软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(5):938−950 [doi: 10.3724/SP.J.1001.2011.03817]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种 $O(2.983^n)$时间复杂度的最优联盟结构生成算法*

刘惊雷+, 张 伟, 童向荣, 张振荣

(烟台大学 计算机科学与技术学院,山东 烟台 264005)

$O(2.983^n)$ Time Complexity Algorithm for Optimal Coalition Structure Generation

LIU Jing-Lei+, ZHANG Wei, TONG Xiang-Rong, ZHANG Zhen-Rong

(School of Computer Science and Technology, Yantai University, Yantai 264005, China)

+ Corresponding author: E-mail: Jinglei_liu@sina.com

Liu JL, Zhang W, Tong XR, Zhang ZR. $O(2.983^n)$ time complexity algorithm for optimal coalition structure generation. *Journal of Software*, 2011,22(5):938−950. http://www.jos.org.cn/1000-9825/3817.htm

Abstract: First, this paper establishes an effective partition relationship in the finite integer set and an effective splitting relationship in the coalition set, and devises an EOCS (effective optimal coalition structure) algorithm, which only evaluates bipartite effective splittings of coalition to find the optimal value from bottom to top, so it decreases the number of bipartite splitting. Secondly, the correctness of the EOCS algorithm is proved based on the Kleene closure function. Moreover, this paper proves that the EOCS lower bound is $\Omega(2.818^n)$ by the integration limit theorem, and discovers that the EOCS upper bound is $O(2.983^n)$ by the time serial analysis technique. Finally, this paper compares the EOCS algorithm with other algorithms to point out that the EOCS algorithm can find optimal coalition structure in $O(2.983^n)$ time whether the coalition values meet which probability distributions or not. The DP (dynamic programming) algorithm and the IDP (improved dynamic programming) algorithm proposed by Rothkopf and Rahwan can find an optimal solution in $O(3^n)$. The EOCS algorithm's design, correctness proof, and time complexity analysis are all improvements of Rothkopf and Rahwan's related work.

Key words: optimal coalition structure; effective bipartite splitting; Kleene closure; upper and lower bound of time complexity; integration limit theorem; time series analysis

摘 要: 首先,在有限整数集上建立有效拆分关系,在联盟集上建立有效二部分解关系,并设计了一种 EOCS (effective optimal coalition structure)算法.该算法采用自底向上方式,只对具有有效二部分解关系的联盟进行二部分解来求联盟的优值,从而降低了二部分解的数量.随后,利用函数的克林闭包特性证明了EOCS算法的正确性,利用积分极限定理证明了 EOCS 算法时间复杂度的下界是$\Omega(2.818^n)$,用时间序列分析方法求出了 EOCS 算法的上界是$O(2.983^n)$.最后,将EOCS算法与其他算法作了对比,指出无论联盟值满足何种概率分布,EOCS算法都能在$O(2.983^n)$时间内找出最优联盟结构.Rothkopf提出的DP(dynamic programming)算法和Rahwan提出的IDP(improved dynamic programming)算法能够在$O(3^n)$时间内求出最优联盟结构.所作的 EOCS 算法设计、正确性证明、时间复杂度的上下界分析都是对 Rothkopf 及 Rahwan 等人相关工作的改进和提高.

关键词: 最优联盟结构;有效二部分解;克林闭包;时间复杂度的上下界;积分极限定理;时间序列分析

* 基金项目: 国家自然科学基金(60496323); 山东省教育厅科技计划(J07JYJ24)
收稿时间: 2009-04-05; 修改时间: 2009-07-29; 定稿时间: 2010-01-05

中图法分类号: TP301 文献标识码: A

多 Agent 系统中,由于单个 Agent 资源和能力的限制,为了高效地完成给定任务,多个 Agent 需要组成联盟以便进行高效的合作.例如,现实世界的“拼车”一族为了降低自己乘车费用,从而多个人组成联盟,以求得整体和单个费用的降低;经济学中的组合拍卖,投标者可将多个拍卖品组合在一起集中竞拍,以取得较低的竞拍价钱;分布式传感器网络中,多个传感器可组成联盟,以共同跟踪感兴趣的目标[1].

Agent 联盟的研究问题有 3 类[2].(1) 联盟结构的生成问题.该问题研究如何对多个 Agent 进行划分,使得整个联盟结构的收益最大化.该问题关注的是整个 Agent 系统的收益,而不是单个或部分 Agent 的收益.(2) 联盟值的优化问题.该问题描述多个 Agent 组成联盟,合伙经营他们的资源、任务和能力,以使这些 Agent 得到的收益最大化.它关注自利的小组织的局部收益.(3) 联盟值的分配问题.该问题描述如何在组成联盟的多个 Agent 之间进行收益分配.它追求 Agent 合作完成并取得收益后,如何公平地分配这些收益,从而使得 Agent 能够建立长期联盟.本文研究 Agent 联盟的第 1 个问题.

近年来,有越来越多的学者在研究 Agent 联盟的第 1 个问题[2−14].Sandholm[2]的工作最具开创性,他首次提出了一种可保证最坏情况下的最优值的求解方法.但它是 Anytime 算法,其求出最优值需搜索 $O(n^n)$的空间.Rothkopf[3]及刘惊雷等人[4]利用 DP(dynamitic programming)算法求解最优联盟结构,解决了大量重复子问题的计算.张新良等人在假定联盟值满足合作收益独立(新加入的 Agent 不影响加入前其他 Agent 之间的收益大小关系)[5]的前提下,对联盟结构图进行简化,从而减少搜索量.其本质是在限定条件下的联盟结构生成问题,并且它是一种 Anytime 算法.当要找出最优值时,其时间复杂度为 $O(n^n)$.Dang 等人[6]第一个提出了不以层为搜索单位的任一时间算法.Shehory 等人[7]提出了对联盟大小进行限制,利用贪心算法求最优联盟结构,但它不保证一定能够得出最优值.Rahwan 等人[8]提出了 IDP(improved dynamitic programming)算法,它可以降低搜索次数.文献[8]只有实验描述,没有算法的设计和分析,但他们在文献[9]中指出该算法的复杂度为 $O(3^n)$.Rahwan 等人还给出了一种在满足某个质量要求下的算法[10,11].它可以在任一时间得出较优值,但要得到最优值仍需搜索 $O(n^n)$的空间.

最近两年还出现了一些基于正整数拆分的最优联盟结构生成算法[9−14],本质上说,它们都是文献[12]所给出的整数拆分图和联盟结构图的代数性质的应用.其中,文献[9−11]设计的算法是正整数的 $k(k=2,3,\ldots,n-1)$部拆分在 Anytime 算法中的应用.由于这些算法所给出的联盟值都满足某种概率分布,因此最坏情况下需搜索 $O(n^n)$的空间;文献[12]所给出的算法是利用整数的二部拆分来遍历联盟结构图,该算法所给出的联盟值也满足某种概率分布,但它不是 Anytime 算法,最坏情况下只搜索 $O(3^n)$的空间.本文所设计的 ECOS(effective optimal coalition structure)算法是利用正整数的有效二部拆分来遍历联盟结构图.该算法与文献[9−11]的区别在于,无论联盟值满足何种分布,EOCS 都能够在确定的 $O(2.983^n)$时间内找到最优值.

目前,联盟结构生成算法的分析大多还停留在仅给出一些实验数据.本文的特色在于:

(1) 在有限整数集上定义了有效拆分关系,在联盟集上定义了有效二部分解关系,设计了 EOCS 算法,它只对具有有效分解关系的联盟进行二部分解,从而减少分解次数.

(2) 借助于函数的克林闭包证明了算法的正确性,利用积分极限定理证明了 EOCS 算法的下界为 $\Omega(2.818^n)$,利用时间序列分析方法求出 EOCS 算法的上界为 $O(2.983^n)$.

(3) 对目前求最优联盟结构算法做了对比,阐述了 EOCS 算法的优点在于它对联盟值的概率分布不作任何假设,因此无论输入数据满足何种统计规律,该算法都能在确定的 $O(2.983^n)$时间内找出最优值.

1 问题描述及基本定义

1.1 最优联盟结构的相关概念

定义 1. 设 Agent 的集合 $A=\{1,2,3,\ldots,n\}$,联盟 S 是 A 的一个非空子集;联盟结构 CS 是 A 的分解,它由 A 的

互不相交的非空子集族组成,且这些子集的并等于 A,即

$$CS:=(\cup CS=A)\wedge\forall B\forall C((B,C\in CS\wedge B\neq C\neq\varnothing)\rightarrow B\cap C=\varnothing).$$

L 是 A 可能形成的所有联盟结构,即 $L=\bigcup_{k=1}^{n}L_k$,其中,L_k 代表联盟结构图中第 k 层的所有顶点.

定义 2. 联盟值 $V(C)$是多个 Agent 组成联盟 C 时产生的可能收益,它映射为一个实数;联盟结构值 $V(CS)$是联盟结构 CS 中所有联盟值的和,即 $V(CS)=\sum_{C\in CS}V(C)$;最优联盟结构 $CS^*=\text{argmax}_{CS\in L}V(CS)$.

定义 3. 若正整数 $m=m_1+m_2$,且 $m_1\geqslant m_2$,则称将 m 二部拆分成 m_1 和 m_2,或者 m 的一个二部拆分是$\langle m_1,m_2\rangle$;若联盟 $S=S_1\cup S_2$ 且 $S_1\cap S_2=\varnothing$,$|S_1|\geqslant|S_2|$,则称将 S 二部分解为 S_1 和 S_2,或者 S 的二部分解是$\langle S_1,S_2\rangle$.

定理 1. n 个 Agent 可形成 2^n-1 个联盟,可形成 $2^{n-1}-1$ 个二部分解[4],整数 n 有$\lfloor n/2\rfloor$个二部拆分.

例 1:设 $A=\{1,2,3,4\}$,这 4 个 Agent 可形成 15 个联盟,$C_1=\{1\}$,$C_2=\{2\}$,$C_3=\{3\}$,$C_4=\{4\}$,$C_5=\{1,2\}$,$C_6=\{1,3\}$,$C_7=\{1,4\}$,$C_8=\{2,3\}$,$C_9=\{2,4\}$,$C_{10}=\{3,4\}$,$C_{11}=\{1,2,3\}$,$C_{12}=\{1,2,4\}$,$C_{13}=\{1,3,4\}$,$C_{14}=\{2,3,4\}$,$C_{15}=\{1,2,3,4\}$;还可以形成如图 1 所示的 15 个联盟结构,其中,L_2 层含 $2^{4-1}-1=7$ 个二部分解.整数 4 的两个二部拆分是$\langle 3,1\rangle$和$\langle 2,2\rangle$.

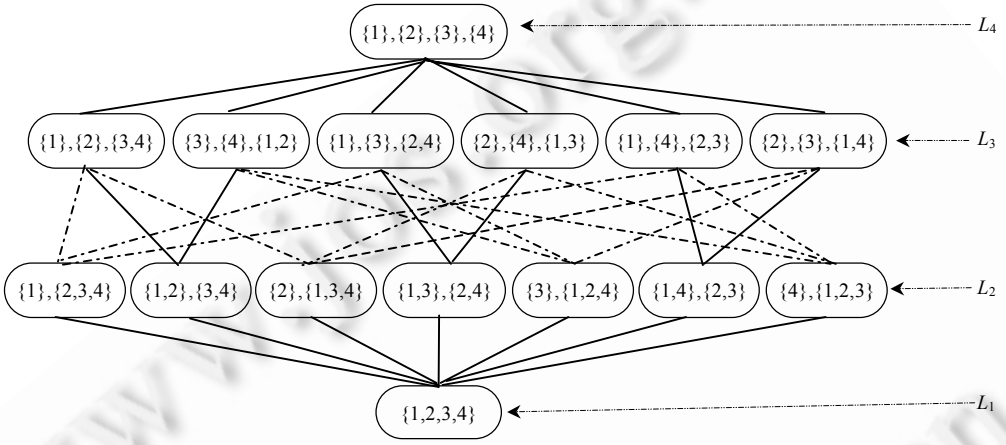


Fig.1 Coalition structure graph of 4 agents

图 1 4 个 Agent 的联盟结构图

联盟结构生成算法是在某种数据模型(主要考虑联盟值 $V(C)$的特性)前提下求解,目前有两类模型:文献[13,14]所提出的算法都是在 PFG(partition function games)模型下的最优联盟结构生成算法,而本文和其他文献[2−12]的算法都是在 CFG(characteristic function games)下的最优联盟结构生成算法.其区别在于,CFG 数据模型中的联盟值满足独立性(即一个联盟值的大小取决于联盟内部成员的合作方式,而与其他联盟无关)或不变性(即联盟 C 的联盟值 $V(C)$在任何一个联盟结构中的值都一样)假设.该特性保证了当联盟 C 确定以后,其联盟值 $V(C)$就能确定,不受其他联盟和所处的联盟结构的影响.而 PFG 数据模型下的联盟值则不满足独立性或不变性假设.

1.2 整数的有效拆分与联盟的有效分解

定义 4. 若$\langle m_1,m_2\rangle$是正整数 m 的二部拆分,则在有限整数集 $M=\{1,2,3,4,\ldots,n\}$上定义的三元关系$\propto$为

$$\propto:=\{\langle m,m_1,m_2\rangle|m,m_1,m_2\in M\wedge((m=n)\vee(m_1\leqslant n-m))\}.$$

它表示当 m 的二部拆分$\langle m_1,m_2\rangle$满足 $m=n$ 或者 $m_1\leqslant n-m$ 时,m 可二部有效拆分成 m_1 和 m_2(或者$\langle m_1,m_2\rangle$是 m 的有效二部拆分,简称有效拆分),记作 $m\propto\langle m_1,m_2\rangle$.

定义 5. 若$\langle S_1,S_2\rangle$是联盟 S 的二部分解,则在联盟集上定义的三元关系$\propto'$为

$$\propto':=\{\langle S,S_1,S_2\rangle||S|\propto\langle|S_1|,|S_2|\rangle\},$$

其中,$|S|$为联盟 S 包含的 Agent 个数.

它表示当 S 的二部分解$\langle S_1,S_2\rangle$满足$|S|\propto\langle|S_1|,|S_2|\rangle$时,联盟 S 可二部有效分解为联盟 S_1 和 S_2(或者$\langle S_1,S_2\rangle$是 S 的有效二部分解,简称有效分解),记作 $S\propto'\langle S_1,S_2\rangle$.

由图 1 可知,顶点{{1},{2},{3,4}}向下的实线仅仅保留与顶点{{1,2},{3,4}}的连接,而与其余顶点的边都删除(用虚线代表).其中,实线就是有效分解连接的边.由于删除许多边,故分解次数降低,但又保证每个顶点至少含有 1 条连接下层顶点的实线边,因此从根顶点可通到联盟结构图中任何一个顶点,保证了图的连通性.

例 2:设 A 包含 5 个 Agent,即$|A|=n=5$,若 $S=\{1,2,3\}$,由于 $3\propto\langle 2,1\rangle$,则 $S\propto'\langle\{2,3\},\{1\}\rangle$,它表示联盟 S 可有效分解成{2,3}和{1}.由于$\neg(4\propto\langle 2,2\rangle)$,因此联盟 $S'=\{1,2,3,4\}$不能有效分解成{1,2}和{3,4};同样,$\neg(4\propto\langle 3,1\rangle)$,因此联盟 S' 不能有效分解成{2,3,4}和{1},即$\neg(S'\propto'\langle\{2,3,4\},\{1\}\rangle)$.

其实,有效拆分和有效分解概念是一个在联盟结构图中如何画出实线与虚线的方法的形式化描述.在图 1 的联盟结构图中,L_k 的一个顶点$\{S_1,S_2,S_3,\ldots,S_k\}(|S_1|\leqslant|S_2|\leqslant|S_3|\leqslant\ldots\leqslant|S_k|)$只与 L_{k-1} 中的顶点$\{S_1\cup S_2,S_3,\ldots,S_k\}$用实线相连,而与 L_{k-1} 中的其他顶点$\{S_i\cup S_j,S_m,\ldots,S_k\}(\langle i,j\rangle\neq\langle 1,2\rangle)$用虚线相连.用实线相连的是有效分解,用虚线相连的不是有效分解.

1.3 联盟的有效分解性质

定理 2. 设 S,S_1 和 S_2 是 A 的联盟,$\langle S_1,S_2\rangle$是 S 的二部分解,其中,$|A|=n,|S|=m,|S_1|=m_1,|S_2|=m_2$,

(1) 若 $m\in[2,\lceil n/2\rceil]$,则 S 的所有二部分解$\langle S_1,S_2\rangle$都是有效分解;

(2) 若 $m\in[\lfloor 2n/3\rfloor+1,n-1]$,则 S 的所有二部分解$\langle S_1,S_2\rangle$都不是有效分解;

(3) 若 $m\in[\lceil n/2\rceil+1,\lfloor 2n/3\rfloor]$,则 S 的部分二部分解$\langle S_1,S_2\rangle$是有效分解;

(4) 若 $m=n$,即 $S=A$ 时,则 A 的所有二部分解$\langle S_1,S_2\rangle$都是有效分解.

证明:这里只证明定理 2(1),其他的类似.

(1) 只需证明$|S|\propto\langle|S_1|,|S_2|\rangle$,即 $m\propto\langle m_1,m_2\rangle$.

(a) 若 n 为偶数,即 $n=2k$,则由 $m\leqslant\lceil n/2\rceil$得 $m\leqslant k$ (1-1)

因为 $m=m_1+m_2$,由步骤(1-1)得 $m_1<k$ (1-2)

又因为 $n-m=2k-m$,由步骤(1-1)得 $n-m\geqslant k$ (1-3)

由步骤(1-2)和步骤(1-3)得 $m_1<n-m$ (1-4)

(b) 若 n 为奇数,即 $n=2k+1$,则由 $m\leqslant\lceil n/2\rceil$得 $m\leqslant k+1$ (2-1)

因为 $m=m_1+m_2$,且 $m_2\geqslant 1$,由步骤(2-1)得 $m_1\leqslant k$ (2-2)

又因为 $n-m=2k+1-m$,由步骤(2-1)得 $n-m\geqslant k$ (2-3)

由步骤(2-2)和步骤(2-3)得 $m_1\leqslant n-m$ (2-4)

由步骤(1-4)和步骤(2-4)可知,无论 n 为偶数还是奇数,都有 $m_1\leqslant n-m$.由定义 4 得$|S|\propto\langle|S_1|,|S_2|\rangle$.再根据定义 5 得 $S\propto'\langle S_1,S_2\rangle$,由 S_1 和 S_2 的任意性可知,S 的所有二部分解$\langle S_1,S_2\rangle$都是有效分解. □

2 EOCS 算法的设计和证明

由定理 2 可知,长度为 m 的联盟 S 以$\lceil n/2\rceil$,$\lfloor 2n/3\rfloor$和 n 为分界线,将长度为 2 到 n 的联盟分为 4 段:当 $m\leqslant\lceil n/2\rceil$时,联盟 S 的所有二部分解都是有效分解;当$\lceil n/2\rceil+1\leqslant m\leqslant\lfloor 2n/3\rfloor$时,联盟 S 具有部分有效分解,随着 m 的增加,S 的有效分解越来越少;当$\lfloor 2n/3\rfloor+1\leqslant m\leqslant n-1$ 时,S 的所有二部分解都不是有效分解;但当 $m=n$ 时,即 A 的所有二部分解都是有效分解.

2.1 EOCS算法的求解过程

表 1 给出了 5 个 Agent 系统的 DP 和 EOCS 算法的求解对比.其中,正常字体是 DP 和 EOCS 都执行的过程,带删除线的只是 DP 执行(EOCS 不执行)的过程,带下划线的只是 EOCS 执行(DP 不执行)的过程.

Table 1 Solving example of DP and EOCS algorithm for 5 agents

表 1 5 个 Agent 的 DP 和 EOCS 算法求解实例

The length of coalition	Coalition	Solving process for C and F	C	F	Remarks
1	{1}	$V(\{1\})=50$	{1}	50	Steps for DP and EOCS
	{2}	$V(\{2\})=40$	{2}	40	
	{3}	$V(\{3\})=30$	{3}	30	
	{4}	$V(\{4\})=20$	{4}	20	
	{5}	$V(\{5\})=60$	{5}	60	
2	{1,2}	$V(\{1,2\})=70$, $F[\{1\}]+F[\{2\}]=90$	{1},{2}	90	
	{1,3}	$V(\{1,3\})=90$, $F[\{1\}]+F[\{3\}]=80$	{1,3}	90	
	…	…	…	…	
	{4,5}	**$V(\{4,5\})=110$**, $F[\{4\}]+F[\{5\}]=80$	{4,5}	110	
3	{1,2,3}	**$V(\{1,2,3\})=160$**, $F[\{1\}]+F[\{2,3\}]=140$ $F[\{2\}]+F[\{1,3\}]=120$, $F[\{3\}]+F[\{1,2\}]=120$	{1,2,3}	160	
	{1,2,4}	**$V(\{1,2,4\})=100$**, $F[\{1\}]+F[\{2,4\}]=120$ $F[\{2\}]+F[\{1,4\}]=120$, $F[\{4\}]+F[\{1,2\}]=110$	{1},{2,4}	120	
	…	…	…	…	
	{3,4,5}	**$V(\{3,4,5\})=190$**, $F[\{3\}]+F[\{4,5\}]=140$ $F[\{4\}]+F[\{3,5\}]=110$, $F[\{5\}]+F[\{3,4\}]=110$	{3,4,5}	190	
4	{1,2,3,4}	**$V(\{1,2,3,4\})=180$**, ~~$F[\{1\}]+F[\{2,3,4\}]-200$~~ ~~$F[\{2\}]+F[\{1,3,4\}]-170$, $F[\{3\}]+F[\{1,2,4\}]-150$~~ ~~$F[\{4\}]+F[\{1,2,3\}]-180$, $F[\{1,2\}]+F[\{3,4\}]-140$~~ ~~$F[\{1,3\}]+F[\{2,4\}]-160$, $F[\{1,4\}]+F[\{2,3\}]-170$~~	~~{1},{2,3,4}~~ **{1,2,3,4}**	~~200~~ 180	Steps only for DP Steps only for EOCS
	…	…	…	…	
	{2,3,4,5}	$V(\{2,3,4,5\})=200$, ~~$F[\{2\}]+F[\{3,4,5\}]-230$~~ ~~$F[\{3\}]+F[\{2,4,5\}]-210$, $F[\{4\}]+F[\{2,3,5\}]-170$~~ ~~$F[\{5\}]+F[\{2,3,4\}]-210$, $F[\{2,3\}]+F[\{4,5\}]-200$~~ ~~$F[\{2,4\}]+F[\{3,5\}]-160$, $F[\{2,5\}]+F[\{3,4\}]-170$~~	~~{2},{3,4,5}~~ {2,3,4,5}	~~230~~ 200	
5	{1,2,3,4,5}	**$V(\{1,2,3,4,5\})=250$**, ~~$F[\{1\}]+F[\{2,3,4,5\}]-280$~~ ~~$F[\{2\}]+F[\{1,3,4,5\}]-280$, $F[\{3\}]+F[\{1,2,4,5\}]-260$~~ ~~$F[\{4\}]+F[\{1,2,3,5\}]-240$, $F[\{5\}]+F[\{1,2,3,4\}]-260$~~ ~~$F[\{1,2\}]+F[\{3,4,5\}]-80$, $F[\{1,3\}]+F[\{2,4,5\}]-270$~~ ~~$F[\{1,4\}]+F[\{2,3,5\}]-230$, $F[\{1,5\}]+F[\{2,3,4\}]-260$~~ ~~$F[\{2,3\}]+F[\{1,4,5\}]-250$, $F[\{2,4\}]+F[\{1,3,5\}]-220$~~ ~~$F[\{2,5\}]+F[\{1,3,4,\}]-250$, $F[\{3,4\}]+F[\{1,2,5\}]-220$~~ ~~$F[\{3,5\}]+F[\{1,2,4\}]-210$, $F[\{4,5\}]+F[\{1,2,3\}]-270$~~	~~{2},{1,3,4,5}~~	~~280~~	
		$V(\{1,2,3,4,5\})=250$, $F[\{1\}]+F[\{2,3,4,5\}]=250$ $F[\{2\}]+F[\{1,3,4,5\}]=260$, $F[\{3\}]+F[\{1,2,4,5\}]=220$ $F[\{4\}]+F[\{1,2,3,5\}]=80$, $F[\{5\}]+F[\{1,2,3,4\}]=240$ $F[\{1,2\}]+F[\{3,4,5\}]=80$, $F[\{1,3\}]+F[\{2,4,5\}]=270$ $F[\{1,4\}]+F[\{2,3,5\}]=230$, $F[\{1,5\}]+F[\{2,3,4\}]=260$ $F[\{2,3\}]+F[\{1,4,5\}]=50$, $F[\{2,4\}]+F[\{1,3,5\}]=20$ $F[\{2,5\}]+F[\{1,3,4,\}]=250$, $F[\{3,4\}]+F[\{1,2,5\}]=220$ $F[\{3,5\}]+F[\{1,2,4\}]=10$, $F[\{4,5\}]+F[\{1,2,3\}]=70$	{1,2},{3,4,5}	280	

表 1 的 EOCS 算法以自底向上方式求 C 和 F 的值.下面以例 3 来描述求最优联盟结构值 $V(CS^*)$和最优联盟结构 CS^*的过程.

例 3:A={1,2,3,4,5},所有的联盟值为 $V(\{1\})=50$,$V(\{2\})=40$,…(如表 1 中带着重号的文字所示),求 A 的最优联盟结构值 $V(CS^*)$及最优联盟结构 CS^*.

解:EOCS 和 DP 算法一样,需要辅助表 C 和 F 存储子问题的解[3,4].$C[S]$(简称联盟 S 的优结构)和 $F[S]$(简称联盟 S 的优值)分别存储联盟 S 的最优联盟结构和最优联盟结构值.EOCS 采用自底向上方式由以下公式求长度为 2,3,4,5 的联盟 S 的优值 $F[S]$和优结构 $C[S]$:

$$F[S]=\mathrm{Max}\left(_{S'\subset S\wedge 1\leqslant|S'|\leqslant\frac{1}{2}\lfloor m\rfloor\wedge S\propto'\langle S-S',S'\rangle}F[S-S']+F[S'],V(S)\right),$$

其中,$m=|S|$.最优联盟结构值保存在 $F[A]$中,CS^*以自顶向下方式由 C 表构造.

(1) 当 $m=1$ 时,表 F 的值就是对应的联盟值,表 C 的值就是对应的联盟,因此,

$$F[\{1\}]=V(\{1\})=50,$$

$$C[\{1\}]=\{1\},\ldots$$

(2) 当 $m=2$ 时,根据定理 2(1)知,长度为 2 的联盟都需进行二部分解的评估,即

$$F[\{1,2\}]=\mathrm{Max}\begin{cases}V(\{1,2\})\\F[\{1\}]+F[\{2\}]\end{cases}=\mathrm{Max}\begin{cases}70\\50+40\end{cases}=\mathrm{Max}\begin{cases}70\\90\end{cases}=90,$$

$$C[\{1,2\}]=\{1\},\{2\},\ldots$$

(3) 当 $m=3$ 时,根据定理 2(1)知,长度为 3 的联盟都需进行二部分解的评估,即

$$F[\{1,2,3\}]=\mathrm{Max}\begin{cases}V(\{1,2,3\})\\F[\{1\}]+F[\{2,3\}]\\F[\{2\}]+F[\{1,3\}]\\F[\{3\}]+F[\{1,2\}]\end{cases}=\mathrm{Max}\begin{cases}160\\50+90\\40+90\\30+90\end{cases}=\mathrm{Max}\begin{cases}160\\140\\130\\120\end{cases}=160,$$

$$C[\{1,2,3\}]=\{1,2,3\},\ldots$$

(4) 当 $m=4$ 时,由定理 2(2)可知,长度为 4 的联盟 S 都不进行二部分解的评估,因此 $F[S]=V(S)$,$C[S]=S$,即

$$F[\{1,2,3,4\}]=V(\{1,2,3,4\})=180,C[\{1,2,3,4\}]=\{1,2,3,4\},\ldots$$

(5) 当 $m=5$ 时,即 $S=A$,由定理 2(4)可知,A 要进行所有二部分解的评估,即

$$F[\{1,2,3,4,5\}]=\mathrm{Max}\begin{cases}V(\{1,2,3,4,5\}),F[\{1\}]+F[\{2,3,4,5\}],F[\{2\}]+F[\{1,3,4,5\}],F[\{3\}]+F[\{1,2,4,5\}]\\F[\{4\}]+F[\{1,2,3,5\}],F[\{5\}]+F[\{1,2,3,4\}],F[\{1,2\}]+F[\{3,4,5\}],F[\{1,3\}]+F[\{2,4,5\}]\\F[\{1,4\}]+F[\{2,3,5\}],F[\{1,5\}]+F[\{2,3,4\}],F[\{2,3\}]+F[\{1,4,5\}],F[\{2,4\}]+F[\{1,3,5\}]\\F[\{2,5\}]+F[\{1,3,4\}],F[\{3,4\}]+F[\{1,2,5\}],F[\{3,5\}]+F[\{1,2,4\}],F[\{4,5\}]+F[\{1,2,3\}]\end{cases}$$

$$=280,$$

$$C[\{1,2,3,4,5\}]=\{1,2\},\{3,4,5\}.$$

所以,最优联盟结构值 $F[A]=280$,采用自顶向下方式构造[3,4]出的最优联盟结构

$$CS^*=\{\{1\},\{2\},\{3,4,5\}\},$$

$$V(CS^*)=V(\{1\})+V(\{2\})+V(\{3,4,5\})=50+40+190=280=F[A].$$

DP 和 EOCS 都求出 $V(CS^*)=F[A]=280$,$CS^*=\{\{1\},\{2\},\{3,4,5\}\}$.但 DP 搜索次数是 90(10+3×10+5×7+15),而 EOCS 的搜索次数是 55(10+3×10+15),减少了 35 次长度为 4 的联盟的二部分解.随着 n 的增加,EOCS 无需评估的次数更多.

2.2 EOCS算法描述

完整的 EOCS 算法如下:

Input:所有的联盟值 $V(S)$.

Output:最优联盟结构 CS^*.

Step 1. //求联盟长度为 1,$\lfloor 2n/3\rfloor+1,n-1\rceil$间的所有联盟 S 的 $C[S]$和 $F[S]$值

Construct all $S\subset A$, where $|S|=1$ **or** $|S|\in\lfloor 2n/3\rfloor+1,n-1\rceil$ **do**

Set $C[S]:=S$, $F[S]:=V(S)$

Step 2. //求长度 $m\in[2,\lceil n/2\rceil]$的所有联盟 S 的优值 $F[S]$,S 的每个二部分解都要评估

For $m:=2$ to $\lceil n/2\rceil$ **do**:

Construct all $S\subset A$ such that $|S|=m$ **do**:

2.1. $F(S):=\mathrm{Max}\left(\underset{S'\subset S\wedge 1\leqslant|S'|\leqslant\frac{1}{2}\times m}{}F[S-S']+F[S']\right)$ //在 S 的所有二部分解中找最大值

2.2. **If** $F[S]\geqslant V[S]$ **then** Set $C[S]:=S^*$ where S^* maximizes the right side of equation in Step 2.1.

2.3. **If** $F[S]<V[S]$ **then** Set $C[S]:=S$, $F[S]:=V(S)$

EndFor

Step 3. //求长度 $m\in[\lceil n/2\rceil+1,\lfloor 2n/3\rfloor]$的联盟 S 的优值 $F[S]$,S 的部分二部分解需要评估

For $m:=\lceil n/2\rceil+1$ to $\lfloor 2n/3\rfloor$ **do**:

Construct all $S\subset A$ such that $|S|=m$ **do**:

3.1. $F(S):=\mathrm{Max}\left(\underset{S'\subset S\wedge 1\leqslant|S'|\leqslant\left\lfloor\frac{1}{2}\times m\right\rfloor\wedge S\propto'\langle S-S',S'\rangle}{}F[S-S']+F[S']\right)$ //在 S 的有效二部分解中找最大值

3.2. **If** $F[S]\geqslant V[S]$ **then** Set $C[S]:=S^*$ where S^* maximizes the right side of equation in Step 3.1.

3.3. **If** $F[S]<V[S]$ **then** Set $C[S]:=S$, $F[S]:=V(S)$

EndFor

Step 4. //求 A 的优值 $F[A]$,需要评估 A 的所有二部分解

4.1. $F(A):=\mathrm{Max}\left(\underset{S'\subset A\wedge 1\leqslant|S'|\leqslant\left\lfloor\frac{1}{2}\times n\right\rfloor}{}F[A-S']+F[S']\right)$ //在 A 的所有二部分解中找最大值

4.2. **If** $F[A]\geqslant V[A]$ **then** Set $C[A]:=S^*$ where S^* maximizes the right side of equation in Step 4.1.

4.3. **If** $F[A]<V[A]$ **then** Set $C[A]:=A$, $F[A]:=V(A)$

Step 5. //自顶向下基于 C 表构造 A 的最优联盟结构 CS^*

Set $CS^*:=\{A\}$

While ($S\in CS^*$) **do**:

If $C[S]\neq S$ **then** Set $CS^*:=CS^*-\{S\}\cup\{C[S]\}$

EndWhile

Return CS^*

在 EOCS 算法中,

Step 1. 设置长度为 1,长度为$\lfloor 2n/3\rfloor+1$~$n-1$ 间联盟 S 的 C 表和 F 表的初始值.

Step 2. 评估长度在 2 与$\lceil n/2\rceil$间的联盟 S 的所有二部分解后得出 $C[S]$和 $F[S]$.

Step 3. 评估长度在$\lceil n/2\rceil+1$ 和$\lfloor 2n/3\rfloor$间的联盟 S 的部分二部分解(有效二部分解)后得出 $C[S]$和 $F[S]$.

Step 4. 评估 A 的所有二部分解后得出 $C[A]$和 $F[A]$,$F[A]$存的是 $V(CS^*)$.

Step 5. 以自顶向下方式基于 C 表构造 A 的最优联盟结构 CS^*.

2.3 EOCS算法的正确性证明

定义 6. 对于任意 $CS_1\in L_i$,$CS_2\in L_{i+1}$,从 L_i 到 L_{i+1} 上的二元关系$\propto''$为

$$\propto'':=\{\langle CS_1,CS_2\rangle|\exists!S_1\exists S_2\exists S_3(S_1\in CS_1,S_2,S_3\in CS_2\wedge(S_1\propto'\langle S_2,S_3\rangle))\},$$

其中 $i\in\{1,2,\ldots,n-1\}$.

(1) 若 CS_1 中存在唯一的联盟 S_1 能够有效分解成 CS_2 中的 S_2 和 S_3,即$\langle S_2,S_3\rangle$是 S_1 的有效二部分解,则 CS_1 和 CS_2 具有联盟结构有效分解关系$\propto''$,记作 $CS_1\propto''CS_2$.

(2) 若 $CS_1\propto''CS_2$,则称 CS_1 为 CS_2 的一个父亲,CS_2 为 CS_1 的一个儿子,即 $CS_1\in Par(CS_2)$,$CS_2\in Son(CS_1)$.

(3) $Son^0(CS)=\{CS\}$,$Son^1(CS)=Son(CS)=\{CS'|CS\propto''CS'\}$,$Son^2(CS)=\{CS'|\exists CS_1\wedge(CS_1\in Son(CS)\wedge CS_1\propto''CS')\}$,…,$Son^{n-1}(CS)=\{CS'|\exists CS_1\wedge(CS_1\in Son^{n-2}(CS)\wedge CS_1\propto''CS')\}$.其中,$Son^0(CS)$表示 CS 本身,$Son^1(CS)$表示 CS 的儿子,CS 可以有多个儿子,$Son^2(CS)$表示 CS 的儿子的儿子(孙子).

(4) 函数 Son 的克林闭包 $Son^*=\bigcup_{i=0}^{n-1}Son^i=Son^0\cup Son^1\cup Son^2\cup\ldots\cup Son^{n-1}$,$Son^*(CS)$表示包含 CS 及 CS 所有后代的集合.

引理 1. 在 A 的联盟结构图中,

(1) 任何一个顶点 CS 不一定都有儿子;

(2) 任何一个顶点 $CS(CS\neq\{A\})$都有父亲.

从图 1 可看出,顶点{{1},{2,3,4}}没有儿子.联盟结构图中任意一个顶点 CS,向下都有一条实线边与其相连,与该实线边关联的顶点就是 CS 的父亲.

引理 2. 在联盟结构有效分解关系$\propto''$下,

(1) $Son^i(\{A\})=L_{i+1}(i=0,1,2,...,n-1)$;

(2) $Son^*(\{A\})=L$.

引理 2(1)表明,A 的儿子是 A 的联盟结构图中 L_2 层中的所有顶点,A 的孙子是 A 的联盟结构图中 L_3 层中的所有顶点,…引理 2(2)表明,A 的克林闭包是 A 的联盟结构图的所有顶点.

引理 3. EOCS 算法在求最优联盟过程中,若访问了某个顶点 CS,则:

(1) 访问了其所有儿子顶点 $Son(CS)$(若儿子顶点存在);

(2) 访问了其所有后代顶点 $Son^0(CS)\cup Son^1(CS)\cup\ldots$

搜索联盟结构图中某个顶点 CS,本质上是搜索在$\propto''$关系下形成的以 CS 为根顶点的子图,故搜索了其所有后代.

定理 3. EOCS 算法结束时,A 的联盟结构图中所有顶点都被遍历,此时,

$$F(A)=\mathrm{Max}_{CS\in Son^*(\{A\})}V(CS)=\mathrm{Max}_{CS\in L}V(CS).$$

证明:根据引理 3(2)知,若 EOCS 算法访问了 CS,则访问了 CS 的所有后代 $Son^0(CS)\cup Son^1(CS)\cup\ldots$,而 EOCS 算法在 Step 4 求 A 的优值 $F[A]$时访问了顶点$\{A\}$,因此 EOCS 算法访问了$\{A\}$的所有后代 $Son^0(\{A\})\cup Son^1(\{A\})\cup Son^2(\{A\})\cup\ldots\cup Son^{n-1}(\{A\})$,而 $Son^0(\{A\})\cup Son^1(\{A\})Son^2(\{A\})\cup\ldots\cup Son^{n-1}(\{A\})=\bigcup_{i=0}^{n-1}Son^i(\{A\})=Son^*(\{A\})$,即 EOCS 访问了 $Son^*(\{A\})$.由引理 2(2)可知,$Son^*(\{A\})=L$,因此 EOCS 算法结束时访问了 L,即 A 的联盟结构图所有顶点都被遍历.此时,$F(A)=\mathrm{Max}_{CS\in Son^*(\{A\})}V(CS)=\mathrm{Max}_{CS\in L}V(CS)$. □

2.4 EOCS的有效分解数目

DP 和 EOCS 算法依次构造包含 m 个 Agent 的联盟 S 求 $C[S]$及 $F[S]$.从 n 个元素中取 m 个元素的取法为 C_n^m,m 个 Agent 有 $2^{m-1}-1$ 个二部分解.DP 算法评估所有二部分解,因此 DP 需进行的二部分解次数:

$$DP(n)=\sum_{m=2}^{n}C_n^m\times(2^{m-1}-1)=\sum_{m=1}^{n}C_n^m\times(2^{m-1}-1).$$

而 EOCS 算法只评估有效分解,它的评估次数计算如下:

若 $N^{m',m''}$表示将大小为 $m=m'+m''$的联盟二部分解成大小为 m'和 m''的二部分解个数,则

$$N^{m',m''}=\begin{cases}C_m^{m'}/2, & \text{if } m'=m''\\ C_m^{m'}, & \text{otherwise}\end{cases}.$$

EOCS 算法在 Step 2 中进行的有效分解次数为

$$St2(n)=\sum_{m=2}^{\lceil n/2\rceil}C_n^m\times(2^{m-1}-1)=\sum_{m=1}^{\lceil n/2\rceil}C_n^m\times(2^{m-1}-1),$$

在 Step 3 中进行的有效分解次数为

$$St3(n)=\sum_{m=\lceil n/2\rceil+1}^{\lfloor 2n/3\rfloor}C_n^m\times\sum_{m''=\lceil m/2\rceil\wedge(m''\leqslant n-m)}^{m-1}N^{m-m'',m''},$$

在 Step 4 中进行的有效分解次数为

$$St4(n)=2^{n-1}-1,$$

因此,EOCS 算法进行的二部分解次数为

$$EOCS(n)=St2(n)+St3(n)+St4(n)=\left[\sum_{m=1}^{\lceil n/2\rceil}C_n^m\times(2^{m-1}-1)\right]+\left[\sum_{m=\lceil n/2\rceil+1}^{\lfloor 2n/3\rfloor}C_n^m\times\sum_{m''=\lceil m/2\rceil\wedge(m''\leqslant n-m)}^{m-1}N^{m-m'',m''}\right]+[2^{n-1}-1].$$

定理 4. 对于含有 n 个 Agent 的系统:

(1) DP 算法进行的二部分解次数

$$DP(n)=\sum_{m=1}^{n}C_n^m\times(2^{m-1}-1).$$

(2) EOCS 算法进行的有效分解次数

$$EOCS(n)=\left[\sum_{m=1}^{\lceil n/2\rceil}C_n^m\times(2^{m-1}-1)\right]+\left[\sum_{m=\lceil n/2\rceil+1}^{\lfloor 2n/3\rfloor}C_n^m\times\sum_{m''=\lceil m/2\rceil\wedge(m''\leqslant n-m)}^{m-1}N^{m-m'',m''}\right]+[2^{n-1}-1].$$

(3) EOCS 算法与 DP 算法进行的分解次数比

$$\frac{EOCS(n)}{DP(n)}=\frac{\left[\sum_{m=1}^{\lceil n/2\rceil}C_n^m\times(2^{m-1}-1)\right]+\left[\sum_{m=\lceil n/2\rceil+1}^{\lfloor 2n/3\rfloor}C_n^m\times\sum_{m''=\lceil m/2\rceil\wedge(m''\leqslant n-m)}^{m-1}N^{m-m'',m''}\right]+[2^{n-1}-1]}{\sum_{m=1}^{n}C_n^m\times(2^{m-1}-1)}.$$

3 EOCS 算法的时间复杂度分析

3.1 二项分布与积分极限定理

定义 7. 设 P 是事件 Event 在每次实验中出现的概率,则 n 次伯努利实验中事件 Event 出现 m 次的概率 $B(m,n,p)=C_n^m p^m\times(1-p)^m$ 为二项分布[15](binomial distribution),其中,$m=1,2,...,n$.

定理 5(积分极限定理)[15]. 设 u_n 是 n 次伯努利实验中事件 Event 出现的次数,而 P 是事件 Event 在每次实验中出现的概率,则有

$$P\left\{a\leqslant\frac{u_n-np}{\sqrt{npq}}<b\right\}=\phi(b)-\phi(a)\approx\phi\left(\frac{b-np+0.5}{\sqrt{npq}}\right)-\phi\left(\frac{a-np-0.5}{\sqrt{npq}}\right),$$

其中, $\phi(x)=\frac{1}{\sqrt{2\pi}}\int_0^x e^{-t^2/2}dt$ 为正态分布函数,$q=1-p$.

3.2 EOCS时间复杂度下界的证明

由定理 4(1)和定理 4(2)可知,

$$\begin{aligned}DP(n)&=\sum_{m=1}^{n}C_n^m\times(2^{m-1}-1)=\sum_{m=1}^{n}[(C_n^m\times 2^{m-1})-C_n^m]=\frac{1}{2}\sum_{m=1}^{n}(C_n^m\times 2^m)-\sum_{m=1}^{n}C_n^m\\&=\frac{1}{2}\sum_{m=0}^{n}(C_n^m\times 2^m)-\sum_{m=0}^{n}C_n^m=\frac{1}{2}\sum_{m=0}^{n}(C_n^m\times 2^m\times 1^{(n-m)})-\sum_{m=0}^{n}C_n^m\\&=\frac{1}{2}(2+1)^n-2^n=\frac{1}{2}3^n-2^n\approx\frac{1}{2}3^n,\end{aligned}$$

$$\begin{aligned}EOCS(n)&=\left[\sum_{m=1}^{\lceil n/2\rceil}C_n^m\times(2^{m-1}-1)\right]+\left[\sum_{m=\lceil n/2\rceil+1}^{\lfloor 2n/3\rfloor}C_n^m\times\sum_{m''=\lceil m/2\rceil\wedge(m''\leqslant n-m)}^{m-1}N^{m-m'',m''}\right]+[2^{n-1}-1]\\&\geqslant\left[\sum_{m=1}^{k}C_n^m\times(2^{m-1}-1)\right]+[2^{n-1}-1]=\left[\frac{1}{2}\sum_{m=0}^{k}C_n^m\times 2^m\right]-\sum_{m=1}^{k}C_n^m+[2^{n-1}-1]\\&=\frac{1}{2}\sum_{m=0}^{k}(C_n^m\times 2^m\times 1^{n-m})+\sum_{m=k+1}^{n-1}C_n^m-1\approx\frac{1}{2}\sum_{m=0}^{k}(C_n^m\times 2^m\times 1^{n-m}).\end{aligned}$$

其中,$k=\lceil n/2\rceil$.

用 $LB(EOCS(n))=\frac{1}{2}\sum_{m=0}^{k}(C_n^m\times 2^m\times 1^{n-m})$ 表示 $EOCS(n)$下界,则可用 $\lim_{n\to\infty}\frac{LB(EOCS(n))}{DP(n)}$ 来证明 EOCS(n)的下界.

因为 $\frac{LB(EOCS(n))}{DP(n)}=\frac{\frac{1}{2}\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{\frac{1}{2}3^n}=\frac{\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{3^n}=\frac{\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{3^m\times 3^{n-m}}=\sum_{m=0}^{k}C_n^m\times\left(\frac{2}{3}\right)^m\times\left(\frac{1}{3}\right)^{n-m}$, 由于 $C_n^m\left(\frac{2}{3}\right)^m\times\left(\frac{1}{3}\right)^{n-m}$ 是定义 7 中的二项分布 $b(m,n,p)$,其中,m 为第 m 次搜索,p=2/3,n 为 Agent 个数,因此,$\sum_{m=0}^{k}C_n^m\times\left(\frac{2}{3}\right)^m\times\left(\frac{1}{3}\right)^{n-m}$ 就是二项分布 $B(m,n,p)$的积分.

因此,$EOCS(n)$下界可用定理 5 来逼近,其方法如下:

$$\begin{aligned}\frac{LB(EOCS(n))}{DP(n)}&=\sum_{m=0}^{k}C_n^m\times\left(\frac{2}{3}\right)^m\times\left(\frac{1}{3}\right)^{n-m}=\phi\left(\frac{b-np+0.5}{\sqrt{npq}}\right)-\phi\left(\frac{a-np-0.5}{\sqrt{npq}}\right)\\&=\phi\left(\frac{\frac{n}{2}-n\times\frac{2}{3}+0.5}{\sqrt{n\times\frac{2}{3}(1-\frac{2}{3})}}\right)-\phi\left(\frac{0-n\times\frac{2}{3}-0.5}{\sqrt{n\times\frac{2}{3}(1-\frac{2}{3})}}\right)=\phi\left(\frac{\frac{2}{3}\times n+0.5}{\sqrt{n\times\frac{2}{9}}}\right)-\phi\left(\frac{\frac{1}{6}\times n-0.5}{\sqrt{n\times\frac{2}{9}}}\right)\\&=\phi\left(\sqrt{2n}+\frac{3\sqrt{2}}{4\sqrt{n}}\right)-\phi\left(\frac{\sqrt{2n}}{4}-\frac{3\sqrt{2}}{4\sqrt{n}}\right)\end{aligned}\tag{1}$$

其中,a=0,b=k=⌈n/2⌉.

令 EOCS 下界为 g^n,其中,2<g≤3,只要 n 趋于无穷大时 $\frac{LB(EOCS(n))}{g^n}$ 的极限为 1,则可用 $\Omega(g^n)$作为 EOCS 算法的时间复杂度下界.

因为 $\lim_{n\to\infty}\frac{\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{g^n}=\lim_{n\to\infty}\frac{\frac{\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{3^n}}{\frac{g^n}{3^n}}=\lim_{n\to\infty}\frac{\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{3^m\times 3^{n-m}}{(g/3)^n}=\lim_{n\to\infty}\frac{\sum_{m=0}^{k}C_n^m\times\left(\frac{2}{3}\right)^m\times\left(\frac{1}{3}\right)^{n-m}}{(g/3)^n}$,

将公式(1)代入,

$$\begin{aligned}\lim_{n\to\infty}\frac{\sum_{m=0}^{k}C_n^m\times 2^m\times 1^{n-m}}{g^n}&=\lim_{n\to\infty}\frac{\phi\left(\sqrt{2n}+\frac{3\sqrt{2}}{4\sqrt{n}}\right)-\phi\left(\frac{\sqrt{2n}}{4}-\frac{3\sqrt{2}}{4\sqrt{n}}\right)}{(g/3)^n}\\&=\lim_{n\to\infty}\frac{\left[\phi\left(\sqrt{2n}+\frac{3\sqrt{2}}{4\sqrt{n}}\right)-\phi\left(\frac{\sqrt{2n}}{4}-\frac{3\sqrt{2}}{4\sqrt{n}}\right)\right]'}{[(g/3)^n]'}\\&=-\lambda\lim_{n\to\infty}\left(\frac{3}{g\times e^{\frac{1}{16}}}\right)^n=1,\end{aligned}$$

其中, $\lambda=\frac{\frac{1}{\sqrt{2\pi}}}{\frac{g}{3}}=\frac{3}{g\times\sqrt{2\pi}}$.

所以, $\frac{3}{g\times e^{\frac{1}{16}}}=1$,即 $g=\frac{3}{\sqrt[16]{e}}=\frac{3}{\sqrt[16]{2.71828183}}=2.81824$.取 g=2.818,则 EOCS 的下界是$\Omega(2.818^n)$.

定理 6(复杂度下界定理). EOCS 算法的时间复杂度下界是$\Omega(2.818^n)$.

3.3 时间序列分析

当 Agent 个数在 13~21 时,根据定理 4 可求出 DP 和 EOCS 算法所进行的有效二部分解次数及比值(见表 2).为了利用时间序列分析方法[16],我们将 Agent 个数用时间 t 表示,实验数据个数用 n 表示.这样,对一组变量 $y(t)=EOCS(t)/DP(t)$进行求解($t_1<t_2<\ldots<t_n$),得到离散数字序列集合 $y(t_1),y(t_2),y(t_3),\ldots,y(t_n)$,我们称其为时间序列.时间序列分析是根据表 2 所示的时间序列数据,通过曲线拟合和参数估计来建立时间复杂度的数学模型,可以采用曲线拟合中非线性最小二乘法的参数估计方法求 EOCS 复杂度的上界.

Table 2 Time serial table of DP and EOCS algorithm

表 2 DP and EOCS 算法的搜索时间序列表

Number of agent	13	14	15	16	17	18	19	20	21
m (searching number by DP)	7.89×10^5	2.37×10^6	7.14×10^6	2.15×10^7	6.44×10^7	1.93×10^8	5.81×10^8	1.74×10^9	5.23×10^9
n (searching number by EOCS)	3.07×10^5	9.65×10^5	2.88×10^6	8.22×10^6	2.57×10^7	7.65×10^7	2.20×10^8	6.83×10^8	2.04×10^9
m/n (%)	39.0	40.6	40.3	38.3	39.8	39.5	37.9	39.2	39.0

3.4 EOCS时间复杂度上界的求取

DP 的时间复杂度为 $O(3^n)$[3,4],而 EOCS 的搜索次数低于 DP,故 EOCS 的复杂度不超过 $O(3^n)$.可设 EOCS 的上界为 kg^n,其中,$2<g\leqslant3$,现在利用时间序列分析方法求 k 与 g 的值.

当 Agent 的个数为 t 时,设 EOCS 算法的搜索次数 $y'=k'g^t$,而 DP 算法的搜索次数为 $y''=k''3^t$,其中,k'和 k''为特定常数.

$$y=\frac{y'}{y''}=\frac{k'\times g^t}{k''\times3^t}=\left(\frac{k'}{k''}\right)\times\left(\frac{g}{3}\right)^t=k\times x^t \tag{2}$$

其中, $k=\frac{k'}{k''}$, $x=\frac{g}{3}$.

对 $y=k\times x^t$ 两边取对数得 $\mathrm{Log}_{10}(y)=\mathrm{Log}_{10}(k)+t\times\mathrm{Log}_{10}(x)$.根据最小二乘法原理,按照直线形式的常数确定方法,得到求解 $\mathrm{Log}_{10}(k)$和 $\mathrm{Log}_{10}(x)$的标准方程为

$$\begin{cases}\sum\mathrm{Log}_{10}(y)=n\times\mathrm{Log}_{10}k+\mathrm{Log}_{10}(x)\sum t\\ \sum t\times\mathrm{Log}_{10}(y)=\mathrm{Log}_{10}k\times\sum t+\mathrm{Log}_{10}(x)\sum t^2\end{cases} \tag{3}$$

方程组(3)所需的辅助数据见表 3,其中,第 2 列数据为表 2 最后一行数据.

Table 3 Data of least squares method

表 3 最小二乘法的数据

t	Y (ratio of searching number)	log$y$	$t\times$log$y$	t^2
13	3.90E−01	−4.09E−01	−5.32E+00	1.69E+02
14	4.06E−01	−3.91E−01	−5.48E+00	1.96E+02
15	4.03E−01	−3.95E−01	−5.92E+00	2.25E+02
16	3.83E−01	−4.17E−01	−6.67E+00	2.56E+02
17	3.98E−01	−4.00E−01	−6.80E+00	2.89E+02
18	3.95E−01	−4.03E−01	−7.26E+00	3.24E+02
19	3.79E−01	−4.21E−01	−8.01E+00	3.61E+02
20	3.92E−01	−4.07E−01	−8.13E+00	4.00E+02
21	3.90E−01	−4.09E−01	−8.59E+00	4.41E+02
∑: 153	3.54E+00	−3.65E+00	−6.22E+01	2.66E+03

将表 3 数据代入方程组(3)得 $\begin{cases}9\times \mathrm{Log}_{10}k+153\times \mathrm{Log}_{10}x=-3.65\\ 153\times \mathrm{Log}_{10}k+2660\times \mathrm{Log}_{10}x=-62.2\end{cases}$,解之得 $\begin{cases}\mathrm{Log}_{10}k=-192.4/531\\ \mathrm{Log}_{10}x=-1.35/531\end{cases}$.

因此,$x=10^{(-135/531)}=0.99417$,根据等式(2)得 $g=3\times x=2.983$,故 EOCS 算法复杂度的上界为 $O(2.983^n)$.

定理 7(复杂度上界近似定理). EOCS 算法的时间复杂度上界是 $O(2.983^n)$.

4 EOCS 算法与其他算法的对比

可从与联盟值的概率分布相关性、是否任一时间算法、找最优联盟结构的时间复杂度、搜索方式来判断联盟结构生成算法的优劣.表 4 给出了 EOCS 与其他算法在这 4 个方面的对比.

Table 4 Comparison of EOCS algorithm and other algorithms for optimal coalition structure

表 4 EOCS 与其他最优联盟结构算法的对比

Algorithm	Related with probability distribution of coalition value?	Is anytime algorithm?	Time complexity of finding optimal coalition structure	Searching method
Rothkopf's DP	No	No	$O(3^n)$	Bottom-Top
Rahwan's IDP	No	No	$O(3^n)$	Bottom-Top
Sandholm's anytime algorithm	No	Yes	$O(n^n)$	Horizontal
Jennings's anytime algorithm	No	Yes	$O(n^n)$	Vertical
Zhang XL's SCS algorithm	Yes	Yes	$O(n^n)$	Bottom-Top
EOCS	**No**	**No**	$\boldsymbol{O(2.983^n)}$	**Bottom-Top Local-Global**

从表 4 中可看出,EOCS 的优点是,该算法与联盟值的概率分布无关,即无论联盟值满足何种概率分布,EOCS 算法都能在确定的 $O(2.983^n)$时间内求出最优联盟结构.这些优点的产生都得益于 EOCS 算法的搜索方法,EOCS 算法不仅自底向上,而且充分利用了 A 的子联盟的联盟结构图与全局 A 的联盟结构图的关系,将一些通向最优值的重复路径裁剪掉.这些搜索方法都是联盟结构图的一些新的性质,这将在未来的工作和报告中再作进一步说明.

5 结论和未来的工作

本文基于联盟的有效分解关系设计了一种有效的联盟结构生成算法——EOCS 算法,而且基于函数的克林闭包证明了 EOCS 算法的正确性.最后,借助于积分极限定理证明了 EOCS 的下界是 $\Omega(2.818^n)$,利用时间序列分析方法求出了 EOCS 的上界是 $O(2.983^n)$.

未来的工作包括:(1) 进一步探讨基于正整数拆分的联盟结构搜索的代数性质,重点考虑整数的 k 部拆分与联盟 k 部分解的数量关系、联盟值的概率分布与联盟结构值的概率分布关系;(2) 将 EOCS 算法的思想应用到 Anytime 算法上,设计一种能够在低于 $O(n^n)$时间内找出最优值的 Anytime 算法,并分析联盟结构图的图性质及其上搜索方法的特性.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是审稿专家对本文图 1 所提出的建议表示衷心的感谢.

References:

[1] Rahwan T, Jennings NR. An algorithm for distributing coalitional value calculations among cooperating agents. Artificial Intelligence, 2007,171(8-9):535−567. [doi: 10.1016/j.artint.2007.03.002]

[2] Sandholm TW, Larson K, Andersson M. Coalition structure generation with worst case guarantees, Artificial Intelligence, 1999, 111(112):209−238. [doi: 10.1016/S0004-3702(99)00036-3]

[3] Rothkopf MH, Pekec A, Harstad RM. Computationally manageable combinatorial auctions. Management Science, 1995,44(8): 1131−1147. [doi: 10.1287/mnsc.44.8.1131]

[4] Liu JL, Tong XR, Zhang W. A kind of method for quick constructing optimal coalition structure. Computer Engineering & Application, 2006,42(4):35−44 (in Chinese with English abstract).

[5] Zhang XL, Shi CY. A dynamic formation algorithm of multi-agent coalition structure. Journal of Software, 2007,18(3):574−581 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/574.htm [doi: 10.1360/jos180574]

[6] Dang VD, Jennings NR. Generating coalition structures with finite bound from the optimal guarantees. In: Proc. of the 3rd Int'l Joint Conf. on Autonomous Agents and Multi-Agent Systems. Berlin: Springer-Verlag, 2004. 564−571. [doi: 10.1109/AAMAS.2004.132]

[7] Shehory O, Kraus S. Methods for task allocation via Agent coalition formation. Artificial Intelligence, 1998,101(1-2):165−200. [doi: 10.1016/S0004-3702(98)00011-3]

[8] Rahwan T, Jennings NR. An improved dynamic programming algorithm for coalition structure generation. In: Proc. of the 7th Int'l Conf. on Autonomous Agents and Multi-Agent Systems. Berlin: Springer-Verlag, 2008. 1417−1420.

[9] Rahwan T, Ramchurn SD, Jennings NR. An anytime algorithm for optimal coalition structure generation. Journal of Artificial Intelligence Research, 2009,34(1-4):521−567.

[10] Rahwan T, Ramchurn SD, Dang VD, Jennings NR. Near-Optimal anytime coalition structure generation. In: Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 2007. 2365−2371.

[11] Rahwan T, Ramchurn SD, Giovannucci A, Dang VD, Jennings NR. Anytime optimal coalition structure generation. In: Proc. of the 22nd Conf. on Artificial Intelligence. San Jose: AAAI Press, 2007. 1184−1190.

[12] Liu JL, Zhang W, Wang LL. Algebra property and application of coalition structure graph. Pattern Recognition and Artificial Intelligence, 2009,22(6):841−847 (in Chinese with English abstract).

[13] Michalak T, Rahwan T. On representing coalitional games with externalities. In: Proc. of the 10th ACM Conf. on Electronic Commerce. New York :ACM Press, 2009. 11−20. [doi: 10.1145/1566374.1566377]

[14] Rahwan T, Michalak T. Coalition structure generation in multi-agent systems with positive and negative externalities. In: Proc. of the 21st Int'l Joint Conf. on Artificial Intelligence. Pasadena, 2009. 257−263.

[15] Li XP. Foundations of Probability Theory. 2nd ed., Beijing: Higher Education Press, 2004. 262−266 (in Chinese).

[16] He SY. Applied Time Series Analysis. Beijing: Peking University Press, 2004. 140−143 (in Chinese).

附中文参考文献:

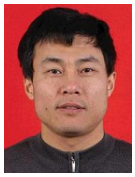
[4] 刘惊雷,童向荣,张伟.一种快速构建最优联盟结构的方法.计算机工程与应用,2006,42(4):35−44.

[5] 张新良,石纯一.多 Agent 联盟结构动态生成算法.软件学报,2007,18(3):574−581. http://www.jos.org.cn/1000-9825/18/574.htm [doi: 10.1360/jos180574]

[12] 刘惊雷,张伟,王玲玲.联盟结构图的代数性质及应用.模式识别与人工智能,2009,22(6):841−847.

[15] 李贤平.概率论基础.第 2 版,北京:高等教育出版社,2004.262−266.

[16] 何书元.应用时间序列分析.北京:北京大学出版社,2004.140−143.



刘惊雷(1970－),男,山西临猗人,副教授,CCF 会员,主要研究领域为程序理论,多 Agent 系统.

童向荣(1975－),男,博士,副教授,CCF 会员,主要研究领域为多 Agent 协商.

张伟(1961－),男,博士,教授,CCF 高级会员,主要研究领域为分布式人工智能.

张振荣(1984－),女,硕士生,主要研究领域为多 Agent 联盟.