

基于虚拟联盟的重叠联盟形成算法

张国富^{1,2,3}, 周 鹏², 蒋建国^{1,2}, 苏兆品², 田敬北², 刘 扬²

(1. 合肥工业大学特种显示技术教育部重点实验室, 安徽合肥 230009; 2. 合肥工业大学计算机与信息学院, 安徽合肥 230009;
3. 合肥工业大学信息与通信工程博士后科研流动站, 安徽合肥 230009)

摘 要: 重叠联盟形成问题是复杂智能系统中极具挑战性的前沿课题. 特别是当一个能力有限的 agent 同时参与了多个不同的任务, 但又不能同时满足这多个任务的需求时, 就会产生资源冲突. 为此, 本文重点研究如何把一个无效的二维二进制编码修正为一个合法的编码, 提出将有效联盟的剩余能力转移给一个动态的虚拟联盟, 由虚拟联盟帮助解决其他无效联盟. 实验结果表明, 本文算法不会丢弃任何无效编码, 在解决激烈的资源冲突时显得更加灵活有效.

关键词: 复杂智能系统; 多 agent 系统; 重叠联盟; 二维二进制编码; 虚拟联盟

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2012) 01-0121-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.01.020

An Algorithm for Overlapping Coalition Formation Based on Virtual Coalition

ZHANG Guo-fu^{1,2,3}, ZHOU Peng², JIANG Jian-guo^{1,2}, SU Zhao-pin², TIAN Jing-bei², LIU Yang²

(1. Key Laboratory of Special Display Technology, Hefei University of Technology, Ministry of Education, Hefei, Anhui 230009, China;

2. School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China;

3. Information and Communication Engineering Postdoctoral Research Station, Hefei University of Technology, Hefei, Anhui 230009, China)

Abstract: Overlapping coalition formation is a very challenging topic in complex intelligent systems. When an agent with insufficient resources may join in several different tasks but can not satisfy those tasks' need at the same time, resource conflicts will take place. To tackle such scenarios, this paper mainly focuses on how to revise an invalid two-dimensional binary encoding into a valid one to resolve conflicts over the usage of joint resources. A dynamic virtual coalition is proposed to inherit the surplus resources of all feasible coalitions to join and help other invalid coalitions. Finally, contrastive experiments are made to show that the proposed algorithm does not discard any invalid encoding and is more effective to resolve possible resource conflicts.

Key words: complex intelligent systems; multi-agent systems; overlapping coalitions; two-dimensional binary encoding; virtual coalition

1 引言

复杂系统作为系统科学中的一个前沿方向, 近年来一直是人们关注的一个热点. Science 在 2009 年 7 月推出了复杂系统与网络专辑, 在这一期里, 代表人文领域应用最新成果的一些研究表明^[1,2], 仿真方法已逐渐成为复杂系统应用研究的一个重要手段. 其中, 基于智能体(agent)的观点^[3]更是提供了一种强有力的工具、技术和途径来提高人们概念化和实现各种复杂系统的能力, 也提供了一种自然和艺术的方式来表述一系列多样化的问题^[2].

特别是随着多 agent 系统(Multi-Agent Systems, MAS)

理论不断发展, 基于 MAS 的复杂智能系统(complex intelligent systems)仿真技术正在蓬勃兴起^[4~7], 以适应计算机支持的协同工作等应用需求, 因而使得对 MAS 中的联盟研究也变得越来越重要, 因为当某个资源有限的 agent 遇到其自身无法完成的任务时, 它不得不与系统中的其它 agent 进行交互和协作, 组成一个团队(team)来共同承担这个任务, 这样的团队即为联盟(coalition).

2 相关工作

联盟形成(coalition formation)是联盟一切活动的基础, 如何形成一个稳定均衡的联盟, 使联盟朝着稳定的

方向发展,是 MAS 理论的热点课题,正被国内外的众多学者所关注和研究,且其应用已涉及到国民经济众多领域,如虚拟企业^[8]、电力传输^[9]、无线网络^[10]、多机器人协作^[11]、资源调度与分配^[12]、灾害应急^[13]等.然而,现有大多数研究都是着眼于非重叠联盟(disjoint coalition)^[14~19],即要求一个 agent 在任意时刻都只能加入一个联盟.

应该看到,由于一些复杂系统本身的计算资源有限,将一个 agent 局限于一个联盟中,很难发挥 agent 参与任务求解的积极性,且往往存在巨大的资源浪费,会造成资源利用率不高和系统的不稳定,而且对于某些资源较多的 agent,完全可以同时参与多个任务的求解以谋求更多的利益.如图 1 所示,假设 agent 代表的是虚拟企业,为了获取丰厚的商业利润,它和其他的合作伙伴组成多种商业联盟承担了多个项目,这时它就需要把它的资源同时分配给不同的项目(对应不同的联盟).同样,这种“重叠”性在电力传输^[9]、无线网络^[10]、资源调度与分配^[12]中也是很自然和普遍的.因此,考虑重叠联盟(overlapping coalition),即允许一个 agent 把自己的资源同时分给多个不同的联盟,力所能及地求解多个不同的任务,这更能符合现实世界的实际情况,更有利于增强单个 agent 求解任务的能力,实现资源重组和优势互补,从而提高整个系统的资源利用率和任务求解效率^[20].

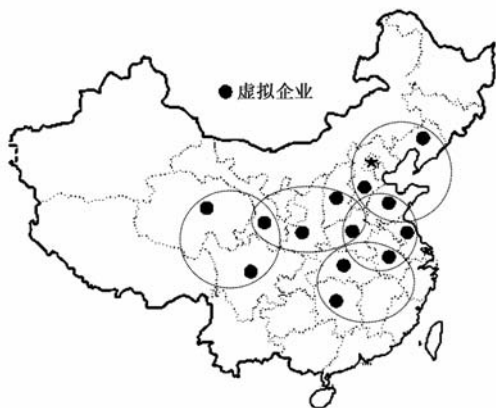


图1 虚拟企业中的重叠商业联盟

然而,自巴伊兰大学的 Shehory 和 Kraus^[20]提出重叠联盟形成(Overlapping Coalition Formation, OCF)问题之后,国内外对 OCF 问题的研究出现了近十年的真空.究其原因,重叠联盟允许一个 agent 可以同时参与多个联盟,即一个联盟的若干成员可能会同时属于其他联盟,这就可能带来潜在的资源冲突问题,因为当多个联盟同时竞争某一资源有限的 agent,而该 agent 又无法同时满足如此多的联盟的请求时,系统就会产生资源冲突和联盟死锁^[20],从而导致系统崩溃.因此,OCF 问题已

成为 MAS 中公认的难点课题,一度令广大学者望而却步.直到 2006 年,南安普敦大学的 Jennings^[21]教授领导的 IAM Group 将重叠联盟成功应用于传感器网络中的多目标跟踪问题,并显示出了比非重叠联盟的优势,OCF 问题才再度引起一些学者的注意和兴趣.

其中,Chalkiadakis 等^[22,23]从对策论中引入“核”的概念来研究 OCF 的建模及其稳定性.但是上述工作都过分理想的假设每个 agent 的资源是有限的,即在重叠联盟中的分配不存在冲突现象,显然这与现实世界的某些实际情况是相悖的.而且,Chalkiadakis 等^[22]在其研究成果中更是悲观的认为:由于重叠联盟要比非重叠联盟情况复杂的多,解空间异常庞大,因此不可能得到 OCF 问题的最优解,甚至近似最优解.

为此,张国富等^[24]基于虚拟 agent 技术改进了 Shehory 和 Kraus 的工作,可以为串行执行的多个任务生成相应的求解联盟,蒋建国等^[25]基于冲突消解策略研究面向并发的多个任务的复杂联盟生成.最近, Lin 和 Hu^[26]在国际 agent 领域最顶级会议 AAMAS 会议上提出将所有有效联盟(即能完成任务的联盟)的剩余资源转移给一个选定的 agent,并由该 agent 继承所有有效联盟的资源并代表这些有效联盟中的成员去帮助其他无效联盟(即不能完成任务的联盟),从而可以避免求解过程中的资源冲突,其研究结论相比已有的确定性算法要进步了很多,给广大学者指明了一条可行的研究思路.但是,其所设计的算法对于编码是否有效判断过于武断,导致大量本来是合法的编码被算法随意丢弃,降低了种群的多样性,直接影响解的优越性,因此其算法(后称 LH 算法)并没有在 AAMAS 会议上引起多少学者的共鸣.

总的来说,相比于非重叠联盟理论的硕果累累^[14~19],目前 OCF 理论刚刚起步,还处于探索阶段,仍有诸多问题需要进一步深入研究.本文在总结前人工作的基础上,重点研究无效编码的修正算法,并从理论上论证和分析算法的合理性与有效性,试图以一种更加有效的方式解决重叠联盟形成问题.

3 OCF 数学模型

设 MAS 中有 n 个 agent, $A = \{a_1, \dots, a_n\}$, 需要合作求解 m 个任务, $T = \{t_1, \dots, t_m\}$. m 个任务有个优先级排序,例如 $t_1 < t_2 < \dots < t_m$, 表示任务之间的相对重要程度,其中 $t_1 < t_2$ 表示任务 t_1 要比 t_2 重要或至少和 t_2 一样重要.

定义 1 (agent) 每个 $a_j \in A$ 都有一个 r 维初始能力向量 $B_j = [b_1^j, \dots, b_r^j]$, $0 \leq b_k^j < \infty$, $j = 1, \dots, n$, $k = 1, \dots, r$, $r \in \mathbb{N}$, 用于表征 a_j 执行 r 种特定动作的能力大小.

定义 2 (task) 每个 $t_i \in T$ 都均有一定的能力需求

$$D_i = [d_1^i, \dots, d_r^i], 0 \leq d_k^i < \infty, i = 1, \dots, m.$$

定义 3 (coalition) 联盟 $C_i \subset A, C_i \neq \emptyset$, 为任务 t_i 对应的求解联盟, 具有能力向量 $B_{C_i} = [b_1^{C_i}, \dots, b_r^{C_i}]$. 联盟 C_i 可以完成任务 t_i 的必要条件是:

$$b_k^{C_i} \geq d_k^i, k = 1, \dots, r \quad (1)$$

两个不同的联盟 C_{i_1}, C_{i_2} 为重叠联盟, 当且仅当 $C_{i_1} \cap C_{i_2} \neq \emptyset$, 即有某些 agent 同时参与了任务 t_{i_1}, t_{i_2} .

定义 4 联盟 C_i 的值用一个特征函数 $\nu(C_i) \geq 0$ 给出^[14~19]:

$$\nu(C_i) = \pi(t_i) - \theta(C_i) - \varepsilon(C_i) \quad (2)$$

其中, $\pi(t_i)$ 为完成任务 t_i 所获得的报酬, 通常为给定的一个常数; $\theta(C_i)$ 指联盟成员总能力折合的成本, 在重叠联盟中 a_j 可能同时属于多个不同的联盟, 而 a_j 不可能对其所在的多个联盟都是贡献其能力向量 B_j , 所以在重叠联盟中 $\theta(C_i)$ 通常为联盟中各成员实际贡献的能力之和, 也就是对应的任务的能力需求, 即 $\theta(C_i) = \sum_{k=1}^r d_k^i$, 这一点与非重叠联盟是不同的; $\varepsilon(C_i)$ 指联盟 C_i 中的各 agent 成员协作求解任务 t_i 过程中的通信开销, 一般 a_{j_1} 与 a_{j_2} 之间的通信成本为 ξ_{j_1, j_2} , 通常为给定的一个常数, 满足 $\xi_{j_1, j_1} = 0, \xi_{j_1, j_2} = \xi_{j_2, j_1}$, 若 $C_i = \{a_{j_1}, a_{j_2}, a_{j_3}\}$, 则 $\varepsilon(C_i) = \xi_{j_1, j_2} + \xi_{j_1, j_3} + \xi_{j_2, j_3}$.

重叠联盟形成问题就是对任务序列 t_1, \dots, t_m , 在式(3)的前提下同时给出 m 个求解联盟 C_1, \dots, C_m , 使得系统总收益 ν_{MAS} 尽可能大, 显然这是一个复杂的组合优化问题.

$$\sum_j B_j \geq \sum_i D_i \quad (3)$$

$$\nu_{\text{MAS}} = \sum_{i=1}^m \nu(C_i) \quad (4)$$

4 二维二进制编码

从直观上看, 重叠联盟形成问题就是如何在 n 个 agent 中选取部分 agent 组成联盟值相对较优的 m 个联盟去求解任务, 这恰似一个二维组合优化问题. 而二维二进制编码不仅方式简单、容易理解, 而且从根本上与本文的重叠联盟形成问题的二维本质相适应, 从而为问题的求解奠定了良好的基础, 并为设计出性能优良的搜索算法提供了极为广阔的空间, 因此本文重点研究这种编码方式.

如图 2 所示, 二维二进制编码为一个 $m \times n$ 的 0-1 二值矩阵. 编码的每一行均对应着一个任务, 而每一列则对应着一个 agent. 设 Δ_{ij} 为位于第 i 行和第 j 列的元素, 若 Δ_{ij}

	a_1	\dots	a_j	\dots	a_m
t_1	1	\dots	1	\dots	0
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
t_i	1	\dots	0	\dots	1
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
t_m	0	\dots	1	\dots	1

图2 二维二进制编码

$= 1$, 表示 a_j 参与求解任务 t_i , 即该 agent 加入到任务 t_i 的求解联盟 C_i 中去; 若 $\Delta_{ij} = 0$, 则表示 a_j 不参与任务 t_i , 即该 agent 被排除在任务 t_i 的求解联盟 C_i 之外.

由于重叠联盟允许一个 agent 同时加入多个联盟, 因此编码中的每一列可能会有多个“1”, 这就可能带来如下两个问题:

(1) 对于每一行, 对应着一个任务求解联盟 C_i , 有可能 C_i 能力不够, 不足以承担其对应的任务, 即不满足式(1), 这时 C_i 对于 t_i 来说是一个无效联盟;

(2) 对于每一列, 对应着一个 agent a_j , 有可能这一列有多个“1”, 即有多个联盟同时竞争 a_j , 而 a_j 的能力又非常有限, 不可能同时满足这么多联盟的需求, 这时就会产生资源冲突和联盟死锁.

其实只要在编码中存在上述问题中的任何一个, 该编码就会是一个非法的编码, 即对应的是一个不可行解. 从进化计算的角度我们知道, 过多的非法编码会降低种群的多样性, 大大延缓算法的演化, 从而影响所得联盟的优越性. 为了解决上述问题, Lin 和 Hu^[26] 将所有有效联盟的剩余能力转移给一个固定的 agent, 由该 agent 代表若干有效联盟参与其他任务的求解, 以避免求解过程中的资源冲突, 但是 LH 算法仅凭编码的第一行是一个无效联盟或者有效联盟的剩余能力不能帮助完成某一无效行就丢弃整个编码, 这是极其随意和盲目的, 因为可能其他行的有效联盟的剩余能力可以满足第一行的需求, 虽然在 LH 算法中对所有丢弃的编码可以重新生成一个新编码再来检查, 但如此盲目的反复操作会大大滞后算法的执行, 影响算法的效率. 出现上述问题的根源在于 LH 算法是着眼于联盟来考虑问题, 而没有从编码本身来思考, 从而不能发挥二维二进制编码本身的优势. 因此, 本文将着重从编码的角度来考虑问题, 如果编码是非法的, 则将其修正为一个合法的编码, 而不会因任何原因丢弃它.

5 编码修正算法

5.1 算法描述

定义 5 二值函数 $flag(i)$ 表示编码中第 i 行的情况, 如果 $flag(i) = 1$, 则第 i 行是有效的, 否则 $flag(i) = 0$, 表示第 i 行是个无效行, 即对应的是一个无效联盟.

定义 6 继承有效联盟剩余能力的联盟 C^* 称为转移联盟, 因为 C^* 实际上是不存在的, 而且是变化的, 因此 C^* 是一个动态的虚拟联盟(virtual coalition).

基于上述定义, 本文对非法编码的修正算法可以描述如下:

Step 1 对每个 $i = 1, \dots, m, flag(i) \leftarrow 0$. 选择优先级最高且未检查的一行 i , 如果 $B_{C_i} = \sum_{j \wedge \Delta_{ij}=1} B_j \geq D_i$, 第 i

行有效, $flag(i) \leftarrow 1$, 此时对每个 $j = 1, \dots, n$, 如果 $\Delta_{ij} = 1$, 则对每个 $i^* = 1, \dots, m, i^* \neq i$, 执行 $\Delta_{i^*j} \leftarrow 0$; 如果 $B_{C_i} < D_i$, 则第 i 行无效.

Step 2 如果还有未检查的行, 则转 Step1. 如果所有行经检查均无效, 则选择优先级最高的一行, 在该行随机挑选一个“0”置为“1”, 并重复这个操作直到该行有效为止.

Step 3 $C^* \leftarrow \emptyset, B_{C^*} \leftarrow 0$, 转移所有有效联盟的剩余能力: 对每个 $i = 1, \dots, m$, 如果 $flag(i) = 1$, 则将其对应的有效联盟 C_i 的剩余能力 $B_{C_i} - D_i$ 转移给虚拟联盟 C^* , 即 $C^* \leftarrow C^* + C_i, B_{C^*} \leftarrow B_{C^*} + (B_{C_i} - D_i)$.

Step 4 对于剩下的无效行, 依次选取一个优先级最高的无效行 i' . 如果 $B_{C_{i'}} + B_{C^*} \geq D_{i'}$, 即 C^* 加入后 $C_{i'}$ 可以完成任务; 如果 $B_{C_{i'}} + B_{C^*} < D_{i'}$, 则 C^* 加入后 $C_{i'}$ 仍然无效, 此时在第 i' 行随机挑选一个自由可用的 agent, 即没有参加任何有效行的“0”, 将该“0”置为“1”, 并重复这个操作直到 $B_{C_{i'}} + B_{C^*} \geq D_{i'}$ 为止.

Step 5 转移剩余能力 $B_{C^*} \leftarrow B_{C^*} + B_{C_{i'}} - D_{i'}$, $flag(i') \leftarrow 1$. 对每个 $j = 1, \dots, n$, 如果 $\Delta_{i'j} = 1$, 则对每个 $i^* = 1, \dots, m, i^* \neq i'$, 且 $flag(i^*) = 0$, 执行 $\Delta_{i^*j} \leftarrow 0$.

Step 6 如果还有无效行没有处理, 转 Step4, 否则结束算法.

需要注意的是:

(1) 在 Step1 中, 每找到一个有效行 i , 就要检查该行的各个“1”所在的列, 解决多个联盟竞争同一个 agent 可能带来的资源冲突问题, 若 $\Delta_{ij} = 1$ 且第 j 列中还有其他多个“1”, 则为了保证优先级高的任务首先得到响应, 同时避免资源竞争, 将第 j 列中其他行的“1”全部置为“0”;

(2) Step2 是为了保证在编码中至少有一个有效行, 这样就不用丢弃该编码, 保证修正算法的正常运行;

(3) Step3 是将所有有效行的剩余能力转移给虚拟联盟 C^* , 避免了 agent 能力的浪费, 如果某列 agent 还有剩余能力, 则只允许该 agent 的剩余能力(通过转移给 C^*)参与竞争其他任务. 因此, 即使该 agent 参与了多个任务, 那也是在承担某任务之后的剩余能力参与了其他任务, 从而可以有效避免资源冲突;

(4) Step4 是对无效行进行修正, 因为充分考虑了联盟的剩余能力, 所以只要在式(3)的条件下, 就可以完全保证任何无效行都能修正成为一个有效行.

5.2 性能分析

命题 1 给定任意一个 $m \times n$ 的重叠联盟问题, 本文修正算法的复杂度至多为 $o(n^3)$.

证明 在本文的修正算法中:

(1) 检查每一行是否有效时, 需要对每一维都进行

判断, 最多可能需要遍历所有的 agent 以计算联盟能力. 每找到一个有效行, 就要检查该行的各个“1”所在的列(至多 n 列), 解决其他(至多 $m-1$ 个)联盟竞争同一个 agent 可能带来的资源冲突问题, 因此复杂度为 $o(m \times (r \times n + n \times m)) = o(n^3)$.

(2) 转移有效联盟的剩余能力时, 至多有 m 个有效联盟, 而每行至多需要遍历所有的 agent, 而且每一维都需要统计, 故复杂度至多为 $o(m \times n \times r) = o(n^3)$.

(3) 修正无效行时, 最多有 $m-1$ 个无效行, 而每选中一个无效行, 首先需要统计该行的实际能力, 这最多可能需要遍历所有的 agent, 如果在 C^* 加入后仍然不能完成任务, 则又要挑选新的合适的 agent 加入此任务, 且至多挑选 $n-1$ 个. 一旦该行有效, 又要检查该行的每个“1”所在的列(至多 n 列), 解决其他(至多 $m-2$ 个)无效联盟竞争同一个 agent 可能带来的资源冲突问题. 因此, 复杂度至多为 $o((m-1) \times (n \times r + (n-1) \times r + n \times (m-2))) = o(n^3)$.

综上, 本文算法的复杂度为 $o(n^3)$, 显然只是一个多项式复杂度.

命题 2 任意一个 $m \times n$ 的二维二进制编码都能被本文算法修正为一个合法编码.

证明 本文算法旨在先找出有效行, 再解决无效行, 因此这里从行的角度出发, 采用数学归纳法加以证明:

(1) 当 $m = 1$ 时, 编码中仅有一行, 由式(3)可以确保只要挑选有限的“0”置为“1”即可完成任务.

(2) 假设当 $m = p$ 时命题成立, 即 $p \times n$ 的二维二进制编码能被本文算法修正为一个合法编码, 经过修正后 p 行均是有效行.

(3) 则当 $m = p + 1$ 时, 由式(2)可得前 p 行可以被修正为有效行, 我们只需证明第 $p + 1$ 行也能被修正为一个有效行即可. 如果 $B_{C_{p+1}} + B_{C^*} \geq D_{p+1}$, 即虚拟联盟 C^* 加入后 C_{p+1} 可以完成任务, 只要前 p 行的转移联盟 C^* 加入, 第 $p + 1$ 行即为有效, 命题得证. 如果 $B_{C_{p+1}} + B_{C^*} < D_{p+1}$, 则需要挑选一些自由可用(即没有加入任何有效行)的 agent 加入 C_{p+1} 以实现 $B_{C_{p+1}} + B_{C^*} \geq D_{p+1}$. 为了说明一定存在自由可用的 agent 可供挑选, 这里采用反证法: 假设在第 $p + 1$ 行没有自由可用的 agent, 即第 $p + 1$ 行的“0”均参加了其他有效行, 因 $B_{C_{p+1}} + B_{C^*} < D_{p+1}$, 这里虚拟联盟 C^* 继承的是 A 中除了 C_{p+1} 以外的所有 agent 的剩余能力, 因此有 $B_{C^*} = (B_{C_1} - D_1) + \dots + (B_{C_p} - D_p)$. 而 $B_{C_{p+1}} + B_{C^*} = (B_{C_1} - D_1) + \dots + (B_{C_p} - D_p) + B_{C_{p+1}}$, 即 $B_{C_{p+1}} + B_{C^*} = (B_{C_1} + \dots + B_{C_p} + B_{C_{p+1}}) - (D_1 + \dots + D_p)$, 又 $(B_{C_1} + \dots + B_{C_p} + B_{C_{p+1}}) =$

$\sum_j B_j$, 且 $B_{C_{p+1}} + B_{C^*} < D_{p+1}$, 故 $\sum_j B_j - (D_1 + \cdots + D_p) < D_{p+1}$, 即 $\sum_j B_j < (D_1 + \cdots + D_p + D_{p+1})$, 因 $(D_1 + \cdots + D_p + D_{p+1}) = \sum_i D_i$, 所以有 $\sum_j B_j < \sum_i D_i$, 与式(3)矛盾, 得证。

因此, 当 $m = p + 1$ 时 $(p + 1) \times n$ 的二维二进制编码也能被本文算法修正为一个合法编码。

综上可得, 任意一个 $m \times n$ 的二维二进制编码都能被本文算法修正为一个合法编码。

6 实验结果与分析

考虑虚拟企业中的伙伴选择问题: 假设某集团公司承包了表 1 所示的四个工程(每个工程需要两种不同的资源), 表 2 给出了十个候选合作伙伴(每个候选企业各拥有一定量的所需资源), 现集团需要决策出四个项目联盟以获得最大的利润. 以式(4)为适应度函数, 以离散粒子群算法为实验平台^[26], 对比分析 LH 算法和本文算法. 采用标准的实验参数: 粒子数为 20, 粒子最大速度为 6.0, 最大迭代次数为 500. 两种算法各独立运行 50 次.

表 1 工程需求状况				
工程	t_1	t_2	t_3	t_4
资源需求量	[32, 15]	[18, 51]	[60, 23]	[30, 64]
优先级	$t_1 < t_2 < t_3 < t_4$			

表 2 企业资源状况										
企业	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
资源拥有量	[15, 23]	[7, 27]	[13, 19]	[14, 15]	[19, 24]	[29, 18]	[27, 18]	[10, 19]	[16, 21]	[5, 7]

50 次实验的结果见表 3, 最终的项目合作联盟见表 4, 图 3 为每次实验耗费的时间, 图 4 示出了两种算法所得最优解的进化曲线, 图 5 给出了每次实验搜索到的最佳值, 图 6 列出了每次实验丢弃的编码数。

表 3 50 次实验的结果				
算法	利润	丢弃编码数	运行时间(s)	
LH 算法	最大值	1069	6986	0.187
	最小值	1024	521	0.14
	平均值	1042	1965	0.155
本文算法	最大值	1174	0	0.172
	最小值	1140	0	0.14
	平均值	1158	0	0.155

表 4 项目合作联盟				
算法	t_1	t_2	t_3	t_4
LH 算法	$\{a_3, a_5\}$	$\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_9\}$	$\{a_4, a_6, a_7\}$	$\{a_1, a_2, a_9\}$
本文算法	$\{a_3, a_5\}$	$\{a_1, a_2, a_3, a_5\}$	$\{a_4, a_6, a_7\}$	$\{a_2, a_4, a_6, a_7, a_9\}$

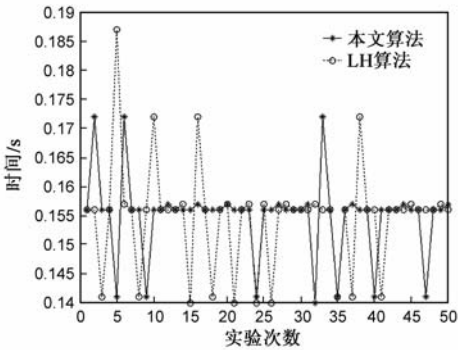


图3 每次实验耗费的时间

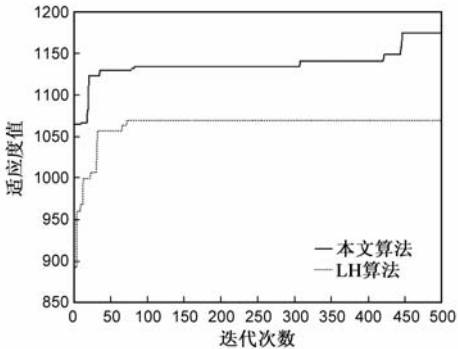


图4 最优解对应的进化曲线

由表 3 可以看出, 本文算法的效果要明显优于 LH 算法, 且在整个运行过程中没有丢弃任何非法编码, 而 LH 算法获得的利润较低(见表 4), 这是因为 $C_2 = C_1 + C_3 + C_4$, 为 C_1, C_3, C_4 的剩余能力组成的转移联盟, 因此 C_2 虽然能够完成工程 t_2 , 但联盟规模过于庞大, 带来巨大的联接成本, 从而降低了合作联盟的优越性. 在本文算法中, 只有第一行为有效行, 虚拟联盟 $C^* = \{a_3, a_5\}$, 修正第二行时, C^* 加入后第二行仍然为无效行, 且经比较其各维能力均小于等于任务的需求, 即不能完成任务. 因此, a_2 被随机选中, 此时第二行有效. 因原 C^* 的各维能力均小于等于任务的需求, 所以第二行的剩余能力应该是 a_2 的, 虚拟联盟 $C^* = \{a_2\}$. 同理第三行的剩余能力应该是 a_2, a_6, a_7 的, 虚拟联盟 $C^* = \{a_2, a_6, a_7\}$, 第四行的剩余能力应该是 a_4, a_9 的, 虚拟联盟 $C^* = \{a_4, a_9\}$. 这样处理的好处就尽可能的减少联盟规模, 从而降低成员之间的联接成本, 增加公司的总利润。

由图 3 可以看出, 本文的修正算法并没有带来较大的耗时, 且稍低于 LH 算法, 后者耗时的根源在于对任意无效编码都丢弃并重复生成一个新编码, 直到该编码有效为止. 而大量非法编码的丢弃会大大阻碍了算法的进化, 如图 4 所示, 本文算法在 450 代左右还在进化, 而 LH 算法在 100 代左右就基本不进化了. 因此, 本文算法解的质量要明显优于 LH 算法, 如图 5 所示。

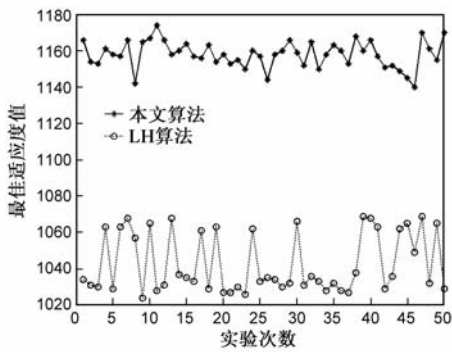


图5 每次实验搜索到的最佳值

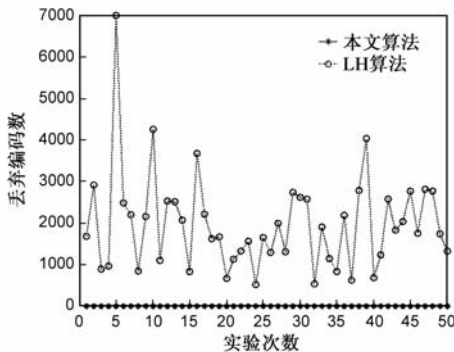


图6 每次实验丢弃的编码数

而且,我们在实验中发现:

(1)对于 agent 数和任务数越来越多时,特别是 agent 的总能力和任务的总需求很接近时,LH 算法的效率会大幅降低,因为这种情况下资源冲突会很激烈,导致大量编码会被抛弃,而本文的算法却不会丢弃任何编码,如图 6 所示。

(2)只要满足式(3),就可以将有效行的剩余能力转移出去帮助那些无效行。每列最多允许出现一个“1”,如果该列 agent 还有剩余能力,则只允许该 agent 的剩余能力(通过转移给虚拟联盟 C^*)参与竞争其他任务,这就避免了因多个联盟竞争同一个 agent 而引起的资源冲突。因此,本文算法不会因任何原因丢弃任何非法编码,而且任何一个非法编码都可以修正为一个合法的编码。

(3)当虚拟联盟 C^* 加入某无效行后能完成任务,则剩余能力有可能是 C^* 的,也有可能是原无效行中成员的,因为在原无效行中可能某些维上不能完成任务,而在另一些维上可以完成任务。另外,当 C^* 加入某无效行后仍然不能完成任务,就需要挑选新的自由可用的 agent 加入直到该行有效,此时这个新的有效联盟的剩余能力可能是新加入的那些 agent 的剩余的,也可能是原 C^* 剩余的,还有可能是原无效行中成员剩余的,需要区别加以对待。

7 结论及进一步工作

本文对重叠联盟形成问题做了一些试探性的研

究,提出了二维二进制编码的修正算法。与 LH 算法相比,本文算法的效果要更好,可以同时为多个任务快速生成相应的求解联盟,而且可以将任何一个无效编码修正为一个有效的编码,同时不会丢弃任何无效编码,从而能在一定程度上提高系统的搜索效率和求解质量。在今后的工作中,我们将重点研究二维交叉、变异算子的设计和修正。

参考文献

- [1] J Bohannon. Counterterrorism's new tool: 'metanetwork' analysis[J]. Science, 2009, 325(5939): 409 - 411.
- [2] F Schweitzer, G Fagiolo, et al. Economic networks: The new challenges[J]. Science, 2009, 325(5939): 422 - 425.
- [3] 张发, 宣慧玉, 等. 复杂系统多主体仿真方法论[J]. 系统仿真学报, 2009, 21(8): 2386 - 2390.
Zhang Fa, Xuan Hui-yu, et al. Methodology of multi-agent based simulation for complex systems[J]. Journal of System Simulation, 2009, 21(8): 2386 - 2390. (in Chinese)
- [4] K Tumer, A Agogino. Multiagent learning for black box system reward functions[J]. Advances in Complex Systems, 2009, 12(4-5): 475 - 492.
- [5] S S Manvi, M S Kakkasageri. Multicast routing in mobile ad hoc networks by using a multi-agent system[J]. Information Sciences, 2008, 178(6): 1611 - 1628.
- [6] 陈志, 王汝传, 等. 一种无线传感器网络的多 Agent 系统模型[J]. 电子学报, 2007, 35(2): 240 - 243.
Chen Zhi, Wang Ru-chuan, et al. Multi-agent system model for wireless sensor networks[J]. Acta Electronica Sinica, 2007, 35(2): 240 - 243. (in Chinese)
- [7] 翟社平, 魏娟丽, 等. 基于概念语义协商的服务 Agent 划分[J]. 电子学报, 2010, 38(9): 2030 - 2034.
Zhai She-ping, Wei Juan-li, et al. Service agents partition based on concepts semantic negotiation[J]. Acta Electronica Sinica, 2010, 38(9): 2030 - 2034. (in Chinese)
- [8] Guofu Zhang, Jianguo Jiang, et al. Searching for overlapping coalitions in multiple virtual organizations[J]. Information Sciences, 2010, 180(17): 3140 - 3156.
- [9] J M Zolezzi, H Rudnick. Transmission cost allocation by cooperative games and coalition formation[J]. IEEE Trans on Power Systems, 2002, 17(4): 1008 - 1015.
- [10] 李翠莲, 杨震, 等. 分组多用户检测联盟模型与联盟形成算法研究[J]. 电子学报, 2010, 38(10): 2447 - 2452.
Li Cui-lian, Yang Zhen, et al. Research on group multiuser detection coalition models and coalition formation algorithm[J]. Acta Electronica Sinica, 2010, 38(10): 2447 - 2452. (in Chinese)
- [11] L Vig, J A Adams. Multi-robot coalition formation[J]. IEEE Trans on Robotics, 2006, 22(4): 637 - 649.

- [12] K T Seow, K M Sim, et al. Coalition formation for resource coallocation using BDI assignment agents[J]. IEEE Trans on Systems, Man and Cybernetics Part C-Applications and Reviews, 2007, 37(4): 682 – 693.
- [13] Zhaopin Su, Jianguo Jiang, et al. Path selection in disaster response management based on Q-learning [J]. International Journal of Automation and Computing, 2011, 8(1): 100 – 106.
- [14] T Agotnes, W van der Hoek, et al. Reasoning about coalitional games[J]. Artificial Intelligence, 2009, 173(1): 45 – 79.
- [15] T Rahwan, S D Ramchurn, et al. An anytime algorithm for optimal coalition structure generation[J]. Journal of Artificial Intelligence Research, 2009, 34(1): 521 – 567.
- [16] 蒋建国, 夏娜, 等. 基于能力向量发挥率和拍卖的联盟形成策略[J]. 电子学报, 2004, 32(12A): 215 – 217.
Jiang Jian-guo, Xia Na, et al. The coalition formation strategy based on capability vector contribution-rate and auction[J]. Acta Electronica Sinica, 2004, 32(12A): 215 – 217. (in Chinese)
- [17] 蒋建国, 夏娜, 等. 一种基于蚁群算法的多任务联盟串行生成算法[J]. 电子学报, 2005, 33(12): 2178 – 2182.
Jiang Jian-guo, Xia Na, et al. An ant colony algorithm based multi-task coalition serial generation algorithm[J]. Acta Electronica Sinica, 2005, 33(12): 2178 – 2182. (in Chinese)
- [18] 蒋建国, 尹翔, 等. 基于历史行为的 agent 联盟策略[J]. 电子学报, 2007, 35(8): 1485 – 1489.
Jiang Jian-guo, Yin Xiang, et al. Strategy of agent coalition based on historicbehavior[J]. Acta Electronica Sinica, 2007, 35(8): 1485 – 1489. (in Chinese)
- [19] 李剑, 景博, 等. 一种基于奖励机制的 agent 联盟形成策略[J]. 电子学报, 2008, 37(12A): 71 – 75.
Li Jian, Jing Bo, et al. A strategy to form agent coalition based on encouragement [J]. Acta Electronica Sinica, 2008, 37(12A): 71 – 75. (in Chinese)
- [20] O Shehory, S Kraus. Methods for task allocation via agent coalition formation[J]. Artificial Intelligence, 1998, 101(1 – 2): 165 – 200.
- [21] V D Dang, R K Dash, et al. Overlapping coalition formation for efficient data fusion in multi-sensor networks[A]. AAAI-06, the 21st National Conf on Artificial Intelligence [C]. Boston, USA: AAAI Press, 2006. 635 – 640.
- [22] G Chalkiadakis, E Elkind, et al. Overlapping coalition formation[A]. The 4th International Workshop on Internet and Network Economics[C]. Shanghai, China: Springer-Verlag, 2008. 307 – 321.
- [23] G Chalkiadakis, E Elkind, et al. Cooperative games with overlapping coalitions [J]. Journal of Artificial Intelligence Research, 2010, 39: 179 – 216.
- [24] 张国富, 蒋建国, 等. 基于离散粒子群算法求解复杂联盟生成问题[J]. 电子学报, 2007, 35(2): 323 – 327.
Zhang Guo-fu, Jiang Jian-guo, et al. Solutions of complicated coalition generation based on discrete particle swarm optimization[J]. Acta Electronica Sinica, 2007, 35(2): 323 – 327. (in Chinese)
- [25] 蒋建国, 张国富, 等. 基于离散粒子群求解复杂联盟的并行生成[J]. 电子与信息学报, 2009, 31(3): 519 – 522.
Jiang Jian-guo, Zhang Guo-fu, et al. Complicated coalition parallel generation based on discrete particle swarm optimization[J]. Journal of Electronics & Information Technology, 2009, 31(3): 519 – 522. (in Chinese)
- [26] C F Lin, S L Hu. Multi-task overlapping coalition parallel formation algorithm[A]. AAMAS 2007, the 6th Int Joint Conf on Autonomous Agents and Multi-agent Systems[C]. Honolulu, Hawaii, USA: ACM Press, 2007. 1260 – 1262.

作者简介



张国富 男, 1979年3月出生于安徽合肥. 博士, IEEE会员. 2008年在合肥工业大学获工学博士学位, 现为信息与通信工程博士后科研流动站在站博士后, 从事联盟形成理论方面的有关研究.
E-mail: zgf@hfut.edu.cn



周鹏 男, 1987年3月出生于安徽蚌埠. 硕士研究生, 研究方向为进化计算、多agent系统.



蒋建国 男, 1955年10月出生于安徽黄山. 教授, 博士生导师, 全国信息与电子学科研究生教育委员会、本科生教育委员会常务委员, 安徽省计算机学会副理事长. 研究方向为数字图像处理、智能信息系统.