 s, Host 需要 30.4 s. 值得注意的是,此类方法通常在单台机器上对 DAA 的各协议实体进行性能测试,不计通信开销. 这和在实际的运行环境进行测试是有差别的,毕竟原型系统不是实际运行环境. 而且协议中涉及到的多个实体如果均在同一个平台上运行,会相互影响彼此的运行效率,因此该方法测试的性能负荷分布并不准确.

(2)运算符号化统计法. 该方法是将 DAA 协议中的各主要运算符号化,然后统计符号次数并求和. 例如只需将 DAA 中涉及的主要运算的表示符号约定:G_n:一指数模 n 运算,例如 $g^a \bmod n$(G_n 是 G_n^1 的简写);G_n^2:两指数连乘模 n 运算,例如 $g^a h^b \bmod n$;G_n^3:三指数连乘模 n 运算,例如 $g^a h^b y^c \bmod n$;G_n^i:依上述 G_n,G_n^2,G_n^3 的定义类推;G_Γ:一指数模 Γ 运算,例如 $g^a \bmod \Gamma$;G_Γ^2:两指数连乘模 Γ 运算,例如 $g^a h^b \bmod \Gamma$;G_Γ^3:三指数连乘模 Γ 运算,例如 $g^a h^b y^c \bmod \Gamma$;G_Γ^i:依上述 G_Γ,G_Γ^2,G_Γ^3 的定义类推;P_c:生成一个大素数的运算;P_v:验证一个数为素数的运算;H:Hash 运算;则 DAA 协议的性能分布如表 1.

表 1 基于运算符号化统计法的 DAA 性能负荷分布

阶段	参与方	主要计算开销
申请加入(Join)	TPM	$3 \cdot G_\Gamma + 2 \cdot G_n^3 + (i+3) \cdot H$
	Issuer	$j \cdot G_\Gamma + 2 \cdot G_n + 1 \cdot G_n^4 + 1 \cdot G_\Gamma^2 + 1 \cdot P_c + 2 \cdot H$
	Host	$1 \cdot G_\Gamma + 1 \cdot G_n^2 + 1 \cdot P_v + 3 \cdot H$
进行签名(Sign)	TPM	$3 \cdot G_\Gamma + 1 \cdot G_n^3 + 1 \cdot H$
	Host	$1 \cdot G_\Gamma + 1 \cdot G_n + 1 \cdot G_n^2 + 2 \cdot G_n^3 + 1 \cdot G_n^4 + 2 \cdot H$
签名验证(Verify)	Verifier	$4 \cdot G_\Gamma^2 + 2 \cdot G_n^4 + 1 \cdot G_n^6 + j \cdot G_\Gamma + 4 \cdot H$

实际上,有关 DAA 扩展和改进的大量文献(如文献[5-11])均采用该方法进行性能负荷分析. 因为该方法简单,且在同类运算性能的比较上非常有效,如 $G_n < G_n^2 < G_n^3$,$G_\Gamma < G_\Gamma^2 < G_\Gamma^3$ 等. 但该方法在不同类运算之间无法进行比较. 因此,当各协议实体的不同类型运算较多时,该方法不能定量估算性能负荷及其比例关系,不便于各阶段、各实体性能负荷的比较分析.

鉴于以上原因,提出了以机器周期为基本性能单位的性能负荷分布测量方法——归一化统计法(Normalized Statistics, NS). 该方法需要首先分析 DAA 协议中的各种复杂运算,针对不同的运算选用当前性能较好的算法,然后统计各个算法中大整数单精度乘法、单精度加法、读内存、写内存等基本运算的数目,最后通过汇总并转换得出 DAA 协议中各实体以机器周期为单位的性能负荷分布和总性能负荷. 理论分析表明,该方法不仅能相对准确、精细、有效地定量计算出 DAA 协议中各实体的性能负荷和总的性能负荷,而且测出的性能负荷具有平台无关性. 最后为了说明该方法的有效性,将 NS 方法应

用于有关可信计算匿名证明的一个典型方案的性能负荷估算.

2　DAA 协议流程分析

本节将依据文献[1]对 DAA 协议进行流程分析. 文献[1]重点描述 DAA 实现认证和匿名的复杂运算和步骤,但协议流程描述并不详细和完整. 如发布 DAA 证书时发布方对谁提供零知识证明协议证明 DAA 证书构造正确? 谁对 DAA 证书进行了签名? 等等. 本节将对此进行补充,使得 DAA 更明确和完整.

2.1　DAA 协议的常数和假设

DAA 协议涉及到的安全参数和长度要求包括,l_n为 RSA 模数长度,长度为 2048 位;l_f为 TPM 秘密 ID 的长度,长度为 104 位;l_e为指数 e 的长度,长度为 368 位;$l_{e'}$为选择 e 的区间长度,长度为 128 位;l_v为随机数 v 的长度,为 2536 位;l_ϕ为零知识协议安全常数,长度为 80 位;l_H为 Hash 函数输出长度,即 SHA-1,长为 160 位;l_Γ模数 Γ 的长度,长度为 1632 位;l_ρ为 ρ 的长度,与 Γ 一起应用于假名,长度为 208 位.

假设 1.　(Strong RSA Assumption)不存在可行的算法,对任意随机的 RSA 模数 n 和一个随机元素 $u\in Z_n^*$,计算出 $e>1$ 和 v,使得 $v^e\equiv u \bmod n$.

假设 2.　(Decisional Diffie-Hellman Assumption)假设 Γ 为 l_Γ比特的素数,ρ 为 l_ρ比特的素数且 $\rho|\Gamma-1$. 设 $\gamma\in Z_\Gamma^*$ 为阶等于 ρ 的元素,对足够大的 l_Γ 和 l_ρ,不存在可行的算法可以区分四元组$\{(\delta,\delta^a,\delta^b,\delta^{ab})\}$和$\{(\delta,\delta^a,\delta^b,\delta^c)\}$,其中 δ 为$\langle\gamma\rangle$中的一个随机元素,a、b 和 c 为区间$[0,\rho-1]$内的随机元素.

2.2　DAA 协议初始化

DAA 协议初始化是在发布方和可信第三方之间进行的. 发布方在生成 DAA 公钥的同时向可信第三方提供一个非交互式的零知识证明,证明其公钥的合法性. 这对于其后参与者在发布方作弊时也能保持匿名起到了关键的作用.

1. 发布方选择一个安全素数乘积构成的 RSA 模数 $n=pq$,其中 $p=2p'+1$,$q=2q'+1$,p,q,p',q',都是素数,n 长度为 l_n;

2. 选择 QR_n的一个随机生成元 g';

3. 选择随机整数 x_0、x_1、x_Z、x_S、x_h、$x_g\in[1,p'q']$,并计算

$$g=g'^{x_g} \bmod n,\quad h=g'^{x_h} \bmod n,\quad S=h^{x_s} \bmod n$$

$$Z=h^{x_Z} \bmod n,\quad R_0=S^{x_0} \bmod n,\quad R_1=S^{x_1} \bmod n;$$

4. 提供一个非交互式的零知识证明发布方公钥构造正确. 即发布方对 g 和 h 相对于 g'的离散对数作零知识证明,对 S 和 Z 相对于 h 的离散对数作零知识证明,对 R_0 和 R_1 对 S 的离散对数作零知识证明;

(1) 发布方选择随机数 r_{x_g}, r_{x_h}, r_{x_S}, r_{x_Z}, r_{x_0}, $r_{x_1}\in[1,p'q']$,计算

$$t_{x_g}=g'^{r_{x_g}} \bmod n,\ t_{x_h}=g'^{r_{x_h}} \bmod n,\ t_{x_S}=h^{r_{x_s}} \bmod n,$$

$$t_{x_Z}=h^{r_{x_Z}} \bmod n,\quad t_{x_0}=S^{r_{x_0}} \bmod n,\quad t_{x_1}=S^{r_{x_1}} \bmod n;$$

(2) 可信第三方选择一个随机比特串 $n_{TI}\in\{0,1\}^{l_H}$ 作为防重放攻击的 nonce,发送给发布方;

(3) 发布方也选择一个随机比特串 $n_{IT}\in\{0,1\}^{l_H}$ 作为防重放攻击的 nonce,并计算

$$c_{x_g}=H(t_{x_g}\|n_{TI}\|n_{IT}),\ c_{x_h}=H(t_{x_h}\|n_{TI}\|n_{IT}),$$

$$c_{x_S}=H(t_{x_s}\|n_{TI}\|n_{IT}),\ c_{x_Z}=H(t_{x_Z}\|n_{TI}\|n_{IT}),$$

$$c_{x_0}=H(t_{x_0}\|n_{TI}\|n_{IT}),\ c_{x_1}=H(t_{x_1}\|n_{TI}\|n_{IT});$$

(4) 发布方计算

$$s_{x_g}=r_{x_g}+c_{x_g}x_g,\ s_{x_h}=r_{x_h}+c_{x_h}x_h,\ s_{x_S}=r_{x_S}+c_{x_S}x_S,$$

$$s_{x_Z}=r_{x_Z}+c_{x_Z}x_Z,\ s_{x_0}=r_{x_0}+c_{x_0}x_0,\ s_{x_1}=r_{x_1}+c_{x_1}x_1,$$

将$(c_{x_g},s_{x_g},c_{x_h},s_{x_h},c_{x_S},s_{x_S},c_{x_Z},s_{x_Z},c_{x_0},s_{x_0},c_{x_1},s_{x_1},n_{IT})$传给可信第三方;

(5) 可信第三方验证 $s_{x_g}<(p'q'\times2^{h+1})$,$s_{x_h}<(p'q'\times2^{h+1})$,$s_{x_S}<(p'q'\times2^{h+1})$,$s_{x_Z}<(p'q'\times2^{h+1})$,$s_{x_0}<(p'q'\times2^{h+1})$,$s_{x_1}<(p'q'\times2^{h+1})$,并计算

$$H(g'^{s_{x_g}}t_{x_g}^{-c_{x_g}} \bmod n)=c_{x_g},\ H(g'^{s_{x_h}}t_{x_h}^{-c_{x_h}} \bmod n)=c_{x_h},$$

$$H(h^{s_{x_s}}t_{x_s}^{-c_{x_s}} \bmod n)=c_{x_S},\quad H(h^{s_{x_z}}t_{x_z}^{-c_{x_z}} \bmod n)=c_{x_z},$$

$$H(S^{s_{x_0}}t_{x_0}^{-c_{x_0}} \bmod n)=c_{x_0},\quad H(S^{s_{x_1}}t_{x_1}^{-c_{x_1}} \bmod n)=c_{x_1};$$

5. 选择素数 Γ 为 l_Γ比特,ρ 为 l_ρ比特且 $\Gamma=r\rho+1$. 选择一个随机数 $\gamma'\in_R Z_\Gamma^*$ 使得 $\gamma'^{(\Gamma-1)/\rho}\neq1 \bmod\Gamma$,记 $\gamma=\gamma'^{(\Gamma-1)/\rho} \bmod\Gamma$;

6. DAA 发布者将$(n,g',g,h,S,Z,R_0,R,\gamma,\Gamma,\rho)$发送给可信第三方,可信第三方用其私钥对此签名,随后发布方公布发布者公钥.

2.3　DAA-Join 协议

设发布者公钥 $PK_1=(n,g',g,h,S,Z,R_0,R_1,\gamma,\Gamma,\rho)$,发布方对 PK_1的签名使用的公钥是 PK_1',假设发布方基名是 bsn_1. *DAAseed* 为 TPM 产生 (f_0,f_1)的恒定常量.

1. 平台计算 $\zeta_1=(H(1\|bsn_1))^{(\Gamma-1)/\rho} \bmod\Gamma$,将结果返给 TPM;

2. TPM 检查 $\zeta_1^\rho=1 \bmod\Gamma$,计算其秘密 ID,

$$f=H(H(DAAseed\|H(PK_1'))\|cnt\|1) \bmod\rho,$$

记 $f_0=LSB_{l_f}(f)$,$f_1=CAR_{l_f}(f)$,选择一个随机数 $v'\in_R\{0,1\}^{l_\phi+l_n}$,计算 $U=R_0^{f_0}R_1^{f_1}S^{v'} \bmod n$,$N_1=\zeta_1^{f_0+f_1^2\,l_f} \bmod\Gamma$,将 U 和 N_1发送给平台,平台转发给发布方;

3. 发布方检查撤销列表中的所有 f_0 和 f_1,验证 $N_1\neq$

$\zeta_1^{f_0+f_1^{2^{l_f}}} \bmod \Gamma$. 发布方同时还检查该平台以前使用的 N_1，如果平台位于撤销列表中，发布方终止 Join 协议；

4. TPM 通过零知识协议向发布方证明其拥有 f_0、f_1 和 v' 以及 U 和 N_1 的正确构成；

(1) TPM 选择随机数 $r_{f_0}, r_{f_1} \in_R \{0,1\}^{l_f+l_\phi+l_H}$ 和 $r_{v'} \in_R \{0,1\}^{l_f+2l_\phi+l_H}$，计算 $\widetilde{U}=R_0^{r_{f_0}}R_1^{r_{f_1}}S^{r_{v'}} \bmod n$，$\widetilde{N}_1=\zeta_1^{r_{f_0}+r_{f_1}^{2^{l_f}}} \bmod \Gamma$，将结果发送给平台；

(2) 发布方选择一个随机比特串 $n_i \in \{0,1\}^{l_H}$ 作为防重放攻击的 nonce，发送给平台；

(3) 平台计算 $c_h=H(n \| R_0 \| R_1 \| S \| U \| N_1 \| \widetilde{U} \| \widetilde{N}_1 \| n_i)$，返回结果给 TPM；

(4) TPM 选择 TPM 端的 nonce，$n_t \in \{0,1\}^{l_\phi}$，计算 $c=H(c_h \| n_t)$；

(5) TPM 计算 $s_{f_0}=r_{f_0}+cf_0$，$s_{f_1}=r_{f_1}+cf_1$，$s_{v'}=r_{v'}+cv'$；

(6) TPM 发送 $c, n_t, s_{f_0}, s_{f_1}, s_{v'}$ 给平台，平台转发给发布方；

(7) 发布方计算

$$\widetilde{U}=U^{-c}R_0^{s_{f_0}}R_1^{s_{f_1}}S^{s_{v'}} \bmod n, \widetilde{N}_1=N_1^{-c}\zeta_1^{s_{f_0}+s_{f_1}^{2^{l_f}}} \bmod \Gamma,$$

然后验证

$$c=H(H(n \| R_0 \| R_1 \| S \| U \| N_1 \| \widetilde{U} \| \widetilde{N}_1 \| n_i) \| n_t),$$

$s_{f_0}, s_{f_1} \in \{0,1\}^{l_f+l_\phi+l_H+1}$，$s_{v'} \in \{0,1\}^{l_n+2l_\phi+l_H+1}$；

5. 发布方选择随机数 $\hat{v} \in_R \{0,1\}^{l_v-1}$ 和素数 $e \in_R [2^{l_e-1}, 2^{l_e-1}+2^{l_{e'}-1}]$，计算 $v''=\hat{v}+2^{l_v-1}$，$A=\left(\frac{Z}{US^{v''}}\right)^{1/e} \bmod n$；

6. 发布方向平台证明其 A 计算正确(防止发布方得到 A 后，选择一个 $b \notin \langle h \rangle$，且 $b^e=1 \bmod n$，返回 Ab 给平台，可用于后继跟踪平台)；

(1) 平台选择一个随机的 nonce，$n_h \in \{0,1\}^{l_H}$，将结果发送给发布方；

(2) 发布方随机生成 $r_e \in_R [0, p'q']$，计算 $\tilde{A}=(Z/US^{v''})^{r_e} \bmod n$；

(3) $c'=H(n \| Z \| S \| U \| v'' \| A \| \tilde{A} \| n_h)$ 和 $s_e=r_e-c'/e \bmod p'q'$，并将结果 c', s_e, A, e, v'' 发送给平台；

(4) 平台验证 e 是素数且位于区间 $[2^{l_e-1}, 2^{l_e-1}+2^{l_{e'}-1}]$，计算 $\hat{A}=A^{c'}(Z/US^{v''})^{s_e} \bmod n$，验证 $c'=H(n \| Z \| S \| U \| v'' \| A \| \hat{A} \| n_h)$；

7. 平台发送 v'' 给 TPM；

8. TPM 计算 $v=v'+v''$，保留 v、f_0 和 f_1.

2.4 DAA-Sign 协议

1. 根据验证方是否提供 bsn_v，平台计算 $\zeta \in_R \langle \gamma \rangle$ 或 $\zeta=(H_\Gamma(1 \| bsn_v))^{(\Gamma-1)/\rho} \bmod \Gamma$；

2. 平台随机选择整数 $w, r \in \{0,1\}^{l_n+l_\phi}$，计算 $T_1=Ah^w \bmod n$ 和 $T_2=g^w h^e (g')^r \bmod n$. TPM 计算 $N_v=\zeta^{f_0+f_1^{2^{l_f}}} \bmod \Gamma$ 并将结果发送给平台；

3. TPM 和平台合作生成一个零知识证明，证明 T_1、T_2 来自于 DDA 证书，且 N_v 计算中应用的 f 为该 DAA 证书的秘密；

(1) TPM 随机选择 $r_v \in_R \{0,1\}^{l_v+l_\phi+l_H}$，计算

$\widetilde{T}_{1t}=R_0^{r_{f_0}}R_1^{r_{f_1}}S^{r_v} \bmod n$，$\tilde{r}_f=r_{f_0}+r_{f_1}^{2^{l_f}} \bmod \rho$，$\widetilde{N}_v=\zeta^{\tilde{r}_f} \bmod \Gamma$，TPM 将 $\widetilde{T}_{1t}$、$\widetilde{N}_v$ 发送给平台；

(2) 平台随机生成

$r_e \in_R \{0,1\}^{l_{e'}+l_\phi+l_H}$，$r_{ee} \in_R \{0,1\}^{2l_e+l_\phi+l_H+1}$，

$r_w, r_r \in_R \{0,1\}^{l_n+2l_\phi+l_H}$，$r_{ew}, r_{er} \in_R \{0,1\}^{l_e+l_n+2l_\phi+l_H+1}$，

计算

$$\widetilde{T}_1=\widetilde{T}_{1t}T_1^{r_e}h^{-r_{ew}} \bmod n, \widetilde{T}_2=g^{r_w}h^{r_e}g'^{r_r} \bmod n,$$

$$\widetilde{T}_2'=T_2^{-r_e}g^{r_{ew}}h^{r_{ee}}g'^{r_{er}} \bmod n;$$

(3) 平台计算

$$c_h=H(n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho \| \zeta \| T_1 \| T_2 \| N_v \| \widetilde{T}_1 \| \widetilde{T}_2 \| \widetilde{T}_2' \| \widetilde{N}_v \| n_v),$$

并将结果发送给 TPM，TPM 选择一个随机的 nonce，$n_t \in \{0,1\}^{l_\phi}$，计算 $c=H(H(c_h \| n_t) \| b \| m)$，并将 n_t，c 返回给平台；

(4) TPM 计算

$$s_v=r_v+cv, s_{f_0}=r_{f_0}+cf_0, s_{f_1}=r_{f_1}+cf_1,$$

并将 3 个结果发送给平台；

(5) 平台计算

$s_e=r_e+c(e-2^{l_e-1})$，$s_{ee}=r_{ee}+ce^2$，$s_w=r_w+cw$，

$s_{ew}=r_{cw}+cwe$，$s_r=r_r+cr$，$s_{er}=r_{er}+cer$；

(6) 平台最终输出对 m 的签名：

$$\sigma=(\zeta,(T_1,T_2),N_v,c,n,(s_v,s_{f_0},s_{f_1},s_e,s_{ee},s_w,s_{ew},s_r,s_{er})).$$

2.5 DAA-Verification 协议

1. 计算

$$\hat{T}_1=Z^{-c}T_1^{s_c+c\times 2^{l_e-1}}R_0^{s_{f_0}}R_1^{s_{f_1}}S^{s_v}h^{-s_{ew}} \bmod n,$$

$$\hat{T}_2=T_2 g^{s_w}h^{s_c+c\times 2^{l_e-1}}g'^{s_r} \bmod n,$$

$$\hat{T}_2'=T_2^{-(s_e+c2^{l_e-1})}g^{s_{ew}}h^{s_{ee}}g'^{s_{er}} \bmod n,$$

$$\hat{N}_v=N_v^{-c}\zeta^{s_{f_0}+s_{f_1}^{2^{l_f}}} \bmod \Gamma;$$

2. 验证

$$c=H(H(H((n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho \| \zeta \| T_1 \| T_2 \| N_v \| \widetilde{T}_1 \| \widetilde{T}_2 \| \widetilde{T}_2' \| \widetilde{N}_v \| n_v) n_t) \| b) \| m)$$

以及 $N_v, \zeta \in \langle \gamma \rangle$，$s_{f_0}, s_{f_1} \in \{0,1\}^{l_f+l_\phi+l_H+1}$ 和 $s_e \in \{0,1\}^{l_{e'}+l_\phi+l_H+1}$；

3. 如果验证方提供了基名，验证

$$\zeta \equiv (H(1 \| bsn_v))^{(\Gamma-1)/\rho} \bmod \Gamma;$$

4. 检查撤销列表上所有的 f_0、f_1，验证

$$N_v \neq \zeta^{f_0+f_1^{2^{l_f}}} \bmod \Gamma.$$

3　DAA 协议的性能估算新方法——NS 方法

NS 方法的步骤如下：(1) 首先进行复杂运算的统计；(2) 确定主要运算及其算法选择；(3) 基本运算统计；(4) 以机器周期数为基本单位估算性能负荷分布.

3.1　DAA 协议的复杂运算统计

NS 方法的第 1 步是进行复杂运算的统计. DAA 包括初始化阶段、Join 协议阶段、Sign 阶段和验证阶段，我们在进行复杂运算统计时，实体仅考虑发布方 Issuer、Host 平台、TPM 以及验证方 Verifier. DAA 协议是以大数运算为主，主要的复杂运算包括大素数选择、大随机数产生、模指数运算、SHA-1 运算、大整数乘法和大整数加法等. 下面将各种复杂运算按照实体进行统计，可以很容易地看出各个协议实体在整个协议过程中的复杂运算消耗. 由于篇幅关系，统计过程略. 统计结果见表 2. 其中 l_c 是 TPM 撤销列表的长度.

表 2　DAA 各实体的主要运算统计总表

运算 实体	大素数 选择	大随机数 产生	模指数 运算	SHA-1 运算	大整数 乘法运算	大整数 加法运算
Host	0 次	9 次	20 次	4 次	9 次	6 次
TPM	0 次	7 次	14 次	4 次	6 次	8 次
Issuer	5 次	18 次	24 次	6 次	7 次	8 次
Verifier	0 次	0 次	$17+l_c$ 次	4 次	0 次	0 次

3.2　DAA 复杂运算的算法选择

NS 方法的第 2 步是确定主要运算及其算法选择. 从表 2 可以看出运算次数较多是大整数模指数运算、大随机数运算和大整数乘法运算. 其它运算如大素数选择、SHA-1 运算、大整数加法等运算次数较少，因此，在性能估算时，主要考虑大整数模指数运算、大随机数产生运算和大整数乘法运算.

另外，对主要运算的算法选择时，应遵循一个基本的原则，即选择的算法应该是该运算使用得最多、最广泛的算法，并已经应用于 GMP、NTL、Crypto++、LibTomCrypt (LibTomMath)、OpenSSL 以及 miracl 等库中.

根据以上原则，大随机数产生运算选择 Lehmer 提出的线性同余法算法. 如算法 1 所示.

算法 1.

$$\begin{cases} x_{i+1}=(ax_i+c)\bmod m \\ r_i=\dfrac{x_i}{m} \end{cases},$$

其中 a 是乘子，c 是增量，x_0 为种子，m 为模数，当 $a\approx x_i$，该算法主要运算是一次模乘运算. 著名的粒子输运 Monte Carlo 程序 MCNP、MORSE 和 KENO 的随机数发生器均基于该方法.

模指数运算选择 Montgomery 模指数算法[12]，该算法是目前最好的模指数算法之一，由滑动窗口指数算法结合 Montgomery 模乘法，如算法 2 所示.

算法 2.　Montgomery 模指数算法.

输入：整数 $m=(m_{l-1}\cdots m_1 m_0)_b$，$gcd(m,b)=1$，$R=b^l$，$m'=-m^{-1}\bmod b$，$e=(e_l e_{l-1}\cdots e_1 e_0)_2$，$e_l=1$，整数 x，$1\le x<m$

输出：$x^e \bmod m$

1. $\bar{x}\leftarrow Mont(x,R^2\bmod m)$，$A\leftarrow R\bmod m$；
2. 对于 i 从 t 递减到 0，执行：
   2.1. $A\leftarrow Mont(A,A)$；
   2.2. 若 $e_i=1$，则 $A\leftarrow Mont(A,x)$；
3. $A\leftarrow Mont(A,1)$；
4. 返回 A.

该算法主要运算如表 3. 其中 $t+1$ 表示指数 e 的二进制长度，l 表示以 b 为基的数的长度.

表 3　Montgomery 模指数算法的主要运算次数

步骤	Montgomery 模乘法次数	单精度乘法次数
1	1	$2l(l+1)$
2	$\frac{3}{2}t$	$3tl(l+1)$
3	1	$l(l+1)$

算法 1 和 2 的关键运算均是模乘运算，因此模乘运算算法的选择非常关键. 在众多的模乘算法中，选择 CIOS 算法[13]，该算法将多精度乘法和模约减算法完美地融合为一体，在读写内存等方面节省了许多资源，如算法 3 所示.

算法 3.　CIOS 模乘算法.

输入：整数 $m=(m_{n-1}\cdots m_1 m_0)_b$，$x=(x_{n-1}\cdots x_1 x_0)_b$，$y=(y_{n-1}\cdots y_1 y_0)_b$，$gcd(m,b)=1$，$R=b^n$，$m'=-m^{-1}\bmod b$

输出：$xyR^{-1}\bmod m$

1. $A\leftarrow 0$；
2. 对于 i 从 0 到 $n-1$，执行：
   2.1. $A\leftarrow A+x_i y$；
   2.2. $u_i\leftarrow a_0 m'\bmod b$；
   2.3. $A\leftarrow (A+u_i m)/b$；
3. 如果 $A\ge m$，则 $A\leftarrow A-m$；
4. 返回 A.

算法 3 包含单精度乘法、单精度加法、读内存以及写内存的次数如表 4. 其中 k 表示以 b 为基的大整数的位数.

表 4 COIS 模乘算法的基本运算及次数统计表

基本运算	单精度乘法	单精度加法	读内存	写内存
CIOS 算法	$2k^2+k$	$4k^2-k-1$	$4k^2+7k$	$2k^2+4k$

多精度乘法运算选择效率较高的 Comba 算法[14]. 如算法 4 所示.

算法 4. Comba 多精度乘算法.

输入:两个整数 x 和 y,长度分别为 n 和 t,基为 b

输出:乘积的 b 进制表示 $x\times y=(w_{n+t-1}\cdots w_1 w_0)_b$

1. $(v_2 v_1 v_0)_b=0$;

2. 对于 i 从 0 到 $n+t-2$,执行:

2.1. $(v_2 v_1 v_0)_b\leftarrow(v_2 v_1 v_0)_b+\sum_{j=0}^{i}x_j y_{i-j}$;

2.2. $w_i\leftarrow v_0$, $v_0\leftarrow v_1$, $v_1\leftarrow v_2$, $v_0\leftarrow 0$;

3. $w_{n+t-1}\leftarrow v_0$;

4. 返回$(w_{n+t-1}\cdots w_1 w_0)_b$.

算法 4 包含单精度乘法、单精度加法、读内存以及写内存的次数如表 5. 其中 k 表示以 b 为基的大整数的位数.

表 5 Comba 多精度乘算法的基本运算及次数统计表

操作	单精度乘法	单精度加法	读内存	写内存
Comba 算法	k^2	$2k^2-2$	$2k^2$	$2k$

由于算法 1 的实现主要包括算法 3,根据表 4,得算法 1 包含的基本运算统计表 6.

表 6 大随机数的基本运算及次数统计表

操作	单精度乘法	单精度加法	读内存	写内存
Moktgomery 模乘算法	$2k^2+k$	$4k^2-k-1$	$4k^2+7k$	$2k^2+4k$

由于算法 2 的实现主要包括算法 3 和 4,根据表 4、表 5 得算法 2 包含的基本运算统计表 7.

表 7 Moktgomery 模乘算法的基本运算及次数统计表

操作	单精度乘法	单精度加法	读内存	写内存
Moktgomery 模乘算法	$(2+1.5t)\times(2k^2+k)$	$(2+1.5t)\times(4k^2-k-1)$	$(2+1.5t)\times(4k^2+7k)$	$(2+1.5t)\times(2k^2+4k)$
单精度乘法次数	$3l(l+1)(t+1)$	0	0	0

3.3 DAA 协议各阶段的基本运算统计

NS 方法的第 3 步是基本运算统计. NS 方法的基本运算是单精度乘法、单精度加法、读内存和写内存. 之所以选择这些运算作为 NS 方法的基本运算,是因为这些基本运算的机器周期很容易确定. 由于当前的多数计算机为 32 位机型,因此,选择 $b=32$. 如果机型为 64 位,则 $b=64$,则统计的相关结果减半.

3.3.1 DAA Join 阶段的基本运算统计

在 Join 阶段,协议实体包含 Host 平台、TPM 和 Issuer. Join 阶段的基本运算统计表 8.

表 8 DAA 协议 Join 阶段的基本运算次数统计表

	单精度乘法	单精度加法	读内存	写内存
Host	114385681	226020918	233330857	117121378
TPM	383098821	376894638	389356930	195455631
Issuer	560984641	552135689	569901665	286058980

3.3.2 DAA Sign 阶段的基本运行统计

在 Sign 阶段,协议实体包含 Host 和 TPM. Sign 阶段的基本运算统计表 9.

表 9 DAA 协议 Sign 阶段的基本运算次数统计表

	单精度乘法	单精度加法	读内存	写内存
Host	735976684	727407463	750465815	376673079
TPM	203641660	201146242	207878436	104359354

3.3.3 DAA Verification 阶段的基本运算统计

在 Verification 阶段,协议实体包含 Verifier,取 $l_c=0$. Verification 阶段的基本运算统计表 10.

表 10 DAA 协议 Verification 阶段的基本运算次数统计表

	单精度乘法	单精度加法	读内存	写内存
Verifier	813090596	803349273	829421714	416337084

3.4 DAA 协议的性能负荷分布及总性能负荷的估算与分析

NS 方法的最后一步是以机器周期数为基本单位估算性能负荷分布.

定义 1. 令执行一次单精度乘法运算的机器周期数为 θ_0,执行一次单精度加法的机器周期数为 θ_1,读内存的机器周期数为 θ_2,写内存的机器周期数为 θ_3. 在 DAA 协议的某一阶段某实体运行单精度乘法 n_0 次,运行单精度加法 n_1 次,读内存 n_2 次,写内存 n_3 次,则该实体在此阶段运行时所需机器周期总数为

$$T=n_0\times\theta_0+n_1\times\theta_1+n_2\times\theta_2+n_3\times\theta_3 \quad (1)$$

假定 DAA 整个协议均运行在 X86 指令系统上,通常 X86 指令系统执行一次单精度乘法运算的机器周期数了 $\theta_0=2$,加法的机器周期数 $\theta_1=2$,读内存的机器周期数 $\theta_2=1$,写内存的机器周期数 $\theta_3=1$.

由式(1)根据表 8 在 Join 阶段各实体的机器周期总数如表 11. 图 1 为 Join 阶段的性能分布图. 从图 1 可以看出,在该阶段 Verifier 的性能开销占 0%,TPM 占 33.48%,Host 占 16.58%,Issuer 占 49.94%.

表 11　Join 阶段的各协议实体的机器周期总数

协议实体	机器周期数
TPM	2 104 799 479
Host	1 031 265 433
Issuer	3 082 201 305
Verifier	0

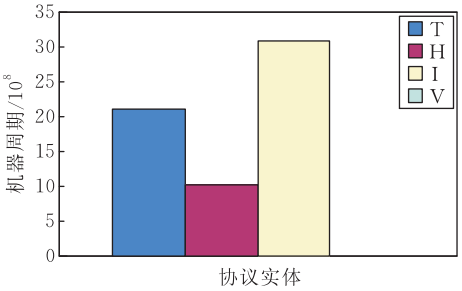


图 1　DAA Join 阶段性能负荷分布

由式(1)根据表 9 在 Sign 阶段各实体的机器周期总数如表 12. 图 2 为 Sign 阶段的性能分布图. 从图 2 可以看出，在该阶段 Issuer、Verifier 的性能开销占 0%，TPM 占 21.68%，Host 占 78.32%.

表 12　Sign 阶段的各协议实体的机器周期总数

协议实体	机器周期数
TPM	1 121 813 594
Host	4 053 907 188
Issuer	0
Verifier	0

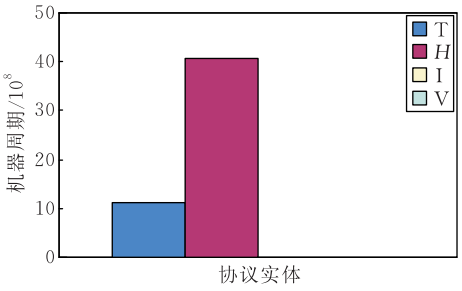


图 2　DAA Sign 阶段性能负荷分布

由式(1)根据表 10 在 Verify 阶段各实体的机器周期总数如表 13. 图 3 为 Verify 阶段的性能分布图. 从图 3 可以看出，在该阶段 TPM、Issuer 的性能开销占 0%，Verifier 占 100%.

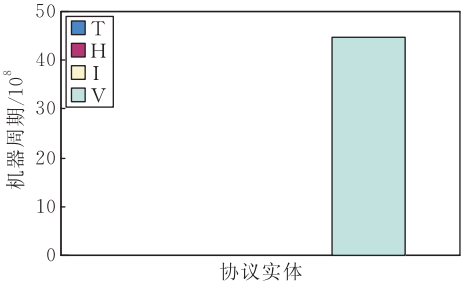


图 3　DAA Sign 阶段性能负荷分布

表 13　Verify 阶段的各协议实体的机器周期总数

协议实体	机器周期数
TPM	0
Host	0
Issuer	0
Verifier	4 478 638 536

由式(1)根据各实体的机器周期总数如表 14. 图 4 为总性能分布图. 从图 4 可以看出 DAA 整个协议完成各实体的总性能开销分布. TPM 占 16.08%，Host 占 17.46%，Host 占 27.51%，Issuer 占 31.81%，Verifier 占 24.22 %.

表 14　各实体的机器周期总数

协议实体	机器周期数
TPM	3 226 613 073
Host	5 085 172 621
Issuer	5 878 593 311
Verifier	4 478 638 536

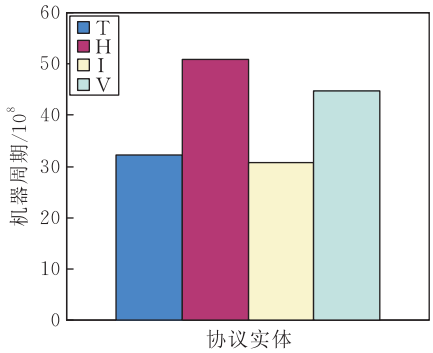


图 4　DAA 总性能负荷分布

显然，采用此方法得出的性能分布与单机系统仿真环境实测法得出的结果完全不一致[2-3].

另外，还可以对各实体在协议的各阶段之间的性能开销做一个比较：

(1) 对于 TPM，显然有 $TPM_{\text{Join}} > TPM_{\text{Sign}} > TPM_{\text{Verify}}$，其中，Join 阶段占 65.23%，Sign 阶段占 34.77%，其它两阶段占 0%.

(2) 对于 Host，显然有 $Host_{\text{sign}} > Host_{\text{Join}} > Host_{\text{Verify}}$，其中，Sign 阶段占 79.72%，Join 阶段占 20.28%，其它两阶段占 0%.

(3) 而对于 Verifier，仍有 $Verifier_{\text{Verify}} > Verifier_{\text{sign}} = Verifier_{\text{Join}}$ 其中，Verify 阶段占了 100%，其它三阶段占 0%.

4　NS 方法的应用

为了验证 NS 方法的有效性，本节将 NS 方法应用于可信计算匿名证明的文献[11]典型方案的性

能负荷估算，并将该方案与原方案进行比较.

首先统计文献[11]中的复杂运算. 文献[11]主要运算包括 $E(F_q)$ 的点加、标量乘[15]和非双线形映射，而 $E(F_q)$ 的点加、标量乘和非双线形映射的主要开销是求逆(I 表示)、模乘法(M 表示)和模平方(S 表示). 仿射坐标表示时：两不同点相加(点的加法)，需要两次模乘法，1 次模平方，1 次求逆，即 $2M+1S+1I$；相同点相加(点的倍点)，需要 2 次乘法，2 次模平方，1 次求逆，即 $2M+2S+1I$. 对采用 Tate 对计算非双线形映射[16]，需要 6 次 M_k，8 次 M_b，2 次 S_k，$(19+2k)$ 次 M，8 次 S，4 次 I，即 $6M_k+8M_b+2S_k+(19+2k)M+8S+4I$，其中 $M_k \approx k^{1.6}M$，$M_b=kM$，$S_k=2^kS$，k 是 F_{p^k} 嵌入系数. 表 15 是文献[11]中协议各阶段的主要运算统计表.

表 15 文献[11]方案的复杂运算统计

运算阶段		运算		
		点加	标量乘	映射
Join	TPM	0	3	0
	Host	1	0	6
	Issuer	2	6	0
Sign	TPM	0	1	0
	Host	0	7	1
Verify	Verifier	1	5	5

然后确定主要运算及其算法. 模乘法运算选用第 3 节中的 CIOS 算法；模平方算法选用 Montgomery 模平方算法的改进算法 OMMS1，适合通用 32 位处理器计算大整数；对于模逆算法，根据费马定理，当 q 是素数，a 是正整数且不能被 q 整除时有 $a^{-1}=a^{q-2} \bmod q$. 因此，对 $E(F_q)$ 内元素求逆都可和模指数运算统一，因此模逆算法选用 Moktgomery 模乘算法.

第 3 步是的基本运算统计. 取嵌入系统 $k=1$，$E(F_q)$ 中元素长度为 160 位，表 16 是协议各阶段运算统计表.

表 16 文献[11]方案的基本运算统计

运算阶段		运算			
		单精度乘法	单精度加法	读内存	写内存
Join	TPM	85260	81492	116871	59325
	Host	648150	569688	818022	423120
	Issuer	14718600	12667320	18191235	9422790
Sign	TPM	40770	55374	79127	38428
	Host	502075	429533	616892	319865
Verify	Verifier	786260	772375	1107453	562379

最后以机器周期数为基本单位估算性能负荷分布. 按照第 3 节的方法，表 17 是协议各阶段各协议实体的机器周期总数. 表 18 是各实体的总性能负荷分布.

表 17 文献[11]方案的性能负荷分布

阶段		负荷
		机器周期数
Join	TPM	509700
	Host	3676818
	Issuer	82385865
Sign	TPM	309843
	Host	2799973
Verify	Verifier	4787102

表 18 文献[11]方案各实体的总性能负荷分布

协议实体	机器周期总数
TPM	819543
Host	6476791
Issuer	82385865
Verifier	4787102

另外，可以将最初的原始方案和文献[11]方案作定量的比较：

(1) 对于两方案总开销，文献[11]方案机器周期总数的理论结果为 94469301，最初的原始方案机器周期总数的理论结果为 18669017541. 显然，文献[11]方案的性能远远高于最初的原始方案的整体性能，高 3 个数量级.

(2) 对于 TPM，文献[11]方案总开销为 819543，最初的原始方案的总开销为 3226613073. 显然，文献[11]方案对 TPM 的性能要求远远低于最初的原始方案，低 4 个数量级.

(3) 对于 Host，文献[11]方案总开销为 6476791，最初的原始方案的总开销为 5085172621. 显然，文献[11]方案对 Host 的性能要求远远低于最初的原始方案，低 3 个数量级.

(4) 对于 Issuer，文献[11]方案总开销为 82385865，最初的原始方案的总开销为 5878593311. 显然，文献[11]方案对 Host 的性能要求远远低于最初的原始方案，低两个数量级.

(5) 对于 Verifier，文献[11]方案总开销为 4787102，最初的原始方案的总开销为 4478638536. 显然，文献[11]方案对 Verifier 的性能要求远远低于最初的原始方案，低 3 个数量级.

5 NS 方法与其它方法的比较

本节将 NS 方法与“仿真环境模拟测试法”和“符号化运算统计法”进行比较分析.

(1) NS 方法与“仿真环境模拟测试法”相比，具有如下特点：

一方面，NS 方法不受实际测试环境、开发技

术、运行平台、网络环境等方面的影响.因此,应用本方法进行同一方案中不同协议实体之间、不同阶段以及不同方案之间性能的比较和分析要比“仿真环境模拟测试法”更方便和准确.

另一方面,“仿真环境模拟测试法”只需要在理解协议的基础上借助于一些大数运算库和加密库就可以开发实体原型进行仿真测试,理论分析难度要比 NS 方法小.但需要用户熟悉大数运算库和加密库的编程.

(2) NS 方法与“符号化运算统计法”相比,具有如下特点:

NS 方法更有效.NS 方法将所有复杂运算转化和汇总为以机器周期为单位的性能负荷分布和总性能负荷,便于不同阶段、不同实体之间的比较分析.克服了符号化统计法在不同运算间不能直接进行比较的缺点.

NS 方法精细和准确.NS 方法由于深入分析了各主要运算具体算法的内部特点,将各种负责运算归结为单精度乘、单精度加、读内存和写内存等基本操作.这样得出的结果要比符号运算统计法更精细和准确.

NS 方法更适用.NS 方法不仅适合同一方案不同实体之间的性能对比,更合适于不同 DAA 方案之间的性能对比.如第 4 节.

NS 方法理论分析难度高.与“符号化运算统计法”相比,NS 方法不仅需要分析算法中的主要运算,而且还要分析算法的实现过程以及基本运算的次数,所以,NS 方法理论分析难度高于“符号化运算统计法”的理论分析难度.

从以上的分析可以看出,尽管本文是针对原方案提出的性能估算新方法,但该方法也适用于以后相关的其它新方案.特别适合与计算类型较多、方案之间需要精细和准确比较的场合.

6　总　结

性能问题是阻碍 DAA 协议的广泛应用的瓶颈.本文提出的以机器周期为基本性能单位的性能负荷分布测量方法具有重要的意义.该方法需要首先分析 DAA 协议中的各种复杂运算,针对不同的运算选用当前性能较好的算法,然后计算各个算法中大整数单精度乘法、加法、读内存、写内存等基本运算的数目,最后通过汇总并转换得出 DAA 协议中各实体以机器周期为单位的性能负荷分布和总的性能负荷.

我们下一步的工作将在两个方面开展:(1)将该方法应用于其它方案的性能分析以及各方案之间的性能分析与比较;(2)进一步优化 DAA 协议.

参　考　文　献

[1] Brickell E, Camenisch J, Chen L. Direct anonymous attestation//Proceedings of the 11th ACM Conference on Computer and Communications Security. Washington, DC, USA, 2004: 132-145

[2] Camenisch J, Lysyanskaya A. A signature scheme with efficient protocols//Proceedings of the 3rd International Conference on Security in Communication Networks. Amalfi, Italy, 2003: 268-289

[3] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems//Proceedings of Advances in Cryptology-CRYPTO'86. London, UK, 1987: 186-194

[4] Chen Xiaofeng, Feng Dengguo. Direct anonymous attestation for next generation TPM. Journal of Computers, 2008, 3(12): 43-50

[5] Backes Michael, Maffei Matteo, Unruh Dominique. Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, California, USA, 2008: 202-215

[6] Brickell Ernie, Chen Liqun, Li Jiangtao. A new direct anonymous attestation scheme from bilinear maps//Proceedings of the 1st International Conference on Trusted Computing. Villach, Austria, 2008: 166-178

[7] Brickell Ernie, Chen Liqun, Li Jiangtao. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. International Journal of Information Security, 2009, 8(5): 315-330

[8] Brickell Ernie, Li Jiangtao. Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities//Proceedings of the 6th ACM Workshop on Privacy in the Electronic Society. Alexandria, VA, USA, 2007: 21-30

[9] Brickell Ernie, Li Jiangtao. A pairing-based DAA scheme further reducing TPM resources//Trust and Trustworthy Computing. Lecture Notes in Computer Science 6101. Heidelberg: Springer, 2010: 181-195

[10] Chen Liqun. A DAA scheme requiring less TPM resources//Proceedings of the 5th China International Conference on Information Security and Cryptology. Beijing, China, 2009: 211-219

[11] Chen Liqun, Morrissey Paul, Smart Nigel P. Pairings in trusted computing//Proceedings of the 2nd Internation Conference on Pairing-Based Cryptography. Egham, UK, 2008: 1-17

[12] Menezes A, van Oorsehot P, Vanstone S. Handbook of Applied Cryptography. CRC Press, 1996

[13] Koe C K, Aear T, Kaliski B. Analyzing and Comparing Montgomery multiplication Algorithms. IEEE Micro, 1996, 16(3): 26-33

[14] Comba Paul G. Exponentiation cryptosystems on the IBM PC. IBM System Journal, 1990, 29(4): 526-538

[15] Skakai Y, Sakurai K. Efficient scalar multiplication on elliptic curves with direct computations of several doublings. IEICE Transactions on Fundamentals of Electronics, 2001, E84-A (1): 120-129

[16] Abdulwahed M Ismail, Mohamad Rushdan MD Said, Kamel Ariffin Mohd Atan et al. Bilinear pairing computation using the extended double-base chains algorithm. International Journal of Mathematical Analysis, 2010, 4(19): 929-941

TAN Liang, born in 1972, Ph. D., professor. His research interests include trusted computing, network security.

MENG Wei-Ming, born in 1985, M. S. candidate. His research interest is trusted computing.

ZHOU Ming-Tian, born in 1937, professor. His research interests include network computing, information security.

Background

Performance estimate of Direct Anonymous Attestation Scheme is important problem for trusted Computing, which is the focus of researches in information security field.

Performance problem is a most important problem to Direct Anonymous Attestation Scheme in TCG. The original DAA scheme's computation, is much more complex than the Privacy CA, is based on the modular exponentiations, modular squarings and multiplications, of which the great problem is complex and great time consumption. Due to the limited computing power of the TPM, the complexity of the DAA protocol is not only to the disadvantage of its practical application, but also to seriously hinder development of trusted computing widely and deeply. So the study of a simpler and more efficient DAA protocol is very significant. It is very necessary and important to analyze and measure performance to every entity quantitatively for optimizing DAA.

Currently, performance estimate of Direct Anonymous Attestation Scheme generally takes on "simulation method" or "operator statistics method", but these two methods can't exactly estimate the performance of DAA.

In this paper, a new measurement method, called Normalized Statistics method, which takes the machine period as the basic performance unit, is put forward. When using this method, all complex calculates in DAA protocol must be found out and statistic, and better algorithms to every complex calculate are chosen, moreover, to each algorithms, we need to compute the sum for each basic operation, such as multiplication of single big integer, addition of single big integer, reading and writing memory, and so on. Finally, the every entity performance and the whole performance burden in DAA, whose unit is the machine period, are summed. The theoretical analysis rusults show that the performance estimate is exact, meticulous and effective by this method which is independent of actual platform. For proving availability of the method, we apply it to estimate performance of other one DAA scheme.

This work is support by the National Natural Science Foundation of China under Grant No. 60970113 and Sichuan Funds for Distinguished Young Scientists under Grant No. 0110028. They aim to find out an optimizing DAA scheme to decrease complex and time-consuming of the original DAA scheme, and to improve the practical application of the DAA scheme.