

基于合作技能博弈的单任务联盟结构生成算法

史强, 夏阳, 王磊

(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

摘要: 提出一种用于单任务最优联盟结构生成算法 STCSG。利用合作技能博弈(CSGs)模型和超图生成合作技能超图(skill hypergraph),根据 STSG 中最优联盟结构特性,具体讨论了当每个 agent 最多只能拥有一个技能和一个技能最多被两个 agent 共同拥有两种情况下搜索合作技能超图的策略,从而求得最优联盟结构。实验结果表明该算法搜索效率较高,时间复杂度为 $O(n^2)$ 。

关键词: 多智能体系统; 联盟结构; 合作技能博弈; 超图; 单任务技能博弈; 合作技能超图

中图分类号: TP18 **文献标志码:** A **文章编号:** 1001-3695(2012)07-2509-04

doi:10.3969/j.issn.1001-3695.2012.07.028

Single task coalition structure generation algorithm based on CSGs

SHI Qiang, XIA Yang, WANG Lei

(School of Computer Science & Technology, China University of Mining & Technology, Xuzhou Jiangsu 221116, China)

Abstract: This paper presented single task coalition structure generation (STCSG) algorithm to generate optimal single task coalition structure. It used CSGs model and hypergraph to generate skill hypergraph. According to the optimal coalition structure characteristics of STSG, specifically in both cases when each agent had only one skill and a skill was at most shared by two agent, it discussed the search strategy of skill hypergraph to obtain the optimal coalition structure. Experimental results show that search efficiency of the algorithm is high, and the time complexity is $O(n^2)$.

Key words: multi-agent system; coalition structure(CS); coalitional skill games(CSGs); hypergraph; single task skill game (STSG); skill hypergraph

0 引言

在多 agent 系统(multi-agent system, MAS)中智能体之间的合作是一个关键问题。在 1993 年欧洲“多 agent 世界中自治 agents 建模”研讨会上,针对智能体之间的合作问题, Ketchpel、Shechory & Kraus、Zlotkin & Rosenschein 分别发表的三篇论文不约而同地提到了联盟生成方法。

随着 MAS 的发展,研究人员在研究联盟生成问题方面取得了许多成果。Kraus^[1]于 2001 年提出分布式人工智能和 MAS 领域的许多问题与资源分配和约束有关,这使得资源分配和约束具有更加一般的研究意义。Wooldridge 等人^[2,3]于 2006 最早提出 CRGs (coalitional resource games),从资源约束的角度分析结盟问题的计算复杂性,探讨 CRGs 与 QCGs (qualitative coalitional games)关系,特别是在何种情况下两者之间是有可能转换的。

针对 agent 资源和能力的浪费问题,张国富等人^[4]提出复杂联盟和虚拟 agent 的概念,设计一种基于多粒子群协同优化的复杂联盟串行生成算法,实现一个 agent 可以加入多个联盟和一个联盟可以承担多个任务。Larson 等人^[5]对联盟结构生成的边界问题进行了研究,得出该问题为 NP-hard,并证明为建立起边界搜索联盟结构图的最底二层是必要且充分的,接着给出一个从顶层开始逐层搜索的任一时间算法。苏射雄等人^[6]提出基于局部最优 agent 联盟结构生成算法(OCS 算法),极大

地降低了搜索空间。梁军等人^[7]针对混合蚁群遗传算法进化代数多、效率低等问题,提出基于混合蚁群遗传算法的 agent 联盟求解算法。

针对生成单任务 agent 联盟,夏娜等人^[8]创新地引入“第 2 种信息素”对蚁群算法进行改进,不再易于陷入局部极小。文献[9]利用群体历史优质解,在最优粒子变异的基础上,采用多种群并行搜索,防止陷入局部极值,并对粒子群进行筛选以加快粒子群的收敛速度。文献[10]用差异演化算法求解单任务 agent 联盟。

单任务联盟结构生成的研究也是有意义的,因为在 MAS 中并不是任何时候都是多任务或者任务之间有优先级顺序的,并且在联盟结构中每个 agent 只参加一个联盟,系统中 agent 的能力不足以支持两个或更多任务的并行执行。

在多 agent 系统中,单任务最优联盟结构应该包含最多可以完成此任务的联盟,在实际应用中可以认为针对完成某个特定的任务,其最优联盟结构具有最多的功能集合备份。

本文基于 2008 年 Bachrach 等人^[11]提出合作技能博弈(CSGs)模型,结合单任务联盟结构特性,提出一种单任务多 agent 联盟结构生成算法。

1 问题描述

1.1 合作技能博弈模型

本文采用 CSGs 模型对联盟结构形成问题进行描述。

收稿日期: 2011-12-13; 修回日期: 2012-02-14

作者简介:史强(1985-),男,江苏徐州人,硕士,主要研究方向为多 agent 系统、联盟结构生成、电子商务(shiqiang@cumt.edu.cn);夏阳(1961-),男,江苏徐州人,硕士,主要研究方向为网络计算、电子商务;王磊(1979-),男,硕士,主要研究方向为网络计算、云计算。

CSGs 是一种合作博弈的受限形式,每个代理商拥有所需要的技能集合来完成各种任务。每个任务需要一套技能来完成,而且只有在联盟的 agents 包含完成任务所需的技能时才可以完成相应的任务。对于一个联盟收益只取决于它可以完成的任务子集。文献[11]给出了合作技能域与合作技能博弈的定义。

定义 1 合作技能域 (coalitional skill domain)^[11] 包括 agents 集合 $I = \{a_1, \dots, a_n\}$, 任务集合 $T = \{t_1, \dots, t_m\}$, 技能集合 $S = \{s_1, \dots, s_k\}$ 。

每个 a_i 有一个技能集合 $S(a_i) \subset S$, 每个任务 t_j 需要一个技能集合 $S(t_j) \subset S$ 来完成, 用 $S(C) = \cup_{a_i \in C} S(a_i)$ 表示联盟 C 的技能集合。如果 agents 的某个联盟 $C \subset I$ 可以完成任务 t_j , 并且完成任务 t_j 的所需技能全是联盟 C 包含的, 称 $S(t_j) \subset S(C)$, 表示为 $\text{perform}(C, t_j)$ 。联盟 C 的可执行任务集合为 $T(C) = \{t_j \in T | \text{perform}(C, t_j)\}$, 任务集合 T 的任务子集 $T' \subset T$, 其技能集合表示为 $S(T') = \cup_{t_j \in T'} S(t_j)$, 任务值函数将一个联盟完成的任务子集映射为一个实数值 $u: 2^T \rightarrow R$ 。一般假设可以任意地放弃某些任务, 不执行它们, 这样完成任务的总效用 u 也会减少, 所以如果 $T_1 \subset T_2$, 则 $u(T_1) < u(T_2)$ 。

定义 2 合作技能博弈 (CSGs) 是在一个合作技能域中的合作博弈 $\langle I, v \rangle$ 。其中: 参与者 I 是合作技能域中的 agents, 联盟的特征方程 v 是联盟可以完成任务的效用值 $V(C) = u(T(C))$ 。

1.2 单任务技能博弈(STSG)

在合作技能博弈 (CSGs) 中有一种很特殊的形式: 任务集合 T 中只有一个任务 t 。在这种情况下, 特征方程 u 只需在联盟结构中分辨哪些联盟结构可以完成任务 t , 放弃那些完成任务 t 不需要的技能。如果一个联盟结构 C 包含完成任务 t 所需的全部技能, 即表示此联盟结构 C 可以完成任务 t 。

定义 3 带权任务技能博弈 (weighted task skill game, WTSG)^[11]。它是带有任务权值方程 $w: T \rightarrow Q$ 的合作技能博弈 (CSGs)。 $\forall T' \subseteq T, W(T') = \sum_{t \in T'} w(t)$, 特征方程 u 表示为 $u(T') = w(T')$ 。

定义 4 单任务技能博弈^[11]。它是只有一个带有单位权值任务 t 的带权任务技能博弈 (WTSG), 若 $S(C) = S(t)$, 则 $v(C) = 1$; 否则 $v(C) = 0$ 。

1.3 联盟结构

联盟结构 (CS) 是由 Aumann 等人^[12] 在 1974 年提出的, 该模型实现了将 agents 分配到不同的集合中同时进行高效的工作。

定义 5 联盟结构是 agents 集合 I 的一个划分, 它是 I 的互不相交的非空子集簇, 并且这些子集的并集等于 I , 即

$$CS = (C_1, \dots, C_l), \cup_{i=1}^l C_i = I, \text{ 且 } C_i \cap C_j = \emptyset, i \neq j$$

联盟结构值 $V(CS)$ 是联盟结构 CS 中所有联盟值的和, 即 $V(CS) = \sum_{C_i \in CS} V(C_i)$

$CS(I)$ 表示 agents 集合 I 所有可能联盟结构组成的集合。对于给定的一个 CSGs, 最优联盟结构 $CS^* \in CS(I)$, 对任意 $CS \in CS(I), v(CS) \leq v(CS^*)$ 。

下面给出示例, 如表 1 所示。

表 1 I 对应的联盟结构及其假设 $V(CS)$

名称	agents 集合 I 的联盟结构	$V(CS)$
CS ₁	$\{\{a_1\}, \{a_2\}, \{a_3\}\}$	3
CS ₂	$\{\{a_1\}, \{a_2, a_3\}\}$	2
CS ₃	$\{\{a_2\}, \{a_1, a_3\}\}$	2
CS ₄	$\{\{a_3\}, \{a_1, a_2\}\}$	2
CS ₅	$\{\{a_1, a_2, a_3\}\}$	1

表 1 中 agents 集合 $I = \{a_1, a_2, a_3\}$, agents 集合 I 对应的所有联盟结构及其联盟结构值, 则 $CS(I) = \{CS_1, CS_2, CS_3, CS_4, CS_5\}$, 最优联盟结构 CS^* 为 CS_1 。

1.4 合作技能超图

定义 6 合作技能超图是一个超图 (hypergraph)^[13] $G = \langle V, E \rangle$, 根据一个给定的合作技能博弈 (CSGs), agents 作为此超图的顶点集合, 即 $V = I$, 每个 agent 表示为一个顶点 $i \in V$, 技能集合 S 形成此超图的超边集合, 即每个技能 s 表示为一条单独的超边 $e_s \in E, e_s$ 连接那些表示拥有技能 s 的 agents 顶点。 $n = \max_{s \in S} |e_s|$ 表示拥有同一个技能的 agents 的最大数量。连接顶点 $i \in V$ 的超边集合表示为 $E_i \subset E$ 。 $m = \max_{i \in V} |E_i|$ 表示合作技能超图所有顶点对应超边集合中边数最大值, 即合作技能博弈中一个 agent 最多可拥有的技能的数量。

图 1(a) 表示顶点 $V = \{a, b, c, d, e, f\}$, 其中 a 和 d 有共同技能, c 和 f 有共同技能, a, b 和 c 有共同技能, d, e 和 f 有共同技能, 容易得到 $n=3, m=2$; (b) 表示 $V = \{a, b, c, d, e\}$, 其中 a 和 d 有共同技能, b 和 e 有共同技能, c 独自有一个技能, $n=2, m=1$ 。

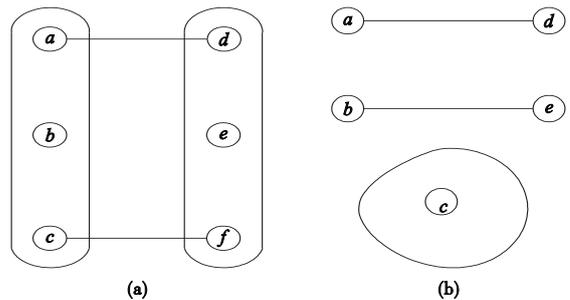


图 1 合作技能超图示例

2 STCSG 联盟结构生成算法

本算法针对单任务技能博弈, 结合合作技能超图和 STSG 中最优联盟结构特性, 具体讨论了当每个 agent 最多只能拥有一个技能和一个技能最多被两个 agent 共同拥有两种情况下联盟结构生成的策略。本算法在合理的时间内可以达到相当不错的效率。

2.1 STCSG 算法基本假设

Agents: 每个 agent 拥有的技能都是完成任务 t 所需要的技能, 没有多余的技能, 且 agents 拥有完成任务 t 所需的全部技能。

等效原则: 若合作技能超图 G 中, $\exists v_1, v_2 \in V$ 由同一条超边 e 连接, 则表示在 STSG 中存在两个 agent 具有相同的技能, 且两个 agent 在执行需要此技能的任务时具有相同效果的。

技能不重复原则: 在某个联盟中, 若已经存在带有技能 s 的 agent, 则不允许向此联盟添加带有技能 s 的 agent, 除非此 agent 带有联盟必需且其他 agent 没有的技能 s' 。

2.2 STSG 中最优联盟结构特性

文献[14]证明,即使在 STSG 中,计算其最优联盟结构也是 NP-Complete 问题。本文将合作技能超图引入 STSG 中,为了降低问题复杂度,设定 $n = \max_{s \in S} |e_s| = 2$, 即讨论一个技能最多被两个 agent 同时拥有的情况下最优联盟结构特性。

定理 1 给定一个 STSG, 包含完成任务 t 所需的全部技能, 并且其合作技能超图 G 中 $n = \max_{s \in S} |e_s| = 2$, 当且仅当 G 是二分图, 联盟结构的最大值 $\max_{CS \in CS(t)} v(CS)$ 为 2; 否则为 1。

证明 首先讨论 G 是二分图。如图 2(a) 所示, 所有超边连接两部分点集 V_l 和 V_r (其中 $V_l = \{a, b, c\}, V_r = \{d, e, f\}$), $\forall v_i \in V_l$, 总是唯一 $\exists v_j \in V_r$, 满足 v_i, v_j 由同一条超边 e_{ij} 连接, 容易得出在此 STSG 中, 每一个技能 s_{ij} 只被 agent $_i$ 和 agent $_j$ 拥有, 所以最多只有两个联盟可以覆盖所有技能, 不妨设为联盟 C_1 和 $C_2, CS = \{C_1, C_2\}$ 。根据 STSG 定义可知, 任一联盟 C 完成任务 t , 其联盟值 $v(C) = 1$, 则联盟 C_1 和 C_2 的联盟值分别为 $V(C_1) = 1, V(C_2) = 1$ 。可以得出以下结论:

$$\max_{CS' \in CS(t)} v(CS') = V(CS), V(CS) = V(C_1) + V(C_2) = 1 + 1 = 2$$

若 G 不是二分图, 且不含单点, 如图 2(b) 所示, 不能找到同时覆盖所有技能的两个联盟, 则联盟结构最大值为 $V(CS) = 1$ 。

最后讨论 G 含有单点的情况, 如图 2(c) 所示, 不妨设此单点为 v' , 存在一条超边 e' 只围绕 v' , 表示有一个技能 s' 只被相应的 agent 拥有, 以不会存在两个以上联盟同时拥有技能 s' , 则联盟结构最大值为 $V(CS) = 1$ 。

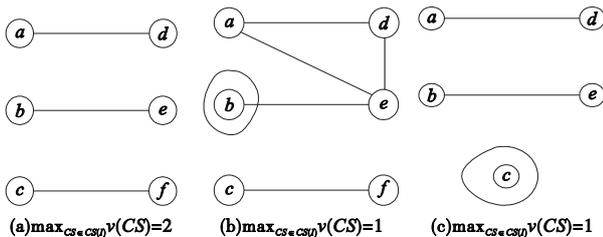


图2 联盟结构的最大值

引理 1 给定一个 STSG, 并且其合作技能超图 G 中 $n = \max_{s \in S} |e_s| = 2$, 当且仅当 G 是二分图, 则 G 的点集二分为两部分构成最优联盟结构 CS^* 。

如图 2(a) 所示, 最优联盟结构 $CS^* = \{\{a, b, c\}, \{d, e, f\}\}$ 。

引理 2 给定一个 STSG, 包含完成任务 t 所需的全部技能, 若其合作技能超图 G 中含有单点, 且此单点的超边对应的技能为任务 t 所需, 则联盟结构的最大值 $\max_{CS \in CS(t)} v(CS)$ 为 1。

如图 2(c) 所示, 当 c 与 a, b 构成联盟, 或者 c 与 d, e 构成联盟, $\max_{CS \in CS(t)} v(CS) = 1$ 。

以上的定理和引论是从限制每一个技能最多被两个 agent 同时拥有的角度来降低问题复杂度的, 对于每个 agent 所具有的技能数量并没有限制。下面设定 $m = \max_{i \in V} |E_i| = 1$, 即讨论每个 agent 只能拥有一个技能的情况下最优联盟结构特性。

定理 2 给定一个 STSG, 包含完成任务 t 所需的全部技能, 并且其合作技能超图 G 中 $m = \max_{i \in V} |E_i| = 1$, 联盟结构的最大值 $\max_{CS \in CS(t)} v(CS) = \min_{s \in S} |e_s|$ 。

证明 当合作技能超图 G 中 $m = \max_{i \in V} |E_i| = 1$, 即每个 agent 只拥有一个技能时, 图中超边不会出现连接和交叉情况,

其中存在一个技能 s 被最少数量的 agents 集合 I' 共同拥有, I' 中每个 Agent 都具有此技能 s , 那么 $\forall C \in CS^*$ 且 $V(C) = 1, \exists i \in I',$ 且 $i \in C$, 即在最优联盟结构中, 任意一个有效联盟都含有 I' 中的一个 agent, 所以 $V(CS^*)$ 等于集合 I' 中 agent 数量。如图 3 所示, 假设存在两个联盟 $C_1 = \{a, c, f\}$ 和 $C_2 = \{b, d, g\}, V(CS^*) = V(C_1) + V(C_2) = 1 + 1 = 2$ 。

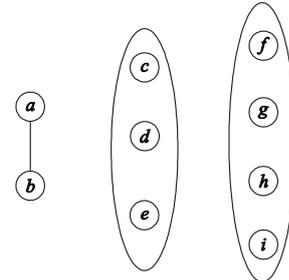


图3 $\max_{i \in V} |E_i| = 1$ 合作技能超图示例

2.3 STCSG 算法描述

结合上述 STSG 中最优联盟结构特性, 给出以下算法。

输入: 合作技能超图 $G = \langle V, E \rangle$ 。

输出: 在合作技能超图 G 中求出最优联盟结构 CS^* 及其 $V(CS^*)$ 。

a) 初始化, 设置 $CS^* = \emptyset, V(CS^*) = 0, \text{EdgeIDs} = \{1, 2, \dots, m\}$ 及 $\text{agents} = A, \text{RestOfAgents} = \emptyset, \text{RestOfEdgeIDs} = \emptyset, C_{\text{temp}} = \emptyset$ 。

b) 首先验证每个 agent 是否只拥有一个技能, 即验证 $\max_{i \in V} |E_i| = 1$ 。

```
for(int i = 1; i <= |A|; i++)
    if(|E_i| = 1) return true; else return false;
```

c) 如果 $\max_{i \in V} |E_i| = 1$, 则针对每个 agent 只拥有一个技能的情况(图 3), 计算出每条超边所包含点数量, 即针对每一个 $j \in \text{EdgeIDs}$ 计算其 $|e_j|$, 然后按照顺序排列。

d) 由定理 2 得 $\min_{s \in S} |e_s|$ 决定最优联盟结构中有效联盟个数, 并认为具有相同技能的 agent 是等效的。

```
for(int i = 1; i <= min_{s in S} |e_s|; i++)
    for(int j = 1; j <= |EdgeIDs|; j++)
```

从编号为 j 的超边包围的点中任意取出一 v_j , 将其表示的 agent 添加到联盟 C_i 中。

```
CS* = CS* union {C_i};
V(CS*) = V(CS*) + 1;
```

```
将剩下的所有的点表示的 agent 添加到联盟 C_0 中, 且 V(C_0) = 0;
CS* = CS* union {C_0}; V(CS*) = V(CS*) + V(C_0);
```

```
return CS* 和 V(CS*);
```

程序终止。

e) 如果 $\max_{i \in V} |E_i| \neq 1, \max_{s \in S} |e_s| = 2$, 则针对一个技能最多被两个 agent 同时拥有的情况(图 2), 根据二分图定理当且仅当无向图 G 的每一个回路的长度均为偶数时, G 才是一个二分图; 如果无回路, 相当于任一回路的长度为 0, 0 视为偶数。首先判断输入的合作技能超图 $G = \langle V, E \rangle$ 是否为二分图。如果合作技能超图 G 为二分图, 则由引理 1 可以得到 G 的点集二分为两部分 C_1 和 C_2 , 构成最优联盟结构 $CS^*, V(C^*) = 2$ 。

f) 如果合作技能超图 G 不是二分图, 那么此问题就演变为在此 STSG 中搜索最优的联盟问题。首先将超图 G 中的所有单点划分到子图 G_D 中, 不包括单点的无回路部分划分到子图

G_B 中,子图 G_D 和 G_B 中 agents 集合分别标记为 D, B ,其超边编号集合分别标记为 EI_D, EI_B 。RestOfAgents = $A \setminus (B \cup D)$, RestOfEdgeIDs = EdgeIDs $\setminus (EI_D \cup EI_B)$ 。

针对子图 G_D ,将其对应的 agents 集合 D 加入到联盟 $C_{temp} = C_{temp} \cup D$ 中。

针对子图 G_B ,采用 e)中算法得到其点集的二分的一半 B_{half} ,并将其加入到联盟 $C_{temp} = C_{temp} \cup B_{half}$ 中。

g)为保证要尽量少的 agent 加入联盟,所以在剩下的带回路子图 G_C 中先从具有对最多超边交叉的点 v 开始搜索(如果存在相等的超边交叉的点,那么随机选取其中一个点)。连接点 v 的超边编号集合为 $EI_v = \{e_{v1}, e_{v2}, \dots, e_{vk}\}$,将超点 v 所表示的 agent $_v$ 加入到联盟 C_{temp} 中, $C_{temp} = C_{temp} \cup \{agent_v\}$, RestOfEdgeIDs = RestOfEdgeIDs $\setminus EI_v$, RestOfAgents = RestOfAgents $\setminus \{agent_v\}$,并在子图 G_C 中除去相应的点 v 及其超边,返回到 g),直到 RestOfEdgeIDs = \emptyset 。

返回联盟 C_{temp} 。

h)将不在 C_{temp} 中的 agents 加入到联盟 C_0 中且 $V(C_0) = 0$ 。

i)根据返回的联盟 C_{temp} 和联盟 C_0 , $CS^* = \{C_{temp}\} \cup \{C_0\}$, $V(CS^*) = 1$ 。

程序终止。

2.4 时间复杂度分析

a)的时间复杂度为 $O(1)$ 。

b) ~ d)针对每个 agent 只拥有一个技能的情况,其时间复杂度为 $O(n)$ 。

e)判断输入的合作技能超图 $G = \langle V, E \rangle$ 是否为二分图,其时间复杂度为 $O(n^2)$ 。

f)生成子图过程其时间复杂度为 $O(n)$,针对子图 G_D 将其对应的 agents 集合 D 加入到联盟中的时间复杂度为 $O(n)$,针对子图 G_B 采用 e)中算法,其时间复杂度为 $O(n^2)$,所以步骤 f)的时间复杂度为 $O(n^2)$ 。

g)在剩下的子图中查找尽量少的点使其覆盖到所有超边,然后将其代表的 agents 加入联盟中,其时间复杂度为 $O(n^2)$ 。

h)和 i)的时间复杂度为 $O(1)$ 。

所以整个算法的时间复杂性为 $O(n^2)$ 。

3 实验结果

为验证算法的有效性,笔者在计算机上进行了模拟计算。实验取两组数据进行测试。

第一组数据:首先考虑在每个 agent 只拥有一个技能情况下,即 $\max_{i \in V} |E_i| = 1$ 时,设 agent 数目为 $n = 10$,并设定每个 agent 所具有的技能,如表 2 所示。

表 2 agent 所具有的技能情况 1

agent	skill	agent	skill
a_1	s_1	a_6	s_3
a_2	s_2	a_7	s_4
a_3	s_2	a_8	s_4
a_4	s_3	a_9	s_4
a_5	s_3	a_{10}	s_4

随机给出五个任务组成的任务集合 $T = \{t_1, t_2, t_3, t_4, t_5\}$, 集合 T 中每个任务对于技能的需求不同,如表 3 所示。

表 3 每个任务所需有技能情况

task	skill
t_1	s_1, s_2, s_3, s_4
t_2	s_2, s_3, s_4
t_3	s_1, s_3
t_4	s_2, s_4
t_5	s_3, s_4

经过计算,实验结果如表 4 所示。

表 4 集合 T 中单任务联盟结构生成实验结果

task	$V(CS)$	联盟结构 CS
t_1	1	$\{\{a_1, a_2, a_4, a_7\}, \{a_3, a_5, a_6, a_8, a_9, a_{10}\}\}$
t_2	2	$\{\{a_2, a_4, a_7\}, \{a_3, a_5, a_9\}, \{a_1, a_6, a_8, a_{10}\}\}$
t_3	1	$\{\{a_1, a_3\}, \{a_2, a_4, a_5, a_7, a_6, a_8, a_9, a_{10}\}\}$
t_4	2	$\{\{a_2, a_7\}, \{a_3, a_9\}, \{a_1, a_4, a_5, a_6, a_8, a_{10}\}\}$
t_5	3	$\{\{a_4, a_7\}, \{a_5, a_9\}, \{a_6, a_8\}, \{a_1, a_2, a_3, a_{10}\}\}$

第二组数据:考虑 $\max_{s \in S} |e_s| = 2$,即一个技能最多有两个 agent 共同拥有的情况,设 agent 数目为 $n = 30$,并且设定每个 agent 所具有的技能,如表 5 所示。

表 5 agent 所具有的技能情况 2

agent	skill	agent	skill
a_1	s_1, s_3	a_{16}	s_9
a_2	s_1, s_2	a_{17}	s_{12}, s_{26}, s_{32}
a_3	s_2, s_4, s_5, s_6	a_{18}	s_{13}, s_{32}, s_{27}
a_4	s_7	a_{19}	s_{14}
a_5	s_8	a_{20}	s_{30}
a_6	s_9	a_{21}	s_{17}, s_{22}
a_7	s_{10}, s_{12}	a_{22}	s_{18}, s_{22}, s_{23}
a_8	$s_{10}, s_{11}, s_{13}, s_{14}$	a_{23}	s_{19}, s_{21}, s_{23}
a_9	s_{11}	a_{24}	s_{24}
a_{10}	s_{15}	a_{25}	s_{24}, s_{25}
a_{11}	s_3, s_4, s_{16}, s_{17}	a_{26}	s_{25}
a_{12}	$s_5, s_{16}, s_{18}, s_{19}, s_{20}$	a_{27}	s_{26}, s_{27}, s_{28}
a_{13}	s_6, s_{20}, s_{21}	a_{28}	s_{28}, s_{29}
a_{14}	s_7	a_{29}	s_{29}
a_{15}	s_8	a_{30}	s_{31}

给出任务 t ,完成此任务 t 需要表 5 中所有的技能,即完成任务 t 需要技能集 $S = \{s_1, s_2, \dots, s_{32}\}$ 。因为此算法中有些部分为随机,经过 10 次计算,实验结果如表 6 所示(其中有效联盟 C 指可以完成任务的联盟,有效联盟中 agent 个数越好)。

表 6 单任务 t 联盟结构生成实验结果

task	$V(CS)$	min C	max C	近似最优联盟结构 CS^*
t	1	16	17	$\{\{a_2, a_4, a_5, a_6, a_8, a_{10}, a_{11}, a_{12}, a_{13}, a_{17}, a_{20}, a_{22}, a_{25}, a_{27}, a_{29}, a_{30}\}, \{a_1, a_3, a_7, a_9, a_{14}, a_{15}, a_{16}, a_{18}, a_{19}, a_{21}, a_{23}, a_{24}, a_{26}, a_{28}\}\}$

通过第一组数据实验可以得出将此算法推广到串行任务集的求解中也是可行的。在第二组数据实验中,STCSG 联盟结构算法可以高效准确地得到单任务 t 的近似最优联盟结构。通过这两组数据实验,STCSG 联盟结构算法在(下转第 2516 页)

(上接第 2512 页)求解单任务联盟结构生成问题中是简洁高效的,并且具有一定的推广性,在大规模多 agent 系统中具有广阔的应用空间。

4 结束语

本文基于合作技能博弈,引入超图概念表述 CSGs 中各元素关系,结合单任务联盟结构特性,针对性地提出了 STCSG 联盟结构生成算法。实验结果表明本算法简洁、快速、有效,并且具有一定的推广性。

如何将本文算法的思想应用到大规模多 agent 系统中针对多任务的联盟结构生成问题是有待研究的工作。

参考文献:

- [1] KRAUS S. Strategic negotiation in multi-agent environments [M]. Cambridge: MIT Press, 2001: 67-113.
- [2] WOOLDRIDGE M, DUNNE P E. On the computational complexity of qualitative coalitional games [J]. *Journal of Artificial Intelligence*, 2004, 158(1): 27-73.
- [3] WOOLDRIDGE M, DUNNE P E. On the computational complexity of coalitional resource games [J]. *Journal of Artificial Intelligence*, 2006, 170(10): 835-871.
- [4] 张国富, 蒋建国, 夏娜, 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, 2(12): 323-327.
- [5] LARSON K, SANDHOLM T. Anytime coalition structure generation;

an average case study [J]. *Journal of Experimental and Theoretical Artificial Intelligence*, 2000, 12(1): 23-42.

- [6] 苏射雄, 胡山立, 林超峰, 等. 基于局部最优的联盟结构生成算法 [J]. *计算机研究与发展*, 2007, 44(2): 277-281.
- [7] 梁军, 程显毅. 基于混合蚁群遗传算法的 agent 联盟求解 [J]. *计算机科学*, 2009, 36(5): 227-231.
- [8] 夏娜, 蒋建国, 魏星, 等. 改进型蚁群算法求解单任务 agent 联盟 [J]. *计算机研究与发展*, 2005, 42(5): 734-739.
- [9] 许波, 余建平. 基于 QPSO 的单任务 agent 联盟形成 [J]. *计算机工程*, 2010, 36(19): 168-170.
- [10] 武志峰, 黄厚宽. 用差异演化算法求解单任务 agent 联盟 [J]. *计算机研究与发展*, 2006, 43(Z1): 186-189.
- [11] BACHRACH Y, ROSENSCHEIN J S. Coalitional skill game [C] // Proc of the 7th International Joint Conference on Autonomous Agents and Multi-agent Systems. 2008: 1023-1030.
- [12] AUMANN R, DREZE J. Cooperative games with coalition structures [J]. *International Journal of Game Theory*, 1974, 3(4): 217-237.
- [13] GOTTLÖB G, LEONE N, SCARCELLO F. Hypertree decompositions and tractable queries [J]. *Journal of Computer and System Sciences*, 2002, 64(3): 579-627.
- [14] BACHRACH Y, MEIR R, JUNG K, *et al.* Coalitional structure generation in skill games [C] // Proc of the 24th AAAI Conference on Artificial Intelligence. Atlanta: AAAI Press, 2010: 703-708.