带时间约束实时任务图模型上可调度性分析算法研究

孙景昊 关 楠 邓庆绪
(东北大学信息科学与工程学院 沈阳 110819)

摘 要 带时间约束的实时任务图(TCDRT)模型具有接近于时间自动机的丰富表达性,但是其关联的可调度性分析(SA)问题却是强 NP 困难的.目前的研究仅关注一类约束个数为常数 K 的易解模型:K-TCDRT,且局限于 SA 问题的图转换求解方法.这种间接求法使得问题的计算复杂度随约束宽度呈指数倍增长.该文研究 TCDRT 模型上可调度性分析问题的直接求解方法,为两个核心子问题给出新的理论结果:第一,针对需求上界函数(DBF)的计算问题,提出了考虑时间约束的路径需求结构,并据此设计了新的动态规划算法,其时间复杂度与约束宽度无关;第二,对于可调度分析上界 T 的限定问题,从理论上证明了该问题是伪多项式时间可解的,且计算复杂度不再与 K 指数相关,这使得文中算法性能较已有结果有指数级提升.更进一步地,该文方法还蕴含着一类新的 TCDRT 易解模型.该类模型突破了约束个数必须为常数的局限,其分析难度也较 K-TCDRT 有指数倍地下降.

关键词 时间约束;实时任务图;可调度性分析;需求上界函数;动态规划
中图法分类号 TP301 **DOI 号** 10.11897/SP.J.1016.2016.02481

The Digraph Real-Time Task Model with Timing Constraints: Schedulability Analysis Revisited

SUN Jing-Hao GUAN Nan DENG Qing-Xu
(*School of Information Science and Engineering, Northeastern University, Shenyang* 110819)

Abstract The digraph real-time task with timing constraints (TCDRT) is one of the most expressiveness models in real-time community, but its corresponding schedulability analysis (SA) is strongly NP-hard problem. Present researchers focus on a tractable TCDRT model where the number of constraints is bounded by a constant K, and the only known method for the TCDRT model uses a transformation into an equivalent DRT model, which leads to a high complexity that is exponential in the width of the constraints. This work analyzes the schedulability of the TCDRT model directly in order to achieve a much lower complexity. First, we propose a dynamic program to deal with the demand bound computation problem for the K-bounded TCDRT case and refine the complexity result to a better bound that has no relation with the width of constraints. Second, we prove that the schedulability of bound T can be computed within a pseudo-polynomial time, and the corresponding computation complexity is drastically linear in K instead of being exponential. Furthermore, our approach also indicates another tractable TCDRT model that is not necessary to postulate the K-bounded constraints.

Keywords timing constraints; digraph real-time tasks; schedulability analysis; demand bound function; dynamic programming

收稿日期:2015-01-06;在线出版日期:2015-06-18.本课题得到国家"九七三"重点基础研究发展规划项目预研项目(2014CB360509)、国家自然科学基金(61300022,61300194,61472072)、中央高校基本科研业务费(N130423007,N130504008)、河北省自然科学基金(F2013501048)资助.**孙景昊**,男,1985 年生,博士,讲师,主要研究方向为实时系统调度算法,最优化理论和时间自动机.E-mail:jhsun@mail.dlut.edu.cn.
关 楠,男,1981 年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为实时系统、嵌入式系统.**邓庆绪**(通信作者),男,1970 年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为实时嵌入式系统、可重构计算、物联网.E-mail:dengqx@mail.neu.edu.cn.

1 引 言

在实时系统验证领域,形式化模型更关注系统的时间行为和不确定性等非功能特征,是系统可调度性分析的理论基础.为便于理论分析,形式化模型通常将系统抽象为一组独立且并发的实时任务 $T=\{\tau_1,\cdots,\tau_N\}$,每个任务 τ_i 都将产生无限数量的作业序列.其中,最经典的实时系统模型是 Liu 和 Layland(1973)[1]提出的周期任务模型.该模型上定义的可调度性分析(Schedulability Analysis,SA)问题存在多项式时间算法.然而,模型在刻画系统行为方面却受到极大局限:实时任务的截止期必须和周期相等;任务释放的作业具有统一的执行时间,且释放时刻必须严格限定在周期的整数倍.作为另一个极端,时间自动机(Timed Automata,TA)[2-3]能够精确刻画系统的时间特征,是实时系统最强有效的形式化模型之一.但是,TA 上 SA 问题却是强 NP 困难的,甚至是不可判定性问题[4].

一种优良的实时任务模型需在可表达性和易解性之间寻求一个折衷[5].之前的研究通常把 SA 问题是否具有(伪)多项式时间算法,作为衡量一个模型是否易解的标准[6].在 TA 和易解模型之间划定一条精确的界限,是实时系统领域面临的一个挑战性问题[7].在过去的 40 年间,众多学者致力于提出尽可能接近 TA 的易解模型,将模型上 SA 问题的难度限定在弱 NP 困难问题之内,探究伪多项式时间算法的存在性,并努力尝试优化和提高算法的性能.

偶发任务模型是直接扩展自 Liu&Layland 模型的首个系统模型[8].该模型刻画了实时任务释放作业时刻的不确定性.与 Liu&Layland 模型不同,偶发任务模型上定义的 SA 问题成为 NP 困难问题,不再具有多项式时间算法.Baruah 等人(1990)[9]将此类任务系统的利用率 U 限定在一个严格小于 1 的常数 c 内,并提出了伪多项式时间算法.文献[9]为之后的实时系统可调度性分析提供了统一的算法框架:首先,独立地分析每个实时任务的行为特征;然后,将这些分析结果线性组合在一起,构成整个任务系统的判定结果.具体来说,文献[9]引入需求上界函数(Demand Bound Function,DBF)的概念,将 SA 问题归结为两个子问题来解决.

(1)对于任意时间段 t,如何计算任务 τ_i 的需求上界函数 $dbf_{\tau_i}(t)$?(整个任务系统的需求上界函数 $dbf(t)=\sum_{i=1}^{N}dbf_{\tau_i}(t)$).

(2)如何给定 t 的上界 T,使得:对于任意的 $t\leq T$,若 $dbf(t)\leq t$,则可判定系统是可调度的.

在偶发任务模型中,SA 问题的计算复杂性取决于问题(2)中给出的上界 T.文献[9]最先给出的上界 T 如下式所示,其中,D_i 为任务 τ_i 的截止期,P_i 为任务 τ_i 的周期,U 为任务系统的利用率.在之后的研究中,上界 T 又称为可调度性分析上界[10].

$$T=\max\left\{D_1,\cdots,D_N,\max_{1\leq i\leq N}\{P_i-D_i\}\frac{U}{1-U}\right\}.$$

最近又有学者进一步缩小了该上界的范围[10],如表 1 中所示.但新的上界仍然是关于 $1/U$ 的伪多项式规模函数.这意味着,当利用率 U 趋于 1 时,在最坏情况下 SA 算法将求解指数多个 DBF 计算问题.这使得基于 DBF 的分析方法具有一定的局限性.但是,同时也注意到,DBF 方法的优点是:将困难的 SA 问题分解为若干易于求解的 DBF 计算子问题.偶发任务模型上关联的 DBF 计算问题具有 $O(1)$ 的时间复杂度[9],这极大地降低了 SA 问题的难度.在更多任务模型的研究中,学者依然期望获得较低时间复杂度的 DBF 计算方法.但是,随着模型的可表达性增强,DBF 计算问题将变得越来越困难.

广义多帧任务(Generalized Multi-frame Task,GMF)模型[11]是偶发任务模型的一个重要扩展.GMF 模型允许任务中具有多种类型的作业,各类作业具有不同的执行时间;同一任务中不同类型的作业以线性序释放.在 GMF 模型中,DBF 计算问题不再具有 $O(1)$ 时间可解的特性.为求解该问题,Baruah 等人(1999)[11]提出了一个后来称之为"需求二元组"(Demand Pair,DP)[12]的数据结构,并基于该结构给出了多项式时间的动态规划算法.在之后的研究中,DP 结构被应用到更复杂的任务模型,如复发实时任务(Recurring Real-time Task,RRT)模型[13]和非循环 GMF(non-cyclic GMF)模型[14].值得注意的是,在 RRT 和非循环 GMF 模型中,根据 DP 结构设计的动态规划算法不再是多项式时间的,而是退化成伪多项式时间算法.其中,RRT 模型中计算 DBF 的复杂度与时间上界 T 无关[15];非循环 GMF 模型中计算 DBF 的复杂度是关于 T 的线性函数[14],如表 1 所示.

表 1　任务模型上可调度性分析问题的计算复杂性

任务模型名称	DBF 计算问题复杂度	可调度性分析上界 T	
		伪多项式规模上界 T	计算利用率 U 的复杂度
偶发任务模型	$O(1)$	$\max\left\{(D_1-P_1),\cdots,(D_N-P_N),\dfrac{\sum_{i=1}^{N}(P_i-D_i)U_i}{1-U}\right\}$	$O(N)$
GMF 模型	$O(n^2\log n)$	$\dfrac{U}{1-U}\max_{i=1,\cdots,N}\{P^i_{\text{sum}}-D^i_{\min}\}$	$O(N)$
RRT 模型	$O(n^3E_i^{\max})$	$\dfrac{2\sum_{i=1,\cdots,N}E_i}{1-U}$	$O(Nn)$
non-cyclic GMF 模型	$O(nT)$	$\dfrac{\sum_{i=1,\cdots,N}E_i^{\max}}{1-U}$	$O(N)$
DRT 模型	$O(nT^2)$	$\dfrac{\sum_{i=1,\cdots,N}E_i^{\text{sum}}}{1-U}$	$O(Nn^3)$
TCDRT 模型	$O\left(n\prod_{k=1}^{K}\gamma_kT^2\right)$	$\dfrac{\sum_{i=1,\cdots,N}E_i^{\text{sum}}}{1-U}$	$O\left(Nn^3\prod_{k=1}^{K}\gamma_k^3\right)$

注：表 1 中用到的变量说明如下：n 为 τ_i 对应的任务图的节点个数；P^i_{sum} 为任务 τ_i 中所有类型作业的周期和；D^i_{sum} 为任务 τ_i 中所有类型作业的最小截止期；$E_i^{\max}$ 为任务 τ_i 中所有作业的最大执行时间；E_i 为任务 τ_i 对应的 DAG 图中从汇点到源点的最大需求路径的需求量；E_i^{sum} 为任务 τ_i 中所有作业的执行时间之和.

2011 年，Stigge 等人[12]提出实时任务图(Digraph Real-time Task，DRT)模型. DRT 模型的表达能力更加丰富，以上各类任务模型均可看做 DRT 模型的特例. 例如，GMF 中的每个任务模型是仅包含单个简单回路的 DRT 图；RRT 模型对应有向无环图(Directed Acyclic Graph，DAG)；非循环 GMF 模型对应有向完全 DRT 图[5]. 对于 DRT 模型，研究者期望找到有效的 DBF 计算方法，其时间复杂度至多限定为 T 的线性函数. Stigge 等人将 DP 结构扩展为“需求三元组”(Demand Triple，DT)结构，并基于此设计了新的伪多项式时间动态规划算法，但是时间复杂度是关于 T 的二次方函数. 本文将提出一种新的数据结构，将动态规划算法的复杂度降低到 T 的线性函数.

迄今为止，DRT 模型是与 TA 最接近的易解模型. 注意到，DRT 模型与 TA 最大的差别是没有考虑时间约束需求. 因此，Stigge 等人(2011)[7]又进一步研究了带时间约束(Timing Constraints)的 DRT(简写为 TCDRT)模型：任务 τ_i 关联一组时间约束，每个约束形如 $(from_k,to_k,\gamma_k)$，表示 τ_i 释放作业 $from_k$ 和 to_k 的时间间隔应至少与 γ_k 相等(其形式化定义详见第 2 节). TCDRT 模型属于难解模型，Stigge 等人[7]通过构造带有自环约束的 DRT 特例，将 SA 问题在伪多项式时间内规约为哈密顿路问题，从而证明了在 TCDRT 模型上 SA 问题是强 NP 困难的. 同时，Stigge 等人[7]也指出，当时间约束个数为常数 K 时，TCDRT 模型上 SA 问题变为弱 NP 困难问题，并提出了一个将 TCDRT 转化为 DRT 的方法. 这种图转化方法间接地给出了 K 约束 TCDRT 模型上 SA 问题的伪多项式时间判定算法，但同时也使得算法复杂度关于时间约束宽度 γ_k 呈指数倍增长. 无论是计算 DBF，还是限定时间上界 T，算法的时间复杂度都是关于 $\prod_{k=1}^{K}\gamma_k$ 的函数，如表 1 所示.

本文致力于给出 TCDRT 模型上 SA 问题的直接求解方法，旨在降低可调度性分析算法的时间复杂度，并找到 TCDRT 模型更多的易解特例. 首先，对于 DBF 计算问题，本文提出了更加高效的动态规划算法，其时间复杂度不再与时间约束宽度 γ_k 相关，而是简化为约束个数 K 的函数. 更进一步地，当模型中任意两时间约束 $(from_k,to_k,\gamma_k)$ 和 $(from_{k'},to_{k'},\gamma_{k'})$ 满足 $to_k\neq to_{k'}$ 时，本文中 DBF 计算方法的时间复杂度与约束个数 K 也无关，直接简化为 T 的线性函数. 另外，对于如何限定时间上界 T 的问题，本文从理论上证明了：存在一个算法能够在伪多项式时间内给出 T 的精确上界，并且算法的时间复杂度不但与 $\prod_{k=1}^{K}\gamma_k^3$ 无关，甚至不再与 K 指数相关. 因此，与文献[7]中算法相比，本文算法的性能得到指数倍提升.

本文第 2 节介绍 TCDRT 的系统模型和可调度性分析问题的形式化定义；第 3 节提出 DBF 计算问题的动态规划算法；第 4 节从理论上证明了可调度

分析上界限定问题的伪多项式时间可解性;最后是总结.

2 系统模型和问题定义

2.1 带时间约束的实时任务图

实时任务系统定义为由一组有限数量且相互独立的实时任务构成的集合$\mathcal{T}=\{\tau_1,\cdots,\tau_N\}$.$\mathcal{T}$中任意实时任务$\tau$都可由有向图$D(\tau)=(V_\tau,A_\tau)$表示,其中$V_\tau$是节点集,$A_\tau$是弧集.图$D(\tau)$中的节点$\{v_1,\cdots,v_n\}$与$\tau$释放的所有作业类型一一对应.每个节点$v_i$关联两个整形参数$e(v_i)$和$d(v_i)$,分别用来表示作业$J_i$的最坏执行时间和相对截止期.注意到,与传统调度问题[16]不同,任务τ在理论上可以任意多次释放作业J_i,规定每次释放的作业J_i具有相同的$e(v_i)$和$d(v_i)$.若作业J_i在t时刻释放,则该作业必须在$t+d(v_i)$时刻之前完成.$t+d(v_i)$又称为作业J_i的绝对截止期.另外,图$D(\tau)$中的每条弧(v_i,v_j)表达了τ释放作业J_i和J_j的先后顺序,并且弧(v_i,v_j)上关联的非负参数$p(v_i,v_j)$表示作业J_i和J_j释放时刻的最小间隔(又称为周期[12]).假设任务τ中每个作业J_i关联的截止期$d(v_i)$均小于等于相应的周期$p(v_i,v_j)$.满足此类假设的系统,称为约束截止期任务系统[9].

以上即为实时任务图(DRT)[12]的定义.在本文中,任务图$D(\tau)$还关联一组时间约束$C(\tau)=\{(from_1,to_1,\gamma_1),\cdots,(from_k,to_k,\gamma_k)\}$.$C(\tau)$中的每个约束$(from_i,to_i,\gamma_i)$要求遍历任务图$D(\tau)$上两节点$from_i$和$to_i$的时间间隔不得小于$\gamma_i$个时间单位.为表达方便,将$from_i$称为约束的左节点,$to_i$称为约束的右节点,$\gamma_k$称为约束的宽度.若约束集的势$k$为常数$K$,则任务$\tau$简记为$K$-TCDRT任务.特殊地,若$K=0$,则$K$-TCDRT任务即简化为一般的DRT任务.

例1. 一个3-TCDRT任务τ如图1所示.τ可释放6种不同类型的作业$J_1,\cdots,J_6$,在图中分别对应节点$v_1,\cdots,v_6$.节点间的实线弧表示任务τ释放作业的周期,虚线弧则表示时间约束.例如,约束$(v_6,v_2,11)$表示任务τ释放作业J_4和J_2的时间间隔不能小于11个时间单位;自环约束$(v_3,v_3,9)$表示任务τ连续释放两个J_3的时间间隔不能小于9个时间单位.

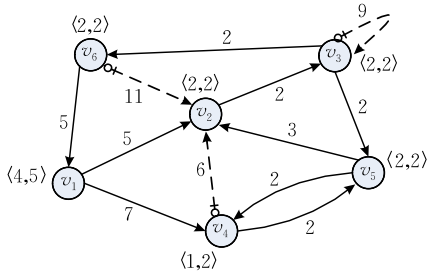


图1 包含3个时间约束的TCDRT任务τ

2.2 带时间约束任务图的语义模型

TCDRT任务τ释放的作业序列与有向图$D(\tau)$中的路径一一对应.任务τ释放作业J_i的行为触发路径对$D(\tau)$中的相应节点v_i进行遍历.可将任务τ释放并执行作业J_i的行为用一个三元组$(r_i,e(v_i),r_i+d(v_i))$来表示,其中,r_i是作业J_i的释放时刻;$e(v_i)$和$d(v_i)$是节点v_i关联的两个参数,分别对应作业J_i的执行时间和相对截止期.对于任务τ释放的任意一个作业序列$\sigma=[(r_1,e(v_{[1]}),r_1+d(v_{[1]})),\cdots,(r_i,e(v_{[i]}),r_i+d(v_{[i]})),\cdots]$,图$D(\tau)$中相应的路径定义为$\pi=[(r_1,v_{[1]}),\cdots,(r_i,v_{[i]}),\cdots]$,其中,$v_{[i]}$为路径中第$i$个遍历的节点,$r_i$为遍历节点$v_{[i]}$的时刻.由任务图定义可知,序列$\sigma$中作业的释放时间间隔需要满足周期和集合$C(\tau)$中的约束.这些时间约束可在路径$\pi$中由如下不等式精确刻画.

(1)对于任意$i=1,2,\cdots$:$r_{i+1}-r_i\geq p(v_{[i]},v_{[i+1]})$.

(2)对于任意$1\leq i<j$,且$(v_i,v_j,\gamma_l)\in C(\tau)$:$r_j-r_i\geq\gamma_l$.

另外,对于有限长度的路径$\pi=[(r_1,v_{[1]}),\cdots,(r_l,v_{[l]})]$,分别给出路径需求和长度的定义如下:

(1)路径需求:$e(\pi)=\sum_{i=1}^{l}e(v_{[i]})$.

(2)路径长度:$d(\pi)=r_l-r_1+d(v_{[l]})$.

例2. 图1中任务τ释放的一个作业序列为$\sigma_1=[(0,1,2),(3,1,6),(6,2,9)]$.该序列对应的路径为$\pi_1=[(0,v_4),(3,v_5),(6,v_2)]$.首先,观察路径$\pi_1$中相邻两节点的遍历时刻,遍历节点$v_4$和$v_5$以及$v_5$和$v_2$的时间间隔均为3.注意到,弧$(v_4,v_5)$和$(v_5,v_2)$关联的周期参数分别为2和3.易知,路径$\pi_1$中遍历相邻节点的时间间隔均大于等于相应弧关联的周期.这说明作业序列σ_1满足周期约束.另外,注意到路径π_1中包含节点v_4和v_2,这两节点的遍历时间间隔需受到$(v_4,v_2,6)\in C(\tau)$的约束.可计算得遍历v_4和v_2的时间间隔为6,恰巧等于时间约束$(v_4,v_2,6)$的宽度.这意味着作业序列σ_1在满足周期约束的同时,也满足$C(\tau)$中的时间约束.最后,路径π_1的需求和长度分别为4和9.

2.3 可调度性分析问题

本文关注 TCDRT 任务系统的可调度性，定义如下.

定义 1. 可调度性[7] (Schedulability). TCDRT 任务集 T 在可抢占的单处理器系统上是可调度的，当且仅当 T 释放的所有作业序列集合均可在该单处理器平台上由一种可抢占式策略成功调度，使得序列集中的每个作业均在其截止期之前完成.

根据定义 1，对于序列集合中任意一个作业 $J=(r,e,d)$. 可知，作业 J 在 r 时刻释放，且执行作业 J 需要占用 e 个单位时间的处理器资源. 若作业 J 能成功调度，则 J 需要在 d 时刻之前完成，即 J 占用处理器的时间需要落在时间段 $[r,d]$ 内. 可调度性分析问题是指，对于给定的 TCDRT 任务系统 $\mathcal{T}$，是否存在一个判定算法 A：若 $\mathcal{T}$ 能够被某种策略成功调度，则算法 A 回答“是”；否则，若 $\mathcal{T}$ 不能被任何调度策略成功调度，则算法 A 回答“否”. 在单处理器系统中，EDF[17-18] 是最优的调度策略[19]. 因此，单处理器上任务系统 $\mathcal{T}$ 的可调度性分析问题又等价为系统 $\mathcal{T}$ 是否可以被 EDF 策略成功调度的判定问题.

可调度性分析的算法框架建立在一种被称为需求上界函数的基础之上. 下面首先给出需求上界函数的定义.

定义 2. 需求上界函数 (Demand Bound Function, DBF). 对于 $\mathcal{T}$ 中的任意一个任务 τ 和一个时间段长度 t，$dbf_\tau(t)$ 表示任务 τ 在长度为 t 的时间段内可能产生的最大执行时间需求，其计算公式如下：

$$dbf_\tau(t) = \max\{e(\pi) \mid \pi \text{ 是 } D(\tau) \text{ 中路径，且 } d(\pi) \le t\}.$$

另外，对于整个任务系统 $\mathcal{T}$，定义需求上界函数如下：

$$dbf(t) = \sum_{\tau \in \mathcal{T}} dbf_\tau(t).$$

易知，在长度为 t 的时间段内，处理器至少要提供 $dbf(t)$ 个时间单位的计算资源用来执行任务系统 $\mathcal{T}$ 中已释放的作业. 否则，$\mathcal{T}$ 中的任务就有错失截止期的危险. 根据 DBF 的定义，能够很自然地推导出系统可调度的一个充要条件，如定理 1 所述.

定理 1[7]. TCDRT 任务系统 $\mathcal{T}$ 在单处理器上是可调度的，当且仅当：$\forall t \ge 0$：$dbf(t) \le t$.

根据定理 1，TCDRT 系统 $\mathcal{T}$ 的可调度性分析问题即等价为一个判定问题：寻找一个违反以上性质的 t_f，即 $dbf(t_f) \ge t_f$. 若存在这样的 t_f，则判定系统 $\mathcal{T}$ 不可调度；否则，系统 $\mathcal{T}$ 可调度. 该判定问题又被分解为两个关键子问题：(1) DBF 计算问题；(2) t_f 的上界限定问题. 若以上两问题均可在 (伪) 多项式时间内求解，并且问题 2 中 t_f 的上界限定在伪多项式规模内，则可断定 TCDRT 即为易解模型. 然而，文献[7]从理论上证明了 TCDRT 是难解模型，不存在伪多项式时间的可调度性分析算法，并且指出，直接解决以上两个子问题将面临如下挑战：

(1) 通过遍历图 $D(\tau)$ 来计算 BDF，需要共同考虑周期和时间约束集 $C(\tau)$ 对路径长度的影响.

(2) 更重要的是，注意到在 DRT 模型中，通过检查任务图中的简单回路，即可获得 t_f 的精确上界. 然而，该结论在 TCDRT 模型中却不成立. 这使得 DRT 中用于确定 t_f 上界的伪多项式时间算法在 TCDRT 中不再适用.

考虑到以上困难，文献[7]通过图转换间接给出了问题的求解方法：先将 TCDRT 等价转换为 DRT 模型，再应用 DRT 中的现有方法分析系统的可调度性. 这种间接方法带来了额外的复杂性，计算时间与约束宽度的连乘积 $\prod_{k=1}^{K} \gamma_k^3$ 成正比. 在接下来的两节中，将为 TCDRT 模型提出一种能够直接分析可调度性的方法，其计算复杂度也将改进为与 $\prod_{k=1}^{K} \gamma_k^3$ 不再相关.

3 求解 DBF 的动态规划算法

在之前的研究中，基于需求二元组 (DP) 结构的动态规划算法广泛用于各类实时任务模型的 DBF 计算，并且算法复杂度至多是关于可调度性分析上界 T 的线性函数[11,13-15]. 面对更复杂的 DRT 模型，DP 结构同样有效，基于 DP 的 DBF 计算方法仍被限定在伪多项式时间内，但算法复杂度却变为 T 的二次方[12]. 更加困难的是，以上算法结果并不能轻易扩展到 TCDRT 模型[7]. 迄今为止，尚未见到在 TCDRT 中计算 DBF 的伪多项式时间算法.

本节为 TCDRT 模型中的 DBF 计算问题提出一种伪多项式时间的动态规划算法. 注意到 DRT 模型是满足 $K=0$ 条件的 TCDRT 特例. 因此，在接下来的 2.1 节，首先从基本的 DRT 模型入手，以图论的角度设计新的数据结构，为 DRT 模型提出更高效的动态规划算法，使算法复杂度由 T 的平方降低到 T 的线性函数. 然后，在 2.2 节考虑 $C(\tau)$ 中时间约束对动态规划递推方程的影响，将 DRT 中的

算法扩展到 TCDRT 模型中.

3.1 DRT 模型上 DBF 的计算方法

由定义 2 可知,对于任意给定的任务 τ 和时间长度 t,计算需求上界函数 $dbf_\tau(t)$ 等价为在任务图 $D(\tau)$ 中寻找一条长度小于等于 t 的路径,使得路径需求最大.本节设计一个动态规划算法,通过多条最优子路径之间的组合,得到最优路径.其中,每条最优子路径所关联的特征量均可抽象为如下数据结构.

定义 3. 最大路径需求(Maximum Path Demand, MPD).给定图 $D(\tau)$ 中两节点 v_i 和 v_j,以及时间段长度 t,$e_{D(\tau)}(v_i,v_j,t)$ 表示在图 $D(\tau)$ 中分别以 v_i 和 v_j 为起点和终点,且长度小于等于 t 的路径对应的最大路径需求.

根据最优化原理,MPD 结构 $e_{D(\tau)}(v_i,v_j,t)$ 可由以下定理中的递推方程计算得出.

定理 2. 给定图 $D(\tau)$ 中两节点 v_i 和 v_j,$p^*(v_i,v_j)$ 表示顺次遍历 v_i 和 v_j 的最小时间间隔.对于任意的时间段长度 $t\geq p^*(v_i,v_j)+d(v_j)$:

$$\begin{aligned}e_{D(\tau)}(v_i,v_j,t)=\max\{&e_{D(\tau)}(v_i,v_j,t-1),\\&e_{D(\tau)}(v_i,v_l,t-p(v_l,v_j)-d(v_j)+\\&d(v_l))+e(v_j)\mid(v_l,v_j)\in A\}\end{aligned}\tag{1}$$

特殊地,对于 $i=j$,且 $t=d(v_i)$:$e_{D(\tau)}(v_i,v_j,t)=e(v_i)$.另外,对于任意的 $t<p^*(v_i,v_j)+d(v_j)$:$e_{D(\tau)}(v_i,v_j,t)=-\infty$.

证明. 对于 $t<p^*(v_i,v_j)+d(v_j)$ 的情况.由于在图 $D(\tau)$ 中不存在一条从 v_i 到 v_j 的路径,其长度小于 $p^*(v_i,v_j)+d(v_j)$,易知 $e_{D(\tau)}(v_i,v_j,t)=-\infty$.另外,对于 $t\geq p^*(v_i,v_j)+d(v_j)$ 的情况,将从两方面证明递推方程的正确性.

首先,考虑 $t=p^*(v_i,v_j)+d(v_j)$ 的情况.图 $D(\tau)$ 中一定存在一条从 v_i 到 v_j,且长度等于 t 的路径 π.若将周期参数看作弧上的权值,路径 π 即对应图 $D(\tau)$ 中由 v_i 到 v_j 的最短路径.计算 $e_{D(\tau)}(v_i,v_j,t)$ 的一个自然的枚举思路即在 v_i 到 v_j 的最短路径集合 ϕ 中找出最大需求的那条路径.可将最短路径集合划分为若干子集$\{\phi_l\mid(v_l,v_j)$ 是 $D(\tau)$ 中的弧$\}$.其中,ϕ_l 表示图 $D(\tau)$ 中以弧 (v_l,v_j) 结尾的路径子集.集合 ϕ_l 中需求最大的路径定义为 π_l,则 π_l 的需求 $e(\pi_l)$ 可看成两部分的加和:(1)从 v_i 到 v_l 的最大路径需求 $e_{D(\tau)}(v_i,v_l,t')$,其中 t' 的取值为 $t-p(v_l,v_j)-d(v_j)+d(v_l)$,$t$ 和 t' 的关系如图 2 所示;(2)终点上的需求 $e(v_j)$.因此,$e_{D(\tau)}(v_i,v_j,t)=\max\{e(\pi_l)\mid\phi_l\in\phi\}=\max\{e_{D(\tau)}(v_i,v_l,t')+e(v_j)\mid(v_l,v_j)\in A_\tau\}$.另一方面,根据递推公式得,$e_{D(\tau)}(v_i,v_j,t)=\max\{e_{D(\tau)}(v_i,v_j,t-1),e_{D(\tau)}(v_i,v_l,t')+e(v_j)\mid(v_l,v_j)\in A_\tau\}$.由于 $t-1<p^*(v_i,v_j)+d(v_j)$,有 $e_{D(\tau)}(v_i,v_j,t-1)=-\infty$.这种情况下,仅式(1)等号右边第二项起作用.综上可知,由枚举方法求得的最优解和递推公式得到的解值相同,故在 $t=p^*(v_i,v_j)+d(v_j)$ 时,定理 2 中结论正确.特殊地,当 $i=j$ 时,有 $p^*(v_i,v_j)=0$.易知,$t=d(v_i)$ 是 $t=p^*(v_i,v_j)+d(v_j)$ 情况的特例,故 $e_{D(\tau)}(v_i,v_i,d(v_i))=e(v_i)$ 自然成立.

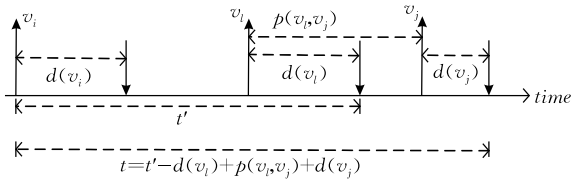


图 2 时间段长度 t 和 t' 的关系示意

接下来考虑 $t>p^*(v_i,v_j)+d(v_j)$ 的情况.一方面,证明路径存在性:图 $D(\tau)$ 中存在一条从 v_i 到 v_j,且长度至多为 t 的路径.由递推方程可知,$e_{D(\tau)}(v_i,v_j,t)$ 可由两种可能的情况计算得到:(1)若 $e_{D(\tau)}(v_i,v_j,t)=e_{D(\tau)}(v_i,v_j,t-1)$,可知图 $D(\tau)$ 中存在一条从 v_i 到 v_j,且长度至多为 $t-1$ 的路径.易知,从 v_i 到 v_j,且路径长度至多为 t 的路径显然存在;(2)若 $e_{D(\tau)}(v_i,v_j,t)$ 由方程右边第二项 $\max\{e_{D(\tau)}(v_i,v_l,t')+e(v_j)\mid(v_l,v_j)\in A_\tau\}$ 导出,则图 $D(\tau)$ 中一定存在一条从 v_i 到 v_l 的路径 $\pi'=([r_1,v_i],\cdots,[r_k,v_l])$,且路径长度 $d(\pi')$ 至多为 $t'=t-p(v_l,v_j)-d(v_j)+d(v_l)$,其中,$v_l$ 是 v_j 的直接前驱节点.将路径 π' 与弧 (v_l,v_j) 合并成一条新的路径 $\pi=([r_1,v_i],\cdots,[r_k,v_l],[r_{k+1},v_j])$.易知,新路径的长度 $d(\pi)\leq t'-d(v_l)+p(v_l,v_j)+d(v_j)=t$.综上,图 $D(\tau)$ 中一定存在从 v_i 到 v_j,且长度至多为 t 的路径.

另一方面,证明 $e_{D(\tau)}(v_i,v_j,t)$ 对应从 v_i 到 v_j 且长度至多为 t 的最大需求路径.定义 $\pi^*=([r_1,v_i],\cdots,[r_k,v_l],[r_{k+1},v_j])$ 为从 v_i 到 v_j 且长度至多为 t 的最大需求路径,假设路径 π^* 的需求 $e(\pi^*)\geq e_{D(\tau)}(v_i,v_j,t)$.由于路径 π^* 可看做是 $\pi^*_{il}=([r_1,v_i],\cdots,[r_k,v_l])$ 和 $\pi^*_{lj}=([r_k,v_l],[r_{k+1},v_j])$ 两段子路径的组合,路径长度和需求分别可展开为 $d(\pi^*)=d(\pi^*_{il})-d(v_l)+d(\pi^*_{lj})$ 和 $e(\pi^*)=e(\pi^*_{il})-e(v_l)+e(\pi^*_{lj})$.注意到子路径 π^*_{lj} 为弧 (v_l,v_j),则有 $d(\pi^*_{lj})\geq p(v_l,v_j)+d(v_j)$ 和 $e(\pi^*_{lj})=e(v_l)+e(v_j)$ 成立.又因为 $d(\pi^*)\leq t$,则有 $d(\pi^*_{il})\leq t'=t-p(v_l,v_j)-d(v_j)+d(v_l)$ 成立.即路径 π^*_{il} 的长度小于等于 t'.根据定义 3,有

$e(\pi_{il}^{*})\leq e_{D(\tau)}(v_i,v_l,t')$，代入到$e(\pi^{*})$的计算式中得：$e(\pi^{*})\leq e_{D(\tau)}(v_i,v_l,t')+e(v_j)\leq e_{D(\tau)}(v_i,v_j,t)$. 推出与假设矛盾，定理得证.　　证毕.

根据定理 2，提出计算需求上界函数 $dbf_\tau(t)$ 的动态规划算法如下.

算法 1.　计算 $dbf_\tau(t)$ 的动态规划算法.

输入：任务图 $D(\tau)$ 及时间段长度 t

输出：DRT 任务 τ 的需求上界函数集合$\{dbf_\tau(k)|k\leq t\}$

BEGIN

1. FOR each $v_i,v_j\in V_\tau$
2. 以周期作为弧的权值，计算从 v_i 到 v_j 的最短路径，并将路径权值存入 $p^{*}(v_i,v_j)$；
3. REPEAT $1\leq k\leq t$：
4. FOR each $v_i\in V_\tau$
5. FOR each $v_j\in V_\tau$
6. IF $k<p^{*}(v_i,v_j)+d(v_j)$ THEN
7. $e_{D(\tau)}(v_i,v_j,k):=-\infty$；
8. ELSE IF $i=j$ and $k=d(v_i)$ THEN
9. $e_{D(\tau)}(v_i,v_j,k):=e(v_i)$；
10. ELSE
11. IF $k-1\geq p^{*}(v_i,v_j)+d(v_j)$ THEN
12. $e_{\max}:=e_{D(\tau)}(v_i,v_j,k-1)$；
13. FOR each $(v_l,v_j)\in A_\tau$
14. IF $k\geq p(v_l,v_j)+d(v_j)-d(v_l)$ THEN
15. $e_{\max}:=\max\{e_{\max},e_{D(\tau)}(v_i,v_l,k-p(v_l,v_j)-d(v_j)+d(v_l))+e(v_j)\}$；
16. $dbf_\tau(k):=\max\{dbf_\tau(k),e_{\max}\}$；

END

算法 1 按照时间段长度 k 从 1 到 t，顺次计算需求上界函数 $dbf_\tau(k)$. 根据定义 2 可知，对于给定的 k，若要计算 $dbf_\tau(k)$，需要先获得每对节点 v_i 和 v_j 关联的 MPD 结构 $e_{D(\tau)}(v_i,v_j,k)$. 算法在 4～16 行计算了所有节点对关联的 MPD 值，并在其中选取一个最大值存储在 $e_{\max}$ 中. 根据定理 2，对于任意两节点 v_i 和 v_j，计算 $e_{D(\tau)}(v_i,v_j,k)$ 需要分 3 种情况讨论：(1) 当 $k<p^{*}(v_i,v_j)+d(v_j)$ 时，$e_{D(\tau)}(v_i,v_j,k):=-\infty$，对应算法 6，7 行；(2) 当 $i=j,k=d(v_i)$ 时，$e_{D(\tau)}(v_i,v_j,k):=e(v_i)$，对应算法 8，9 行；(3) 此外，在一般情况下，$e_{D(\tau)}(v_i,v_j,k)$ 需要由递推式(1)来计算，对应算法 10～15 行.

定理 3.　算法 1 是伪多项式时间的，且计算复杂性是关于时间段长度 t 的线性函数.

证明.　对于图 $D(\tau)$ 中任意的两个节点 v_i 和 $v_j\in V_\tau$，以及任意的时间段长度 $k\leq t$，算法 1 均需要计算一个 MPD 结构 $e_{D(\tau)}(v_i,v_j,k)$. 易知，$D(\tau)$ 中节点对个数至多为 $n\times n$，时间段长度 k 上界限定为 t. 因此，算法 1 中需要计算的 MPD 个数最多为 $O(n^2t)$. 另外，对于任意的长度 k，需求上界函数 $dbf_\tau(k)$ 可在所有 $e_{D(\tau)}(v_i,v_j,k)$ 都计算完后，由 $\max\{e_{D(\tau)}(v_i,v_j,k)|v_i,v_j\in V_\tau\}$ 计算得到，见算法第 16 行. 综上，算法 1 是伪多项式时间的，其时间复杂度为 t 的线性函数 $O(n^2t)$.　　证毕.

与之前的研究相比，算法 1 将 DRT 中 DBF 计算问题的复杂度由 t 的平方降低为 t 的线性函数，进一步提升了求解效率. 更重要的是，算法 1 还可以扩展到 TCDRT 模型，而以往 DRT 中求解 DBF 的动态规划算法在 TCDRT 中却不再适用[7].

3.2　TCDRT 模型上 DBF 的计算方法

本节将基于MPD的动态规划算法扩展到TCDRT模型中. 与 DRT 模型不同，在 TCDRT 中构造一个 MPD 结构 $e_{D(\tau)}(v_i,v_j,t)$ 需要额外考虑 $C(\tau)$ 中的时间约束. 接下来，针对路径终点 v_j 是否为某个时间约束的右节点，对 $e_{D(\tau)}(v_i,v_j,t)$ 的递推方程分两种情况进行讨论. 对于任意给定的两节点 v_i 和 v_j，以及时间段长度 t 有：

情况 1.　从 v_i 到 v_j 的路径不以任何时间约束的右节点结尾，即对于任意的 $(from_k,to_k,\gamma_k)\in C_\tau$，均有 $v_j\neq to_k$ 成立. 此时，MPD 结构 $e_{D(\tau)}(v_i,v_j,t)$ 仍然可由定理 2 中的递推方程(1)计算得到.

情况 2.　路径以某时间约束的右节点结尾，即存在 $(from_k,to_k,\gamma_k)\in C_\tau$，使得 $v_j=to_k$. 此时，计算 $e_{D(\tau)}(v_i,v_j,t)$ 则需要考虑作用于终点 v_j 上的时间约束. 不失一般性，设时间约束 $(from_k,to_k,\gamma_k)$ 的右节点 $to_k=v_j$，并且左节点 $from_k=v_l$. 假设从 v_i 到 v_j 的路径中不止一次遍历节点 v_l，则只需判断离终点 v_j 最近的一次遍历 v_l 的时间是否满足时间约束 $(from_k,to_k,\gamma_k)$，如例 3 中所示. 因此，对于图 $D(\tau)$ 中的每个节点，本节只关注其在路径中最后一次被遍历的时刻.

例 3.　如图 1 中 3-TCDRT 任务 τ，考虑 $C(\tau)$ 中的时间约束 $(v_4,v_2,6)$. 图中存在一条路径 $\pi=[(0,v_4),(2,v_5),(4,v_4),(6,v_5),(8,v_4),(10,v_5),(14,v_2)]$，如图 2 所示. 路径 π 遍历终点的时刻为 14. 注意到，路径 π 三次遍历节点 v_4，其遍历时刻按离终点 v_2 由远及近顺序排列分别为 0，4 和 8. 显然，只需检查离 v_2 最近一次对 v_4 进行遍历的时刻，即可判断 π 是否满足时间约束 $(v_4,v_2,6)$. 即由 $14-8\geq 6$ 可判定整条路径 π 满足约束 $(v_4,v_2,6)$.

考虑更为一般的情况，假设 $C(\tau)$ 中有多个时间

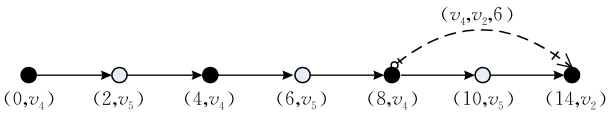


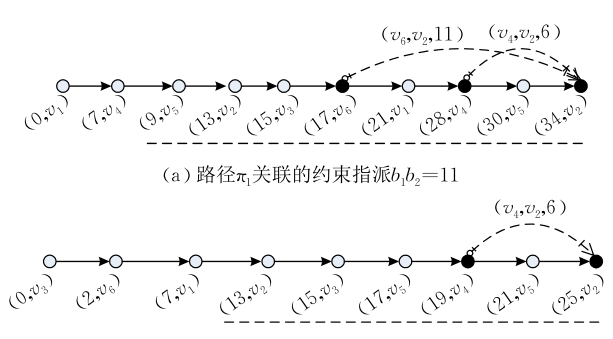
图 3 时间约束$(v_4,v_2,6)$中的左节点 v_4 在路径 π 中被多次遍历

约束都作用于路径终点 v_j. 所有将 v_j 作为右节点的时间约束构成集合 $C(v_j)=\{(v_{i1},v_j,\gamma_1),\cdots,(v_{ikj},v_j,\gamma_{kj})\}$. 对于每条以 v_j 为终点的路径 π,都对应一个 $C(v_j)$到 π 的指派,定义如下.

定义 4. 约束指派(Constraint Assignment, CA). 对于给定的节点 v_j 和一条以 v_j 为终点的路径 π,约束集合 $C(v_j)=\{(v_{i1},v_j,\gamma_1),\cdots,(v_{ikj},v_j,\gamma_{kj})\}$ 到路径 π 的指派定义为 $assign_j(\pi)=b_1\cdots b_l\cdots b_{kj}$. 其中,$b_l$ 为一个二进制位:若路径 π 包含节点 v_{il},且在最后一次遍历 v_{il} 之后除终点外再未遍历 v_j,则 $b_l=1$;否则,$b_l=0$.

通常将路径 π 的约束指派 $assign_j(\pi)$ 记为 $b_1\cdots b_l\cdots b_{kj}$ 的十进制形式. 为表达方便,可直接用十进制指派赋值来标识路径 π,即 $\pi=b_1+2b_2+\cdots+2^{kj-1}b_{kj}$. 约束指派精确标识了那些作用于路径终点的时间约束,以下是约束指派赋值的例子.

例 4. 考虑图 1 中节点 v_2,在 $C(\tau)$中以 v_2 为右节点的约束子集为 $C(v_2)=\{(v_4,v_2,6),(v_6,v_2,11)\}$. 分别用二进制位 b_1 和 b_2 代表 $C(v_2)$中的约束 $(v_4,v_2,6)$ 和 $(v_6,v_2,11)$. 路径 $\pi_1=[(0,v_1),(7,v_4),(9,v_5),(13,v_2),(15,v_3),(17,v_6),(21,v_1),(28,v_4),(30,v_5),(34,v_2)]$以 v_2 结尾,且路径 π_1 中包含 $C(v_2)$中两约束的左节点 v_4 和 v_6. 注意到,路径 π_1 在最后一次遍历 v_4 和 v_6 后,以及到达终点 v_2 之前,再也没有对节点 v_2 进行遍历,如图 4(a)所示. 根据定义 4,约束 $C(v_2)$到 π_1 的指派赋值为 $b_1b_2=11$. 另外,图 4(b)同样给出一条以 v_2 结尾且包含 v_4 和 v_6 的路径 $\pi_2=[(0,v_3),(2,v_6),(7,v_1),(13,v_2),(15,v_3),(17,v_5),(19,v_4),(21,v_5),(25,v_2)]$. 注意到,路径 π_2 在最后一次遍历 v_6 之后并且在到达终点 v_2 之前,又遍历了节点 v_2. 因此,约束 $(v_6,v_2,11)$ 对应的二进制位 b_2 应取 0. 路径 π_2 对应的约束指派应赋值为 $b_1b_2=10$.



(a) 路径π_1关联的约束指派$b_1b_2=11$

(b) 路径π_2关联的约束指派$b_1b_2=10$

图 4 关联不同约束指派的路径实例

基于定义 4,DRT 中的路径需求定义可进一步扩展为 TCDRT 模型中的指派路径需求.

定义 5. 指派路径需求(Assigned Path Demand, APD). 给定时间段长度 t,以及图 $D(\tau)$的任意子图 D,对于图 D 中任意一条从 v_i 到 v_j 的路径 π,$e_D(v_i,v_j,assign_j(\pi),t)$表示路径 π 在满足 $C(v_j)$中的时间约束,且长度 $d(\pi)\leq t$,时,所获得的最大需求,其中 $assign_j(\pi)$为 $C(v_j)$到路径 π 的约束指派.

根据以上定义,对于图 D 中任意的两节点 v_i 和 v_j,以及时间段长度 t,相应的 MPD 结构 $e_D(v_i,v_j,t)$可由式(2)计算得出

$$e_D(v_i,v_j,t)=\max\{e_D(v_i,v_j,assign_j(\pi),t)\mid 0\leq\pi\leq 2^{k_j}-1\} \tag{2}$$

式(2)中的 π 表示从 v_i 到 v_j 路径关联约束指派 $b_1\cdots b_l\cdots b_{kj}$ 对应的十进制. 易知,$b_1\cdots b_l\cdots b_{kj}$ 的取值范围从 0…0 到 1…1,对应的十进制从 0 到 $2^{kj}-1$. 式(2)说明,一旦求得所有的 APD 结构 $e_D(v_i,v_j,assign_j(\pi),t)$,即可获得 MPD 结构 $e_D(v_i,v_j,t)$. 显然,当 k_j 为常数时,在 TCDRT 模型中计算 MPD 结构 $e_D(v_i,v_j,t)$是伪多项式时间的.

对于给定的约束指派 $assign_j(\pi)=b_1\cdots b_l\cdots b_{kj}$,APD 结构 $e_D(v_i,v_j,assign_j(\pi),t)$的计算方法主要基于对路径集合进行划分的思想. 考虑所有从 v_i 到 v_j,且关联指派 $assign_j(\pi)$的路径,这些路径构成的集合定义为 ϕ. 另外定义 ϕ 关联的节点集定义为 $V(\phi)=\{v_{il}\mid(v_{il},v_j,\gamma_l)\in C(v_j)\wedge b_l\in assign_j(\pi)\wedge b_l=1\}$. 显然,$\phi$ 中每条路径均会对 $V(\phi)$中的所有节点进行遍历. 定义 ϕ 的子集 $\phi_l=\{\pi\mid\pi\in\phi$ 且 $V(\phi)$中节点在 π 中最早完成最后一次遍历的是 $v_{il}\}$. 易知,$\bigcup_{l=1}^{k_j}\phi_l=\phi$,且对于任意 $l\neq l'$,有 $\phi_l\cap\phi_{l'}=\varnothing$ 成立. 可知,路径集合 ϕ 存在一个划分$\{\phi_l\mid v_{il}\in V(\phi)\}$. 根据定义 5,$e_D(v_i,v_j,assign_j(\pi),t)$对应 ϕ 中长度小于 t,且具有最大需求的路径. 显然,$e_D(v_i,v_j,assign_j(\pi),t)$也可通过对划分集合进行如下操作获得.

首先,分别在每个子集 ϕ_l 中找出长度小于等于 t 的路径,计算这些路径的最大需求值 $e_D(\phi_l,t)$,具体的递推方程如下:

$$e_D(\phi_i,t)=\max\{e_D(v_i,v_{il},t')+e_{D[v_{il},v_j]}(v_f,v_j,assign_j(\pi-2^{l-1}),$$

$$t-t'-p(v_{il},v_f))\,|\,(v_{il},v_f)\in A$$

且　　$$t-t'\geq\gamma_l+d(v_j)\}\tag{3}$$

如图 5 所示，对于 ϕ_l 中任意一条路径 π，式(3)以最后一次遍历节点 v_{il} 为分界，将 π 分解成三部分：(1)从 v_i 到 v_{il} 且满足时间约束的路径 π_1；(2)衔接弧 $(v_{il},v_f)\in A$，其中，A 表示图 D 的弧集；(3)从 v_f 到 v_j 且满足时间约束的路径 π_2，并且 π_2 中再未遍历节点 v_{il}，且仅在结尾遍历 v_j，为了保证这一点，定义 π_2 为图 $D[v_{il},v_j]$ 上的路径，其中 $D[v_{il},v_j]$ 为图 D 在删去 v_{il} 的入弧和 v_j 的出弧后得到的子图. 以图 1 中的任务为例，$D(\tau)$ 的一个子图 $D[v_4,v_2]$ 在例 5 中给出.

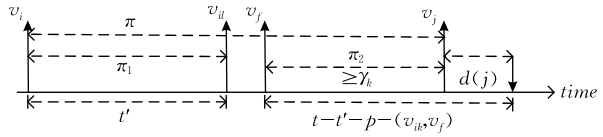


图 5　路径 π 分解为 3 段子路径：π_1，(v_{il},v_f) 和 π_2

例 5.　考虑图 1 中的任务图 $D(\tau)$，令图 $D=D(\tau)$. 则 D 的子图 $D[v_4,v_2]$ 如图 6 所示. 在图 $D[v_4,v_2]$ 中，图 1 中 v_4 的入弧和 v_2 的出弧均被删掉. 易知，图 $D[v_4,v_2]$ 上的任何路径中除起点和终点外，不可能再出现节点 v_4 和 v_2.

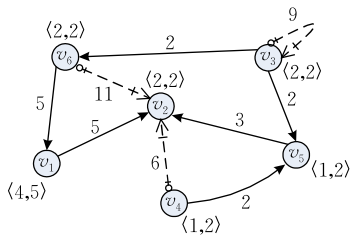


图 6　$D(\tau)$ 的子图 $D[v_4,v_2]$

由定义 4 可知，将子路 π_1 从路径 π 中去除后，对整条路径关联的约束指派没有影响. 但是，若再去掉衔接弧 (v_{il},v_f)，由于去掉了所有 v_{il} 节点，剩余路径关联的约束指派中相应二进制位 b_l 将由 1 置为 0. 即路径 π_2 的约束指派为 $assign_j(\pi)-2^{l-1}$. 另外，由于路径 π 的长度要求小于等于 t，以上三段子路径的长度之和也应小于等于 t. 若路径 π_1 的长度定义为 t'，衔接弧的周期为 $p(v_{il},v_f)$，则 π_1 的长度需小于等于 $t-t'-p(v_{il},v_f)$，如图 5 所示. 最后，由子路径定义可知，尽管 $C(\tau)$ 中的时间约束在 π_1 和 π_2 中分别得到满足. 但是，路径 π 关联的时间约束 (v_{il},v_j,γ_l) 跨越了两段子路径 π_1 和 π_2，需要在式(3)中额外加入限定条件 $t-t'\geq\gamma_l+d(v_j)$ 以体现约束 (v_{il},v_j,γ_l) 对路径的作用. 综上可知，以 (v_{il},v_f) 为衔接弧的路径其最大需求等于以下两部分之和：(1)π_1 的最大需求 $e_D(v_i,v_{il},t')$；(2)π_2 的最大需求 $e_{D[v_{il},v_j]}(v_f,v_j,t-t'-p(v_{il},v_f))$. 如式(3)所示，考虑图 D 中 v_{il} 的所有出弧作为衔接弧以及所有时间段长度 $t'\leq t-\gamma_l-d(v_j)$，即可求得 ϕ_l 中路径的最大需求 $e_D(\phi_l,t)$.

进一步地，按照以上方法计算每个集合 ϕ_l 中路径的最大需求，即可获得APD结构 $e_D(v_i,v_j,assign_j(\pi),t)$，如式(4)所示.

$$e_D(v_i,v_j,assign_j(\pi),t)=\max\{e_D(\phi_l,t)\,|\,v_l\in V(\phi)\}\tag{4}$$

综上，可给出计算 TCDRT 模型中需求上界函数 $dbf_\tau(t)$ 的动态规划算法：首先，对于所有节点对 v_i 和 $v_j\in V_\tau$，以及每个时间段长度 $k\leq t$，联合应用递推方程(1)～(4)计算相应的 MPD 结构 $e_{D(\tau)}(v_i,v_j,k)$；然后，再在其中选取最大值作为 $dbf_\tau(t)$，如式(5)所示.

$$dbf_\tau(t)=\max\{e_{D(\tau)}(v_i,v_j,k)\,|\,v_i,v_j\in V_\tau\wedge k\leq t\}\tag{5}$$

定理 4.　K-TCDRT 模型中的 DBF 计算问题是伪多项式时间的，且计算复杂性与时间约束宽度无关.

证明.　在计算 $dbf_\tau(t)$ 的过程中，可将需要计算的子结构分为两类：(1)与计算 $dbf_\tau(t)$ 直接相关的 MPD 结构 $e_{D(\tau)}(v_i,v_j,k)$，如式(5)所示. 注意到，这部分 MPD 结构均定义在任务图 $D(\tau)$ 上，个数限定在 $O(n^2t)$ 内；(2)与计算 $dbf_\tau(t)$ 间接相关的 MPD 和 APD 子结构. 在计算每个定义图 $D(\tau)$ 上的 MPD 结构时，需要额外计算一系列新的定义在图 D 上的 MPD 结构 $e_D(v_i,v_j,k)$ 和 APD 结构 $e_D(v_i,v_j,assign_j(\pi),k)$，其中，$D$ 为 $D(\tau)$ 的诱导子图，D 在反复使用递推方程(3)的过程中获得. 注意到，APD 结构 $e_D(v_i,v_j,assign_j(\pi),k)$ 仅在 v_j 是 $C(\tau)$ 中某时间约束的右节点时才需计算. 由定义 4 和 5 可知，对于给定的 v_j，APD 结构中可能的约束指派 $assign_j(\pi)$ 个数至多为 $O(2^{k_j})$. 另外，APD 结构中可能的子图 D 个数仅与约束子集 $C(v_j)$ 的势 k_j 相关. 由子图 $D[v_{il},v_j]$ 的定义可知，对于给定的 v_j，在计算过程中可能生成的子图个数至多为 $O(2^{k_j})$. 因此，在动态规划算法中，需要计算的 APD 结构 $e_D(v_i,v_j,assign_j(\pi),k)$ 个数至多为 $O\left(\sum_{l=1}^{n}4^{k_l}nt\right)$. 又注意到，子图 D 上的 MPD 结构 $e_D(v_i,v_j,k)$ 仅在 v_j 是 $C(\tau)$ 中某时间约束的左节点时才需计算. 因此，需要计算的 MPD 结构 $e_D(v_i,v_j,k)$ 个数至多为 $O\left(\sum_{l=1}^{n}2^{k_l}nt\right)$. 显然，当中每个节点 v_j 作为右节点关联的约束个数为常数 k_j 时，TCDRT 模型中 DBF 计算问题具有伪多项式时间算法，且计算复杂度与约束宽度无关，仅是时间段长度 t 的线性函数.　　证毕.

注意到,之前的间接方法使得 DBF 计算问题的复杂度与约束宽度指数相关,且是 t 的二次函数[7].本节方法使问题的复杂度降低到 t 的一次方,且直接去除了一类问题参数——约束宽度对问题复杂度的影响.与之前研究相比,本文方法在理论上降低了 TCDRT 模型的分析难度.另外,注意到本节方法的计算复杂度仅与节点关联的约束个数 k_j 指数相关.易知,在平均情况下,k_j 是一个远小于 K 的小常数.更进一步地,若图中每个节点关联的约束个数 $k_j \leq 1$,则本节方法的计算复杂度直接简化为与 $C(\tau)$ 中的约束个数 K 不再相关,如定理 5 所述.

定理 5. 若 $C(\tau)$ 中任意两约束的右节点均不相同,则 TCDRT 模型中的 DBF 计算问题的计算复杂性为 $O(7n^2t)$.

证明. 由定理 4 的证明过程可知,定义在图 $D(\tau)$ 上 MPD 结构 $e_{D(\tau)}(v_i,v_j,k)$ 的个数为 $O(n^2t)$,定义在子图 D 上 MPD 结构 $e_D(v_i,v_j,k)$ 和 APD 结构 $e_D(v_i,v_j,assign_j(\pi),k)$ 的个数分别为 $O\left(\sum_{l=1}^{n}2^{k_l}nt\right)$ 和 $O\left(\sum_{l=1}^{n}4^{k_l}nt\right)$.由于 $C(\tau)$ 中任意两约束的右节点均不相同,可知对于任意的 $v_j \in V_\tau$,有 $k_j \leq 1$ 成立.将 $k_j \leq 1$ 代入后可知,若要得到 $dbf_\tau(t)$,共需计算的子结构个数至多为 $O(7n^2t)$.定理得证. 证毕.

4 可调度性分析上界 T 的限定策略

本节给出可调度性分析上界 T 的限定方法.首先介绍要用到的相关概念.

定义 6. 利用率(Utilization).给定 TCDRT 任务集 $\mathcal{T}$,对于每个任务 $\tau \in \mathcal{T}$,在图 $D(\tau)$ 中存在一条回路 $\pi=[(r_1,v_{[1]}),(r_2,v_{[2]}),\cdots,(r_l,v_{[l]})](v_{[1]}=v_{[l]})$,定义利用率如下:

(1) 回路 π 的利用率:$U(\pi)=\sum_{i=1}^{l}e(v_{[i]})/(r_l-r_1)$;

(2) 任务 τ 的利用率:$U(\tau)=\max\{U(\pi)\mid \pi$ 是图 $D(\tau)$ 上满足 $C(\tau)$ 中时间约束的回路$\}$;

(3) 任务系统 $\mathcal{T}$ 的利用率:$U(\mathcal{T})=\sum_{\tau\in T}U(\tau)$.

称利用率等于 $U(\tau)$ 的回路为密集回路.

以上定义与 DRT 模型中的利用率[12]定义类似.基于密集回路的利用率定义,能够成功用于限定 DRT 的可调度性分析上界 T.根据文献[12]提供的方法,可以通过搜索简单回路寻找密集回路,进而在伪多项式时间内得到任务 τ 的利用率 $U(\tau)$.但是,在 TCDRT 模型中,密集回路却不再局限于简单回路[7].这给计算任务利用率 $U(\tau)$ 带来挑战.本节从新的角度,给出 TCDRT 模型中密集回路的伪多项式时间求解方法.

本节方法基于对时间约束在路径中位置的观察.如图 7 所示,对于作用于路径 π 的任意两个时间约束 (v_i,v_j,γ_1) 和 $(v_{i'},v_{j'},\gamma_2)$,两约束之间的位置关系可分为以下两类:

(1) 约束独立.如图 7(a)所示,时间约束 (v_i,v_j,γ_1) 和 $(v_{i'},v_{j'},\gamma_2)$ 在 π 中起作用的路径段分别为 π_{ij} 和 $\pi_{i'j'}$.由于这两段子路径没有重合部分,故称两约束彼此独立,记为 $(v_i,v_j,\gamma_1)\bigcap(v_{i'},v_{j'},\gamma_2)=\varnothing$.

(2) 约束覆盖.如图 7(b)所示,有时间约束作用的两段路径 π_{ij} 和 $\pi_{i'j'}$ 有重合,则称时间约束部分覆盖,记为 $(v_i,v_j,\gamma_1)\bigcap(v_{i'},v_{j'},\gamma_2)\neq\varnothing$.特殊地,如图 7(c)所示,路径 π_{ij} 是 $\pi_{i'j'}$ 的子路径,则称约束 (v_i,v_j,γ_1) 被 $(v_{i'},v_{j'},\gamma_2)$ 完全覆盖,记为 $(v_i,v_j,\gamma_1)\subset(v_{i'},v_{j'},\gamma_2)$.

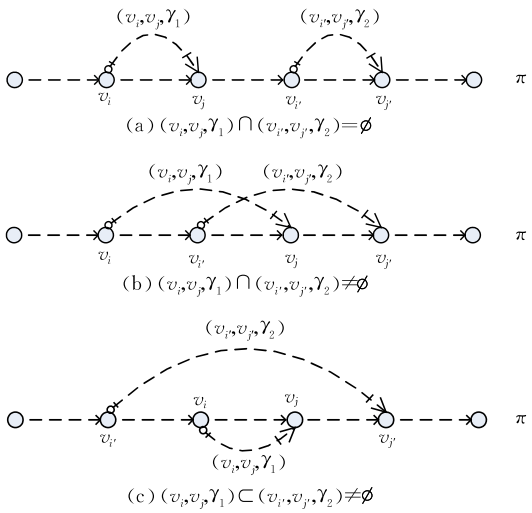


图 7 两时间约束不同的位置关系

定理 6. 在密集回路中,一个时间约束以不被其他约束完全覆盖的形式至多出现一次.

证明. 假设在密集回路 π 中,时间约束 (v_i,v_j,γ_1) 以不被其他约束完全覆盖的形式出现了两次,如图 8 所示.易知,π 中存在两个子回路 $\pi_1=(v_i,\cdots,v_j,\cdots,v_i)$ 和 $\pi_2=(v_j,\cdots,v_i,\cdots,v_j)$.若删去其中任何一条子回路,则 π 中将会只包含一个不被其他约束完全覆盖的约束 (v_i,v_j,γ_1).不失一般性,选择将 π_1 从回路 π 中删去,得到 $\pi-\pi_1$.显然,$\pi-\pi_1$ 仍然是一个回路.由于 π_1 中的约束 (v_i,v_j,γ_1) 不被其他任何约束完全覆盖,可知一定不存在一个约束的作用域完全包括 π_1.因此,回路 π 在删去 π_1 后,并不影响时间约束的满足性.易知,π_1 和 $\pi-\pi_1$ 均是满足时间约束的独立子回路.由定义 6 可知,回路 π 的

利用率 $U(\pi)=(e(\pi_1)+e(\pi-\pi_1))/(d(\pi_1)+d(\pi-\pi_1))$，而回路 π_1 和 $\pi-\pi_1$ 的利用率分别为 $U(\pi_1)=e(\pi_1)/d(\pi_1)$ 和 $U(\pi-\pi_1)=e(\pi-\pi_1)/d(\pi-\pi_1)$. 易知，$U(\pi_1)$ 和 $U(\pi-\pi_1)$ 中至少有一个大于等于 $U(\pi)$. 所以，可以通过删去独立回路 π_1 和 $\pi-\pi_1$ 来获得比回路 π 更密集的回路，这与假设相矛盾. 定理得证.　　证毕.

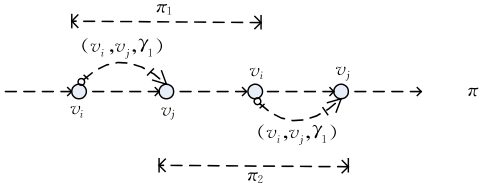


图 8　(v_i,v_j,γ_1)在路径 π 中以不被其他约束完全覆盖的方式出现了两次

由定理 6 可得密集回路的长度上界，如定理 7 所述.

定理 7.　图 $D(\tau)$ 中密集回路的长度不超过 $d_{\max}=(2K+1)\sum_{l=1}^{K}\gamma_l+3K\sum_{(v_i,v_j)\in A_\tau}p(v_i,v_j)$.

证明.　首先，对于密集回路 π 中每个不被其他约束完全覆盖的时间约束$(from_l,to_l,\gamma_l)$，受该约束作用的子路径长度至多为 $P+\gamma_l$，其中，P 为图 $D(\tau)$ 上满足时间约束的最长简单路径加上最长简单回路的长度. 否则，约束$(from_l,to_l,\gamma_l)$作用的路径段内一定会出现冗余回路 π'，删去 π' 后不影响约束$(from_l,to_l,\gamma_l)$的满足性. 可知，子回路 π' 和 $\pi-\pi'$ 中至少有一个回路的利用率大于等于 π 本身，这与 π 是密集回路相矛盾. 根据引理 6，$C(\tau)$中的每个时间约束在 π 中以不被其他约束完全覆盖的形式至多出现一次. 因此，π 中受时间约束作用的路径段的总长度一定小于等于 $KP+\sum_{l=1}^{K}\gamma_l$. 又因为 $P\leq 2\sum_{l=1}^{K}\gamma_l+2\sum_{(v_i,v_j)\in A_\tau}p(v_i,v_j)$，故 π 中受到约束作用的子路径长度上界为$(2K+1)\sum_{l=1}^{K}\gamma_l+2K\sum_{(v_i,v_j)\in A_\tau}p(v_i,v_j)$. 另外，易知 π 中不受任何约束作用的子路径也至多有 K 段，其中每段子路径的长度存在一个上界为 $\sum_{(v_i,v_j)\in A_\tau}p(v_i,v_j)$. 综上可知，图 $D(\tau)$中密集回路的长度上界为$(2K+1)\sum_{l=1}^{K}\gamma_l+3K\sum_{(v_i,v_j)\in A_\tau}p(v_i,v_j)$. 证毕.

定理 7 蕴含了一个计算任务利用率 $U(\tau)$的方法：为图 $D(\tau)$中所有长度小于等于 $d_{\max}$的回路计算需求和长度，得到回路利用率，并在其中选取最大者作为任务利用率 $U(\tau)$，如式(6)所示：

$$U(\tau)=\max\{e_{D(\tau)}(v_i,v_i,k)/k\mid v_i\in V_\tau\wedge k\leq d_{\min}\}\quad(6)$$

由式(6)可知，求解 $U(\tau)$的核心是计算 MPD 子结构 $e_{D(\tau)}(v_i,v_i,k)$. 可以应用 2.2 节提出的方法在伪多项式时间内给出每个 $e_{D(\tau)}(v_i,v_i,k)$的值. 又因为 $d_{\max}$是伪多项式规模的，所以可在伪多项式时间内计算出任务利用率 $U(\tau)$.

在计算出每个任务 τ 的利用率 $U(\tau)$后，整个任务集 $\mathcal{T}$的利用率 $U(\mathcal{T})$可根据定义 6 由所有 $U(\tau)$线性加和获得. 显然，$U(\mathcal{T})>1$ 能够充分说明 $\mathcal{T}$不可调度. 但是 $U(\mathcal{T})\leq 1$，却不能推出 $\mathcal{T}$可调度. 根据定理 1，当 $U(\mathcal{T})\leq 1$ 时，需要确定一个可调度分析上界 T，只要对于所有 $t\leq T$，有 $dbf(t)\leq t$ 成立，即可保证 $\mathcal{T}$可调度. 反言之，若存在 $t_f\leq T$，使得 $dbf(t_f)>t_f$，则说明 $\mathcal{T}$不可调度. 对上界 T 的直观解释如图 9 所示. 由图中可以看出，上界 T 即为 t_f 的上界，可由 DBF 的线性上界曲线与对角线求交获得. 因此，首先在定理 8 中给出 DBF 的线性上界.

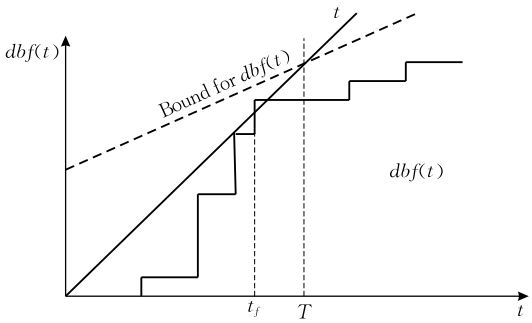


图 9　任务集的需求上界函数

定理 8.　对于任意 TCDRT 任务 τ 和时间段长度 t，有 $dbf_\tau(t)\leq t\cdot U(\tau)+\sum_{v_i\in V_\tau}e(v_i)$.

证明.　任务图 $D(\tau)$中任意一条路径 π 均可看做是一条简单路和若干回路的组合. 易知，$D(\tau)$中任何简单路的需求均小于等于 $\sum_{v_i\in V_\tau}e(v_i)$. 另外，$D(\tau)$中任意回路的利用率均小于 $U(\tau)$. 所以，若 $d(\pi)\leq t$，则 π 中所有回路的总需求必小于等于 $t\cdot U(\tau)$. 综上可知，对于任意 $d(\pi)\leq t$ 的路径 π，有 $e(\pi)\leq t\cdot U(\tau)+\sum_{v_i\in V_\tau}e(v_i)$成立. 定理得证.　　证毕.

由定理 8 可推导出 t_f的上界 T，如定理 9 所述.

定理 9.　对于任意 $U(\mathcal{T})<1$ 的 TCDRT 任务集 $\mathcal{T}$，可调度分析上界 $T=\sum_{\tau\in\mathcal{T}}\sum_{v_i\in V_\tau}e(v_i)/(1-U(\mathcal{T}))$.

证明.　若 $\mathcal{T}$ 不可调度，则存在 $t_f\leq T$，使得 $dbf(t_f)>t_f$. 由定理 8 得：$dbf_\tau(t_f)\leq t_f\cdot U(\tau)+\sum_{v_i\in V_\tau}e(v_i)$. 注意到，$\mathcal{T}$在 t_f 处的需求上界函数 $dbf(t_f)=$

$\sum_{\tau\in T} dbf_\tau(t_f)$. 故有 $t_f \leq \sum_{\tau\in T}\sum_{v_i\in V_\tau} e(v_i)/(1-U(T))$ 成立. 证毕.

根据定理 9,只需检查小于等于 T 的时间段长度 t,即可判定任务集的可调度性. 注意到 $U(T)$ 严格小于 1,故 T 是伪多项式界的. 据此,可得本文的主要结论如下.

定理 10. 对于任意 $U(T)<1$ 的 TCDRT 任务集 T,若 T 中任务图上每个节点关联的约束个数均为常数,则可调度分析算法是伪多项式时间的,且计算复杂度是约束宽度的线性函数.

证明. 根据定理 4 可知,在任务图中每个节点关联的约束个数为常数时,任务集 T 上 DBF 计算问题是伪多项式时间的. 又因为,在可调度性分析算法中,计算 DBF 的次数被限定在 T 内. 根据定理 9,可调度性分析上界 T 是伪多项式规模,为约束宽度的线性函数. 综上可知,可调度性分析算法是伪多项式时间的. 证毕.

另外,定理 9 结合定理 5 容易得出如下推论.

定理 11. 对于任意 $U(T)<1$ 的 TCDRT 任务集 T,若时间约束集合 $C(\tau)$ 中任意两约束的右节点均不相同,则可调度分析算法是伪多项式时间的,且计算复杂度是约束宽度的线性函数.

证明. 根据定理 5 可知,$C(\tau)$ 中任意两约束的右节点均不相同时,任务集 T 上 DBF 计算问题是伪多项式时间的. 又因为,在可调度性分析算法中,计算 DBF 的次数被限定在 T 内. 根据定理 9,可调度性分析上界 T 是伪多项式规模,为约束宽度的线性函数. 综上可知,可调度性分析算法是伪多项式时间的. 证毕.

定理 10 和定理 11 各给出了一类易解的 TCDRT 模型. 注意到,之前研究给出的结论是:任务图中时间约束总数为常数的 TCDRT 模型是易解的[7]. 而定理 10 中将这个限制条件进行了松弛,仅需任务图中每个节点关联的约束个数为常数即可保证 TCDRT 的易解性. 另外,定理 11 中的限定条件则直接与约束个数无关,而是只与时间约束设定的位置有关.

5 总 结

本文研究了带时间约束的实时任务图(TCDRT)模型,为 TCDRT 模型提出了新的可调度性分析算法. 与之前的研究相比,本文方法去除了一类问题参数——约束宽度对算法性能的影响,将算法复杂度由约束宽度的指数函数降低为线性函数. 另外,之前的研究指出,任务图中约束个数为常数的 TCDRT 是易解模型. 在本文中,该结论被扩展到更一般的 TCDRT 模型:仅需任务图中每个节点关联的约束个数为常数,即可保证 TCDRT 模型满足易解性. 更进一步地,本文还提出了一类新的 TCDRT 易解模型. 在新模型中,时间约束个数不再受到局限,易解性只受到约束在图中位置的影响. 该类模型可视为继文献[7]发现 K-TCDRT 之后,发现的第二类 TCDRT 易解模型.

参 考 文 献

[1] Liu C L, Layland J W. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM, 1973, 20(1): 46-61

[2] Alur R, Dill D L. A theory of timed automata. Theoretical Computer Science, 1994, 126(2): 183-235

[3] Tan Guo-Zhen, Sun Jing-Hao, Wang Bao-Cai, et al. Solving Chinese postman problem on time varying network with timed automata. Journal of Software, 2011, 22(6): 1267-1280(in Chinese)
(谭国真, 孙景昊, 王宝财等. 时变网络中国邮路问题的时间自动机模型. 软件学报, 2011, 22(6): 1267-1280)

[4] Fersman E, Krcal P, Pettersson P, et al. Task automata: Schedulability, decidability and undecidability. Information and Computation, 2007, 205(8): 1149-1172

[5] Stigge M, Wang Yi. Models for real-time workload: A survey// Proceedings of the Conference Organized in Celebration of Professor Alan Burns' Sixtieth Birthday. Heslington, UK, 2013: 133-160

[6] Baruah S. The non-cyclic recurring real-time task model// Proceedings of the 31st IEEE Real-Time Systems Symposium (RTSS). San Diego, USA, 2010: 173-182

[7] Stigge M, Ekberg P, Guan Nan, et al. On the tractability of digraph-based task models//Proceedings of the 23rd Euromicro Conference on Real-Time Systems (ECRTS). Porto, Portugal, 2011: 162-171

[8] Ka A, Mok L. Fundamental Design Problems of Distributed Systems for the Hard-Real-Time Environment [Ph. D. dissertation]. Massachusetts Institute of Technology, USA, 1983

[9] Baruah S K, Rosier L E, Howell R R. Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. Real-Time Systems, 1990, 2(4): 301-324

[10] Zhang F, Burns A. Schedulability analysis for real-time systems with EDF scheduling. IEEE Transactions on Computers, 2009, 58(9): 1250-1258

[11] Baruah S, Chen D, Gorinsky S, et al. Generalized multiframe tasks. Real-Time Systems, 1999, 17(1): 5-22

[12] Stigge M, Ekberg P, Guan Nan, et al. The digraph real-time task model//Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). Chicago, USA, 2011: 71-80

[13] Baruah S K. A general model for recurring real-time tasks//Proceedings of the 19th IEEE Real-Time Systems Symposium (RTSS). Madrid, Spain, 1998: 114-122

[14] Baruah S. Preemptive uniprocessor scheduling of non-cyclic GMF task systems//Proceedings of the 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). Macau, China, 2010: 195-202

[15] Chakraborty S, Erlebach T, Thiele L. On the complexity of scheduling conditional real-time code//Dehne F, Sack J-R, Tamassia R, eds. Algorithms and Data Structures. Lecture Notes in Computer Science 2125. Berlin Heidelberg: Springer, 2001: 38-49

[16] Sun Jing-Hao, Deng Qing-Xu, Meng Ya-Kun. Two-stage workload scheduling problem on GPU architectures: Formulation and approximation algorithm. Journal of Software, 2014, 25(2): 298-313(in Chinese)
(孙景昊，邓庆绪，孟亚坤. GPU 上两阶段负载调度问题的建模与近似算法. 软件学报，2014，25(2)：298-313)

[17] Xia Jia-Li, Chen Hui, Yang Bing. A real-time task scheduling algorithm based on dynamic priority. Chinese Journal of Computers, 2012, 35(12): 2685-2695(in Chinese)
(夏家莉，陈辉，杨兵. 一种动态优先级实时任务调度算法. 计算机学报，2012，35(12)：2685-2695)

[18] Li Qi, Ba Wei. Two improved EDF dynamic scheduling algorithms in soft real-time systems. Chinese Journal of Computers, 2011, 34(5): 943-950(in Chinese)
(李琦，巴巍. 两种改进的 EDF 软实时动态调度算法. 计算机学报，2011，34(5)：943-950)

[19] Dertouzos M L. Control Robotics: The procedural control of physical processes//Proceedings of the International Federation for Information Processing. Stockholm, Sweden, 1974: 807-813

SUN Jing-Hao, born in 1985, Ph.D., lecturer. His research interests include real-time scheduling, optmization and timed automata.

GUAN Nan, born in 1981, Ph. D., professor. His research interests include real-time system and embeded system.

DENG Qing-Xu, born in 1970, Ph. D., professor. His research interests include embeded real-time system, reconfigurable computing and cyber-physical system.

Background

Nowadays real-time embedded systems are designed for specific control functions in more and more domains ranging from portable devices such as smartphones, to large stationary installations like traffic lights, to systems controlling nuclear power plants. Especially, we witness the rapid development and deployment of cyber-physical system (CPS) such as autonomous vehicle and smart power grid, which dramatically propels this growth, and the demand for embedded systems will expand rapidly in the very near future. Contrary to non-timed systems, the real-time systems are time constrained and often safety critical. Indeed, the execution of such a system must produce the output not only correctly but also on time.

In order to ensure systems' high credibility, many formal models have been proposed, and with these formal descriptions, real-time systems may be successfully designed and analyzed. Almost all of the researches have been published in the well-known conferences and journals, such as RTSS, RTAS, ECRTS and Real-time systems, etc. In the very beginning, many classical results were exhibited for several simple linear task models. Most recently, the scholars pay more attention to more complex digraph-based models. For example, the directed acyclic graph (DAG) based non-cyclic recurring real-time task model (2010) and the digraph real time task (DRT) model (2011), etc. The digraph real-time task model with timing constraints (TCDRT) is the most expressiveness model conrrently known in real-time community, but its corresponding schedulability analysis is strongly NP-hard problem, which precludes the existence of any pseudo-polynomial algorithms. Present researches only focus on a special instance of TCDRT model, where the number of constaints is a constant K, and the only known analysis method transforms the TCDRT model into a corresponding DRT model, which leads to a high complexity that is exponential in the width of the constaints.

This work analyzes the schedulability of the TCDRT model directly in order to achieve a much lower complexity. The computation complexity result is refined to a better bound that has no relation with the width of constraints. Furthermore, our approach also indicates another tractable TCDRT model that is not necessary to postulate the *K*-bounded constraints.

This work is partly supported by the National Basic Research Program (973 Program) of China under Grant No. 2014CB360509, the National Natural Science Foundation of China under Grant Nos. 61300022, 61300194, 61472072, the Fundamental Research Funds for the Central Universities Nos. N130423007, N130504008, and the Natural Science Foundation of Hebei Province of China under Grant No. F2013501048.