软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(2):264−279 [doi: 10.13328/j.cnki.jos.004881]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

分布式约束优化方法研究进展*

段沛博, 张长胜, 张 斌

(东北大学 信息科学与工程学院,辽宁 沈阳 110819)

通讯作者: 张斌, E-mail: zhangbin@ise.neu.edu.cn

摘 要: 多 agent 系统作为分布式人工智能研究领域的重要分支,已被广泛应用于多个领域中复杂系统的建模.而分布式约束优化作为一种多 agent 系统求解的关键技术,已成为约束推理研究的热点.首先对其适用性进行分析,并基于对已有算法的研究,总结出采用该方法解决问题的基本流程,在此基础上,从解的质量保证、求解策略等角度对算法进行了完整的分类;其次,根据算法分类结果以及执行机制,对大量经典以及近年来的分布式约束优化算法进行了深入分析,并从通信、求解质量、求解效率等方面对典型算法进行了实验对比;最后,结合分布式约束优化技术的求解优势给出了分布式约束优化问题的实际应用特征,总结了目前存在的一些问题,并对下一步工作进行了展望.

关键词: 多 agent 系统;分布式约束优化;约束规划;优化算法

中图法分类号: TP18

中文引用格式: 段沛博,张长胜,张斌.分布式约束优化方法研究进展.软件学报,2016,27(2):264−279. http://www.jos.org.cn/1000-9825/4881.htm

英文引用格式: Duan PB, Zhang CS, Zhang B. Research progress in distributed constraint optimization method. Ruan Jian Xue Bao/Journal of Software, 2016,27(2):264−279 (in Chinese). http://www.jos.org.cn/1000-9825/4881.htm

Research Progress in Distributed Constraint Optimization Method

DUAN Pei-Bo, ZHANG Chang-Sheng, ZHANG Bin

(School of Information Science and Engineering, Northeastern University, Shenyang 110819, China)

Abstract: Multi agent system, one of important branches of distributed artificial intelligence, has been widely applied to modeling a serious of complex systems in diverse research fields. Significant research effort has sought to solve constraint programming with distributed constraint optimization which is a popular framework for multi agent system. The contributions of this research proceed from previous work in the following ways. First, based on the existing research, the applicability of distributed constraint optimization is analyzed, and general process of distributed constraint optimization algorithms is extracted. Second, a relatively complete classification of algorithms is provided from the perspective of quality assurance and solving strategies. Next, considering execution mechanism, a thorough analysis of a large number of classic algorithms proposed in recent years is carried out. Moreover, the experimental analysis of some typical algorithms with the metrics of communication, solution quality and efficiency is provided. Finally, combining the advantage of distributed constraint optimization technology, the application characteristics of distributed constraint optimization problem are proposed, and future work is discussed.

Key words: multi agent system; distributed constraint optimization; constraint programming; optimization algorithm

* 基金项目: 国家自然科学基金(61572116, 61572117); 国家科技支撑计划(2014BAI17B00); 宁夏回族自治区自然科学基金(NZ13265); 中央高校东北大学基本科研专项基金(N120804001, N120204003)

Foundation item: National Natural Science Foundation of China (61572116, 61572117); National Key Technology Research and Development Program of China (2014BAI17B00); Natural Science Foundation of Ningxia Hui Autonomous Region (NZ13265); Fundamental Research Funds for the Central Universities of Northeastern University (N120804001, N120204003)

收稿时间: 2014-10-14; 修改时间: 2014-12-15; 采用时间: 2015-01-21; jos 在线出版时间: 2015-11-06

CNKI 网络优先出版: 2015-11-06 13:28:54, http://www.cnki.net/kcms/detail/11.2560.TP.20151106.1328.002.html

© 中国科学院软件研究所 http://www.jos.org.cn

多 agent 系统(multi-agent system,简称 MAS)作为分布式人工智能(distributed artificial intelligence,简称 DAI)的一个重要分支,利用分布式约束推理解决 agent 之间合作问题.MAS 中每个 agent 是自主的,通过协调这些自治的 agent 行为,使这些 agent 达到共同或各自不同的目标.在 MAS 中,agent 是一种可以感知环境变化并且通过内部执行机构反馈给环境的节点.在 MAS 的具体应用中,关键的一个问题是如何加强 agent 之间的有效合作,使 agent 与其他 agent 交换本地决策信息的同时能够根据其他 agent 的决策信息进行分析、推理,进而求得问题的最优解[1].

虽然将回溯搜索以及约束一致性检查两种基本思想引入到启发式方法中能够解决 MAS 问题,但是随着硬件和网络技术的发展以及分布式计算环境广泛地应用,越来越多支持 agent 自治性特点的建模、求解技术逐渐成为研究热点[2].对比这些建模求解方法,分布式约束优化问题(distributed constraint optimization problem,简称 DCOP)的建模方式及其相关算法已在 MAS 领域中得到一定应用并形成一种研究趋势,其相关成果一直是人工智能顶级会议《The Association for the Advancement of Artificial Intelligence》和《International Joint Conference on Artificial Intelligence》的热点,并设有多个专题对此进行讨论,国内也有很多学者致力于该类问题的研究,诸如约束满足问题的求解研究[3−5]等.

随着 DCOP 研究的深入,许多具备分布式求解特点的实际问题被建模成为 DCOP 进行求解.为了能够更直观地了解DCOP在实际问题中的应用特征,图1给出了一个具体的会议排班问题案例,问题中包含3个会议:$\{m_1, m_2, m_3\}$、3 个与会人员:$\{a_1, a_2, a_3\}$.为便于理解,假设每个会议持续时间为 1 个小时,与会者档期及所需参加会议见表 1.

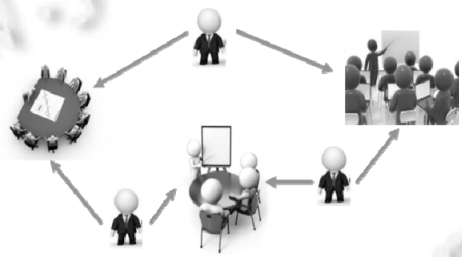


Fig.1 Three meeting schedule problem with 3 participants

图 1 3 人 3 会议安排问题

Table 1 Schedule time of participants

表 1 与会者档期时间

与会者	会议	档期(时)
a_1	m_1/m_3	8-9/10-11/11-12
a_2	m_1/m_2	9-10/10-11/13-14/17-18
a_3	m_2/m_3	9-10/11-12//13-14

会议安排问题具备如下特点:

- 每个与会者只关心个人会议安排情况,个人的档期与其他与会者无关.
- 会议时间的通信协商仅发生在同一会议与会者之间.
- 当部分与会者的档期发生变化时,需要及时获取新的会议安排.

结合该典型实例可以看出,DCOP 求解技术解决的一类实际问题通常具备如下特征:

- 自治性强:难以建立全局模型求解.每个子系统由于固定地理位置等原因,只能根据自身信息或者局部信息进行决策,使得系统无法获取全局信息,难以建立全局模型进行求解,如无线传感器网络.
- 限制性通信:难以获取全局约束.在不同的实际应用领域中,由于通信代价、通信隐私、通信范围的局限,使得通信仅能发生在相邻子系统间.整个系统难以搜集全局的约束,如社交网络.
- 时效性强:难以适应环境动态变化.一类实际问题对于求解的实时性要求较高.一旦周围环境发生局部

© 中国科学院软件研究所 http://www.jos.org.cn

变化时,通过更新全局信息将延迟求解速度,如灾难救援等问题.

对于这类问题,集中式求解方法的处理方式是:设置中央处理节点,将各节点的信息直接或间接发送至中央处理节点进行处理;最后,通过集中决策得到问题的解.然而,集中式求解算法主要存在以下 3 个方面的缺点:

(1) 容易暴露个体内部的隐私信息.比如无线传感器网络在军事中的应用,每个传感器的感知内容通过集中式的传递,会造成信息泄露等安全性威胁.

(2) 难以适应个体内部环境的动态变化.比如在会议安排问题中,个别与会者的档期发生变化时,中央处理节点要重新利用更长时间进行全局会议时间分配的调整;相应地,更多的额外信息会在个体之间进行传递.

(3) 较差的鲁棒性.中央处理节点需要收集到所有节点的消息后才会进行集中决策,如果决策过程中发生错误,需要从全局方面变更个体的内部状态.

相应地,DCOP 求解方法则适应于求解该类问题,其求解目的是获得问题的最优解或者次优解,并能使这些解在质量上得到一定的保证

本文第 1 节介绍 DCOP 的定义,并给出 DCOP 的求解流程.第 2 节根据 DCOP 求解流程对目前的 DCOP 算法进行分类,并结合具体的典型算法进行算法间的对比分析.第 3 节介绍 DCOP 相应的应用.最后,本文在总结中给出一些 DCOP 未来的扩展和研究工作.

1 分布式约束优化问题及求解过程

1.1 分布式约束优化问题

MAS 解决实际问题的方式可以理解为一种基于 agent 的协作问题,而分布式约束则可以描述领域对象的性质、相互关系、任务要求、目标,因此可以作为一种有效的方法表示这种 agent 间的协作关系.所以,研究者将分布式约束作为建模此问题的一个关键范式.分布式约束优化满足问题(distributed constraint satisfaction problem,简称 DCSP)作为早期求解 MAS 的技术,从表 2 中可以看出,虽然 DCSP 较集中式求解方法在在通信代价、隐私性、容错性以及并行性等方面有了不可替代的优势,但是其求解局限性使 DCOP 的研究逐渐成为重点.

Table 2 Comparison between centralized and DCSP algorithms

表 2 集中式与 DCSP 算法比较

求解类型	集中式	DCSP
通信代价	全局通信	局部通信
隐私性	暴露隐私	保护隐私
容错性	局部错误对全局影响大	仅需调整局部错误
并行性	难以实现并行	易实现并行
求解质量	解的质量较好	无解、多解
通信代价	全局通信	局部通信

表 3 是使用 DCSP 求解方法得到的图 1 中实例的两个解,从表 3 中可以看出,如果与会者想尽早结束会议,那么解 1 要好于解 2,在这种情况下,DCSP 求解方法难以在两个解中进行取舍.事实上,当问题中变量的所有赋值组合都不能满足所有约束时,DCSP 也无法给出相对较好的次优解.因此,多解或无解问题成为 DCSP 在实际应用中面临的困难.

Table 3 Solutions of DCSP

表 3 DCSP 求得的解

会议	与会者	开会时间(时)	
		解 1	解 2
m_1	a_1 & a_2	10-11	10-11
m_2	a_2 & a_3	9-10	13-14
m_3	a_1 & a_3	11-12	11-12

© 中国科学院软件研究所 http://www.jos.org.cn

造成 DCSP 算法中无解或者多解的原因,是由于约束只有满足和不满足两种状态,DCSP 相关解决方案的目的是找到满足所有约束的解.因此,多解之间不存在优劣比较,使算法无法在多个解中进行取舍.同理,只要一个约束是违反规则的,DCSP 相关解决方案无法根据解的优劣选择次优解.DCOP 在 DCSP 的基础之上,将约束状态从满足与不满足扩展到约束函数,约束函数是对约束更加具体的量化处理.DCOP 定义存在若干形式上有细微差异的版本[6−8].在本文中,我们采用 DCOP 的五元组模型:

定义 1(DCOP). 分布式约束优化问题是一个五元组$\langle A,X,\alpha,D,F\rangle$,其中,

- A 表示 agent 集合;
- $X=\{x_1,x_2,\ldots,x_n\}$表示 agent 包含的变量集合;
- $\alpha:X\to A$ 表示每一个 $x_i\in X$ 有且只属于一个 $a_j\in A$;
- $D=\{d_1,d_2,\ldots,d_n\}$表示每个变量有限取值空间的集合,对于每个 $x_i\in X$ 有且只有一个 $d_i\in D$ 与之对应;
- $F=\{f_1,f_2,\ldots,f_m\}$表示效应函数集合.一个效应函数 $f_{ij}:d_i\times d_j\to R^+\cup\infty$表示两个 agent 或两个变量间的关联关系.

定义 2(DCOP 的目标函数). 设 $F:D\to R$(R 是实数集)是一组效应函数之和,一个分布式约束优化问题的目标函数为

$$F(x)=\sum_{x_i,x_j\in X} f_{ij}(x_i,x_j),x_i,x_j\in D \tag{1}$$

定义 3(DCOP 的解). agent 通过选择变量的赋值使全局目标函数最小,这样的一组变量值分配情况称作一个分布式约束优化问题的解,即

$$X^*=\min F(x) \tag{2}$$

从 DCOP 的定义中我们可以看出,DCSP 是 DCOP 的特殊情况.

- 在约束方面,变量和约束分布在不同自治的 agent 中,但 DCSP 的约束形态仅为满足或不满足.只要使 DCOP 中的约束函数代价值为 0(不满足)或 1(满足),即可代表这两种形态.
- 在目标函数方面,如果使 DCOP 中所有的约束代价值之积为 1,则当前的变量赋值情况即为全局满足所有约束的一个解.

为便于研究,在目前的 DCOP 建模过程中,我们假设每个 agent 有且仅控制一个变量[9];同时,我们只考虑二元约束,即约束关系仅存在两个变量之间.

1.2 分布式约束优化问题求解过程

通过对 DCOP 求解算法执行过程的分析、总结,图 2 给出了 DCOP 算法的基本求解流程.

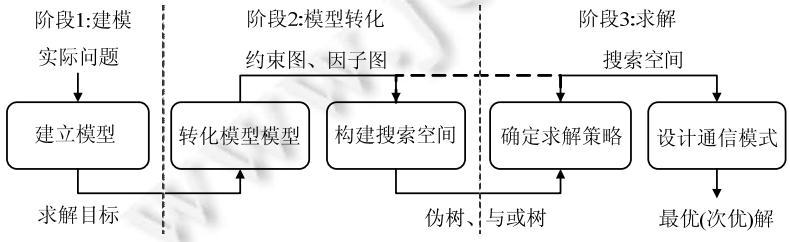


Fig.2 Basic process of solution method for DCOP

图 2 DCOP 求解方法基本流程

该流程主要分为 3 个阶段:第 1 阶段是将实际问题建模成为 DCOP,在流程中主要体现为模型的建立;第 2 阶段是对建模的 DCOP 进行预处理,包含模型转化和搜索空间的构建;第 3 阶段是算法求解,在流程中包括求解策略的确定以及通信模式的设计.

在 DCOP 求解过程中,不同算法之间的差异主要体现在模型建立、模型转换、搜索空间构建、求解策略、

© 中国科学院软件研究所 http://www.jos.org.cn

通信方式等过程中的解决策略上,不同的求解算法在一个或几个流程中采取不同的策略,形成不同的求解效果.

在模型的建立阶段,一个实际问题首先被建模成为一个五元组模型.然而,同一个问题通常有若干建模方式,不同的建模方式导致问题在形成 DCOP 模型时所确定的问题的 agent 划分、变量从属情况、约束划分、目标建立有所不同.这些区别将直接导致模型转化后形成的问题规模有所差异,进而影响算法的求解效率.在目前的大部分 DCOP 算法中,模型的转化是必要的;但是在一些算法中,模型转化后的图结构可以直接作为搜索空间,不需要进行树形结构的建立.

在模型转化中,一个 DCOP 一般被转化成为一个图结构.根据 DCOP 算法求解策略的不同,目前大部分 DCOP 算法通常将一个 DCOP 模型转化为约束图或因子图.图 3 给出了一个简单的会议安排问题经过建模以后转化成为的约束图 3(a)和因子图 3(b).两者最大的区别在于:约束图中只含有变量(agent)节点,而因子图中还包含函数节点.

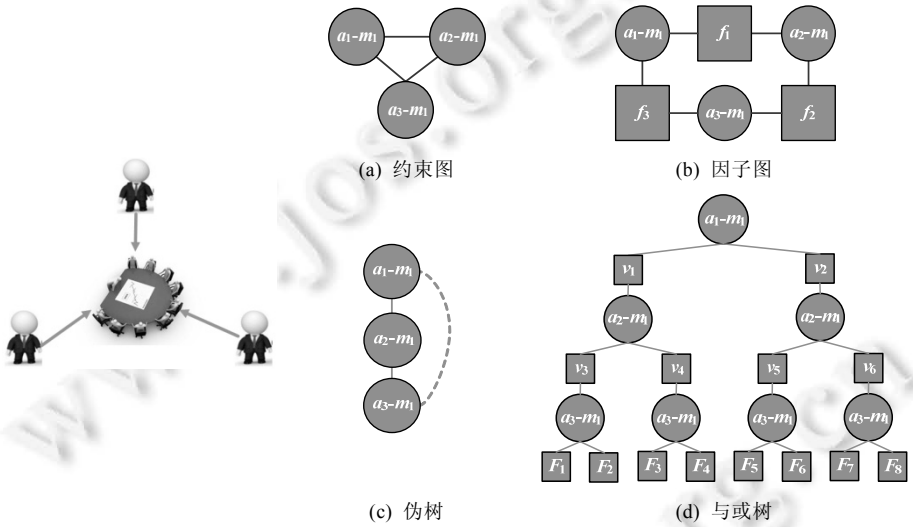


Fig.3 Transformation and the determination of search space based on DCOP model

图 3 DCOP 的模型转化及搜索空间的确定

搜索空间的构建,主要为了确定 agent 之间的次序关系,以满足消息传递的范围.目前,DCOP 根据求解策略的不同,主要在图结构的基础上将其构建成一棵伪树或者(pseudo tree)与或树(AND/OR tree),图 3(c)和图 3(d)分别为图 3(a)中约束图的伪树和与或树.伪树可以有效地提高搜索,其高效性使不同分支中节点之间没有约束限制,因此具有很强的独立性.这样对以一个约束图进行重构后,允许算法在独立的枝条上进行并行搜索.一个伪树边的集合被划分为两种子集:一种称为树边(tree edge),这些边是一个约束图通过深度优先搜索转化成伪树时的路径;另一种称为回溯边(back edge).树边连接的点为父节点和子节点,而回溯边连接的点称为伪父节点(pseudo parent)和伪子节点(pseudo children).与或树提供了问题中所有变量的解的组合情况,便于剪枝处理;同时,每个分支的独立性也满足算法进行并行搜索.

DCOP 完备算法通常采用两种求解策略:一种是搜索方法,如最佳优先、深度优先、回溯和分支界限搜索等;另一种是基于推理的方法,基于推理的求解策略允许 agent 计算从相邻 agent 中聚合约束的代价值以逐步减少问题规模.在一些非完备算法中,为了更快地获得次优解,随机策略、约束问题求解中的求解框架也相应地运用到了 DCOP 的求解过程中.

为了支持不同的求解策略,agent 之间的通信方式也有差异,在基于搜索的算法中,经常采用异步通信方式,而在基于推理的算法中,则常采用同步通信方式;此外,agent 之间的通信范围也有所不同,比如部分集中式算法中,信息会传送至中央节点,而完全分散式的算法则常常采用局部通信等.针对目前 DCOP 算法在流程各个阶段

© 中国科学院软件研究所 http://www.jos.org.cn

采用的不同机制,本文对其中的关键策略进行了归纳总结,见表 4.

Table 4 Critical strategies adopted in the solution progress of DCOP algorithms

表 4 DCOP 算法在求解流程中采用的关键策略

求解流程	关键策略
模型建立	三元组模型:$\langle X,D,F\rangle$; 五元组模型:$\langle A,X,\alpha,D,F\rangle$
模型转化	约束图/因子图
搜索空间	约束图/因子图/伪树/与或树
求解策略	基于搜索(最佳优先/深度优先/分支界限/回溯); 基于推理/随机策略/基于约束问题求解框架
通信模式	同步局部通信/异步局部通信/集中式通信

2 DCOP 求解算法分析

为了使算法能够解决 DCOP,我们还需要让 DCOP 算法满足以下几个需求:

(1) 完备性.寻找满足所有约束或违反约束最少并且代价值最优的解.

(2) 分布性.支持 agent 分布性特点,保护 agent 的内部隐私.

(3) 异步性.保证各 agent 不必相互等待,确保系统的健壮性.

为了满足以上求解条件,提出了一系列优秀的 DCOP 算法,如 ADOPT[10]、DPOP[11]算法等.DCOP 算法根据 agent 对于消息的响应与处理方式,分为同步与异步两类.在同步算法中,每个 agent 当且仅当接收到特定消息后,才可以进行本地的信息处理与决策.虽然同步机制保证了算法求解的一致性,但却增加了 agent 的等待时间.相比之下,异步算法虽然避免了 agent 过多的等待时间,但却难以保证算法求解过程中的一致性,因此通常需要进行回溯求解.为了能够更好地了解这些算法的异同,本文并不从同步与异步的角度对算法进行归类分析,而是在文献[12]的基础上,根据算法的执行过程对完备与非完备算法进行进一步的划分.

完备与非完备算法的最大区别在于求解质量的不同:完备算法求得的是全局最优解,而非完备算法求得的是最优解或是次优解.造成求解质量不同的主要原因在于算法选取的求解策略有所差异,这将直接影响 agent 的通信范围、消息传递的类型、数量以及消息处理的时间等执行过程,进而使 agent 根据局部信息给出的决策不同,最终影响 agent 内部变量的赋值情况.通常,为了能够找到问题的最优解,完备算法需要消耗过多的计算和通信资源,这些消耗的资源会随着问题规模的扩大而呈非线性增长.而非完备算法则能够有效地避免这些问题,又因为有些实际问题对于求解的实时需求高于精度需求,或因实际问题受物理条件的制约(如存储空间小),此时,采用拥有良好稳定性和健壮性的非完备算法可以在时间、资源代价较小的情况下求得次优解.

完备算法主要分为部分集中式算法和完全分散式算法.两类算法最大的区别在于通信的模式不同.

- 部分集中式算法融入了集中式求解特点,一些消息将会被传递至中央处理节点;而后,中央处理节点将会集中处理消息,并根据这些消息进行变量的赋值决策.
- 在完全分散式算法中,消息仅在相邻 agent 间进行传递;然后,每个 agent 根据局部消息进行决策.

因此,通信模式的不同导致消息传递的范围、agent 对于消息的接收和处理方式不同,造成两种算法在求解效率上有所差异.虽然消息的集中处理会使 agent 提供的变量赋值决策更加准确,但消耗更多的时间;而分散式则与其相反.部分集中式求解算法的代表是 OptAPO 算法[13],由于该类算法难以使 agent 间进行完全意义上的并行决策,所以以往的研究重点主要集中于完全分散式算法.在完全分散式求解算法中,由于求解策略的不同,一类算法采用搜索策略,一类算法采用动态规划策略.

针对非完备算法,本文从质量保证角度进行了归类分析.非完备算法主要分为无质量保证和基于质量保证两类.两类算法最大的区别在于次优解的可靠程度和保障程度.虽然在两类算法的执行过程中,每个 agent 的变量赋值决策都基于局部相邻 agent 的赋值情况,但是无质量保证的非完备算法通常采用随机决策的方式,以一定的概率对变量进行赋值的变更;而基于质量保证的非完备算法则是通过上下界或局部最优等限制条件给出更

© 中国科学院软件研究所 http://www.jos.org.cn

严谨的赋值变更决策.虽然无质量保证算法会降低求解时间,但这种求解策略容易导致算法陷入局部最优,也使求得的解难以得到质量保证,在许多实际问题中,无质量保证的解的实际意义不强,比如 DSA[14]等算法.因此,目前的研究热点主要集中在基于质量保证的算法中.在该类算法中,由于求解策略对于所需搜索空间的构建以及空间内的划分方式不同,一类算法在执行前就可以根据搜索空间的区域划分情况以及问题规模来确定求解质量,我们将其归类为 off-line 算法;另一类算法需要根据每次对搜索空间内所有子问题的求解结果来确定求解质量,我们将其归类为 on-line 算法.综上,本文形成了如图 4 所示的 DCOP 算法的分类体系.

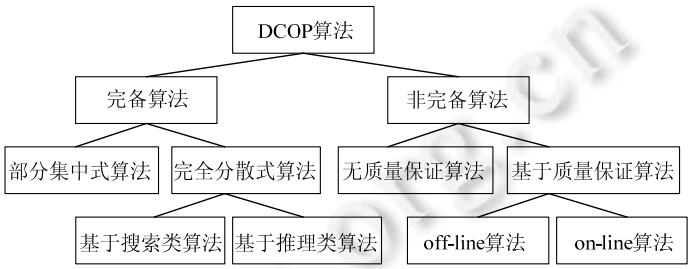


Fig.4 Classification of DCOP algorithm

图 4 DCOP 算法的分类

本文的第 2.1 节和第 2.2 节将对基于搜索以及推理类的算法进行详细的描述.本文的第 2.3 节和第 2.4 节将对基于质量保证的 off-line 算法和 on-line 算法进行详细的描述.

2.1 基于搜索的典型算法

这类算法主要采用搜索的求解策略进行解的构建,不同的搜索策略将会影响算法的求解效率.同时,算法采用树形的搜索空间,使每个子树根节点之间不存在约束,便于子树中 agent 进行异步搜索.这类算法容易造成 agent 间传递大量消息,因此,很多算法的改进目标就是避免大量的消息传递.搜索效率较好的算法主要采用最佳优先搜索或者深度优先分支界限搜索策略.

2.1.1 最佳优先搜索

最佳优先搜索可以看成是对宽度优先搜索的一种改进.与宽度优先一样,最佳优先也总是保留一组继续向下搜索的可选择路径.此外,最佳优先的一个重要原理就是根据评价函数的计算结果中代价最小的那条路径向下搜索.在搜索过程中,通过不断地放弃代价较大的路径,从而找到代价最小的问题求解答案.由于最佳优先搜索在寻找最优路径时采用的是最小下界,而最小下界的计算可以不依靠全局信息,因此,这种搜索策略更适合异步搜索.

然而,这种搜索策略带来的问题在于agent在搜索过程中无法保证当前得到的局部解是否会引导全局最优,因此,有可能会舍弃导致全局最优的局部解.在这种情况下,回溯搜索则必不可少,目的是探寻已经被舍弃的局部解.而这种回溯的频繁发生会导致异步搜索里 agent 之间传递的消息数量增加.

在 DCOP 算法中,ADOPT,ADOPT-ng[15]采用的是这种搜索策略.ADOPT 是第一种在伪树结构中进行完全异步搜索的DCOP算法,该算法通信的消息数量随着问题规模增加呈指数增长[16];ADOPT-ng是ADOPT的改进算法,该算法通过在消息中引入包含阈值和约束关系等更多的信息量,帮助算法提高搜索速度、降低回溯频率,从而达到提升求解效率的目的.

2.1.2 深度优先分支界限搜索

分支界限法源于运筹学中的整数规划问题,是使用最为广泛的约束优化方法.其具体做法是:算法以深度优先的方式搜索解空间,而且如标准回溯方法一样执行,只是在为变量赋值时,需要计算一下当前的代价函数值,如果代价函数值越界,那么当前搜索路径以下的子树将被立刻修剪掉.开始时,算法通常先找到一个次优解,随着搜索过程的进行不断被修改.分支界限法的效率取决于代价函数值的大小以及是否能够更早地发现一个好

© 中国科学院软件研究所 http://www.jos.org.cn

的边界.

与最佳优先搜索策略相比,深度优先分支界限搜索可以有效地避免反复重建内存中已经舍弃的局部解,因此可以有效地降低 agent 之间由于回溯搜索而进行的大量消息传递,进而提高搜索效率.

在 DCOP 算法中,NCBB[17],AFB[18]以及 BnB-ADOPT+-AC[19]采用的是这种搜索策略.NCBB 令不同的 agent 并行搜索不相交的搜索空间,以使分支界限搜索策略可以更有效地剪切无效的搜索子树,但是 agent 之间的信息传递以及根据信息进行界值计算的过程是同步执行的;AFB 在搜索过程中引入了当前局部赋值(current partial assignment,简称 CPA)消息,agent 可以异步地根据该消息进行同步的代价值界限估计,进而提早检测出回溯需求,但是搜索过程是同步的.因此,NCBB 与 AFB 并非完全异步的完备算法,这增加了 agent 的等待时间,影响了求解效率.BnB-ADOPT 采用的是 ADOPT 中的消息传递和通信框架,利用深度优先分支界限搜索策略在与或树中进行解的构建;BnB-ADOPT+-AC 在 BnB-ADOPT 的基础上去除了冗余信息并在搜索中保持软弧一致性,增强了 agent 之的通信质量.

2.2 基于推理的典型算法

基于推理的 DCOP 算法主要采用动态规划求解策略进行解的构建,DPOP 是最先采用这种思想进行求解的完备算法,后续算法均在 DPOP 基础上进行的改进,而求解策略仍然是动态规划.与搜索类算法相比,这类算法的消息数量是呈线性增长的,而消息容量却容易呈指数增长,因此,大部分基于 DPOP 的改进算法主要采用更合理的数据结构,减少消息对空间的占用.

2.2.1 动态规划求解过程

利用动态规划进行求解的基本算法 DPOP 主要采用伪树结构,并且从小范围开始构建局部解,并适用到越来越大的范围,最终得到全局解.这类算法主要包含以下 3 个求解阶段:

- 伪树生成阶段:DPOP 算法利用深度优先搜索方式,结合现有的分布式伪树构造算法将约束图转化成一棵伪树.
- UTIL 消息传播阶段:UTIL 消息是一系列值的矩阵,由子节点发送至父节点,矩阵中每一维代表一个 agent 的一个变量.UTIL 消息在沿着树形结构至底向上传递的过程中,列出了和全局优化有关的所有子问题所有不同取值组合情况以及代价值.
- VALUE 消息传播阶段:VALUE 消息由父节点传送至子节点,包含父节点内部变量的最佳取值.在该阶段中,根节点率先根据获取的 UTIL 消息进行内部变量的最佳赋值,然后,子节点依次根据父节点传送的 VALUE 消息以及内部存储的 UTIL 消息,为节点内的变量进行最佳赋值,直到发送到叶子节点为止.

DPOP 算法只需要线性的消息数,但消息的容量却随着约束图 induced width 的增加呈指数增长[20].产生这种消息容量急剧膨胀的原因主要由于许多实际问题中包含的硬约束都可以用来协助过滤无用解的空间占用,但是动态规划求解策略却没有利用硬约束对求解空间进行有效的修剪,因此,UTIL 消息中冗杂了很多无用解.为了能够解决过量的空间占用问题,改进的 DPOP 算法主要从两方面入手:一方面,借鉴搜索过程中的剪枝策略减少无用解;另一方面,通过平衡消息数量与空间占用、限制 UTIL 消息矩阵维度等手段,避免容量的指数增长.

2.2.2 改进的动态规划求解算法

O-DPOP[21]借鉴了搜索类算法中消息数量随伪树 induced width 呈线性增长的特点,令 agent 按照最佳优先顺序对变量的取值进行探索,减少了类似于 DPOP 算法中频繁出现不可能解的糟糕情况.O-DPOP 算法在 UTIL 消息传播阶段中,利用(ASK/GOOD)消息代替了原算法中的 UTIL 消息.在该阶段执行过程中,效应值将沿着伪树结构,不断迭代地自底向上进行传播,在每个迭代过程中,每个节点都要向子节点反复发送 ASK 消息询问子节点的取值情况,并根据子节点发送回来的 GOOD 消息对本节点的当前取值进行效应值评估,直到可以确认当前取值为最佳后,该节点再向高级节点发送 GOOD 消息.当伪树中的根节点可以根据足够的信息进行赋值决策时,该阶段的迭代过程结束.虽然该算法在最坏情况下与 DPOP 的执行效率相同,但是平均执行效果都优于 DPOP 算法.

MB-DPOP(k)[20]算法利用 bound memory 的思想限制了 UTIL 消息的最大容量,在具体执行过程中,MB-

© 中国科学院软件研究所 http://www.jos.org.cn

DPOP(k)基于 DPOP 算法,在伪树生成阶段与 UTIL 消息传播阶段之间添加了标识阶段.在该阶段中,MB-DPOP(k)根据设定的参数 k,利用 cycle-cuts 技术中的有界传播思想,将伪树中的节点划分成以 cycle-cut 节点作为根节点的若干子树.其中,每个子树的 induced width 小于或等于 k.这样的划分模式虽然使算法中传递的 UTIL 消息矩阵容量不超过 d^k(d 为变量的取值空间大小),但也使其不能像 DPOP 算法一样在全局上执行完整的推理,进而降低了每个子树内变量赋值的准确性.针对这一问题,MB-DPOP(k)通过增加节点间的消息传递数量来平衡消息容量减小带来的信息损失.

H-DPOP[22]在算法中借用约束决策图(constrained decision diagram,简称 CDD)来舍弃不可能导致全局最优的变量赋值组合.约束决策图是对二元决策图的一种一般化推广,其最大特点就是利用紧凑的数据结构将约束推理和一致性技术结合在一起,更好地表示了 n 元约束关系.在 DPOP 算法的第 2 阶段中,H-DPOP 用 CDD 消息代替 UTIL 消息,一个 CDD 消息包含以下 3 部分内容:

- *CDDTree*:CDDTree 表示了消息中所有变量有效的取值组合.CDDTree 中的每一层有且仅包含 1 个变量,每一个路径代表变量间取值的一个组合.
- *UtilArray*:UtilArray 存储了 CDDTree 中每一个路径下的效应值.
- *DimensionArray*:DimensionArray 存数了该消息中涉及的变量集合.

2.3 off-line典型算法

大部分 off-line 算法主要根据定义的标准对搜索空间进行区域划分,形成若干独立区域,并在求解过程中保证这些独立区域内的变量赋值最优.这类算法的求解特点是,能够根据问题规模(变量、约束数量等)给出解的质量保证.

通常,off-line 算法的求解过程主要分为以下 3 个阶段:

- 初始化阶段:agent 之间通过传递初始信息确定在所给标准下所有独立划分区域的集合.
- 优化阶段:每个独立区域内的变量在保证边缘变量当前值的情况下,计算得到当前区域内的变量最优赋值.
- 执行阶段:独立区域间进行信息通信,并根据信息调整区域内赋值情况以获得全局.

其中,第 2 阶段和第 3 阶段循环执行,直到算法达到终止条件为止.结合图 2 的流程分析可知,DCOP 求解算法的搜索空间主要为图结构和树结构,因此,目前 off-line 算法中的区域划分标准也主要基于这两种数据结构.

2.3.1 基于图的区域划分

该类算法主要包括 K-Opt[23],DALO-t[24]以及 C-Opt[25]算法.这些算法之间的主要区别在于划分的标准不同.

定义 4(*K* 节点区域最优分配). 给定一个 DCOP,若存在这样的一组赋值集合 A,使得改变 A 中任意不超过 K 个 agent 赋值后,所得到的新赋值集合 A'的全局代价值都小于 A,那么我们称 A 为当前 DCOP 的 K 最优分配. K-Opt 是第一种提供解的质量保证的 off-line 算法.该算法利用节点数量 k 作为标准,对约束图进行独立区域划分.该算法求得的变量赋值集合 A_{kop} 的代价值 $R(A_{kop})$与全局最优分配为 A^*的代价值 $R(A^*)$的关系为(n 为变量数量,m 为最小约束参数)

$$R(A_{kop}) \geqslant \frac{\binom{n-m}{k-m}}{\binom{n}{k}-\binom{n-m}{k}} R(A^*) \tag{3}$$

定义 5(*T* 距离区域最优分配). 给定一个 DCOP,若存在这样的一组赋值集合 A,使得改变A中任意距离不超过 T 的 agent 的赋值后,所得到的新赋值集合 A'的全局代价值都小于 A,那么我们称 A 为当前 DCOP 的 T 最优分配.DALO-t(T-Opt)则是利用图距离 t 作为标准,对约束图进行独立区域划分.该算法得到的变量分配 A_{top} 的代价值 $R(A_{top})$与全局最优分配为 A^*的代价值 $R(A^*)$的关系为(n 为变量数量,m 为最小约束参数)

$$R(A_{top}) \geqslant \frac{(m+t-1)}{n} R(A^*) \tag{4}$$

© 中国科学院软件研究所 http://www.jos.org.cn

C-Opt 对 *K*-Opt 以及 *T*-Opt 两类算法中采用的区域划分标准进行了一般化,其划分目的是找出变量距 $t\in T_n$ (其中,$T_n=\{1,2,\ldots,n\}$),且内部变量数目不超过 s 的区域集合.同时,该算法给出了基于区域划分的 off-line 算法求解质量保证的普遍性公式,其中,$|C|$为独立区域数量,cc_*为区域内最小约束数量,nc_*为区域间最小约束数量. $R(A_{cop})$为算法求得的最优变量赋值 A_{cop} 的代价值,$R(A^*)$为与全局最优分配 A^*的代价值:

$$R(A_{cop})\geqslant\frac{cc_*}{|c|-nc_*}R(A^*) \tag{5}$$

2.3.2 基于树的区域划分

P-Opt[26]借助约束图生成的伪树结构,通过移除树种的部分边,将问题转化成一个可以利用完备算法求解的问题规模,进而提升求解效率.该算法主要分为两个阶段:

- 阶段 1:通过移除伪树中的部分边,生成一棵子树,使该子树代表的子图的 induced width 在给定的参数 p 之内.
- 阶段 2:借助任意完备算法,在该子树上求解问题的解.

若用 $w(G,o)$表示约束图的宽度,$r_{\max}$ 表示变量间约束函数的最大值,$R(A^*)$为与全局最优分配 A^*的代价值, $R(A)$为算法求得的最优变量赋值 A 的代价值,则该算法求得的解的质量为

$$R(A^*)-R(A)\leqslant r_{\max}\times\sum_{k}^{w(G,o)-p}(|X|-(k+1)) \tag{6}$$

从以上次优解与最优解的关系中也可以看出:通过 off-line 算法得到的次优解仅与问题的划分标准和问题规模有关,与算法的运行过程无关.即,在算法运行前就可以确定解的质量范围.

2.4 On-Line典型算法

On-Line 算法的求解思想与完备算法类似,主要根据子问题的求解情况逐步构建更优的全局解,因此,算法每次循环的执行结果不仅用来寻找次优解,同时也在不断为次优解提供上下界.与完备算法不同的是,为了能让 on-line 算法适合快速求解大规模的 DCOP,在求解过程中,不同的求解策略导致子问题的构建方式以及子问题间的通信方式会有所差异,因此,on-line 算法对求解质量的保证也有一定的差别.

2.4.1 基于 Divide-and-Coordinate 框架

DaCSA[27]与 DaQED[28]是利用 Divide-and-Coordinate(DaC)框架进行求解的典型 on-line 算法.该算法主要分为以下 3 个阶段:

- 分解阶段:以 agent 为单位分解原问题,使每个 agent 单独成为一个子问题.
- Divide 阶段:每个 agent 并行独立地对内部变量的赋值情况进行协调,直到内部赋值最优.
- Coordinate 阶段:agent 之间通过信息的交换来改变子问题内变量的赋值.

在 DCOP 中,约束有两种存在方式:仅属于 agent 内部变量之间的约束为内部约束;连接 agent 之间的约束为接口约束.分解一个二元约束 DCOP 时,由于接口约束被两个来自不同 agent 的内部变量共享,因此,两种算法将接口约束平均分配到所属的两个不同 agent 中.这种分配方式会出现 agent 在 Divide 阶段并行赋值调整时,由接口约束连接的两个变量因分属在不同 agent 中而产生赋值冲突.为了能使 agent 之间在 Coordinate 阶段更好、更快地达到赋值一致性,两种算法均采用了拉格朗日对偶分解以及次梯度方法.不同的是两种算法在分解前对问题进行线性转化时所采用的编码方式:DaCSA 采用的是线性编码方式,而 DaQED 采用的是二次编码方式.

算法循环执行 Divide 阶段与 Coordinate 阶段,直到超过预定时间或求得的解落在一定误差范围内,在每次循环中,算法会根据所有子问题的解,进行上界值(*BestUB*)和下界值(*BestLB*)的估算,以确保求解质量.

2.4.2 基于抽样方法

在 ADOPT 等完备算法中,每个 agent 都根据伪树中相邻 agent 内变量的赋值状况以及当前代价值等信息,更加精确地对其内部变量的取之空间进行有效的搜索.虽然更精确的信息有助于提高搜索质量,但往往搜集信息的过程以及赋值的决策都需要消耗大量时间.为了能够提高搜索效率,基于抽样的算法首先将每个变量的每个可能取值看成一个样本,然后,借助于有限的信息首先对样本进行筛选,最后再通过概率选取等手段在可取的

© 中国科学院软件研究所 http://www.jos.org.cn

样本空间内进行变量赋值,进而削弱了决策和信息搜集带来的时间消耗.DUCT[29]以及 D-Gibbs[30]是利用该思想求解 DCOP 的典型算法,不同样本数量的选择以及不同的迭代次数使算法得到的解的质量有所变化.

DUCT 是一种分布式 UCT 算法.该算法首先构建一棵伪树,伪树中的每个节点都包含该变量所有取值下的样本.从根节点开始,伪树中的每个节点都会选择一个样本,并将样本中变量的取值信息包含在 CONTEXT 消息中依次传递至孩子节点.每个节点都包含 2 个计数器,分别记录在当前 CONTEXT 消息下,样本 d 被选取的次数 $\tau(CONTEXT,d)$以及获取的同一 CONTEXT 消息次数$\tau(CONTEXT)$.同时,节点分别根据 CONTEXT 消息以及样本 d 计算最小代价值 $\hat{u}(CONTEXT)$ 和 $\hat{u}(CONTEXT,d)$.最后,节点根据信息判断置信区间,选择合适的变量取值.

D-Gibbs 是分布式的 Gibbs 算法.该算法是一种基于马尔可夫链的蒙特卡罗算法,用于联合概率分布的近似估计,主要应用在需要进行最大后验概率估计的问题中.为了能够利用 Gibbs 求解 DCOP,D-Gibbs 首先将 DCOP 映射成为一个最大后验概率估计问题.与 DUCT 相同的是,该算法仍然是将问题转换成伪树后,自上而下进行消息传递;不同的是,每个节点进行变量选取的过程有所不同.在 D-Gibbs 算法中,伪树中的每个节点记录样本选取的次数 t 以及获取当前最优变量赋值的迭代次数 t^*.同时,节点分别计算以往迭代过程中出现的最优代价值与当前赋值下的代价值和当前最优代价值的差Δ与Δ^*.最后,算法根据不同迭代过程中得到的与Δ与Δ^*来决定是否改变变量的赋值情况.

2.4.3 基于 Max-Sum 算法

这类算法以 max-sum 算法为基础.Max-Sum 算法是一类消息传递算法,通过将问题转化成为一个因子图(如图 3(b)所示)并在因子图中的两种节点之间传递不同消息来寻找问题的最优解.其中,每个节点都要等待非目标节点外的其他所有相邻节点传递消息后,再根据消息内容进行赋值估计,最后将估计结果传向目标节点.当算法收敛时,每个节点计算代价值,进而得到全局解.算法根据得到的解,提供后验保证.

Bounded max-sum(BMS)[31]算法是能够提供解的质量保证的 max-sum 算法.该算法主要包含 3 个阶段.

- 松弛阶段:首先,算法在原有因子图的基础上为图中的每一个边分配一个权重,这个权重用来衡量一个因子对于最优解决方案的影响程度;然后,算法找到一棵该权重下的最大生成树;接下来,原问题被转化成了一个拥有最大生成树的无环因子图.
- 求解阶段:利用 max-sum 算法求解松弛阶段中转化成的无环因子图.
- 边界阶段:对获得的解进行边界限定.

Weak Improved BMS[32]与 RN-BMS[33]算法都是在 BMS 的基础上进行的改进,与 BMS 不同的是:两种算法都是在松弛阶段中,对因子图的转化方式不同.

Weak Improved BMS 通过计算上界值,提高算法性能:在松弛阶段,算法将属于因子图,但是不属于生成树的约束进行改造,然后再将因子图转化成一个无环图.RN-BMS 在 BMS 的基础上,通过引入信任度的概念提高变量赋值的准确性.在具体执行过程中,RN-BMS 对问题中每两个相邻变量之间设置信任度,表示一个变量对邻变量中每个取值是否导致最优解的可能程度.结合信任度及约束函数,算法在松弛阶段对因子图进行转化.

2.5 其他算法

在 DCOP 算法的研究中,除了在求解效率、求解质量等方面对算法进行改进外,也有研究从实际应用、执行环境等角度给出了相关改进策略.比如 P-DPOP[34],P^2-DPOP[35]等算法针对保护 agent 之间通信隐私,在 DPOP 算法的基础上进行了改进;GD-Gibbs[36]利用 GPU 对 DCOP 中的子问题进行并行求解,以提高算法执行效率;$MO\text{-}DPOP_{LP}$[37]在 DPOP 的基础上进行改进,主要针对多目标 DCOP 进行求解.

2.6 几种DCOP算法比较

从图 4 中的算法分类以及上述的算法分析中可以看出:不同类别的 DCOP 算法存在各自的求解特点和适用性,也存在各自的优点和局限性.表 5 和表 6 是从 DCOP 的求解流程角度,结合具体的 DCOP 算法,分析了算法之间在各流程环节采用的求解策略.表 7 是从消息数量、消息容量两个方面对完备类典型算法的定性对比,表 8 是从完备、求解完备性这 3 个方面对各类典型算法的定性对比.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 5 Solution strategy of some typical complete algorithms

表 5 几种典型完备算法的求解策略

算法	ADOPT	ADOPT-ng	DPOP	MB-DPOP
模型建立	$\langle X,D,F\rangle$	$\langle X,D,F\rangle$	$\langle X,D,F\rangle$	$\langle X,D,F\rangle$
模型转化	约束图	约束图	约束图	约束图
搜索空间	伪树	伪树	伪树	伪树
求解策略	最佳优先搜索	最佳优先搜索	动态规划	动态规划
通信机制	异步局部通信	异步局部通信	同步局部通信	同步局部通信

Table 6 Solution strategy of some typical incomplete algorithms

表 6 几种典型非完备算法的求解策略

算法	DSA	*K*-Opt	DaQED	D-Gibbs
模型建立	$\langle X,D,F\rangle$	$\langle X,D,F\rangle$	$\langle X,D,F\rangle$	$\langle A,X,\alpha,D,F\rangle$
模型转化	约束图	约束图	约束图	约束图
搜索空间	约束图	约束图	约束图	伪树
求解策略	随机搜索	区域最优	DaC 框架	抽样策略
通信机制	同步局部通信	异步局部通信	同步局部通信	同步局部通信

Table 7 Qualitative comparison of some typical complete algorithms

表 7 几种典型完备算法的定性比较

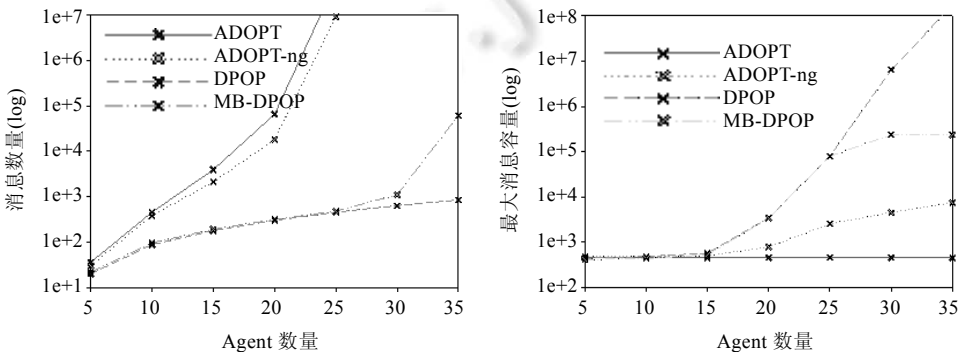
算法	ADOPT	ADOPT-ng	DPOP	MB-DPOP
消息数量	多	较多	较少	少
消息容量	小	较小	大	较大

Table 8 Qualitative comparison of some typical incomplete algorithms

表 8 几种典型非完备算法的定性比较

算法	DSA	*K*-Opt	DaQED	D-Gibbs
求解质量	差	较高	较高	高

为了更准确地对比算法之间的求解优势和性能区别,针对表 4 及表 5 中的算法,本文给出了定量的分析.由于 DCOP 算法的求解侧重有所不同,算法在对比分析时,不能仅考虑单方面测试指标来评析算法优劣,例如同步与异步的 DCOP 算法,不能仅通过执行时间、求解质量进行算法性能衡量;再者,有些算法通过增加通信量或消息容量来提升求解效率,以此换来快速的求解速度,因此,本文在定量分析中,从消息数量、容量、求解时间、求解质量这几个方面对不同种类算法进行综合的实验对比.图 5 是 4 种完备算法的对比,其中,图 5(a)是算法在消息数量方面的比较,图 5(b)是算法在最大消息容量方面的对比.



(a) 消息数量 (b) 最大消息容量

Fig.5 Performance comparison of some typical complete algorithms

图 5 几种典型完备算法的性能比较

© 中国科学院软件研究所 http://www.jos.org.cn

图 6 是 4 种非完备算法在求解质量方面的对比.图 7 是几种完备算法与非完备算法在求解时间方面的对比.本文采用的是变量(agent)数量分别为 5,10,15,20,25,30,35 的随机图染色问题,每个测试结果均为同一测试用例在同一算法在中运行 10 次的平均值.

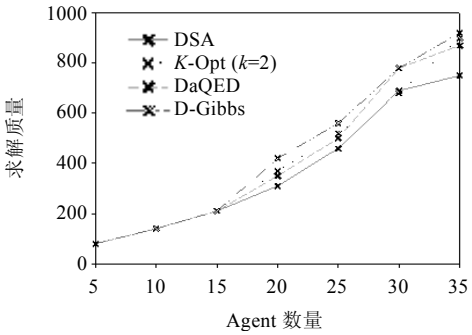


Fig.6 Performance comparison of incomplete algorithms

图 6 非完备算法的性能比较

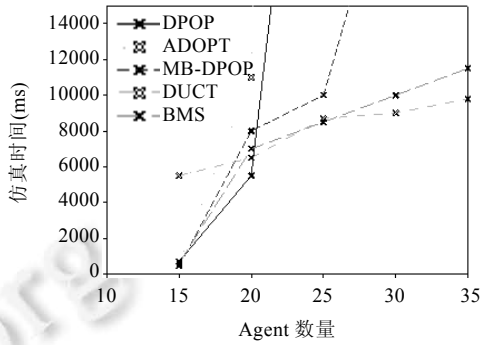


Fig.7 Runtime comparison of complete and incomplete algorithms

图 7 完备与非完备算法的运行时间比较

从图 5 中可以看出:ADOPT 算法执行过程中的消息数量随着 agent 数量的增加呈指数增长;而 ADOPT-ng 中消息的增长量相对于 ADOPT 有明显的减少,但在消息容量上有所增加,这是因为 ADOPT-ng 通过增加信息量提高了搜索效率造成的.DPOP 与 MB-DPOP 的消息数量均为线性变化,但 DPOP 是以最大消息容量的指数增长作为代价,而 MB-DPOP 有效地降低了消息容量在空间方面的开销.总体而言,搜索类算法通信频繁,基于推理类算法则注重消息传递的质量,平衡消息数量增长和消息容量对于空间的需求是提高两类算法执行效率的改进方向.

从图 6 中可以看出:DSA 算法的求解质量在所选取的非完备算法中最差,因为算法基于一种随机策略来改变变量赋值.当 k=2 时,K-Opt 算法的求解质量基本介于 DaQED 和 D-Gibbs 算法之间.事实上,随着 k 的取值增加,算法得到的解的质量会相应提高,但所花销的时间也会随着增加.DaQED 与 D-Gibbs 两种非完备算法作为近年来新提出的算法,显示出了极好的求解效果.从图 7 中可以看出:非完备算法在解决 agent 数量较多的 DCOP 时,所花费的时间明显小于完备算法.

3 DCOP 的应用

除了前文给出的会议安排问题实例,DCOP 同时适用于其他 agent 间的协作问题.这类问题通常难以建立全局模型,或者难以获取全局约束,并且面临环境动态变化等问题.例如,在无线传感器追踪网络应用实例中,每个传感器可感知搜索范围内所有目标,但同一时间仅能追踪目标中的一个,一个目标的位置要由多个传感器共同确定.然而,因为每个传感器的地理位置是固定的,只能根据自身信息或者局部信息进行决策,使得系统无法获取全局信息,难以建立全局模型进行求解.此时,如果以被追踪的目标作为 agent,将同时追踪到该目标的传感器作为变量集合从属于不同 agent 中,就初步形成了该实例的一个 DCOP 模型.

DCOP 的另一个实际应用是分布式资源分配问题.资源分配中将任务或资源分配给 agent,资源的有限性导致 agent 间存在约束关系.文献[38]中给出了形式化的分布式资源配置问题,体现了分布式资源分配问题的动态性和分布性,并提出了一般性的利用分布式约束满足技术的求解策略,agent 采用通信方式,协同地分配共享资源.目前,该问题的 DCOP 形式化定义仍是一项研究热点,文献[39]给出了一种分布式资源分配问题的形式化定义,类似于组合拍卖问题的选择[8]给出了该问题的另一种形式化定义,但是对 agent 内的变量情况和变量间存在的约束做出了相应的假设.

© 中国科学院软件研究所 http://www.jos.org.cn

4 结束语

在多 agent 系统中,每个自治的 agent 之间通过约束相关联,DCOP 的求解目的就是寻找到满足这些 agent 间约束一致行为的最优组合.求解大规模的组合问题通常是 NP 难问题,而这类问题作为一类模型被广泛应用在实际生活中.本文较为全面地评述了 DCOP 的相关研究,给出了 DCOP 的定义、求解流程、算法分析以及 DCOP 的实际应用等.

自 DCOP 提出以来,DCOP 算法的研究成为一项研究热点,大部分算法的提出都是根据执行过程中的某一方面(诸如存储方面、通信方面)有针对性地提高求解性能.然而,虽然算法的求解效果可观,但这些算法更多地集中于完备类别.目前,虽然国内对 DCSP 及其算法的研究已取得了一定成果,但在 DCOP 方面仍有欠缺,影响力较小.

另一方面,随着近年来多agent系统应用范围的拓宽,DCOP的实际应用需求也随之加强.目前,DCOP算法的测试主要采用模拟仿真形式,虽然效果可观,但应对条件严格的实际问题仍有很大不足;其次,如何更好地对实际问题进行 DCOP 建模以便于算法求解,如何解决问题中更复杂的软约束,如何平衡 agent 之间通信数量与质量,如何应对环境动态变化对模型的实时性的影响以及隐私性带给算法的安全隐患等实际问题也亟待解决.

最后,通过对近 10 年 DCOP 算法的统计可以看出,给予质量保证的非完备算法俨然正成为一项研究热点.这些问题都为我们未来的研究提供了方向.

References:

[1] Lesser V, Corkill D. Challenges for multi-agent coordination theory based on empirical observations. In: Proc. of the AAMAS. Richland: IFAAMAS, 2014. 1157−1160.

[2] Marconi M, Helmut P. Multi-User eco-driving training environment based on distributed constraint optimization. In: Proc. of the AAMAS. Richland: IFAAMAS, 2013. 925−932.

[3] Liu JM, Han J, Tang YY. Multi-Agent oriented constraint satisfaction. Artificial Intelligence, 2002,136(1):101−144. [doi: 10.1016/S0004-3702(01)00174-6]

[4] Han J, Chen EH, Cai QS. Multi-Level strategy for maintaining arc consistency in problem solving and its implementation. Ruan Jian Xue Bao/Journal of Software, 1998,9(8):622−627 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/9/622.htm

[5] Chen EH, Zhang ZY, Wang XF. A fast recognition algorithm of CRC constraint networks. Ruan Jian Xue Bao/Journal of Software, 2002,13(5):972−979 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/972.htm

[6] Arnon N, Alon G, Amnon M. Concurrent forward bounding for distributed constraint optimization problems. Artificial Intelligence Archive, 2012,193:186−216. [doi: 10.1016/j.artint.2012.09.002]

[7] Zivan R, Meisels A. Message delay and discsp search algorithms. Annals of Mathematics and Artificial Intelligence, 2006,46:415−439. [doi: 10.1007/s10472-006-9033-2]

[8] Silaghi MC. Using secure DisCSP solvers for generalized Vickrey auctions complete and stochastic techniques. In: Proc. of the Kaelbling and Saffiotti. 2005.

[9] Yokoo M. Distributed Constraint Satisfaction, Foundation of Cooperation in Multi-Agent Systems. Berlin: Springer-Verlag, 2001. [doi: 10.1007/978-3-642-59546-2]

[10] Pragnesh JM, Shen WM. Adopt: Asynchronous distributed constraint optimization with quality Guarantees. Artificial Intelligence, 2005,161:149−180. [doi: 10.1016/j.artint.2004.09.003]

[11] Petcu A, Faltings B. DPOP: A scalable method for multi-agent constraint optimization. Artificial Intelligence, 2005,161:266−271.

[12] Patricia G, Pedro M. Improving BnB-ADOPT+-AC. In: Proc. of the AAMAS. Richland: IFAAMAS, 2012. 273−280.

[13] Mailler R, Lesser VR. Solving distributed constraint optimization problems using cooperative mediation. In: Proc. of the AAMAS. Washington: IEEE Computer Society, 2004. 438−445. [doi: 10.1109/AAMAS.2004.249]

[14] Fitzpatrick S, Meertens LGLT. An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. In: Proc. of the SAGA. Springer-Verlag, 2001. 49−64. [doi: 10.1007/3-540-45322-9_3]

© 中国科学院软件研究所 http://www.jos.org.cn

[15] Silaghi MC, Yokoo M. Nogood-Based asynchronous distributed optimization. In: Nakashima. *et al*., eds. Proc. of the AAMAS. ACM, 2006. 1389−1396. [doi: 10.1145/1160633.1160894]

[16] Thomas L. Distributed constraint optimization: Privacy guarantees and stochastic uncertainty [Ph.D Thesis]. Lausanne: École Polytechnique Fédérale de Lausanne, 2011.

[17] Chechetka A, Sycara K. No-Commitment branch and bound search for distributed constraint optimization. In: Proc. of the AAMAS. ACM, 2006. 1427−1429. [doi: 10.1145/1160633.1160900]

[18] Gershman A, Meisels A, Zivan R. Asynchronous forward-bounding for distributed constraints optimization. In: Proc. of the ECAI. IOS Press, 2006. 133−140.

[19] Gutierrez P, Meseguer P. BnB-ADOPT+ with several soft arc consistency levels. In: Proc. of the ECAI. IOS Press, 2010. 67−72.

[20] Petcu A, Faltings B. MB-DPOP: A new memory-bounded algorithm for distributed optimization. In: Proc. of the IJCAI. AAAI Press, 2007. 1452−1457.

[21] Petcu A, Faltings B. O-DPOP: An algorithm for open/distributed constraint optimization. In: Proc. of the AAAI. AAAI Press, 2006. 703−708.

[22] Kumar A, Petcu A, Faltings B. H-DPOP: Using hard constraints to prune the search space. In: Proc. of the IJCAI. AAAI Press, 2007. 40−55.

[23] Pearce JP, Tambe M. Quality guarantees on *k*-optimal solutions for distributed constraint optimization problems. In: Proc. of the IJCAI. AAAI Press, 2007. 1446−1451.

[24] Kiekintveld C, Yin Z, Kumar A, Tambe M. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. In: Proc. of the AAMAS. Richland: IFAAMAS, 2010. 133−140.

[25] Vinyals M, Shieh E, Cerquides J, Rodriguez-Aguilar JA, Yin ZY, Tambe M, Bowring E. Quality guarantees for region optimal DCOP algorithms. In: Proc. of the AAMAS. Richland: IFAAMAS, 2011. 133−140.

[26] Okimoto T, Joe J, Iwasaki A, Yokoo M, Faltings B. Pseudo-Tree-Based incomplete algorithm for distributed constraint optimization with quality bounds. In: Proc. of the CP. Springer-Verlag, 2011. 660−674. [doi: 10.1007/978-3-642-23786-7_50]

[27] Meritxell V, Marc P, Rodriguez-Aguilar JA, Cerquides J. Divide-and-Coordinate: DCOPs by agreement. In: Proc. of the AAMAS. Richland: IFAAMAS, 2010. 149−156.

[28] Daisuke H, Katsutoshi H. DeQED: An efficient divide-and-coordinate algorithm for DCOP. In: Proc. of the AAMAS. Richland: IFAAMAS, 2013. 556−572.

[29] Ottens B, Dimitrakakis C, Faltings B. DUCT: An upper con_dence bound approach to distributed constraint optimization problems. In: Proc. of the AAAI. Phoenix: AAAI Press, 2012. 528−534.

[30] Nguyen DT, Yeoh W, Lau HC. Distributed Gibbs: A memory-bounded sampling-based dcop algorithm. In: Proc. of the AAMAS. Richland: IFAAMAS, 2013. 167−174.

[31] Rogers A, Farinelli A, Stranders R, Jennings NR. Bounded approximate decentralised coordination via the max-sum algorithm. Artificial Intellegence, 2011,175(2):730−759. [doi: 10.1016/j.artint.2010.11.001]

[32] Rollon E, Larrosa J. Improved bounded max-sum for distributed constraint optimization. In: Proc. of the Constraint Programming. Springer-Verlag, 2012. 624−632. [doi: 10.1007/978-3-642-33558-7_45]

[33] Larrosa J, Rollon E. Risk-Neutral bounded max-sum for distributed constraint optimization. In: Proc. of the Annual ACM Symp. on Applied Computing. ACM, 2013. 92−97. [doi: 10.1145/2480362.2480383]

[34] Boi F, Thomas L, Adrian P. Privacy guarantees through distributed constraint satisfaction. In: Proc. of the IAT. IEEE, 2008. 350−358. [doi: 10.1109/WIIAT.2008.177]

[35] Zang WX, Wang GD, Zhao X, Lars W. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. Journal of Articial Intelligence Research, 2005,161(1-2): 55−87. [doi: 10.1016/j.artint.2004.10.004]

[36] Fioretto F, Campeotto F, Fioretto LDR, Yeoh W, Pontelli E. GD-GIBBS: A GPU-based sampling algorithm for solving distributed constraint optimization problems. In: Proc. of the AAMAS. Richland: IFAAMAS, 2014. 1339−1340.

[37] Okimoto T, Schwind N, Clement M, Inoue K. L_p-Norm based algorithm for multi-objective distributed constraint optimization. In: Proc. of the AAMAS. Richland: IFAAMAS, 2014. 1427−1428.

© 中国科学院软件研究所 http://www.jos.org.cn

[38] Modi PJ, Jung H, Tambe M, Shen WM, Kulkarni S. A dynamic distributed constraint satisfaction approach to resource allocation. In: Proc. of the CP. Springer-Verlag, 2001. 685−700.

[39] Parsons S, Rodríguez-Aguilar JA, Klein M. Auctions and bidding: Aguide for computer scientists. In: Proc. of the ACM Computing Surveys. ACM, 2011. [doi: 10.1145/1883612.1883617]

附中文参考文献:

[4] 韩靖,陈恩红,蔡庆生.求解过程中约束一致性维护的多层次策略研究.软件学报,1998,9(8):622−627. http://www.jos.org.cn/1000-9825/9/622.htm

[5] 陈恩红,张振亚,王煦法.相接行凸约束网络的快速识别算法.软件学报,2002,13(5):972−979. http://www.jos.org.cn/1000-9825/13/972.htm

段沛博(1987－),男,辽宁沈阳人,博士生,主要研究领域为分布式约束优化.

张斌(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,Web 信息处理,数据挖掘.

张长胜(1980－),男,博士,副教授,主要研究领域为约束程序设计,Web 智能信息处理技术.

© 中国科学院软件研究所 http://www.jos.org.cn