

DOI: 10.11992/tis.201804054

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.TP.20180619.1422.002.html>

约束条件下联盟生成研究进展

任子仪, 童向荣

(烟台大学 计算机与控制工程学院, 山东 烟台 264005)

摘 要: 联盟生成是在多 Agent 系统的研究中最重要挑战之一。如何对 Agent 进行划分使所得社会福利最大化是当前面临的主要问题。假设每个 Agent 都具有理性和自利性的特性, 为了追求自身的利益最大化而选择和其他的 Agent 进行联合, 进而使整个系统实现利益的最大化。目前, 联盟生成问题有很大的计算挑战, 即使在进行联盟的时候添加了约束条件, 也需要新的算法来更快更有效地解决该问题。本文主要对约束条件下的联盟生成的研究进行综述, 主要包括 4 部分: 最坏情况有限界联盟生成、动态规划联盟生成求精确最优解、联盟生成求近似最优解和约束条件下联盟生成求最优解。

关键词: 联盟结构; 社会福利; 联盟生成; 约束条件; 特征函数; 联盟结构图; 联盟博弈; 动态规划

中图分类号: TP18 **文献标志码:** A **文章编号:** 1673-4785(2019)03-0413-10

中文引用格式: 任子仪, 童向荣. 约束条件下联盟生成研究进展[J]. 智能系统学报, 2019, 14(3): 413-422.

英文引用格式: REN Ziyi, TONG Xiangrong. Research progress of constrained coalition formation[J]. CAAI transactions on intelligent systems, 2019, 14(3): 413-422.

Research progress of constrained coalition formation

REN Ziyi, TONG Xiangrong

(School of Computer and Control Engineering, Yantai University, Yantai 264005, China)

Abstract: Coalition formation is one of the most important challenges in the research of multiagent systems. Currently, our main problem is how to divide Agent to maximize the social welfare. We assume that each Agent possesses the characteristics of rationality and self-interest to maximize its own interests. An Agent integrates with another Agent, which also maximizes the interest of the whole system. At present, the coalition formation problem presents notable computational challenges. If constraints are added during the coalition process, new algorithms are needed to solve the problem more rapidly and effectively. This paper mainly summarizes the study of coalition structure generation under constraint conditions. This paper comprises four parts: the coalition structure generation with the worst case guaranteed, the use of the dynamic programming to find the exact optimal solution, the near-optimal solution after formation of the coalition structure, and the optimal solution to the constrained coalition formation.

Keywords: coalition structure; social welfare; coalition formation; constraint; characteristic function; coalition structure graph; coalition game; dynamic programming

联盟生成是多 Agent 系统 (multi-agent system, MAS) 研究基本问题之一^[1-2], 主要将 Agent 进行合作或协商, 使其效用增加。Agent 联盟基本上被认为在 MAS 的框架内的一组 Agent, 愿意与这组

中的其他 Agent 合作, 共同合作活动的目的是达到最佳的标准。当 Agent 通过组建联盟共同工作时, 联盟结构生成就会发生^[3]。目标就是求一种最优的联盟结构, 使得所求的社会福利最大, 并返回其值^[4-5]。

Agent 在联盟生成时达到最优方案不一定是全局最优解, 但是应该是 Nash 最优解^[6-7]或者次优解, 因此联盟生成主要包括以下活动:

收稿日期: 2018-04-26. 网络出版日期: 2018-06-20.

基金项目: 国家自然科学基金项目 (61572418); 山东省科技发展计划项目 (2016GGX109004).

通信作者: 童向荣. E-mail: txr@ytu.edu.cn.

1) 联盟结构生成: Agent 之间进行协商生成联盟,但不考虑各个联盟之间的协商。

2) 解决每个联盟优化问题: 将 Agent 的任务和资源集中在一起,共同解决问题。Agent 之间进行协商使联盟的社会福利尽可能最大化,即寻找最优的方法使得联盟本身的效用最大化。

3) 划分值: 在进行收益分配的时候,常用的方法有两种,即公平(每个 Agent 所得到的收益是 Agent 在博弈中的贡献的体现)和稳定(存在没有和其他 Agent 协商而单独形成自己的自私利益联盟)。

同样,存在很多方法寻找和创建一个最优的联盟——协商或搜索以及用于找出解决问题的几种潜在的方法,例如:遗传算法、博弈论、粒子群算法等。但是,随着研究的不断深入,在现实世界的应用中有一些联盟结构是不允许生成的^[8]。目前,联盟研究的热点问题主要是约束条件下联盟生成。在现有研究中,很多文献已经考虑将联盟进行广泛应用,例如:通过形成联盟,自主传感器可以改善某些区域的监控情况^[9];认识无线网络可以增加它们的吞吐量^[10];购买者可以通过批量购买而获得较低的价格^[11];自主连接车辆进行护航工作^[12];形成联盟促进社交网络关系^[13-14]等。

在 Agent 联盟研究发展中, Sandholm 等^[15]最先提出最坏情况有限界联盟结构生成; Yeh 等^[16]首次使用动态规划算法联盟生成并求得精确最优解;随着研究的不断深入, Agent 数量的增加,先前的算法越来越不能满足当前的需求, Dang 等^[17]首次提出建立次优联盟结构并求得次优解; Greco 等^[18]经过进一步研究,提出了一种新的联盟结构生成的方式——约束联盟结构,即在生成联盟的过程对联盟添加约束。

1 基本定义

1.1 主要定义

假设所有的 Agent 都是理性的,并使用合作博弈建模进行 Agent 之间的合作和协商。在博弈中,令 N 是一个 Agent 集,其中 $n = |N|$ 表示 N 中 Agent 的个数,则 $N = (a_1, a_2, \dots, a_n)$, 联盟即 N 中非空的子集。

定义 1 联盟结构。对于任意联盟 C , 在 C 上形成的联盟结构 $CS = \{C_1, C_2, \dots, C_k\}$, 其中 $\cup CS = C$, 并且对于任意 $i, j \in \{1, 2, \dots, k\}, i \neq j$ 都需要满足 $C_i \cap C_j = \emptyset$ 。

例 1 一个集合 $N = \{a_1, a_2, a_3\}$, N 上的联盟一

共有 7 种情况,分别是 $\{a_1\}$ 、 $\{a_2\}$ 、 $\{a_3\}$ 、 $\{a_1, a_2\}$ 、 $\{a_1, a_3\}$ 、 $\{a_2, a_3\}$ 、 $\{a_1, a_2, a_3\}$, 联盟结构一共存在 5 种情况,分别是 $\{\{a_1\}, \{a_2\}, \{a_3\}\}$ 、 $\{\{a_1\}, \{a_2, a_3\}\}$ 、 $\{\{a_2\}, \{a_1, a_3\}\}$ 、 $\{\{a_3\}, \{a_1, a_2\}\}$ 和 $\{a_1, a_2, a_3\}$ 。

定义 2 社会福利。社会福利即为联盟结构 CS 中所有的联盟 $C_i, i \in \{1, 2, \dots, k\}$ 的收益之和,将联盟的值记为 $v(C_i)$, 并且要求 $v(C_i) \geq 0$, 联盟结构的社会福利记为 $V(CS)$, 将具有最大的社会福利联盟结构称为最优联盟结构,记为 CS^* , 其最优社会福利记为 $V(CS^*)$, 则

$$V(CS) = \sum_{i=1}^k v(C_i)$$

定义 3 次加性。对于任意的不相交的联盟 $S, T \in N$, 都存在 $v(S \cup T) \leq v(S) + v(T)$ 。换言之, $\{\{a_1\}, \{a_2\}, \dots, \{a_n\}\}$ 为最大社会福利的联盟结构。

定义 4 特征函数。给定一个 n 个 Agent 的博弈, C 是一个联盟, $v(C)$ 是指 C 和 $N-C$ 两个 Agent 博弈中 C 的最大效用, $v(C)$ 称为联盟 C 特征函数 (characteristic function), 规定 $v(\emptyset) = 0$ 。

1.2 博弈类型

在常用的联盟博弈模型中,通常用货币来表示联盟的价值(或奖励^[19])。在进行博弈的过程中,假设存在一个在 Agent 之间可以自由流动的交换媒介(如货币),每个 Agent 可以根据自己的绩效得到相应的货币,这种博弈被称为“单边支付”博弈或可转移效用 (transferable utility, TU) 博弈。相反,联盟的绩效是用向量表示的,直接指定每个成员的绩效, Agent 只能服从这种分配方案,不能对分配方案进行修改,这种博弈被称为不可转移效用 (non-transferable utility, NTU) 博弈。在多 Agent 系统中,可转移效用博弈受到了很大的关注。

1) 特征函数博弈

特征函数博弈 (characteristic function games, CFGs) 是联盟的价值完全取决于它所包含的 Agent 的值。将特征函数博弈定义为 (N, v) , 其中 N 表示 Agent 数据集, v 是一个函数,称为特征函数,对于每个联盟 C 映射到函数中的值 $v(C)$ 记为 $v: 2^N \rightarrow \mathbf{R}$ 。

2) 分区博弈

在分区博弈 (partition function games, PFGs) 中联盟的价值不仅仅取决于所包含的 Agent 的值,也会受非成员分区的影响。将分区博弈定义为 (N, w) , 其中 N 表示 Agent 数据集, w 是一个分区函数,每个嵌入式联盟 (C, CS) 映射到函数中的值是 $w((C, CS))$ 或 $w(C, CS)$ 。

CFGs 是 PFGs 的子集, 联盟 C 在 CFGs 中有一个精确值, 但在 PFGs 中却存在很多值, 这是因为使用 PFGs 有不同的方法对 N/C 中的 Agent 进行分区。因此, 在进行研究的过程中, 使用 CFGs 更容易得到解决方案, 提高算法的效率。

1.3 空间表示

在联盟结构生成问题中, 通过联盟结构搜索得到最优联盟结构。

1) 联盟结构图

Sandholm 等^[15]提出的联盟结构图中, 每个节点代表一个联盟结构, 并且将节点进行了划分, 表示为 $\Pi_1^C, \Pi_2^C, \dots, \Pi_n^C$, 其中 Π_i^C 表示由 i 个联盟组成的所有的联盟结构的节点; 边缘连接两个联盟结构, 当且仅当它们属于不同的划分, 如 Π_i^C 和 Π_{i-1}^C ; 其中 Π_i^C 中的联盟结构是由 Π_{i-1}^C 中的联盟分裂成两个联盟形成的。图 1 所示为具有 4 个 Agent 联盟结构图。

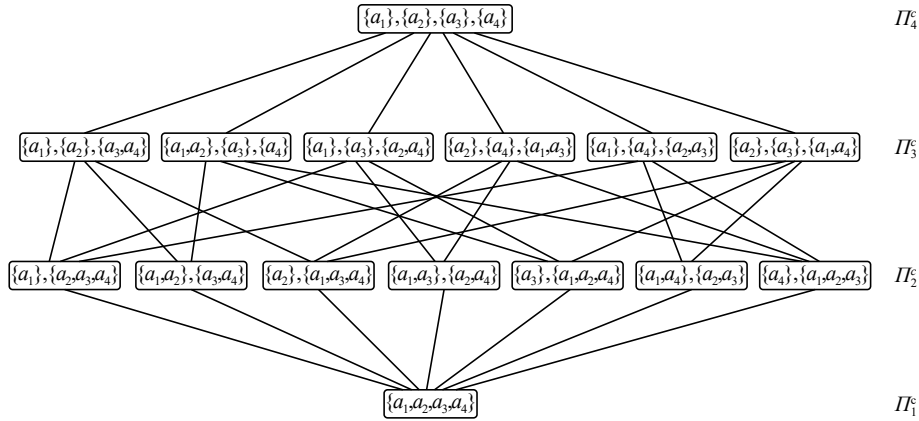


图 1 4 个 Agent 联盟结构

Fig. 1 The coalition structure graph with 4 Agents

2) 整数分区图

在联盟结构图中, 能直接看出具有 i 个联盟所组成的联盟结构的情况。而 Dean 等^[20]提出的整数分区图中, 则是基于联盟结构中所包含的联盟的数量, 将联盟结构中的空间划分成不相交的子空间, 每个子空间是由 n 个整数分割表示的。整数 n 的分区值可以通过递归的方法计算^[21], 用 P^n 表示整数分区。例如: $P^4 = \{\{4\}, \{1, 3\}, \{2, 2\}, \{1, 1, 2\}, \{1, 1, 1, 1\}\}$ 。

假设 $I \in P$, 定义 $\Pi_I^C \in \Pi^C$ 是所有联盟结构组成的子空间, 其中联盟的大小受 I 部分的影响。例如: $\Pi_{\{1,1,2\}}^{[a_1, a_2, a_3, a_4]}$ 表示是由 2 个只有 1 个 Agent 联盟和 1 个有 2 个 Agent 组成的联盟形成的联盟结构。对于这种表示一般用整数分区图来表示。整数分区图也是一个无向图, 其中每个节点代表一个子空间。两个节点 $I, I' \in P^n$ 是通过边连接的, 当且仅当 $i, j \in I$ 时, $I' = (I \setminus \{i, j\}) \cup \{i + j\}$ (\cup 表示多重集的结合操作)。图 2 给出了有 4 个 Agent 整数分区图。

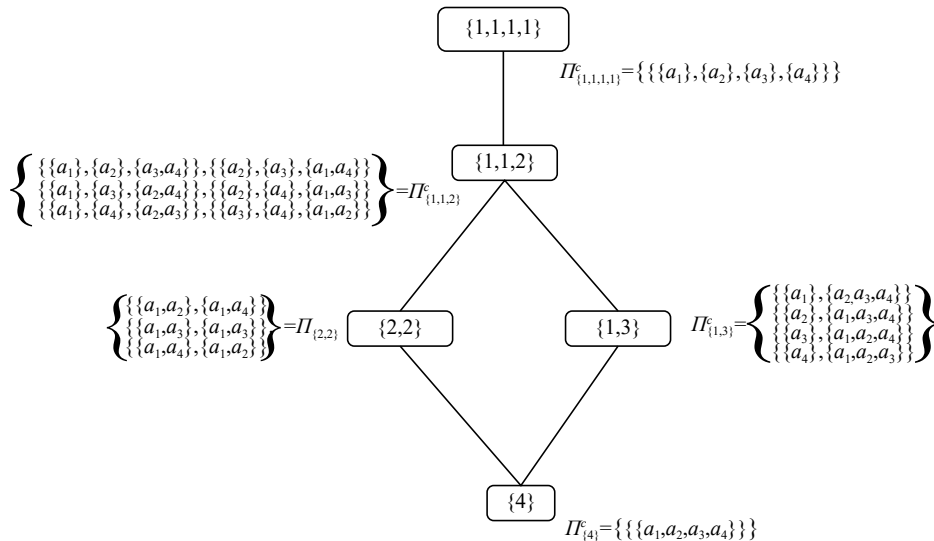


图 2 4 个 Agent 整数分区图

Fig. 2 The integer-partition graph representation of 4 Agents

2 联盟生成的复杂度

2.1 输入值的大小

对于具有 n 个 Agent 的数据集, 会产生 $2^n - 1$ 个非空的子集, 生成 $2^n - 1$ 个联盟, 在进行运算时需要输入 $2^n - 1$ 个值。联盟结构生成算法的输入值和 Agent 的个数之间呈指数关系。

理论上, 在进行输入操作时即可排除一些不合理的联盟, 或者在进行联盟结构生成的时候就可以忽略一部分的输入。但是在接下来进行的操作过程中无法保证存在单一的联盟与其他的联盟进行合作之后的社会福利不是最佳社会福利, 这是因为: 被排除在外的联盟中可能存在比其他的联盟社会福利更大的联盟。

2.2 联盟结构的数量

随着联盟结构不断发展, Agent 数量的增多, 新问题也随之产生。联盟结构的个数表示为

$$\sum_{i=1}^N Z(n, i)$$

其中: n 代表数据集中的 Agent 个数; $Z(n, i)$ 表示具有 i 个联盟组成的联盟结构的个数; $Z(n, i)$ 满足第 2 类 Stirling 数的特征, 即

$$Z(n, i) = iZ(n-1, i) + Z(n-1, i-1)$$

式中: $Z(n, i) = Z(n, 1) = 1$; $iZ(n-1, i)$ 表示在现有存在的联盟中增加一个新的 Agent 形成的联盟结构的个数; $Z(n-1, i-1)$ 表示将一个新的 Agent 加入到已有的联盟中, 因为现在的联盟结构中只有 $i-1$ 个联盟被计算。有以下基本结果^[22-24]:

命题 1 对于具有 n 个 Agent 的集合, 具有 2^{n-1} 个联盟, $O(n^n)$ 和 $\omega(n^{n/2})$ 个联盟结构 (即联盟结构的个数的阶介于 $\frac{n}{2} \sim n$ 之间)。

命题 2 寻找最优联盟结构是 NP 难问题。

随着 Agent 数量的增加, 联盟结构的数量也随之增加。经过实验的证明, 当数据集中 Agent 个数超过 15 时, 使用穷举法列举出所有联盟结构是不可行的。

3 最坏情况有限界联盟结构生成

Dean 和 Boddy 在时间依赖规划 (time dependent planning) 的基础上提出了 Anytime 算法, 其本质是一种反复求解使得结果更加精确的算法。在算法的运行过程中, 能够很快得到一个不精确的解, 然后进行若干次的重复过程, 经过重复后逐步提高解的质量, Anytime 算法一个最显著的优点就是能够很好地权衡计算时间和解的质量。Sandholm 等^[15]使用 Anytime 算法开创性地提出了一种最坏情况有限界联盟结构生成。

命题 1 表明, 在联盟结构图上找到最优的联盟结构是不可行的。所以, 在进行联盟结构图的搜索的时候设定一个界限。

对联盟结构的子集 N 进行搜索, 目的在于寻找搜索中的最优联盟结构, 即局部最优联盟结构, 记:

$$CS_N^* = \arg \max_{CS \in N} V(CS)$$

同时, 为了保证设定的联盟结构在最佳范围内, 即存在一个有限的但是尽可能小的值 k , k 被称为搜索的解的界限, 也是衡量局部最优解 $V(CS_N^*)$ 的标准。能够建立起界限的最小的 k 记为

$$k = \min\{\kappa, \kappa \geq \frac{V(CS^*)}{V(CS_N^*)}\}$$

通常, 如果没有对联盟结构进行完全搜索, 设定一个正确的界限是很难的。这是因为: 在剪枝的过程中虽然可以减少一些联盟结构的搜索, 但是很难保证没有剪掉的比剪掉的具有更优性。在进行社会福利的定义的时候, 令任意一个联盟 C_i 的值 $v(C_i) \geq 0$, 并且一个联盟结构的社会福利 $V(CS) = \sum_{i=1}^k v(C_i)$, 这两点就保证了上述假设是不存在的, 即在没有任何对联盟结构进行搜索的前提下设置一个界限来缩小搜索联盟结构的数量。

3.1 建立界限

本节主要讨论如何尽可能地减少对图的搜索的前提下, 建立一个界限 k 。

定理 1 对于界限 k , 可以只搜索联盟结构图的最低的两级 (图 1 中第 Π_1^C 、 Π_2^C 层)。在这个搜索中, 界限 $k=n$, 需要搜索的联盟结构节点的数量是 2^{n-1} , 这也是能够建立起界限的最小的搜索, 即 $n_{\min} = 2^{n-1}$ 。并且, 没有其他的算法对于联盟结构的搜索的数量会比 2^{n-1} 少。

最底层具有 1 个联盟结构, 第二最底层中具有 $2^n - 2$ 个联盟 (N 中的所有子集, 除掉全集和空集)。在这一层中, 每个联盟结构都有 2 个联盟, 所以一共有 $\frac{1}{2}(2^n - 2)$ 个联盟结构。只搜索最底下两层, 一共搜索 $1 + \frac{1}{2}(2^n - 2) = 2^{n-1}$ 个联盟结果。

3.2 算法描述

算法 1 最坏情况有保证联盟结构生成算法

1) 搜索联盟结构图的最底层;

2) 从联盟结构图的顶层 (第 n 层) 开始, 作宽度优先搜索, 一直搜索到时间不允许为止或搜索完整个联盟结构图;

3) 返回迄今为止所得到的最优的联盟结构。

先前使用特征函数的方式^[25-26]进行联盟结构生成, 算法 1 是一种任意时间算法, 可以在任何时

间中断。如果将第1、2层搜索完成后,还存在剩余时间,则可进一步地搜索缩小界限。

定理2 如果联盟结构图的第1、2层已经被搜索完成并且第1层以及以上的层也被搜索完成 ($l > 3$), 当 $n \equiv h-l \pmod{h}$ 且 $n \equiv l \pmod{h}$ 时, $k = \left\lfloor \frac{n}{h} \right\rfloor$ 是紧的, 否则 $k = \left\lfloor \frac{n}{h} \right\rfloor$ 是紧的, 其中 $h = \left\lfloor \frac{n-l}{2} \right\rfloor + 2$ 。

在 2^{n-1} 个节点被搜索完之前, 无法建立一个界限。当搜索到 $2^{n-1} + 1$ 个节点时, 可以建立界限 $k = n$; 搜索到 $2^{n-1} + 1$ 个节点的时候, 界限变成 $k = \frac{1}{2}n$ 。通过更多实验验证: 当界限 k 变成 $\frac{1}{3}n$ 时, 搜索的层数就会增加2层。当搜索的层数每增加2层, 界限 k 前面的除数就会加1, 若只多搜索一层时, 没有明显效果。

总之, 最坏情况有限界联盟结构生成算法中第2步具有很强的理性, 即界限 k 能够迅速地下降, 同时该算法的收益递减, 如图3所示。

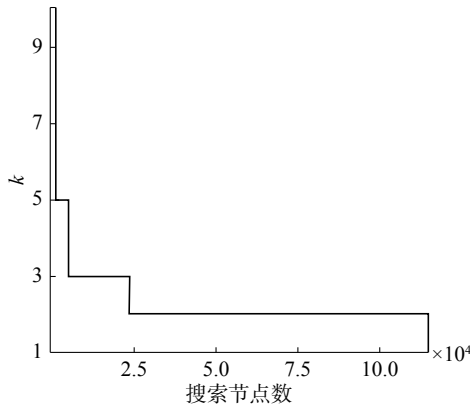


图3 界限 k 作为10个Agent博弈中搜索的函数

Fig. 3 Ratio bound k as a function of search in a 10-Agents game

界限 $k = \frac{1}{2}n$ 的建立与输入的Agent联盟的个数之间呈线性关系, 这是因为输入 2^{n-1} 个数据, 而当界限 $k = \frac{1}{2}n$ 时能够建立 2^{n-1} 个联盟结构, 就是所说的第1、2层和第 n 层上面的节点。

胡山立等^[27-28]对算法1深入研究并进行改进, 给出了解决问题的一种任一时间联盟结构生成算法和给定界限要求的联盟结构生成。Dang等^[29]提出不以层为单位最坏情况有限界的联盟结构生成。

4 动态规划联盟生成求精确最优解

动态规划 (dynamic programming, DP) 算法和分治法类似, 其基本思想是将待求解的问题分解成若干个次级子问题, 在求解的过程中, 先求解子问题, 然后从子问题的解中得到要求解的问题

的答案。将动态规划算法应用于最优化的问题, 一般分为4个步骤来完成:

- 1) 找出最优解的性质, 并刻画其结构特征;
- 2) 递归定义的最优值;
- 3) 以自顶向下的方式计算出最优值;
- 4) 根据计算最优值所得到的信息, 构造最优解。

将动态规划算法应用于联盟中的想法是Yeh在1986年首次提出的^[16]; Rothkopf^[30]及刘惊雷等^[31]使用DP算法对求解最优联盟结构的算法进行了改进, 主要解决存在大量重复子问题计算的问题; Rahwan等^[32]提出了IDP(improved dynamic programming)算法; 张新良等^[33]在2007年提出了一种快速动态生成 (search of coalition structure, SCS) 算法, 主要降低搜索次数。本文中, DP算法主要基于定理3。

定理3 任意给定一个联盟 $C \in N$, $V(CS^*)$ 是最优联盟结构的社会福利, 即 $V(CS^*) = \max_{CS \in \Pi^n} V(CS)$, 则

$$V(CS^*) = \begin{cases} v(C), & |C| = 1 \\ \max\{v(C), \max_{\{C', C''\} \in \Pi^n} (v(C') + v(C''))\}, & \text{其他} \end{cases} \quad (1)$$

使用定理3进行动态规划, 先计算只有一个Agent的联盟, 接着迭代具有2个Agent的联盟, 然后是具有3个Agent的联盟, 一直迭代到具有 n 个Agent的联盟。对于每个联盟 C 都需要使用式(1)计算。从式(1)中易知: 当 $|C| \neq 1$ 时, 需要分别计算 $\max_{\{C', C''\} \in \Pi^n} v(C') + v(C'')$ 和 $v(C)$ 的值, 然后将两个值进行比较, 得到具有较大社会福利的联盟。在此过程中, 将所产生的暂存结果保存在变量 $t(C)$ 中。

通过上述操作, 计算出来的最大 $v(C)$ 就是我们最终要求的 $V(CS^*)$ 。计算 CS^* 的迭代过程具体操作如例2所示。

例2 令数据集 $N = \{a_1, a_2, a_3, a_4\}$, 假设特征函数为

$$\begin{aligned} v(\{a_1\}) &= 30, v(\{a_2\}) = 40, \\ v(\{a_3\}) &= 25, v(\{a_4\}) = 45, \\ v(\{a_1, a_2\}) &= 50, v(\{a_1, a_3\}) = 60, \\ v(\{a_1, a_4\}) &= 80, v(\{a_2, a_3\}) = 55, \\ v(\{a_2, a_4\}) &= 70, v(\{a_3, a_4\}) = 80, \\ v(\{a_1, a_2, a_3\}) &= 90, v(\{a_1, a_2, a_4\}) = 120 \\ v(\{a_1, a_3, a_4\}) &= 100, v(\{a_2, a_3, a_4\}) = 115, \\ v(\{a_1, a_2, a_3, a_4\}) &= 140 \end{aligned}$$

表1表示算法的具体运算过程。由表1知, $t(N) = (\{a_1, a_2\}, \{a_3, a_4\})$, 能够将 N 分裂成 $\{a_1, a_2\}$ 和 $\{a_3, a_4\}$, 而 $t(\{a_3, a_4\}) = \{\{a_3, a_4\}\}$, 这是因为 $\{a_1, a_2\}$ 分裂成 $\{a_1\}$ 、 $\{a_2\}$ 后的社会福利比较大, $\{a_3, a_4\}$ 不需要进行分裂。所以 $CS^* = \{\{a_1\}, \{a_2\}, \{a_3, a_4\}\}$ 。

表1 4个Agent动态规划算法

Table 1 The dynamic programming algorithm for 4 Agents

C	特征函数值	$t(C)$	$v(C)$
$\{a_1\}$	$v(\{a_1\}) = 30$	$\{a_1\}$	30
$\{a_2\}$	$v(\{a_2\}) = 40$	$\{a_2\}$	40
$\{a_3\}$	$v(\{a_3\}) = 25$	$\{a_3\}$	25
$\{a_4\}$	$v(\{a_4\}) = 45$	$\{a_4\}$	45
$\{a_1, a_2\}$	$v(\{a_1, a_2\}) = 50, f(\{a_1\}) + f(\{a_2\}) = 70$	$\{a_1\}, \{a_2\}$	70
$\{a_1, a_3\}$	$v(\{a_1, a_3\}) = 60, f(\{a_1\}) + f(\{a_3\}) = 55$	$\{a_1, a_3\}$	60
$\{a_1, a_4\}$	$v(\{a_1, a_4\}) = 80, f(\{a_1\}) + f(\{a_4\}) = 75$	$\{a_1, a_4\}$	80
$\{a_2, a_3\}$	$v(\{a_2, a_3\}) = 55, f(\{a_2\}) + f(\{a_3\}) = 65$	$\{a_2\}, \{a_3\}$	65
$\{a_2, a_4\}$	$v(\{a_2, a_4\}) = 70, f(\{a_2\}) + f(\{a_4\}) = 85$	$\{a_2\}, \{a_4\}$	85
$\{a_3, a_4\}$	$v(\{a_3, a_4\}) = 80, f(\{a_3\}) + f(\{a_4\}) = 70$	$\{a_3, a_4\}$	80
$\{a_1, a_2, a_3\}$	$v(\{a_1, a_2, a_3\}) = 90, f(\{a_1\}) + f(\{a_2, a_3\}) = 95$ $f(\{a_2\}) + f(\{a_1, a_3\}) = 100, f(\{a_3\}) + f(\{a_1, a_2\}) = 95$	$\{a_2\}, \{a_1, a_3\}$	100
$\{a_1, a_2, a_4\}$	$v(\{a_1, a_2, a_4\}) = 120, f(\{a_1\}) + f(\{a_2, a_4\}) = 115$ $f(\{a_2\}) + f(\{a_1, a_4\}) = 120, f(\{a_4\}) + f(\{a_1, a_2\}) = 115$	$\{a_1, a_2, a_4\}$	120
$\{a_1, a_3, a_4\}$	$v(\{a_1, a_3, a_4\}) = 100, f(\{a_1\}) + f(\{a_3, a_4\}) = 110$ $f(\{a_3\}) + f(\{a_1, a_4\}) = 105, f(\{a_4\}) + f(\{a_1, a_3\}) = 105$	$\{a_1\}, \{a_3, a_4\}$	110
$\{a_2, a_3, a_4\}$	$v(\{a_2, a_3, a_4\}) = 115, f(\{a_2\}) + f(\{a_3, a_4\}) = 120$ $f(\{a_3\}) + f(\{a_2, a_4\}) = 110, f(\{a_4\}) + f(\{a_2, a_3\}) = 110$	$\{a_2\}, \{a_3, a_4\}$	120
$\{a_1, a_2, a_3, a_4\}$	$v(\{a_1, a_2, a_3, a_4\}) = 140, f(\{a_4\}) + f(\{a_1, a_2, a_3\}) = 145$ $f(\{a_1, a_2\}) + f(\{a_3, a_4\}) = 150, f(\{a_3\}) + f(\{a_1, a_2, a_4\}) = 145$ $f(\{a_1, a_3\}) + f(\{a_2, a_4\}) = 145, f(\{a_2\}) + f(\{a_1, a_3, a_4\}) = 150$ $f(\{a_1, a_4\}) + f(\{a_2, a_3\}) = 145, f(\{a_1\}) + f(\{a_2, a_3, a_4\}) = 150$	$\{a_1, a_2\}, \{a_3, a_4\}$	150

定理4 给定一组Agent集,使用动态规划算法求得最优联盟结构的时间复杂度 $O(3^n)$ 。

DP算法在该算法的运行过程中实际做了如下几方面:

- 1) 评估图上的运动;
- 2) 在表 t 中存储最优结构;
- 3) 从底部节点开始向上运动。

5 联盟生成求近似最优解

Rahwan等^[34]将联盟配置与Anytime算法进行结合提出一种联盟结构相似最优解算法;为Albizuri等^[35]基于整数分区对联盟结构进行推广从而对联盟配置(coalition configuration)进行了定义。

5.1 基本定义

在进行求解的过程中,设 $CL_s \in 2^n$, 其中 $s \in \{1, 2, \dots, k\}$ 是联盟列表, CL_i 是联盟列表 CL_s 的

集合(见表2)。设 $G_1, G_2, \dots, G_{s_{cs}} \in \zeta_{cs}$ 是所有的存在的唯一配置的结合(即不存在具有相同联盟大小的两个元素),令 $F: \zeta_{cs} \rightarrow 2^{cs}$ 是一个函数,按照提前设定好的配置返回所有的联盟结构。 N 表示一个列表,里面的每一个元素表示一组具有相同配置的联盟结构。形式上, N 定义为: $N = \{F(G_1), F(G_2), \dots, F(G_{|s_{cs}|})\}$, $N_1, N_2, \dots, N_{s_{cs}} \in N$ 表示 N 中的每个元素(见表3)。

表2 4个Agent联盟列表

Table 2 The coalition list of 4 Agents

CL_s	联盟列表值
CL_1	$\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}$
CL_2	$\{a_1, a_3\}, \{a_1, a_4\}, \{a_2, a_3\}, \{a_2, a_4\}, \{a_3, a_4\}$
CL_3	$\{a_1, a_2, a_3\}, \{a_1, a_2, a_4\}, \{a_1, a_3, a_4\}, \{a_2, a_3, a_4\}$
CL_4	$\{a_1, a_2, a_3, a_4\}$

表3 4个Agent配置表及具体联盟表示

Table 3 The configuration table and specific coalition representation of 4 Agents

联盟数量	配置表及具体联盟表示
1	$G = \{4\}, F(G) = \{\{a_1, a_2, a_3, a_4\}\}$
2	$G = \{2, 2\}, F(G) = \begin{cases} \{\{a_1, a_2\}, \{a_3, a_4\}\} \\ \{\{a_1, a_3\}, \{a_2, a_4\}\} \\ \{\{a_1, a_4\}, \{a_2, a_3\}\} \end{cases}$ $G = \{1, 3\}, F(G) = \begin{cases} \{\{a_1\}, \{a_2, a_3, a_4\}\} \\ \{\{a_2\}, \{a_1, a_3, a_4\}\} \\ \{\{a_3\}, \{a_1, a_2, a_4\}\} \\ \{\{a_4\}, \{a_1, a_2, a_3\}\} \end{cases}$
3	$G = \{1, 1, 2\}, F(G) = \begin{cases} \{\{a_3\}, \{a_4\}, \{a_1, a_2\}\} \\ \{\{a_2\}, \{a_4\}, \{a_1, a_3\}\} \\ \{\{a_2\}, \{a_3\}, \{a_1, a_4\}\} \\ \{\{a_1\}, \{a_4\}, \{a_2, a_3\}\} \\ \{\{a_1\}, \{a_3\}, \{a_2, a_4\}\} \\ \{\{a_1\}, \{a_2\}, \{a_3, a_4\}\} \end{cases}$
4	$G = \{1, 1, 1, 1\}, F(G) = \{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}\}$

5.2 具体实现

算法实现分为3个部分:1)预处理操作;2)构造和选择合适的配置;3)搜索元素。

1) 预处理操作

预处理的主要目的是计算 CL_s 的最大值 \max_s 和平均值 avg_s 并返回计算值,同时将当前搜索社会福利最大的联盟结构返回记为 CS' 。此过程中,主要进行2种操作,分别是评估互补大小和包含配置为 $\{1, 2, \dots, i\}$ 的联盟。

①评估互补大小:将 CL_s 和 CL_{n-s} 称为互补的,如 CL_3 和 CL_1 。在此过程中需要计算由互补

中的联盟组成的不相交的联盟结构的值。

②包含配置为 $\{1, 2, \dots, i\}$ 的联盟:计算 $G = \{1, 2, \dots, i\}$ 配置的联盟结构的值。

经过上述两个操作搜索联盟结构,可以搜索空间的一部分(随着 n 的增加而减小)。表4表示的是8个Agent搜索结果,其中配置 $G = \{8\}$, $G = \{4, 4\}$, $G = \{3, 5\}$, $G = \{2, 6\}$, $G = \{1, 7\}$, $G = \{1, 1, 6\}$, $G = \{1, 1, 1, 5\}$, $G = \{1, 1, 1, 1, 1, 3\}$, $G = \{1, 1, 1, 1, 1, 3\}$, $G = \{1, 1, 1, 1, 1, 1, 2\}$, $G = \{1, 1, 1, 1, 1, 1, 1, 1\}$ 是在预处理操作中需要将被搜索的配置。

2) 构造和选择合适的配置

表4 8个Agent配置表

Table 4 The configuration table of 8 Agents

层数	配置表
$l = 1$	$G = \{8\}$
$l = 2$	$G = \{4, 4\}$ $G = \{3, 5\}$ $G = \{2, 6\}$ $G = \{1, 7\}$
$l = 3$	$G = \{2, 3, 3\}$ $G = \{2, 2, 4\}$ $G = \{1, 3, 4\}$ $G = \{1, 2, 5\}$ $G = \{1, 1, 6\}$
$l = 4$	$G = \{2, 2, 2, 2\}$ $G = \{1, 2, 2, 3\}$ $G = \{1, 1, 3, 3\}$ $G = \{1, 1, 2, 4\}$ $G = \{1, 1, 1, 5\}$
$l = 5$	$G = \{1, 1, 2, 2, 2\}$ $G = \{1, 1, 1, 2, 3\}$ $G = \{1, 1, 1, 1, 1, 3\}$
$l = 6$	$G = \{1, 1, 1, 1, 2, 2\}$ $G = \{1, 1, 1, 1, 1, 3\}$
$l = 7$	$G = \{1, 1, 1, 1, 1, 1, 2\}$
$l = 8$	$G = \{1, 1, 1, 1, 1, 1, 1, 1\}$

在搜索过程中,进行剪枝操作主要目的是减少搜索空间。 G' 为搜索完后找到可能存在最优联盟结构的配置。

①构造独一无二的配置:保证所有的配置的集合等于所有Agent可能产生的整数分区。例如,有5个Agent,可能存在的整数分区的配置 G 为 $\{5\}$, $\{4, 1\}$, $\{3, 2\}$, $\{3, 1, 1\}$, $\{2, 2, 1\}$, $\{2, 1, 1, 1\}$ 和 $\{1, 1, 1, 1, 1\}$ 。

计算每个配置 G 对应于 $F(G)$ 的上界 UB_G 和平均值 Avg_G 。将 $V(CS')$ 和 $UB_G \beta$ 进行比较,如果 G' 中所有的配置都满足前者大于后者,那么该解即为最优解。否则,进行②操作。

②选择合适的配置:搜索过程中满足 $\frac{\text{AVG}}{\text{UB}_G^*} \geq k$, 则选择元素中最小的一个,否则继续进行剪枝工作。

③更新上边界: 进行搜索的 CS^* 是可以计算本身的数值或者将其自身分裂后形成联盟的最大值。例如, 在搜索配置为 $G=\{1, 1, 1, 1, 2, 3, 4\}$ 的元素时, 最大值可能的组合是 $\{1, 1, 3\}$ 。当求配置 $G=\{1, 1, 1, 3, 3, 4\}$ 的上边界时, 在配置 $G=\{1, 1, 3\}$ 的上边界的基础上加上 $\max_1 + \max_3 + \max_4$ 。以该方式对配置进行拆分处理, 能更好地更新上边界。

3) 搜索元素

搜索过程的关键在于构建联盟结构。在该算法中, 构建联盟结构的主要问题是, 在执行时会进行重复操作, 从而形成冗余和无效的联盟结构。

①避免重叠的联盟结构: 重叠的联盟结构是 $\{a_1, a_2\}\{a_2, a_3\}$ 这样。使用快速索引技术^[36]进行解决。

②避免冗余的联盟结构: 冗余的联盟结构是 $\{a_1, a_2\}, \{a_3, a_4\}$ 和 $\{a_3, a_4\}, \{a_1, a_2\}$ 这样。在进行组合的过程中以升序排列, 并且使用联盟中最小的元素 C_k^1 作为关键点, 以确定已经存在的联盟。

③正确性证明: 通过上述步骤可以生成唯一的和有效的解决方案。

6 约束条件下联盟生成求最优解

Frankovi 等^[37]将关注点从所有 Agent 之间的联系转移到只关注给定 Agent 之间建立最优的联盟。Greco 等^[18]经过进一步研究, 提出了一种新的联盟结构生成方式——约束联盟结构, 即在生成联盟的过程对联盟添加约束。

6.1 约束联盟的基础

在进行约束联盟结构形成的过程中, 主要是定义在一对 $\langle N, v \rangle$ 上的, $\langle N, v \rangle$ 称为联盟博弈, 其中 $N = (a_1, a_2, \dots, a_n)$ 是 Agent 集, v 是评估函数。

例 3 假设一个联盟博弈为 $\langle N, v \rangle$, 其中 $N = (a_1, a_2, a_3)$, 评估函数为

$$\begin{aligned} v(\{a_3\}) &= 0 \\ v(\{a_1\}) &= v(\{a_2\}) = v(\{a_1, a_3\}) = v(\{a_2, a_3\}) = 1 \\ v(\{a_1, a_2\}) &= v(\{a_1, a_2, a_3\}) = 3 \end{aligned}$$

易知, 最优联盟结构是 $\{a_1, a_2, a_3\}$ 和 $\{a_1, a_2\}, \{a_3\}$, 这是因为 $v(\{a_1, a_2\}) + v(\{a_3\}) = v(\{a_1, a_2, a_3\}) = 3$ 。

假设在联盟博弈 $\langle N, v \rangle$ 中, Agent 集上定义一个无向图 G , G 称为交互图。联盟 C 是可行的, 当且仅当 C 在 G 的子图上是连通的。

6.2 约束联盟结构生成

令 $\Gamma = \langle N, v \rangle$ 是联盟博弈, $D = (N, E)$ 是联盟博弈上的一个交互图, 其中图中的交点是 N 中的 Agent, 边是节点集, 设 $S \in N$ 是一个可能为空的

Agent 集, 称为枢纽 Agent。规定任意一个可行联盟结构中不允许存在两个及其以上的枢纽 Agent。 $\alpha, \beta: S \mapsto \mathbf{R}$ 是两个评估函数, 而 $x, y \in \mathbf{R}$ 是两个实数, 则 $\sigma = \langle D, S, \alpha, \beta, x, y \rangle$ 是 $\langle N, v \rangle$ 的评估结构。

例 4 给定一个联盟博弈 $\Gamma = \langle N, v \rangle$, 其中 $N = \{a_1, a_2, \dots, a_7\}$ 。评估结构为 $\sigma = (D, \{a_1, a_2\}, \alpha, \beta, x, y)$, 其中图 D 如图 4 所示。

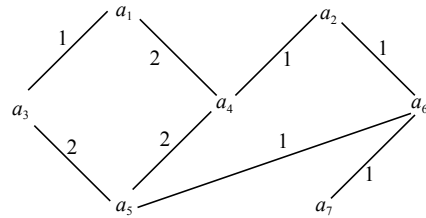


图 4 7 个 Agent 交互图

Fig. 4 The interactive graph of 7 Agents

图 4 中 $C_1 = \{a_1, a_3\}$, $C_2 = \{a_4, a_5, a_6\}$ 是 2 个可行联盟, 而 $\{a_1, a_2\}$ 和 $\{a_5, a_7\}$ 是 2 个不可行联盟, 前者是因为联盟中 2 个 Agent 都是枢纽 Agent, 后者则是因为联盟不是图 4 的子图。

如果 C 是可行联盟结构, 将函数 α 、 β 和实数 x 、 y 通过式 (2) 定义成评估函数 v 的映射 val_σ 。

$$\text{val}_\sigma(v, C) = \begin{cases} \alpha(a_i) \times v(C) + \beta(a_i), & \{a_i\} = C \cap S \\ x \times v(C) + y, & C \cap S = \emptyset \end{cases} \quad (2)$$

例 5 假设图 4 中, 估值函数为 v , C 是一个可行联盟, 则 $v(C)$ 被定义为由 C 所覆盖的边的权值的总和, 即

$$v(C) = \sum_{e \in E, e \in C} w_e$$

$C_1 = \{a_1, a_3\}$, $C_2 = \{a_5, a_6, a_7\}$ 的估值为

$$\begin{aligned} v(C_1) &= w(a_1, a_3) = 1 \\ v(C_2) &= w(a_5, a_6, a_7) = w(a_5, a_6) + w(a_6, a_7) = 1 + 1 = 2 \end{aligned}$$

假设函数 α 、 β 分别为 $\alpha: \{a_1, a_2\} \rightarrow \{1\}$, $\beta: \{a_1, a_2\} \rightarrow \{0\}$, 而实数 $x=0$, $y=-12$, 则 $\text{val}_\sigma(C_1) = 1 \times v(C_1) + 0 = 1$, $\text{val}_\sigma(C_2) = 0 \times v(C_2) - 12 = -12$ 。

6.3 边际贡献网上联盟生成

边际贡献网^[38]在过去的几年里收到了相当大的关注。边际贡献网 (marginal contribution network, MC-net) 是包含一系列代表 Agent 的布尔变量的规则, 每个规则的形式为 $\{\text{pattern}\} \rightarrow \text{value}$, 其中 pattern 可以包含正面的连接或负面的连接, value 是与此 pattern 相关的附加贡献。

例 6 给定一个联盟博弈 $\Gamma = \langle N, v \rangle$, 其中 $N = \{a_1, a_2, \dots, a_5\}$, $v(\{a_1, a_2\}) = v(\{a_2, a_3\}) = v(\{a_1, a_3\}) = 2$; $\exists i \in \{1, 2, \dots, 5\}, v(\{a_i\}) = 0; v(\{a_1, a_2, a_3\}) = 5; \exists C \subseteq \{a_1, a_2, a_3\}, v(C \cup \{a_4\}) = v(C \cup \{a_5\}) = v(C) = v(C \cup \{a_4, a_5\})$; 根据边际贡献网, 给定规则为

$$\begin{aligned}
&\{a_1 \wedge a_2 \wedge \neg a_3\} \rightarrow 2 \\
&\{\neg a_1 \wedge a_2 \wedge a_3\} \rightarrow 2 \\
&\{a_1 \wedge \neg a_2 \wedge a_3\} \rightarrow 2 \\
&\{a_1 \wedge a_2 \wedge a_3\} \rightarrow 5 \\
&\{a_3 \wedge a_5\} \rightarrow 0 \\
&\{a_2 \wedge a_4\} \rightarrow 0
\end{aligned}$$

根据给定的规则, $v(\{a_1, a_1, a_1\}) = 2$ 。

定理5 令 $h \geq 0$ 是一个固定自然数, $\Gamma = \langle N, v \rangle$ 是一个博弈, 其中 v 是独立于不相交成员的评估函数, 交互图 D 所构成的树的高大于 h 。令 $\sigma = \langle D, S, \alpha, \beta, x, y \rangle$, 则 (Γ, σ) 的联盟结构生成的值可以在多项式时间内解决。

约束满足问题具有较强的灵活性, 能够进行推广解决文献[32]中提到的约束问题。

7 结束语

联盟结构生成问题中如何能够更快更精确地进行优化整体性能对于研究人员一直都是挑战。本文针对这一问题进行了全面的综述: 1) 搜索空间的不同表示; 2) 动态规划算法的时候来解决这个问题; 3) 使用近似最优来代替最优解; 4) 使用评估的方法, 主要是添加了约束条件。所有的这些结构都是以一种独立的方式展示出来的。

但是, 在现有联盟生成的研究中, 大多数是以 Agent 之间具有相同的协商资源和态度为前提的, 只考虑了协商中的相同性, 却忽略了协商中的异质性。在未来的工作中, 可以考虑将协商的异质性与约束联盟中的评估结构相结合。

同时, 在约束 K-means 聚类算法的启发下, 可以将一般形式的无法连接的约束进行合理操作后有理性地加入到评估结构中。另外, 可以进行扩展, 将允许联盟生成的概率使用先验知识进行检验或者将 Agent 之间的动态相互信任添加到联盟生成的过程中。

参考文献:

- [1] AZIZ H. Multiagent systems: algorithmic, game-theoretic, and logical foundations by Y. Shoham and K. Leyton-Brown Cambridge University Press, 2008[J]. [ACM SIGACT news](#), 2010, 41(1): 34–37.
- [2] WOOLDRIDGE M. Computational aspects of cooperative game theory[C]//Proceedings of the 5th KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications. Manchester, UK, 2011.
- [3] ELKIND E, RAHWAN T, JENNINGS N R. Computational coalition formation[M]. Cambridge: MIT Press, 2013: 329–380.
- [4] OSBORNE M J, RUBINSTEIN A. A course in game theory[M]. Cambridge: MIT Press, 1994.
- [5] VON NEUMANN J, MORGENSTERN O. Theory of games and economic behavior[M]. Princeton: Princeton University Press, 1972.
- [6] WOOLDRIDGE M, BUSSMANN S, KLOSTERBERG M. Production sequencing as negotiation[C]// Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology. London, UK, 1996: 709–726.
- [7] SANDHOLM T, SIKKA S, NORDEN S. Algorithms for optimizing leveled commitment contracts[C]// Proceedings of the 16th International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 1999: 535–540.
- [8] RAHWAN T, MICHALAK T P, WOOLDRIDGE M, et al. Coalition structure generation: a survey[J]. [Artificial intelligence](#), 2015, 229: 139–174.
- [9] ZHU Han, POOR H V. Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks[C]// Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops. Limassol, Cyprus, 2009: 1–8.
- [10] KHAN Z, LEHTOMÄKI J, LATVA-AHO M, et al. On selfish and altruistic coalition formation in cognitive radio networks[C]//Proceedings of the 2010 Fifth International Conference on Cognitive Radio Oriented Wireless Networks and Communications. Cannes, France, 2010: 1–5.
- [11] LI Cuihong, SYCARA K, SCHELLER-WOLF A. Combinatorial coalition formation for multi-item group-buying with heterogeneous customers[J]. [Decision support systems](#), 2010, 49(1): 1–13.
- [12] MANOOCHEHRI H E, WENKSTERN R Z. Dynamic coalition structure generation for autonomous connected vehicles[C]// Proceedings of 2017 IEEE International Conference on Agents. Beijing, China, 2017: 21–26.
- [13] SLESS L, HAZON N, KRAUS S, et al. Forming k coalitions and facilitating relationships in social networks[J]. [Artificial intelligence](#), 2018, 259: 217–245.
- [14] SLESS L, HAZON N, KRAUS S, et al. Forming coalitions and facilitating relationships for completing tasks in social networks[C]// Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems. Paris, France, 2014: 261–268.
- [15] SANDHOLM T, LARSON K, ANDERSSON M, et al. Coalition structure generation with worst case guarantees [J]. [Artificial intelligence](#), 1999, 111(1/2): 209–238.
- [16] YE D Y. A dynamic programming approach to the complete set partitioning problem[J]. [BIT numerical mathematics](#), 1986, 26(4): 467–474.
- [17] DANG T T, FRANKOVIE B, BUDINNSKA I. Create agent's coalition based on a dynamic programming ap-

- proach[C]//Proceedings of the 15th European Conference on Artificial Intelligence, Workshop “Agent Technologies and Logistics”. Lyon, France, 2002: 16–24.
- [18] GRECO G, GUZZO A. Constrained coalition formation on valuation structures: formal framework, applications, and islands of tractability[J]. *Artificial intelligence*, 2017, 249: 19–46.
- [19] 童向荣, 张伟. 动态联盟收益值的再励学习[J]. *计算机工程与应用*, 2006, 42(6): 85–87.
TONG Xiangrong, ZHANG Wei. Reinforcement learning for the value of dynamic coalition[J]. *Computer engineering and application*, 2006, 42(6): 85–87.
- [20] DEAN T, BODDY M. An analysis of time-dependent planning[C]// Proceedings of the 7th National Conference on Artificial Intelligence. St. Paul, Brazil, 1988: 49–54.
- [21] BRUALDI R A. 组合数学[M]. 冯舜玺, 译. 5 版. 北京: 机械工业出版社, 2012: 180–186.
BRUALDI R A. Introductory combinatorics[M]. FENG Shunxi, trans. 5th ed. Beijing: China Machine Press, 2012: 180–186.
- [22] KARP R M. Reducibility among combinatorial problems[C]// Complexity of Computer Computations. New York, USA, 1972: 85–103.
- [23] HASTAD J. Clique is hard to approximate within $n^{1-\epsilon}$ [J]. *Acta mathematica*, 1999, 182(1): 105–142.
- [24] SANDHOLM T. An algorithm for optimal winner determination in combinatorial auction[C]// Proceedings of the Sixteenth International joint conference on artificial intelligence. Stockholm, Sweden, 1999: 542–547.
- [25] LINIAL N. Game-theoretic aspects of computing[M]// AUMANN R J, HART S. Handbook of Game Theory with Economic Applications. Amsterdam, Netherlands: Elsevier, 1994: 1339–1395.
- [26] LARSON K S, SANDHOLM T W. Anytime coalition structure generation: an average case study[J]. *Journal of experimental & theoretical artificial intelligence*, 2000, 12(1): 23–42.
- [27] 胡山立, 石纯一. 一种任一时间联盟结构生成算法[J]. *软件学报*, 2001, 12(5): 729–734.
HU Shanli, SHI Chunyi. An anytime coalition structure generation algorithm[J]. *Journal of software*, 2001, 12(5): 729–734.
- [28] 胡山立, 石纯一. 给定限界要求的联盟结构生成[J]. *计算机学报*, 2001, 24(11): 1185–1190.
HU Shanli, SHI Chunyi. Coalition structure generation with given required bound[J]. *Chinese journal of computers*, 2001, 24(11): 1185–1190.
- [29] DANG V D, JENNINGS N R. Generating coalition structures with finite bound from the optimal guarantees[C]// Proceedings of 3rd International Joint Conference on Autonomous Agents and Multiagent Systems. New York, USA, 2004: 564–571.
- [30] ROTHKOPF M H, PEKEČ A, HARSTAD R M. Computationally manageable combinatorial auctions[J]. *Management science*, 1998, 44(8): 1131–1147.
- [31] 刘惊雷, 童向荣, 张伟. 一种快速构建最优联盟结构的方法[J]. *计算机工程与应用*, 2006, 42(4): 35–37, 44.
LIU Jinglei, TONG Xiangrong, ZHANG Wei. A kind of method for quick constructing optimal coalition structure[J]. *Computer engineering and applications*, 2006, 42(4): 35–37, 44.
- [32] RAHWAN T, MICHALAK T P, ELKIND E, et al. Constrained coalition formation[C]//Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence. San Francisco, USA, 2011: 719–725.
- [33] 张新良, 石纯一. 多 Agent 联盟结构动态生成算法[J]. *软件学报*, 2007, 18(3): 574–581.
ZHANG Xinliang, SHI Chunyi. A dynamic formation algorithm of multi-agent coalition structure[J]. *Journal of software*, 2007, 18(3): 574–581.
- [34] RAHWAN T, RAMCHURN S D, DANG V D, et al. Near-optimal anytime coalition structure generation[C]// Proceedings of the 20th International Joint Conference on Artificial Intelligence. Hyderabad, India, 2007: 2365–2371.
- [35] ALBIZURI M J, AURRECOECHEA J, ZARZUELO J M. Configuration values: extensions of the coalitional Owen value[J]. *Games and economic behavior*, 2006, 57(1): 1–17.
- [36] RAHWAN T, JENNINGS N R. Distributing coalitional value calculations among cooperative agents[C]// Proceedings of the 20th National Conference on Artificial Intelligence. Pittsburgh, Pennsylvania, 2005: 152–157.
- [37] FRANKOVI B, DANG T, BUDINSKÁ I. Agents’ coalitions based on a dynamic programming approach[J]. *Acta polytechnica hungarica*, 2008, 5(2): 5–21.
- [38] IEONG S, SHOHAM Y. Marginal contribution nets: a compact representation scheme for coalitional games[C]// Proceedings of the 6th ACM Conference on Electronic Commerce. Vancouver, BC, Canada, 2005: 193–202.

作者简介:



任子仪, 女, 1994 年生, 硕士研究生, 主要研究方向为联盟结构生成和数据挖掘。



童向荣, 男, 1975 年生, 教授, 博士, 主要研究方向为多 Agent 系统、分布式人工智能、数据挖掘。主持国家自然科学基金面上项目 2 项, 山东省自然科学基金 1 项。发表学术论文 50 余篇。