

# 分布式约束优化问题研究及其进展

李小玲<sup>1)</sup> 王怀民<sup>1)</sup> 郭长国<sup>2)</sup> 丁 博<sup>1)</sup> 李小勇<sup>1)</sup>

<sup>1)</sup>(国防科学技术大学计算机学院 长沙 410073)

<sup>2)</sup>(中国电子设备系统工程公司 北京 100039)

**摘 要** 多 Agent 协作过程中的许多问题都可以被抽象为分布式约束优化问题(DCOP),如规划、行程安排、分布式控制和资源分配等.这些问题关注于如何通过协调多 Agent 之间的相互决定,以达到一个全局最优决策的目的.相应地,分布式约束优化算法是用来求解此类问题的一种有效方式.该文对分布式约束优化问题进行了综述,首先,阐述了分布式约束优化问题的基本概念,并提出了一种分布式约束优化算法的分类框架.其次,根据该分类框架,介绍了目前已有的分布式约束优化算法,并加以对比分析.此外,分析了分布式约束优化问题的相关应用.最后,指明了分布式约束优化领域的未来研究趋势.

**关键词** 分布式约束优化; Agent; 人工智能; 分布式约束优化算法

中图法分类号 TP301 DOI号 10.11897/SP.J.1016.2015.01656

## Research and Development of Distributed Constraint Optimization Problems

LI Xiao-Ling<sup>1)</sup> WANG Huai-Min<sup>1)</sup> GUO Chang-Guo<sup>2)</sup> DING Bo<sup>1)</sup> LI Xiao-Yong<sup>1)</sup>

<sup>1)</sup>(School of Computer Science, National University of Defense Technology, Changsha 410073)

<sup>2)</sup>(Chinese Electronic Equipment and System Engineering Corporation, Beijing 100039)

**Abstract** Many problems in multi-agent coordination can be abstracted as distributed constraint optimization problem (DCOP), such as planning, scheduling, distributed control, resource allocation, etc. These problems focus on how to coordinate Agents' co-decisions to achieve the aim of global optimality decision, and DCOP algorithm is an effective way to solve these problems. This paper surveys the DCOP, firstly, the basic concepts of DCOP are discussed and a classification framework of DCOP algorithm is proposed. Secondly, according to this framework, the existing various DCOP algorithms are introduced and compared. Thirdly, the related applications of DCOP are introduced. Lastly, the future research trends of DCOP are reviewed.

**Keywords** distributed constraint optimization; Agent; artificial intelligence; distributed constraint optimization algorithm

## 1 引 言

云计算是随着计算、存储以及通信技术的快速

发展而出现的一种崭新的共享基础设施资源的商业计算模型.在云计算中,用户/租户(users/tenants)通过租用云提供商(Cloud Providers,CP)提供的基础设施资源完成其应用请求(WorKload,WK).一

收稿日期:2013-06-17;最终修改稿收到日期:2014-12-28.本课题得到国家“九七三”重点基础研究发展规划项目基金(2011CB3022601)、国家自然科学基金重大项目(90818028,6903043,61202117)、国家“八六三”高技术研究发展计划项目基金(2012AA011201)资助.

李小玲,男,1985年生,博士,中国计算机学会(CCF)会员,主要研究方向为分布式计算、可信软件、数据中心网络、操作系统. E-mail: nudtlixl@163.com. 王怀民,男,1962年生,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为分布计算、可信软件、网络安全. 郭长国,男,1973年生,博士,副研究员,中国计算机学会(CCF)会员,主要研究方向为分布计算、可信软件. 丁 博,男,1978年生,博士,主要研究方向为分布计算、普适计算、软件自适应. 李小勇,男,1982年生,博士,中国计算机学会(CCF)会员,主要研究方向为分布计算、网络距离预测、数据库.

一般来说, CP 希望能够尽可能地提高其资源利用率以谋求自身利益的最大化. 然而, 在很多情况下, 用户/租户可能需要同时租用多个 CP 的资源来完成其 WK 请求, 不同的 CP 由于资源的有限以及相互之间需要互相通信等而使它们之间存在一定的约束. 不同 CP 出于对自身收益最大化的目的, 在分配资源给 WK 时并未考虑它们之间的约束, 使得其分配给用户/租户的资源并非是最合理的, 导致不同 CP 之间的资源分配方式存在一定的冲突(违背了相关的约束), 最终使得全局的资源利用率较低. 我们称将 WK 部署在 CP 提供的基础设施资源上的问题为云计算中的资源分配问题(Resource Allocation, RA)<sup>[1-2]</sup>, 不同 CP 之间在 RA 中因为共同服务于同一 WK 而导致它们在分配资源时存在一定的约束, 如它们之间互相通信等. 如何有效地解决 RA 问题并且使得全局资源利用率最大化是一个非常具有挑战性的工作.

约束是事物之间相互制约\冲突关系的数学建模<sup>[3]</sup>. 约束满足问题(Constraint Satisfaction Problems, CSPs)<sup>[4]</sup>的目标是找到一个“和谐”的解, 即满足问题中所有约束条件的解. 自 Montanari 等人<sup>[5]</sup>首次在图像处理问题中提出了 CSPs 以来, CSPs 作为一种重要的求解方法在人工智能与计算机科学等领域得到了广泛的应用<sup>[6]</sup>, 从 n 皇后、图染色等经典问题到时序安排、计划编制、资源分配等问题, 都可以通过建模成 CSPs 来求解<sup>[4]</sup>.

通过将 RA 建模成 CSPs 问题进行求解, 能够找到一个满足 WK 约束的资源分配方式, 即为一个约束满足解, 但 CSPs 并未考虑解的质量问题, 解的质量在 RA 中指部署 WK 所付出的“代价”, 即部署该 WK 所需要的物理资源的数量. 然而, 在很多情况下, CP 往往希望部署一个 WK 所花费的代价最小, 即分配给该 WK 的资源最少, 在这种提出了最优性要求时, CSPs 则无法适用. 约束优化问题(Constraint Optimization Problems, COPs)<sup>[7-8]</sup>在 CSPs 的基础上进一步提出了最优性的要求. 分布式约束优化问题(Distributed Constraint Optimization Problem, DCOP)是分布式的 COPs, 它由一系列独立的 COPs 组成, 其中每个 COPs 由一个 Agent 所拥有, 不同的 Agent 之间因为一些共同的约束而构成一个整体, 它能够有效地用于求解 RA 问题, 同时能够保证得到的解是全局最优的.

通过将 RA 建模成 DCOP 问题以求解最优分配方案, RA 问题分布于多 Agent 中, 其中每个 CP

即为一个 Agent, 它是一个自治体, Agent 之间相互独立, 分别负责管理自己的内部问题. 每个 Agent 又是智能的, 能够根据环境的不同做出有利的选择. DCOP 的目标就是通过协调多个 Agent 之间的相互决定, 使得整体的利益最大化, 从而达到一个全局最优的目的. 除了 RA 问题外, 现实应用中的许多问题也都可以通过建模成 DCOP 问题进行求解, 如资源调度和任务分配、协同决策问题<sup>[9]</sup>、组合拍卖问题<sup>[10]</sup>、Web 服务组合问题<sup>[11]</sup>、传感器分配问题<sup>[12]</sup>等.

DCOP 是 NP 难问题<sup>[4]</sup>, 对其进行求解是一件既非常困难又非常具有挑战性的工作. 分布式约束优化求解算法(DCOP)是目前能够有效求解 DCOP 问题的算法, 不同算法的主要区别在于求解 DCOP 时所花费的代价以及求得解的质量不同, 如算法时间、空间复杂度以及解是否最优等, 在寻找最优解过程中为了使代价尽可能最小, 研究者们纷纷提出了各种用于求解 DCOP 问题的算法.

近年来重点关注 DCOP 的国际顶级会议有 AAAI (American Association for AI National Conference)、AAMAS(Intl. Conf. on Autonomous Agents and Multi-Agent Systems)、IJCAI (Intl. Joint Conf. on AI)等. DCOP 的主要研究团队有日本九州大学的 Yokoo<sup>[13-14]</sup>、瑞士洛桑理工学院的 Boi Faltings 以及以色列本古里安大学等.

本文以如何有效地对 DCOP 进行求解为目标, 对 DCOP 进行了综述. 本文第 2 节给出 DCOP 的相关形式化描述, 并且介绍如何将常见的经典问题建模为 DCOP; 第 3 节提出 DCOP 算法分类框架; 第 4 节介绍现有的典型 DCOP 算法, 且对比不同算法之间的异同性; 第 5 节分析 DCOP 问题相关的典型应用; 最后, 分析 DCOP 领域未来的发展趋势并总结全文.

## 2 分布式约束优化问题描述

为了更深入地了解分布式约束优化问题及其特点, 我们将首先介绍分布式约束优化问题的定义以及相关应用实例.

### 2.1 分布式约束优化问题定义

**定义 1.** 分布式约束优化问题通常被定义为五元组  $\langle X, D, F, A, m \rangle$ , 其中  $X = \{x_1, \dots, x_n\}$  是变量的集合,  $D = \{D_1, \dots, D_n\}$  是  $X$  中变量所对应的值域集合,  $F = \{f_1, \dots, f_m\}$  是约束函数代价的集合,  $A = \{A_1, \dots, A_k\}$  是 Agent 的集合,  $m$  表示 Agent

与变量之间的管理关系,它是  $A \rightarrow \mathcal{X}$  的一个双射.

其中  $\mathcal{X}$  是变量集合  $X$  的一个划分,表示每个 Agent 所对应的内部变量集合,例如对于 Agent-1,存在变量  $(x_{11}, \dots, x_{1k})$  属于其内部变量. 约束代价函数是指定义违反约束条件所带来的代价,一般包括域内约束和域间约束两种,前者是指同一 Agent 内变量之间的约束,而后者是多个 Agent 之间变量的约束. 同样,DCOP 中的硬约束条件,对于可能的组合赋值为 0,不可能的组合赋值为  $\infty$ .

**定义 2.** DCOP 的目标是通过协商多 Agent 之间的相互决策,对变量进行合理赋值,使得所有违反的约束代价和最小. 其解的形式化的表示如下式(1)所示.

$$X^* = \arg \left\{ \min_X \sum_{f_i \in F} f_i(X) \right\} \quad (1)$$

与 COP 的最大区别在于问题分布于多个 Agent 之中,且它们是相互独立的自治体,Agent 之间只能通过消息传递方式进行交互.

## 2.2 具体问题的 DCOP 建模

多 Agent 领域中的许多真实问题都可以建模成 DCOP 问题,例如 Petcu 等人<sup>[10]</sup>对多 Agent 组合拍卖以及覆盖网优化等问题用 DCOP 进行建模, Maheswaran 等人<sup>[15]</sup>将分布式多事件调度建模为 DCOP 问题进行求解, Sultanik 等人<sup>[16]</sup>实现了多 Agent 任务调度语言 C-TEMS 到 DCOP 的自动转换, Zhang 等人<sup>[12]</sup>给出了移动传感器网络中的 DCOP 的应用等. 下面我们给出 DCOP 的一个具体问题的建模过程,通过它能够让我们更进一步了解 DCOP.

**实例 1.** 分布式组合拍卖问题 (Distributed Combinatorial Auction, DCA) 是多 Agent 领域中资源分配的一种常见方式,它允许不同的报价者对不同的货品组合进行竞价,在货品数量有限的情况下,其目标是如何分配这些货品使得货品拥有者获得的收益最大. DCA 可被定义为三元组  $\langle A, G, B \rangle$ , 其中  $A = \{A_1, \dots, A_k\}$  表示报价者 (Agent) 集合,  $G = \{g_1, \dots, g_n\}$  是所有的货品集合,  $B = \{b_1, \dots, b_k\}$  是所有的报价单集合.

报价单  $b_k$  表示为三元组  $\langle A_i, G_k^i, v_k^i \rangle$ , 其中  $G_k^i$  表示报价者  $A_i$  报价的货品集合,  $v_k^i$  表示  $A_i$  是否获得其所报价的货品  $G_k^i$ . 若  $A_i$  没有获得报价的所有货品,  $v_k^i$  为 1, 否则为 0.

对于  $\forall b_k^i, b_l^m$ , 如果存在货品满足  $g_j \in G$ , 且  $g_j \in G_k^i \wedge g_j \in G_l^m$ , 在这种情况下  $b_k^i$  和  $b_l^m$  至少有一个为

0, 即保证同一货品不能同时被两个不同的报价者获得. 图 1 列出了分布式组合拍卖问题建模成 DCOP 的模型.

如图 1 所示,有 3 个报价者  $\{A_1, A_2, A_3\}$  以及 3 件货品  $\{g_1, g_2, g_3\}$ ,  $A_1$  对货品  $g_1, g_3$  进行报价;  $A_2$  对货品  $g_1, g_2, g_3$  报价;  $A_3$  对货品  $g_2, g_3$  报价. 在将该问题建模成 DCOP 时, Agent 集合为出价者  $\{A_1, A_2, A_3\}$ , 变量集合  $X$  为  $\{g_1^1, g_3^1, g_2^2, g_3^2, g_2^3, g_3^3\}$ , 其中  $g_i^k$  表示报价者  $A_k$  对货品  $g_i$  进行了报价, 值域集合  $D$  为  $\{0, 1\}$ , 其中 1 表示出价者获得该商品, 否则没获得该商品. 映射关系  $M$  如下,  $A_1 \rightarrow \{g_1^1, g_3^1\}$ ,  $A_2 \rightarrow \{g_1^2, g_2^2, g_3^2\}$ ,  $A_3 \rightarrow \{g_2^3, g_3^3\}$ , 分别表示出价者所报价的商品集合.  $F$  表示效益函数集合, 表示一组商品所能够带来的效益, 如对于组合  $\langle g_1^1, g_3^1 \rangle$ , 当其值为  $\langle 1, 1 \rangle$  时, 其效益为 10, 而对于其他的取值, 效益值为 0. 该问题中存在约束是不同的出价者对同一货品进行了报价, 最多只有一个报价者能够获得货品, 其目标是找到一个使得货品拥有者可获得的收益最大的分配方案.

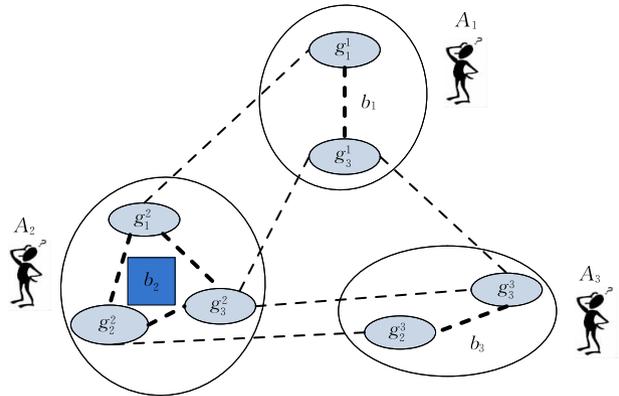


图 1 组合拍卖的分布式约束优化模型

通过对上述实例的建模,我们知道一个现实应用中的问题能够建模成 DCOP 进行求解必须具备以下几个特性:(1)问题分布于多 Agent 之间,在现实应用中,即问题中涉及到多个角色(如 RA 中的云提供商、组合拍卖中的拍卖者)的参与;(2)各个不同的参与者之间存在一定的约束条件(如组合拍卖中货品的数量有限,不同的拍卖者对其进行竞争);(3)Agent 间需要互相协作完成决策,在求解 DCOP 时 Agent 间需要通过大量的消息传递最终达成一致;(4)问题求解中有确定的目标函数(如组合拍卖问题中,如何使得货品拥有者的获益最大),并且要求寻找到最优解. 在满足以上 4 个特性的绝大部分现实应用中的问题都可以考虑建模为 DCOP 问题

进行求解。

### 3 分布式约束优化算法分类

本文通过对现有的分布式约束优化求解算法(DCOP)进行研究分析,提出了一种 DCOP 算法分类

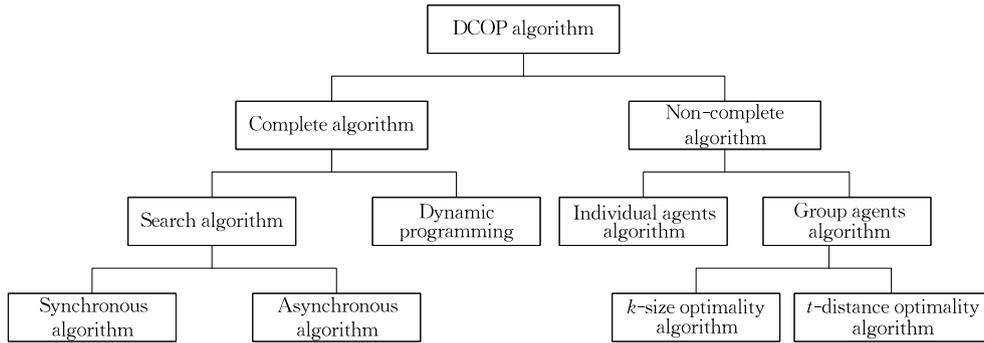


图 2 DCOP 算法分类框架

在完全算法中,根据求解的方式不同,分为基于搜索/回溯的算法(Search algorithm)以及基于动态规划的算法(Dynamic programming)。搜索算法中根据 Agent 之间的通信方式,又可以分为同步搜索算法(Synchronous algorithm)以及异步搜索算法(Asynchronous algorithm)两种,两者的本质区别在于 Agent 之间的通信是否同步。同步算法中 Agent 是同步操作的,而异步算法中 Agent 则是异步工作的。

对于非完全算法,可以分为 Agent 个体决策算法(Individual agents algorithm)和 Agent 组决策的算法(Group agents algorithm)<sup>[17]</sup>。其中,Agent 个体决策算法通过对 Agent 从其他 Agent(相邻的)处收集的信息,对 Agent 自身的决策过程等进行简化,从而加快求解速度;而 Agent 组决策算法是通过若干个 Agent 组成本地组进行决策,而非 Agent 本身单独进行决策。组决策算法又可以分为 *k*-size 最优算法和 *t*-distance 最优算法。

#### 3.1 完全算法

搜索算法是基于分枝定界思想对状态空间进行搜索求解,搜索算法的效率取决于剪枝的效率,算法的剪枝效率高,可以大大地避免对于非解空间的搜索,从而减小算法的时间和空间复杂度。基于动态规划的算法基于桶消元(bucket-elimination)的思想进行求解,根据问题中涉及到约束关系,对变量逐一进行消元,不断简化问题,直到找到解为止。

##### 3.1.1 基于搜索的算法

目前,大部分的 DCOP 求解算法都是基于搜索算法<sup>[18-20]</sup>。基于搜索的算法首先将所有的变量按照

框架。如图 2 所示,根据算法是否对得到的解提供质量保证,可分为完全算法(Complete algorithm)和非完全算法(Non-complete algorithm)。两者的区别在于,在前者中,如果问题存在最优解,那么其保证一定能够找到最优解,而后者则不保证一定能够找到最优解。

一定的结构组织起来,如线性序列<sup>[21]</sup>、任意序列或者 DFS 结构<sup>[22]</sup>。然后在这些结构上结合分枝定界思想进行遍历搜索求解。根据 Agent 之间消息传递是否同步,可分为同步搜索算法和异步搜索算法。

##### (1) 同步搜索算法

Agent 在发送一个消息后必须要在收到相应的回复消息后,才能继续后续的工作,如计算、发送消息等。中间过程中 Agent 一直处于空闲等待状态,而且由于该原因,导致同步搜索算法效率较低。目前常见的同步搜索算法有 SBB<sup>[21]</sup>、dAO-Opt<sup>[10]</sup>、dAOBB<sup>[10]</sup>、dAOBB(*i*)<sup>[10]</sup>等。

##### (2) 异步搜索算法

允许 Agent 在收到回复消息前,继续后面的工作,不需要处于空闲等待状态。相比于同步搜索算法,它的优点是 Agent 不会长时间处于空闲等待状态,一定程度上提高了算法效率,但该类算法往往会因为 Agent 之间的频繁交互产生大量的消息,通信开销会比较大。目前常见的算法有 ADOPT<sup>[9]</sup>、NCBB<sup>[23]</sup>、BnB-ADOPT<sup>[24]</sup>、IDB-ADOPT<sup>[25]</sup>、OptAPO<sup>[26]</sup>、AFB<sup>[27]</sup>等。

搜索算法的优点在于其所需的内存空间为多项式级别,不足之处是会产生大量的消息,导致通信开销较大,尤其是异步搜索算法。如果拥有足够大的内存空间,可以采用 Cache 机制将算法 dAOBB 和 NCBB 扩展为 dAOBB(*i*)和 NCBB(*i*),这能在一定程度上提高算法的效率。

##### 3.1.2 基于动态规划的算法

基于动态规划的 DCOP 算法的基本思想是在

DFS 结构的基础上进行变量消元<sup>[28]</sup>. 基于动态规划的算法在人工智能领域也可以将其归为基于推理的方法, 随着其在 COP 方面的应用, 已经被逐渐地扩展为与其他技术结合使用, 例如与分枝限界估测以及与搜索方法结合等<sup>[29-32]</sup>. 目前常见的算法有 DPOP<sup>[33]</sup> 和 H-DPOP<sup>[34]</sup> 等.

### 3.2 非完全算法

非完全算法(近似算法)无法保证一定能找到最优解, 即使问题存在最优解, 因为该类算法只对部分状态空间进行搜索. 完全算法的时间复杂度往往非常高, 在很多大规模的现实应用中, 求解最优解非常难、代价巨大, 而且有时根据应用需求不需要寻找最优解, 只需要找到一个近似最优解即可. 非完全算法则能够很好地满足这种应用的需求, 花费较小的代价找到一个近似最优解. 其中, DCOP 非完全算法分为 Agent 个体决策的算法和 Agent 组决策的算法.

#### 3.2.1 基于 Agent 个体决策的算法

Agent 个体决策算法主要包括 MGM/DBA (Maximum Gain Message/Distributed Breakout Algorithm)<sup>[35-36]</sup>、ALS\_DisCOP<sup>[37]</sup>、DSA (Distributed Stochastic Algorithm)<sup>[38]</sup> 等, 这类算法通常不为解的质量提供任何保证. 然而在很多的实际应用中关于解的质量问题是一个很重要的考虑因素, 导致这类算法的应用范围具有很大的局限性. 但 2009 年提出的 max-sum<sup>[39]</sup> 算法是一个例外, 针对某些特定问题实例, 它可以提供质量保证, 实验表明其在 95% 情况下可以直接得到最优解. 此类算法可以对解的质量提供一定的保证.

#### 3.2.2 基于 Agent 组决策的算法

Agent 组决策的算法中, Agent 以组的形式去寻找其各自取值, 同一组内的 Agent 通过协调它们之间的取值, 来达到一个局部最优的结果, 主要代表性算法有 KOPT<sup>[40]</sup> 和 DALO<sup>[17]</sup> 算法等. 在组决策算法中, 根据如何将 Agent 集合构成不同的组, 又可以分为  $k$ -size 最优算法和  $t$ -distance 最优算法.

在下面的章节中, 我们将详细对 DCOP 求解算法进行介绍, 包括完全算法和非完全算法, 并且对相关算法进行了对比分析.

## 4 分布式约束优化问题完全算法

如图 2 所示算法分类框架, 本节对每一类中比较常见的 DCOP 完全算法进行了研究. DCOP 算法

在求解 DCOP 时, 首先需要将问题中所涉及的变量按一定的结构组织起来, 如线性序列<sup>[21]</sup>、任意序列、伪树 (Pseudo-Tree) 结构<sup>[10]</sup> 或者 DFS (Depth-First Search Tree) 结构<sup>[22]</sup>, 目前绝大多数 DCOP 算法都是基于 DFS 结构的. 此外, 还有一些研究是基于混合结构进行求解的<sup>[41-42]</sup>.

下面首先介绍 DFS 结构, 然后再分别介绍各类中比较常见的 DCOP 算法. 最后, 对各种算法的复杂性进行对比分析, 有利于我们横向地比较各种算法的优缺点以及适应范围.

### 4.1 DFS 结构

目前大部分的 DCOP 算法都是基于 DFS 结构(伪树结构的一种特例), 但也存在一些是基于随机序列的, 比较典型的有 Yokoo 等人<sup>[43-44]</sup> 提出的 ABT 算法和 AWC 算法, Silaghi 等人<sup>[45]</sup> 提出的 AAS 算法, Gershman 等人<sup>[27,46]</sup> 提出的 AFB 算法等. 也有一些算法是基于顺序序列的, 如 Hirayama 等人<sup>[21]</sup> 提出的 SBB 算法. 下面我们将介绍伪树结构以及 DFS 结构的相关概念.

**定义 3.** 约束图  $G$  所对应的伪树 (Pseudo-Tree) 结构, 根节点是图  $G$  中预先选定的一个节点, 并且图  $G$  中的相邻节点必定处于同一分枝中.

自 Freuder 等人<sup>[22]</sup> 最早将伪树结构引入到约束优化问题中, 从而得到了广泛的应用<sup>[47-49]</sup>. DFS 树可以看作是伪树的一种特例, 其所有的边都属于约束图(即 DFS 中边的数目与约束图一样). 我们可知因为其特有属性, 使得 DFS 非常适合于进行深度优先搜索遍历.

**定义 4.** DFS 结构通常定义如下, 图  $G$  对应的 DFS 结构与约束图  $G$  的节点的数目和边的数目都相同, 并且相邻的节点必定处于同一分枝中.

图 3 为一个简单的约束图  $G$  转换成伪树结构以及 DFS 结构的例子. 由图 3 可知, 对于伪树来说, 允许树的边数目多于原图  $G$  的边数, 图 3(b) 中需要添加两条虚边  $X_4 - X_2$  和  $X_4 - X_6$ , 而图 3(c) 中只包含原图  $G$  中的边.

DFS 结构对于 DCOP 算法的求解效率有一定的影响. 在搜索的算法中, 算法的时间复杂性是 DFS 深度的指数级, 而基于动态规划算法, 时间和空间复杂度都是 DFS 宽度的指数级. 因此, DFS 结构的好坏对于算法的效率有一定的影响. 显然, 对于同一个图  $G$ , 其对应的 DFS 结构并不是唯一的. 目前生成 DFS 结构的算法和策略较多, Chechetka 等

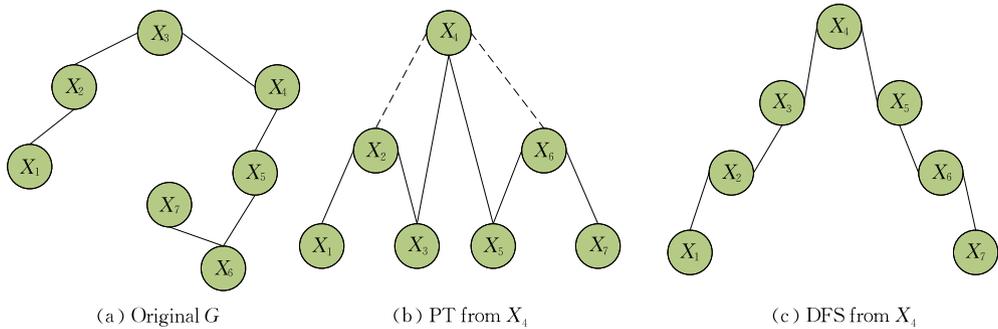


图 3 一个简单问题的伪树结构以及 DFS 结构示例

人<sup>[50]</sup>提出了第 1 个通过使用启发式算法去尽量地降低 DFS 的深度. 对于减小 DFS 宽度方面, 也有较多的研究, 如采用 Maximum Cardinality Set<sup>[51]</sup>、Maximum Degree<sup>[51]</sup> 以及 Min-Fill 策略<sup>[52]</sup> 等都能够一定程度上减小 DFS 的宽度. 对于不同的 DCOP 算法, 根据具体应用的特点, 构造最合适的 DFS 结构是非常重要的.

## 4.2 同步搜索算法

下面我们将从基于线性序列和基于 DFS 结构两方面来介绍 DCOP 算法.

### 4.2.1 基于线性序列结构

基于线性序列结构的同步搜索算法是指在进行搜索之前, 所有的变量按照线性序列进行排列, 然后在此序列上按照一定的搜索策略进行搜索求解.

Hirayamad 等人<sup>[21]</sup>于 1997 年提出了第 1 个 DCOP 完全算法 SBB(Synchronous Branch and Bound), SBB 算法主要是用于解决 DCSP, 有时也可用于求解 DCOP. 其思想很简单, 是经典的分枝定界算法的分布式版本. 在 SBB 算法中, 所有的变量事先都按照线性序列排列好(按照一定的优先级排列, 一般按从高到低的顺序排列), 搜索时按顺序依次对变量进行赋值, 在遇到冲突时, 改变变量赋值, 必要时进行回溯.

SBB 算法在问题规模较大, 包含的变量较多时, 效率一般较低. 因此, SBB 一般只用于解决特定的问题, 使用范围有限. Yokoo 等人<sup>[53]</sup>提出的异步回溯算法 ABT(Asynchronous BackTracking)算法也是基于线性序列的, 但其一般也只用于对约束满足问题进行求解, 对于约束优化问题求解的范围有限. Gershman 等人<sup>[54]</sup>提出的 AFB(Asynchronous Forward Bounding)算法也是基于线性序列结构的, 但其采用的是“同步-异步”混合方式, 在搜索过程中利用异步前瞻(Look Ahead)技术, 减少了一些不必要的搜索.

### 4.2.2 基于 DFS 结构

Nilsson 等人提出的 AND/OR 搜索空间在搜索领域有一定的影响力, 并且已被应用于诸多领域中<sup>[22,48,55]</sup>. Dechter 等人<sup>[49]</sup>通过 AND/OR 图捕获一般的图形模型的搜索空间, 包括约束网络以及信任网络. 在 AND/OR 搜索空间进行搜索求解的效率显然要比在线性序列上求解高, 因为在搜索过程中, 可以对 AND/OR 空间中不同分枝同时进行搜索是其关键所在.

#### (1) dAO-Opt

Collin 等人<sup>[48]</sup>于 1999 年提出了 dAO-Opt 算法, dAO-Opt 是一个同步 AND/OR 搜索算法的扩展, 在 DFS 结构上以深度优先搜索的策略进行分布式搜索. DFS 结构中不同分枝可以同时进行搜索, 这在一定程度上加快了搜索的效率. dAO-Opt 算法引入了局部代价(Local Cost)的概念, 局部代价是指搜索过程中部分解所产生的代价. 在搜索从根节点到叶子节点过程中, 叶子节点接受的消息中将包含所有可能赋值的局部代价, 叶子节点从中选择最好情况的代价, 并返回给其父节点. dAO-Opt 算法的代价在最坏情况下是 DFS 深度的指数级, 相比于线性序列是所有变量的指数级有较大的改善.

#### (2) dAOBB

Marinescu 等人<sup>[55]</sup>提出的 dAOBB(AND/OR Branch and Bound)算法, 针对的是 dAO-Opt 算法中没有考虑剪枝技术而导致需要对全局空间进行搜索的局限性. dAOBB 算法在 dAO-Opt 的基础上结合了剪枝技术的思想, 是 AOBB 算法的分布式版本, dAOBB 在 dAO-Opt 的基础上结合了分枝界限技术, 合理地使用剪枝技术可以避免对不可能求得解的状态空间进行搜索, 一定程度上提高了搜索的效率. 同样, dAOBB 算法也是同步搜索算法.

初始界限选择的好坏对于 dAOBB 算法的效率是有很大的影响的, 因而涌现出了很多与 dAOBB 算

法相关的改进,目的是尽可能地提高其效率.如 Faltings 等人<sup>[56]</sup>提出的 A-DPOP 算法,是一个采用了自适应 minibucket 策略的算法,A-DPOP 算法能够很好地与 dAOBB 算法结合,在界限选择方面能得到更好的结果.同时,Marinescu 等人<sup>[57]</sup>通过将 dAOBB 与 Cache 机制结合,提出了 dAOBB( $i$ )算法,其中  $i$  代表的是 Cache 的大小,采用 Cache 机制能够大大地加快算法的求解速度,效率大大提高.

### (3) HEDA

丁博等人<sup>[58]</sup>针对低约束密度的分布式约束优化问题,提出了一种基于贪婪和回跳思想的算法 HEDA.在该算法中,每个 Agent 基于贪婪原则进行决策,利用低约束密度问题中大多数赋值组合不存在冲突这一特点来加快求解速度,并在必要时通过贪婪求解过程的回跳来保证算法完全性.

## 4.3 异步搜索算法

针对同步搜索算法中,Agent 通常要花大量的时间等待其他 Agent 的消息,导致效率较低的问题,研究者们提出了各种异步搜索算法,有 ADOPT<sup>[9]</sup>、NCBB<sup>[23]</sup>、BnB-ADOPT<sup>[24]</sup>、IDB-ADOPT<sup>[25]</sup>、Opt-APO<sup>[26]</sup>、AFB<sup>[27]</sup>等.

### 4.3.1 ADOPT

Modi 等人<sup>[9]</sup>于 2005 年提出了 DCOP 的异步算法 ADOPT(Asynchronous Distributed Optimization),ADOPT 是 DCOP 领域中第 1 个完全异步算法. ADOPT 是在 DFS 结构上进行异步搜索操作的,根据 DFS 结构划分 Agent 的优先级关系(DFS 中祖先节点的优先级一般要高于子孙节点).该算法的创新之处主要有 3 点:(1)在搜索过程中,每个 Agent 仅仅根据当前已知的信息选择最优的取值,而非全局信息进行选择;(2)通过回溯门槛(backtrack threshold)来有效地改变 Agent 的取值,这能够有效地帮助我们获得之前搜索过的取值;(3)算法能够在找到最优解时及时停止算法的运行,而不需要额外的算法结束机制.

ADOPT 在求解 DCOP 时有很多的好处,但因其计算过程中 Agent 之间自主地进行通信以及更改自身取值信息,每次更改都需要发送信息给其邻居节点,会导致消息数量太多,因此网络开销很大,同时还需要 DFS 宽度的指数级别的内存空间.

### 4.3.2 BnB-ADOPT

南加州大学的 William 等人<sup>[24]</sup>于 2008 年在 AAMAS 上提出了 BnB-ADOPT 算法. BnB-ADOPT

在 ADOPT 上做出了改进,与 ADOPT 算法最大的区别在于将搜索策略从 Best-First 改变为 Depth-First(深度优先)分枝界限搜索.算法 BnB-ADOPT 运行中的消息传递以及通信都是基于 ADOPT 的框架完成的.将搜索策略转变成 Depth-First 分枝界限搜索是因为 Best-First 搜索需要不断重复地构造部分解,这会导致很大的计算开销以及内存空间<sup>[59]</sup>,而 Depth-First 分枝界限搜索则不存在这个问题.

### 4.3.3 IDB-ADOPT

William 等人<sup>[21]</sup>于 2009 年提出的 IDB-ADOPT(Iterative Decreasing Bound ADOPT)算法也是在 ADOPT 算法上做的改进.与 BnB-ADOPT 算法一样,IDB-ADOPT 算法也是基于 Depth-First 搜索策略,不同之处在于 IDB-ADOPT 算法中的 Bound 是逐渐递减的.其基本思想如下,在搜索过程中,消息中携带着一个 Bound(初始值无穷大),每当找到一个解,且其代价小于 Bound 的值时,就用计算出的代价值替换 Bound 的原来值,这样 Bound 的取值就在不断减小,直到不存在一个解,满足其代价比当前 Bound 还小时,此时的解则是最优解.实验表明 IDB-ADOPT 算法在处理大型 DCOP 问题时的效率要比 ADOPT 高.

### 4.3.4 AFB

本古里安大学的 Gershman 等人<sup>[54]</sup>于 2006 年在 ECAI 上提出了 AFB 算法. AFB 与 SBB 算法一样,也是基于变量的线性序列进行搜索,不同之处在于 AFB 是一个“同步—异步”混合算法,将 SBB 算法和异步前瞻(Look Ahead)结合在一起,而 SBB 算法是同步算法.

### 4.3.5 OptAPO

Mailler 等人<sup>[26]</sup>提出的最优异步部分交叉算法 OptAPO(Optimal Asynchronous Partial Overlay)利用了称为调停者(Mediator)的 Agent.调停者在算法执行中动态产生,可以将部分变量及其约束集中起来. OptAPO 算法在调停者的干预下直接地通信约束,以局部地集中问题. OptAPO 允许 Agent 扩展和交叉它们在求解过程中用于本地决策的上下文.

上面介绍的这些搜索算法都是目前 DCOP 问题中比较常见的求解算法,为了进一步了解它们之间的主要区别,表 1 对比了算法中的一些相关属性,包括搜索策略、通信方式、消息传递方式以及所基于的拓扑结构.

表 1 DCOP 搜索算法的属性比较

算法	Search Strategy	Synchronization	Communication	Topology
SBB	Depth-first branch-and-bound	synchronous	Point-to-point with neighbors	Constraint chain
ADOPT	Best-first	asynchronous	Point-to-point with neighbors	Constraint tree
NCBB	Depth-first branch-and-bound	synchronous	Point-to-point with neighbors	Constraint tree
AFB	Depth-first branch-and-bound	asynchronous	broadcast	Constraint chain
BnB-ADOPT	Depth-first branch-and-bound	asynchronous	Point-to-point with neighbors	Constraint tree
dAO-Opt	Depth-first branch-and-bound	synchronous	Point-to-point with neighbors	Constraint tree
dAOBB	Depth-first branch-and-bound	synchronous	Point-to-point with neighbors	Constraint tree
IDB-ADOPT	Depth-first branch-and-bound	asynchronous	Point-to-point with neighbors	Constraint tree

#### 4.4 基于动态规划算法

动态规划(Dynamic Programming)<sup>[60-61]</sup>一直被认为是解决组合拍卖问题<sup>[62]</sup>的有效方法之一. 简单而言, 动态规划是通过逐一进行变量消元进行求解的, 消元过程中主要依据变量之间的约束关系. Dechter 等人提出了动态规划算法的通用框架 BE (Bucket Elimination), 框架中详细介绍了 BE 的消元求解原理. 而 Kask 等人提出了适用于集中式问题的动态规划算法 BTE(Bucket Tree Elimination), BTE 通过在 Bucket Tree 结构上进行消元求解. 以下将详细介绍各种典型的动态规划 DCOP 算法, 包括 DPOP、H-DPOP、M-DPOP、O-DPOP 等.

##### 4.4.1 DPOP

瑞士洛桑联邦理工学院人工智能实验室的 Petcu 等人<sup>[33]</sup>于 2005 年在 IJCAI 会议上提出了 DPOP 算法, 开创了基于动态规划策略在 DCOP 问题中的应用. DPOP 是一个基于动态规划的 DCOP 完全算法, 是集中式 Bucket Elimination 算法<sup>[63]</sup>运用于 DFS 结构的分布式版本. 下面将简单介绍 DPOP 算法的思想.

DPOP 算法大致可以分为 3 个阶段: DFS 构造阶段、UTIL 消息传播以及 VALUE 消息传播阶段, 具体过程包含以下 3 个阶段:

(1) DFS 构造阶段. DPOP 算法也是基于 DFS 结构的, 构造 DFS 的方法有很多, 3.1 节中已经详细介绍了有关 DFS 结构的内容;

(2) UTIL 消息传播阶段. 该阶段是自底向上传播的, 从叶子节点开始沿着伪树中的树边 (tree-edge) 开始向上传播. UTIL 消息的定义如下. UTIL 消息是一个多维矩阵, 其中每一维度代表的是一个变量, UTIL 消息中包括了所有变量取不同值时所带来的效用值;

(3) VALUE 消息传播阶段. 该阶段与 UTIL 消息传播阶段相反, 是自顶向下传播的. 在上一阶段完成之后, 从根节点发起. 每个节点根据收集到的 UTIL 消息, 选择使得效用值达到最大的那个值, 直

到所有的变量都有取值时, 算法结束.

分布式约束优化问题在过去的一些年里一直稳定快速地发展<sup>[64]</sup>. 动态规划方法对于求解 DCOP 问题有很大吸引力, 因为只需要线性级别的消息数量, 但是存在的问题是消息的大小变得非常巨大, Agent 之间交互的消息经常含有许多冗余的信息, DPOP 等算法并没有考虑问题本身的一些硬约束条件.

##### 4.4.2 H-DPOP

Kumar 等人<sup>[34]</sup>针对 DPOP 算法中消息长度指数级等不足, 于 2008 年在 AAAI 会议上首次提出了 H-DPOP 算法, H-DPOP 中通过使用 CDD (Constraint Decision Diagrams)<sup>[65]</sup>技术来剔除一些不可能的赋值(硬约束条件不满足的赋值), 达到压缩 UTIL 消息的目的.

##### 4.4.3 O-DPOP

Petcu 等人<sup>[66]</sup>于 2006 年提出 O-DPOP 算法, O-DPOP 算法主要用于解决开放环境的多 Agent 组合优化问题. 算法与 DPOP 算法类似, 也是基于动态规划策略进行求解, 但是它是增量式的 (incremental), 适合于开放性问题.

##### 4.4.4 M-DPOP

Petcu 等人<sup>[67]</sup>于 2006 年提出的 M-DPOP 算法在求解 DCOP 问题时, 同时考虑了社会选择问题 (Social Choice). 将社会选择问题建模成 DCOP 问题, 通过使用激励相容机制以及纳什均衡等技术来保证 Agent 的可信性 (Faithful), 解决了多 Agent 系统中由于 Agent 的自私等特性给问题带来的麻烦, 同时这个算法也特别适用于许多经济方面的问题.

除上面介绍的几种基于动态规划的算法外, 还有很多其他相关的算法, 如哈佛大学 Greenstadt 等人<sup>[68]</sup>于 2007 年在 AAMAS 上提出的 SSDPOP 算法. 相比于 DPOP 算法, SSDPOP 提供了隐私保护技术, 对 Agent 不愿透入给其他 Agent 的信息增加了一定的保护机制, 能减少隐私丢失等现象的发生. 实验结果表明 SSDPOP 将隐私丢失降低了 29%~

88%左右.此外,还有 S-DPOP<sup>[69]</sup>、A-DPOP<sup>[56]</sup>、PC-DPOP<sup>[10]</sup>、BB-M-DPOP<sup>[10]</sup>算法等.

#### 4.5 算法比较

基于搜索的算法与基于动态规划的算法最主要的区别在于算法的复杂性不同,主要包括时间复杂度以及空间复杂度.基于动态规划的算法的时间和空间复杂度都是树的宽度的指数级别.然而,对于强力搜索只需要线性的内存,但是所需的时间则取决

于问题的结构,即与问题的规模有关.如表 2 所示,我们对某些 DCOP 算法的时间和空间复杂性进行了比较.

从表 2 可以看出,基于动态规划的算法的空间复杂度较大,而基于搜索的算法在时间复杂度上是难以预测的.在实际应用中,针对问题的特点,如问题的规模以及对于时间和空间复杂度的要求,选择一个合适的算法往往能获得较好的效果.

表 2 DCOP 算法复杂性对比

Algorithm	Type	Time complexity	Space complexity
dAO-Opt	Synch-search	Exponential in the depth of DFS	Linear
ADOPT	Asyn-search	Exponential in the number of variables (worst-case)	Polynomial in the number of variables (worst-case)
NCBB	Synch-search	Exponential in the depth of DFS	Polynomial memory
DPOP	Dynamic Programming	Linear number of messages	Exponential in the size of the largest bucket
BnB-ADOPT	Asyn-search	Exponential in the depth of DFS	Polynomial memory
H-DPOP	Dynamic Programming	Linear number of messages	Less memory than DPOP
IDB-ADOPT	Asyn-search	Exponential in the depth of DFS	Polynomial memory

如表 3 所示,给出了 DCOP 算法的内存与消息数量的对比关系,表中 w. c. expon 代表的是图的宽度的指数级 (constraint diagram width).我们可知基于搜索的算法其所需要的内存是比较小的,如

ADOPT、NCBB、AFB 等,都是线性的,但是其消息数量都是指数级的.而基于动态规划的方法,如 DPOP、H-DPOP 等,其消息数量是线性的,但其需要的内存较大,是指数级的.

表 3 DCOP 算法内存与消息数量对比

Memory	Number of Messages			
	linear	polynomial	Worst case exponential	exponential
linear	PC-DPOP(1), A-DPOP(1)	LS-DPOP(1)	ADOPT, NCBB, AFB, SynchBB	MB-DPOP(1)
polynomial	PC-DPOP( $k$ ), A-DPOP( $k$ )	LS-DPOP( $k$ )	NCBB( $k$ ), OptAPO	MB-DPOP( $k$ )
w. c. expon	H-DPOP		O-DPOP	
exponential	DPOP			

## 5 分布式约束优化问题非完全算法

### 5.1 Agent 个体决策算法

Agent 个体决策算法主要有 DBA<sup>[35-36]</sup>、ALS\_DisCOP<sup>[37]</sup>、DSA<sup>[38]</sup>等.最小冲突回溯等约束满足算法中的爬山搜索策略,有时会使得求解过程陷入局部最小(local-minima)状态.local-minima 状态就是一些约束没有被满足从而出现冲突,但是这些冲突的数目不能通过单独改变任何一个变量的值来减小.因此,DBA 算法通过以下两个步骤来实现:第一,始终保证评估值是逐步提高的,相邻的 Agent 对可能会提高的评估值进行通信,只有能够最大提高评估值的 Agent 才去改变自己的取值.如果两个 Agent 不相邻,那么它们可能出现同时改变自己取值的情况;第二,与检测整个 Agent 是否陷入 local-minima 不同的是,每个 Agent 检测其是否处于 quasi-local-

minima 状态,这是比 local-minima 要更弱的条件,并能够通过局部通信而检测到.

DSA 算法中,所有 Agent 之间都是平等的,在处理过程中它们之间没有明显的区分标识,该算法是同步算法.节点在发送消息之后必须要收到返回的消息后,才继续下一步的操作,执行局部的计算,并且改变局部状态.

### 5.2 Agent 组决策算法

在 Agent 组决策算法中,又可以进一步分为  $k$ -size 最优算法<sup>[70]</sup>和  $t$ -distance 最优算法<sup>[71]</sup>,它们的区别主要在于选取同组中的 Agent 的方式不同. $k$ -size 最优算法是随机选取  $k$  个 Agent 构成一个组;而在  $t$ -distance 最优算法中,围绕一个中心 Agent,并将与该 Agent 之间距离小于或等于  $t$  的 Agent 的集合构成一个组.组决策算法有一个特点,假设该组中的 Agent 个数为  $n$ ,如果组决策找到一个局部最优解,我们不可能通过改变小于或等于  $n$  个 Agent

的决策使结果变得更优。

$k$ -size 最优算法中,典型的算法有  $\max\text{-sum}$ <sup>[39,72-73]</sup> 以及 KOPT<sup>[36]</sup> 等。 $\max\text{-sum}$  算法<sup>[39,72-73]</sup> 是一种基于一般性分布式定律 (Generalised Distributive Law, GDL)<sup>[74-75]</sup> 提出的,在  $\max\text{-sum}$  的基础上,又相应地提出了 Fast-Max-Sum (FMS)<sup>[76]</sup> 等算法,FMS 相比于一般的  $\max\text{-sum}$  算法,在求解过程中产生的消息数量更少,时间复杂性也 smaller。Macarthur 等人<sup>[77]</sup> 以 FMS 算法为基础,又相应地提出了 BnB FMS 算法,该算法在 FMS 的基础上,又对解的质量提供一定的保证。然而,这些算法并非对所有的  $k$  都适用,只是  $k$  在一定的范围内有效。而 KOPT 算法则是  $k$ -size 最优算法中的一种特例,它是一种针对任意  $k$  都有效的  $k$ -size 最优算法,而且 KOPT 算法对于得到的解提供一定的保证。

Vinyals 等人<sup>[78]</sup> 提出了一种新的  $k$ -size 最优算法 DaC,该算法的核心思想是分割与协同,即将 Agent 分割成不同的组,然后把每个组看成是一个局部寻找最优的过程。通过协同不同组之间的结果,在一定范围内寻找到一个局部最优解。

Yin 等人<sup>[71]</sup> 提出了  $t$ -distance 最优的评判标准,并在此基础上提出了一种  $t$ -distance 算法 DALO, DALO 是一种异步算法,是对分布式局部搜索算法的改进。DALO 算法主要包括以下 3 个阶段: 第一, Agent 将其局部信息以及约束通过消息发送到距离其  $t$  跳以内的节点,然后将其取值发送到  $t+1$  跳以内的节点上。第二,根据上一阶段收集的信息,所有组中的 leader 节点计算其局部最优解,在计算过程中假设该组之外的 Agent 的取值都保持不变。第三,如果一个 leader 节点找到了一个更好的局部解,该 leader 节点将这个新的取值发送给外部的 Agent。但是这种情况下可能会导致部分出现重合的组中 Agent 存在冲突, DALO 通过使用分布式的锁机制来实现。

Bigdeli 等人<sup>[79]</sup> 提出一种新的  $t$ -distance 最优算法 DGOPT,该算法在 DALO 的基础上,提出了动态添加或者减少同组中 Agent 的组优化策略,在一个 Agent 组中, leader 节点根据边缘分布 (marginal contribution) 的概念添加或者减少同组中 Agent 节点的个数。

Vinyals 等人<sup>[80]</sup> 提出了一种能够适用于  $k$ -size 最优以及  $t$ -distance 最优两种类型的算法的通用框架,并且在该框架下提出了一种新的算法,该算法是 DALO 算法的一种扩展,它能够在  $k$ -size 最优以及

$t$ -distance 两种标准下用于寻找局部最优解。

大多数的  $k$ -size 最优算法只能是 1-size 最优的,如 MGM、DBA 等。目前的组决策算法在  $k \geq 3$  时,其计算复杂度非常大,使得其很难应用于实际问题的求解中。然而,由 Katagishi 等人<sup>[40]</sup> 提出的 KOPT 算法,则属于这类算法的一个特例,其可以是任意 size 最优的,只是不同 size 计算量不同。显然, $n$  个节点时的  $n$ -size 最优就是真正全局最优。但是 KOPT 算法存在以下几个问题: 第一,理论分析结果显示, $k$ -size 最优的解的下限是与图的密集程度成反比<sup>[81]</sup>。大量的研究都证明了  $k$ -size 最优对于得到的解能够提供一定的质量保证,如在节点数量为  $n$  时,有  $R(A) \geq (k-1)/(2n-k-1)R(A^*)$ <sup>[17,71]</sup>, 其中  $R(A)$  为  $k$ -size 最优求得的解,而  $R(A^*)$  为全局最优解,这是一个  $k$ -size 最优解的下限。考虑到不完全算法的时间复杂度相对较小,同时又对解的质量提供一定的保证,使得其被广泛地用于求解一些大型的实际问题当中。

第二,面临着一个消息数量巨大的问题,尤其是当图的密度较大的时候。为了有效地解决这个问题, Yin 等人<sup>[71]</sup> 提出了另外一种衡量解的质量的标准  $t$ -distance 最优,它通过节点间的距离来定义 Agent 组中的成员,且在  $t$ -distance 最优性下,提出了一种异步局部搜索算法,该算法基于 lock/commit 协议,其性能相对于 KOPT 有较大的改善。

针对于  $k$ -size 最优以及  $t$ -distance 最优,两种不同的评价非完全算法解的质量的标准,下面进行了相关比较。

在  $k$ -size 最优中,每一组中的 Agent 的个数是固定的  $k$  个,但是满足  $k$ -size 的组的个数可能会很多。然而  $t$ -distance 最优中每一组中的 Agent 的个数是不固定的,在图的密度较大时,决策组中的 Agent 数量往往会很大,但决策组的个数是有限的,其数量不超过 Agent 的个数。

针对某些特定的环形结构图,我们分析了  $k$ -size 最优和  $t$ -distance 最优对于解的质量的保证情况。首先,在  $k=1$  和  $t=0$  时, $k$ -size 最优以及  $t$ -distance 最优得到的结果是一样的,因为它们都是将每个 Agent 看成是一个独立的组进行决策。其次, $t$ -distance 最优得到的解的质量往往要好于  $k$ -size 最优,因为  $t$ -distance 最优的解必然是  $k$ -size 最优 ( $k=2t+1$ ) 的解,因为每一个  $k$ -size 的决策组都属于  $t$ -distance 中。反之则不是,存在某些  $k$ -size 最优的解,却不是  $t$ -distance 最优的 ( $t=(k-1)/2$ )。

## 6 分布式约束优化问题的相关应用

### 6.1 云计算中的资源映射问题

根据前面的介绍,我们可知云计算中的资源分配问题(RA)<sup>[82-83]</sup>是目前云计算中的一个重点研究方向.在 RA 问题中,如果需要得到资源分配的最优方案时,将 RA 问题建模成 DCOP 进行求解是一种有效的方法.在 RA 中,WK 是指虚拟节点由虚拟链路连接构成的虚拟网络(Virtual Network, VN)<sup>[83]</sup>,而 CP 提供的基础设施可以建模为由物理节点通过物理链路连接构成的物理网络(Substrate Network, SN)<sup>[83]</sup>.其目标是要将 WK 映射到 SN 上,其中虚拟节点映射到 SN 中的物理节点上,虚拟链路映射到 SN 中的物理路径上,同时满足 WK 中节点和链路对资源的约束,如节点对 CPU、memory 等资源有一定的约束,链路对带宽、延迟等资源有一定约束.

图 4 给出了如何将 RA 问题建模成 DCOP 进行求解,如图 4(a)所示,为 RA 中的 WK 请求,图 4(b)、(c)、(d)分别为 3 个 CP 提供的 SN.其中的 1 个约束是要将 WK 中的虚拟节点 1、2、3 分别映射到 CP1、CP2 和 CP3 中,并且满足 WK 中虚拟节点和虚拟链路对资源的约束.

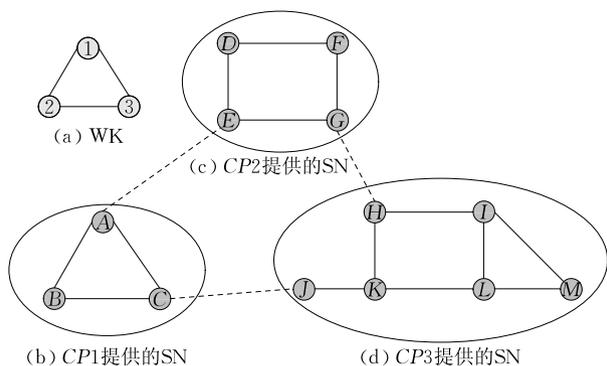


图 4 资源分配问题的 DCOP 模型

在建模成 DCOP 的过程中,其相关的定义如下:Agent 集合为  $\{CP1, CP2, CP3\}$ ,变量集合  $X$  为虚拟节点  $\{1, 2, 3\}$ ,变量  $X$  中变量的值域  $D$  分别为  $\{A, B, C\}$ 、 $\{D, E, F, G\}$  和  $\{H, I, J, K, L, M\}$ .映射关系  $M$  如下:  $CP1 \rightarrow \{1\}$ 、 $CP2 \rightarrow \{2\}$  和  $CP3 \rightarrow \{3\}$ .约束代价函数  $F$  是指映射 WK 产生的代价,而代价又被定义为分配给 WK 的资源数量,一般包括节点所需的 CPU 资源和链路所需的带宽资源等.

在将 RA 建模成 DCOP 问题后,可以采用第 4 节中介绍的算法进行求解,并且可以根据 RA 的具体要求选择算法,如果当前 WK 请求任务比较紧

急,希望部署时间较短,时间复杂度较小的算法应该更为合适,如 DPOP 算法等.如果 WK 中的变量数量较大,则选用搜索算法更为合适,因为其所需要的空间复杂度以及内存更小.出于商业目的考虑,不同 CP 之间的信息往往不希望对外透露,在这种情况下选择 M-DPOP 算法更为合适.如果考虑 CP 提供的 SN 可能出现故障导致 SN 的动态变化,则可进一步将其建模为动态 DCOP 进行求解,这在下面将会展开介绍.

除此之外,很多还有其他寻找 Agent 间共同决策的一致组合问题,比如会议时间安排<sup>[10]</sup>、分布式日常安排<sup>[10]</sup>、传感器资源分配<sup>[12]</sup>等,都可以建模为 DCOP 问题来求解,并且可以根据具体问题的特点选择相应的 DCOP 算法.

### 6.2 动态分布式约束优化问题的研究

随着互联网新型应用的发展,许多问题的应用环境往往是动态变化的.从而要求 DCOP 算法能够有效地解决这种动态变化下的约束优化问题.如上面提到的云计算中的 RA 问题,参与资源分配的 CP 数量可能动态变化以及 CP 内部的基础设施资源也可能动态变化,如 CP 提供的数据中心网络中节点出现故障或者因人工原因导致某些节点不能正常工作时,都会导致其结构发生改变.此外,用户请求 WK 也可能随着用户需求的变化而动态发生改变.例如,以动态会议时间调度问题为例,对于每一个参加会议的成员来说,其行程安排可能随时间会动态发生变化,这种情况下之前已分配的方案,在成员时间发生改变时,未必满足新的要求.在以上情况中,都希望能够有动态 DCOP 算法来有效地解决这些问题.然而,本文提到的 DCOP 算法在求解动态 DCOP 问题上还存在一定的局限性,算法本身以及对算法时间复杂度远远超过了应用的需求.

因此,考虑到现在应用的这种高度动态性,目前有一些研究侧重于 anytime 算法的研究,该类算法能够保证任意时间都能给出问题的解,但其给出的不是问题的最优解,而是近似最优解,与前面介绍的非完全算法有一定的类似之处. DCOP 非完全算法能够在一定程度上解决这种高度动态问题的需求.

## 7 未来的研究趋势

综合分布式约束优化问题目前研究的热点问题以及广泛关注的应用领域,未来的研究趋势主要关注于如下几个方面:

### (1) 动态 DCOP 算法的研究

随着互联网新型应用的发展,许多问题的应用环境不再是恒定不变的,而是可以动态改变的.如动态传感器分配问题,目标的到达可能是随机的、不可预测的,而且目标的位置是可以动态移动的.云计算中的资源分配问题(RA),用户请求往往也是动态变化的,本文介绍的这些 DCOP 算法在解决环境动态改变的问题上存在一定的局限性.目前,动态约束满足问题已经应用于解决某些现实问题<sup>[69-71,83]</sup>,但是它们基本上都是针对集中式环境的,对于分布式动态 DCOP 算法的研究还不成熟.

动态 DCOP 主要面临以下挑战:算法必须能够处理系统中任意状态的变化,如云计算中的资源分配问题中,云用户或者租户往往是动态变化,每时每刻都有请求到来的,其动态性是不可预测的.解决这类高度动态环境下约束优化问题,对于算法的复杂性要求很高,要求其能够以尽可能小的时间代价计算给出合适的解.然而,目前的 DCOP 算法在求解这类现实应用中的许多问题存在明显的不足,也是未来需要重点关注的方向之一.

Petcu 等人提出了 S-DPOP 算法,S-DPOP 算法能够在环境发生变化时做出相应的调整,但其还不能用于解决实际的问题.因此,动态 DCOP 算法的研究必将成为以后该领域研究的一大热点问题.

### (2) 社会选择因素对 DCOP 的影响

DCOP 问题的目标是通过 Agent 的变量序列进行赋值,在满足约束条件的情况下,最大化全局利益.一般情况下,Agent 都会有一些私有信息(通常不希望对外透露),在多 Agent 进行协商决策时,通常都认为其他 Agent 公布出来的信息是可信的,而不考虑 Agent 谎报信息的情况出现.然而,这一说法具有很大的局限性,在现实应用中,Agent 往往是自私的,它对问题有自己的偏好,可能会做出使自己利益最大化的选择,而忽略全局利益最大化的目标,这就是所谓的社会选择(Social Choice)问题.如上面提到的云计算中的资源分配问题,CP 可能通过对外透露虚假的信息以谋求自身的利益最大化,毕竟对于 CP 来说,利益最大化才是其最重要的指标.因此,在这类问题中,其目标是如何在 Agent 自私的情况下使得全局利益最大化.

解决这类社会选择问题可以考虑使用激励相容机制如 Vickrey-Clarke-Groves(VCG)机制以及纳什均衡等技术来保证 Agent 的可信性(Faithful),预防 Agent 谎报个人偏好能力等信息.如 Agent 对外

发布真实信息时,给予相应的奖励,而如果发现 Agent 发布虚假信息时,相应地给予惩罚措施,这在一定程度上能够避免 Agent 发布虚假信息的可能性.但是在一些实际问题中,对于问题的界限确定是非常困难的,这也导致了激励机制在解决该类问题的不足.

文献[10]中将社会选择问题建模成 DCOP 问题并提出了 M-DPOP 算法,最先针对有效社会选择问题提供了可信机制的分布式实现算法,诚实性(faithfulness)保证了所有的 Agent 都应该遵守相关的协议规定,不能为了私人的利益而做假.这一算法也开启对于有效社会选择问题的研究,这也将成为以后研究的一个热点问题.

### (3) DCOP 非完全算法应用的研究

在 DCOP 问题面临的挑战中,代价问题往往是最受关注的.当问题的规模较大时,采用 DCOP 完全算法进行求解时,尽管能够找到最优解,但是其花费的时间、空间代价以及通信开销往往很大,这对许多实时性要求较高的应用往往不适用.如云计算中的资源分配问题(RA),在很多情况下,考虑到用户请求的紧迫性以及 CP 对得到最优解的需求不大时,此时,它们未必需要花费昂贵的代价去得到最优解,而只要得到近似最优解即可.

考虑到以上原因,越来越多的现实应用问题(资源分配问题)都可采用非完全 DCOP 算法进行求解.非完全算法虽然无法找到全局最优解,但是其时间以及通信开销等大大降低,而且一些 DCOP 近似算法能够对解的质量提供一定的质量保证,如  $k$ -size 最优算法能够保证解的下限是  $(k-1)/(2n-k-1)$ <sup>[17]</sup>.因此,考虑到这些应用的实际情况,当其对时间以及通信开销要求较高,往往通过计算其近似最优解来解决.采用 DCOP 近似算法(非完全算法)进行求解优化问题时,由于其通过搜索部分解空间进行求解,既能够保证得到的结果能够在可以承受的范围内,又能够大大地降低求解时所消耗的资源以及求解时间度.这无非是一个更好的选择.

现有的有关 DCOP 非完全算法的研究,大多都还停留在理论层面的研究,针对如何评价这类算法得到的解的质量,提出了  $k$ -size 最优以及  $t$ -distance 最优的评判标准,并且有很多的研究通过理论证明<sup>[17,71]</sup>,证实了它们对解的质量确实有一定的保证.但是,关于如何将 DCOP 非完全算法运用于实际问题中,以及如何根据具体的实际问题选择合适的近似算法是未来的研究热点之一.

## 8 结束语

分布式约束优化问题的研究作为新兴的一个热点研究问题,从完全算法到非完全算法、从搜索算法到基于动态规划的算法、从同步算法到异步算法、从单纯的理论研究到实际系统的有效部署、从单一的算法提出到面向应用的具体实施,可以说,分布式约束优化问题领域的研究是一个非常活跃的方向.从整体上讲,目前在分布式约束优化方面的研究还远未成熟,尚未建立起一套完整的理论体系,而且从技术理论的完善到算法的具体应用还有很大的差距.

本文回顾了近年来学术界在分布式约束优化问题研究领域的主要成果,在一个给定的研究框架下,对分布式约束优化问题进行了综述,详细介绍了常用的一些分布式约束优化算法并加以比较.在本文中对各种算法的定性分析、定量分析以及性能分析较少,主要原因如下:第一,目前很多的 DCOP 算法,虽然能够找到相应算法的论文,但是相应算法的代码我们还没有办法完全获得.第二,不同类型的 DCOP 算法,如完全算法和非完全算法之间,绝大多数是基于不同的平台开发设计的,目前还没有通用的平台能够测试所有的算法,平台的不同对于测试的结果会产生很大的影响.第三,目前 DCOP 研究者在测试他们的算法的性能时,都是基于一些典型的问题,如地图着色问题、Scale-free 问题等,基本没有算法在测试性能时是基于现实应用问题的.在我们未来的研究工作中,将着重针对一个主要的研究问题,并针对该问题对各种算法的性能进行测试比较.本文最后总结并指明了未来研究的趋势.

### 参 考 文 献

[1] Verfaillie G, Schiex T. Solution reuse in dynamic constraint satisfaction problems//Proceedings of the National Conference on Artificial Intelligence. Seattle, USA, 1994: 307-312

[2] Anshul R, Ranjita B, Saikat G. Generalized resource allocation for the cloud//Proceedings of the 3rd ACM Symposium on Cloud Computing. San Jose, USA, 2012: 365-373

[3] Dechter R. Constraint Processing. San Fransisco: Morgan Kaufmann, 2003

[4] Wang Qin-Hui, Chen En-Hong, Wang Xu-Fa. Research and development of distributed constraint satisfaction problems. Journal of Software, 2006, 17(10): 2029-2039(in Chinese)

(王秦辉,陈恩红,王煦法. 分布式约束满足问题研究及其进展. 软件学报, 2006, 17(10): 2029-2039)

- [5] Montanari U. Networks of constraints; Fundamental properties and applications to picture processing. Information Sciences, 1974, 7(2): 95-132
- [6] Kumar V. Algorithms for constraint satisfaction problems; A survey. AI Magazine, 1992, 13(1): 32-44
- [7] Zheng Jian-Guo, Wang Xiang, Liu Rong-Hui.  $\epsilon$ -differential evolution algorithm for constrained optimization problems. Journal of Software, 2012, 23(9): 2374-2387(in Chinese)  
(郑建国, 王翔, 刘荣辉. 求解约束优化问题的  $\epsilon$ -DE 算法. 软件学报, 2012, 23(9): 2374-2387)
- [8] Wang Yong, Cai Zi-Xing, Zhou Yu-Ren, Xiao Chi-Xin. Constrained optimization evolutionary algorithms. Journal of Software, 2009, 20(1): 11-29(in Chinese)  
(王勇, 蔡自兴, 周育人, 肖赤心. 约束优化进化算法. 软件学报, 2009, 20(1): 11-29)
- [9] Modi P J, Shen W, Tambe M. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. Artificial Intelligence, 2005, 161(1-2): 149-180
- [10] Petcu A. A Class of Algorithms for Distributed Constraint Optimization [Ph. D. dissertation]. Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2007
- [11] Cui Shu-Guang. Research on Web Services Composition Online Optimization and Distributed Scheduling [M. S. dissertation]. Hunan Normal University, Hunan, 2007 (in Chinese)  
(崔曙光. Web 服务组合在线优化以及分布式编排研究[硕士学位论文]. 湖南师范大学, 长沙, 2007)
- [12] Zhang W, Xing Z, Wang G, Wittenburg L. An analysis and application of distributed constraint satisfaction and optimization algorithms in sensor networks//Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems. New York, USA, 2003: 185-192
- [13] Yokoo M, Ishida M, Durfee E H, Kuwabara K. Distributed constraint satisfaction for formalizing distributed problem solving//Proceedings of the 12th International Conference on Distributed Computing Systems. Yokohama, Japan, 1992: 614-621
- [14] Yokoo M. Dynamic variable/value ordering heuristics for solving large-scale distributed constraint satisfaction problems //Proceedings of the 12th International Workshop on Distributed Artificial Intelligence. Pennsylvania, USA, 1993: 407-422
- [15] Maheswaran R T, Tambe M, Bowring E, et al. Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling//Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems. New York, USA, 2004: 310-317
- [16] Sultanik E, Modi P J, Regli W C. On modeling multiagent task scheduling as a distributed constraint optimization problem //Proceedings of the 20th International Joint Conference

- on Artificial Interlligence. Hyderabad, India, 2007; 1531-1536
- [17] Kiekintveld C, Yin Z, Kumar A. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds//Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. Toronto, Canada, 2010; 133-140
- [18] Walker R J. An enumerative technique for a class of combinatorial problems//Proceedings of the Symposia in Applied Mathematics. New York, USA, 1960; 91-94
- [19] Knuth D E. Estimating the efficiency of backtrack programs. *Mathematics of Computation*, 1975, 29(129): 121-136
- [20] Golomb S W, Baumert L D. Backtrack programming. *Journal of the ACM*, 1965, 12(4): 516-524
- [21] Hirayama K, Yokoo M. Distributed partial constraint satisfaction problem//Proceedings of the 3rd International Conference on Principles and Practice of Constraint Programming. Linz, Austria, 1997; 222-236
- [22] Freuder E C, Quinn M J. Taking advantage of stable sets of variables in constraint satisfaction problems//Proceedings of the 9th International Joint Conference on Artificial Intelligence. San Francisco, USA, 1985; 1076-1078
- [23] Chechotka A, Sycara K. No-commitment branch and bound search for distributed constraint optimization//Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems. Hakodate, Japan, 2006; 1427-1429
- [24] Yeoh W, Felner A, Koenig S. BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm//Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Estoril, Portugal, 2008; 591-598
- [25] Yeoh W, Koenig S, Felner A. IDB-ADOPT: A depth-first search DCOP algorithm//Proceedings of the 13th Annual ERCIM International Workshop on Constraint Solving and Constraint Logic Programming. Rome, Italy, 2009; 132-146
- [26] Mailler R, Lesser V. Solving distributed constraint optimization problems using cooperative mediation//Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems. New York, USA, 2004; 438-445
- [27] Gershman A, Meisels A, Zivan R. Asynchronous forward bounding for Distributed COPs. *Jouruanl of Artificial Intelligence Research*, 2009, 34(1): 61-88
- [28] Dechter R. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 1999, 113(1-2): 41-85
- [29] Kask K, Dechter R. Branch and bound with mini-bucket heuristics//Proceedings of the 6th International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 1999; 235-426
- [30] Kask K, Dechter R, Larrosa J. Unifying tree decompositions for reasoning in graphical models. *Artificial Intelligence*, 2005, 166(1-2): 165-193
- [31] Larrosa J, Dechter R. Boosting serach with variable elimination in constraint optimization and constraint satisfaction problems. *Constraints*, 2003, 8(3): 303-326
- [32] Rollon E, Larrosa J. Depth-first mini-bucket elimination//Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming. Sitges, Spain, 2005; 563-577
- [33] Petcu A, Faltings B. A scalable method for multiagent constraint optimization//Proceedings of the 9th International Joint Conference on Artificial Intelligence. Edinburgh, Scotland, 2005; 266-271
- [34] Kumar A, Petcu A, Faltings B. H-DPOP: Using hard constraints for search space pruning in DCOP//Proceedings of the 23rd AAAI Conference on Artificial Intelligence. Chicago, USA, 2008; 325-330
- [35] Pearce J P, Tambe M, Maheswaran R T. Solving multiagent networks using distributed constraint optimization. *AI Magazine*, 2008, 29(3): 47-62
- [36] Zhang W, Wang G, Xing Z, et al. Distributed stochastic search and distributed breakout: Properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 2005, 161(1-2): 55-87
- [37] Zivan R. Anytime local search for distributed constraint optimization//Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Estoril, Portugal, 2008; 1449-1452
- [38] Fitzpatrick S, Meertens L. Distributed coordination through anarchic optimization. *Distributed Sensor Networks: A Multiagent Perspective*, 2003, 13(2): 257-295
- [39] Farinelli A, Rogers A, Jennings N. Bounded approximate decentralised coordination using the max-sum algorithm. *Artificial Intelligence*, 2011, 175(2): 730-759
- [40] Katagishi H, Pearce J P. KOPT: Distributed DCOP algorithm for arbitrary  $k$ -optima with monotonically increasing utility//Proceedings of the 9th International Workshop on Distributed Constraint Reasoning. Rhode Island, USA, 2007; 28-41
- [41] He Li-Jian, Zhang Wei. An Agent organization structure for solving DCOP based on the partitions of constraint graph. *Journal of Computer Research and Development*, 2007, 44(3): 434-438(in Chinese)  
(贺利坚, 张伟. 基于约束图分片求解 DCOP 的 Agent 组织结构. *计算机研究与发展*, 2007, 44(3): 434-438)
- [42] Huang Jing, Liu Da-You, Yang Bo, Jin Di. A self-organization based divide and conquer algorithm for distributed constraint optimization problems. *Journal of Computer Research and Development*, 2008, 45(11): 1831-1839(in Chinese)  
(黄晶, 刘大有, 杨博, 金弟. 自组织分治求解分布式约束优化问题. *计算机研究与发展*, 2008, 45(11): 1831-1839)
- [43] Yokoo M, Durfee E H. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 1998, 10(5): 673-685

- [44] Yokoo M. Asynchronous weak-commitment search for solving distributed constraint satisfaction problems//Proceedings of the Principles and Practice of Constraint Programming. Cassis, France, 1995: 88-102
- [45] Silaghi M C, Sam-Haroud D, Faltings B. Asynchronous search with aggregations//Proceedings of the 17th National Conference on Artificial Intelligence. California, USA, 2000: 917-922
- [46] Meisels A, Zivan R. Asynchronous forward-checking on DisCSPs. *Constraints*, 2007, 12(1): 131-150
- [47] Collin Z, Dechter R, Katz S. On the feasibility of distributed constraint satisfaction//Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Aust, 1991: 318-324
- [48] Collin Z, Dechter R. Self-stabilizing distributed constraint satisfaction. *Chicago Journal of Theoretical Computer Science*, 2000, 23(1): 66-78
- [49] Dechter R, Mateescu R. AND/OR search spaces for graphical models. *Artificial Intelligence*, 2007, 171(2/3): 73-106
- [50] Chechetka A, Sycara K. A decentralized variable ordering method for distributed constraint optimization//Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems. New York, USA, 2005: 1307-1308
- [51] Tarjan R E, Yannakakis M. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal of Computing*, 1984, 13(3): 566-579
- [52] Kjaerulff U. Graph-theoretic concepts in computer science//Proceedings of the 28th International Workshop on Graph. Cesky Krumlov, Czech Republic, 2002: 123-135
- [53] Yokoo M, Ishida T. Search algorithms for agents. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 1999, 22(7): 165-199
- [54] Gershman A, Meisels A, Zivan R. Asynchronous forward-bounding for distributed constraints optimization//Proceedings of the 17th European Conference on Artificial Intelligence. Trentino, Italy, 2006: 103-107
- [55] Marinescu R, Dechter R. AND/OR branch-and-bound for graphical models//Proceedings of the 19th International Joint Conference on Artificial Intelligence. San Francisco, USA, 2005: 224-229
- [56] Petcu A, Faltings B. A-DPOP: Approximations in distributed optimization//Proceedings of the International Joint Conference on Artificial Intelligence. Sitges, Spain, 2005: 802-806
- [57] Marinescu R, Dechter R. Memory intensive branch-and-bound search for graphical models//Proceedings of the National Conference on Artificial Intelligence. Boston, USA, 2006: 1200
- [58] Ding Bo, Wang Huai-Min, Shi Dian-Xi, Tang Yang-Bin. Algorithm for distributed constraint optimization problems with low constraint density. *Journal of Software*, 2011, 22(4): 625-639(in Chinese)
- (丁博, 王怀民, 史殿习, 唐扬斌. 低约束密度分布式约束优化问题的求解算法. *软件学报*, 2011, 22(4): 625-639)
- [59] Zhang W, Korf R. Performance of linear-space search algorithms. *Artificial Intelligence*, 1985, 79(2): 241-292
- [60] Bellman R E. *Dynamic Programming*. USA: Princeton University Press, 1957
- [61] Bellman R E, Dreyfus S. *Applied Dynamic Programming*. USA: Princeton University Press, 1962
- [62] De Vries S, Vohra R V. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 2003, 15(3): 284-309
- [63] Dechter R. Bucket elimination: A unifying framework for processing hard and soft constraints. *Constraints: An International Journal*, 1997, 7(2): 51-55
- [64] Faltings B. Distributed constraint programming. *Foundations of Artificial Intelligence*, 2006, 32(2): 699-729
- [65] Cheng K C K, Yap R H C. Constrained decision diagrams//Proceedings of the National Conference on Artificial Intelligence. Pittsburgh, Pennsylvania, 2005: 366-371
- [66] Petcu A, Faltings B. O-DPOP: An algorithm for open/distributed constraint optimization//Proceedings of the International Joint Conference on Artificial Intelligence. Sitges, Spain, 2006: 432-438
- [67] Petcu A, Faltings B, Parkes D. M-DPOP: Faithful distributed implementation of efficient social choice problems. *Journal of Artificial Intelligence Research*, 2008, 32(5): 705-755
- [68] Greenstadt R, Grosz B, Smith M D. SSDPOP: Improving the privacy of DCOP with secret sharing//Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. New York, USA, 2007: 1-3
- [69] Petcu A, Faltings B. Superstabilizing, fault-containing multiagent combinatorial optimization//Proceedings of the National Conference on Artificial Intelligence. Pittsburgh, United States, 2005: 449-454
- [70] Bowring E, Pearce J P, Portway C, et al. On  $k$ -optimal distributed constraint optimization algorithms: New bounds and algorithms//Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Estoril, Portugal, 2008: 607-614
- [71] Yin Z, Kiekintveld C, Kumar A, Tambe M. Local optimal solutions for DCOP: New criteria, bound, and algorithm//Proceeding of the Workshop on Optimization in Multi-Agent Systems. Budapest, Hungary, 2009: 58-69
- [72] Leaute T. *Distributed Constraint Optimization* [Ph.D. dissertation]. Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2011
- [73] Farinelli A, Rogers A, Petcu A, Jennings N R. Decentralised coordination of low-power embedded devices using the max-sum algorithms//Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Estoril, Portugal, 2008: 639-646
- [74] Aji S, McEliece R. The generalized distributive law. *IEEE Transactions on Information Theory*, 2000, 46(2): 325-343

- [75] Vinyals M, Rodriguez-Aguilar J A, Cerquides J. Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law. *Autonomous Agents and Multi-Agent Systems*, 2011, 22(3): 439-464
- [76] Macarthur K. Multi-Agent Coordination for Dynamic Decentralised Task Allocation [ Ph. D. dissertation]. University of Southampton, Southampton, England, 2011
- [77] Macarthur K, Stranders R, Ramchurn S D, Jennings N R. A distributed anytime algorithm for dynamic task allocation in multi-agent systems//Proceedings of the 25th Conference on Artificial Intelligence. San Francisco, USA, 2011: 356-362
- [78] Vinyals M, Pujol M, Rodriguez-Aguilar J A, Cerquides J. Divide-and-coordinate: DCOPs by agreement//Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. Toronto, Canada, 2010: 149-156
- [79] Bigdeli E, Rahmaninia M, Afsharchi M. DGOPT: Dynamic group optimization to find better group formations in DCOPs//Proceedings of the International Joint Conference on Artificial Intelligence. Sitges, Spain, 2006: 324-338
- [80] Vinyals M, Shieh E, Cerquides J. Quality guarantees for region optimal DCOP algorithms//Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems. Taipei, China, 2011: 133-140
- [81] Pearce J P, Tambe M. Quality guarantees on  $k$ -optimal solutions for distributed constraint optimization problems//Proceedings of the International Joint Conference on Artificial Intelligence. Hyderabad, India, 2007: 1446-1451
- [82] Guo C, Lu G, Wang H, Yang S. Secondnet: A data center network virtualization architecture with bandwidth guarantees //Proceedings of the 6th International Conference on Emerging Networking EXperiments and Technologies. Philadelphia, USA, 2010: 26-37
- [83] Bansal N, Lee K, Nagarajan V, Zafer M. Minimum congestion mapping in a cloud//Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing. San Jose, USA, 2011: 267-276



**LI Xiao-Ling**, born in 1985, Ph. D. His research interests include distributed computing, data center network, operating system, and dependable software.

**WANG Huai-Min**, born in 1962, professor, Ph. D. supervisor. His research interests include distributed computing, dependable software, and network security.

## Background

Nowadays, many problems in multi-agent coordination can be abstracted as a distributed constrained optimization problem (DCOP), such as planning, scheduling, distributed control, resource allocation, etc. A major challenge in these applications is how to coordinate Agents' decisions to achieve the aim of global optimality, and DCOP algorithm is an effective means to solve these problems. Consequently, researchers have put forward various DCOP algorithms to solve such problems at the minimal cost. The research of

**GUO Chang-Guo**, born in 1973, Ph. D., associate professor. His research interests include distributed computing and dependable software.

**DING Bo**, born in 1978, Ph. D., research assistant. His research interests include distributed computing, pervasive computing, and software self-adaptive.

**LI Xiao-Yong**, born in 1982, Ph. D., research assistant. His research interests include distributed computing, network distance prediction and database.

DCOP is emerging as a new hotspot of research currently in artificial intelligence area, and this paper surveys the current prevalent DCOP algorithms. This work is partly supported by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302601, the National High Technology Research and Development Program (863 Program) of China under Grant No. 2012AA011201, and the National Natural Science Foundation of China under Grant Nos. 90818028, 6903043, and 61202117.