

文章编号: 2095-2163(2020)10-0167-03

中图分类号: TP18

文献标志码: A

# 一种基于三维编码修正的重叠联盟形成方案

韩露, 聂艳艳, 程文丽, 臧思雨

(安徽理工大学 经济与管理学院, 安徽 淮南 232001)

**摘要:** 重叠联盟形成是人工智能与多 Agent 系统领域中的一个关键问题。由于每个 Agent 拥有的资源有限, 不同任务求解联盟之间会存在潜在的资源冲突。本文将传统的差分进化扩充至三维整数编码, 更直观、有效地完成智能资源体的快速分配, 并针对初始化赋予任意值可能产生的联盟无效与资源冲突问题, 设计了相应的编码修正策略, 以确保任何一个无效编码都能够被修正为一个合法编码。

**关键词:** 多 Agent 系统; 重叠联盟; 三维整数编码; 编码修正

## An overlapping alliance formation scheme based on three-dimensional coding modification

HAN Lu, NIE Yanyan, CHENG Wenli, ZANG Siyu

(School of Economics and Management, Anhui University of Science and Technology, Huainan Anhui 232001, China)

**[Abstract]** The formation of overlapping alliances is a key problem in the field of artificial intelligence and multi-agent systems. As each Agent has limited resources, there are potential resource conflicts between different task solving alliances. In this paper, the traditional differential evolution is extended to the three-dimensional integer coding, so as to realize the fast allocation of intelligent resources more intuitively and effectively. What's more, in order to solve the problem of alliance invalidity and resource conflict caused by initialization assigning arbitrary values, the corresponding encoding correction strategy is designed to ensure that any invalid encoding can be modified into a valid encoding.

**[Key words]** Multi-agent systems; Overlapping coalitions; Three dimensional integer encoding; Encoding repairs

### 0 引言

团队协作已渐成为在面对并行多任务时的首要选择。个人与个人之间、小组与小组之间、企业与企业之间相互合作已成为新时代的常态。由于每个个体拥有的能力不同, 所获的资源不同, 在面对并行的、综合的任务时, 合理组成有效联盟进行任务求解将在一定程度上实现效率最大化、资源浪费最小化以及任务总收益最大化的完美结合<sup>[1]</sup>。

随着计算机科学技术的迅速发展, agent 理论、多 agent 系统 (Multi-agent systems, MAS) 等概念应运而生, 基于 MAS 的联盟研究也受到广泛关注<sup>[2]</sup>。多 agent 系统中的重叠联盟形成 (overlapping coalition formation, OCF), 在求解复杂并行分布式任务时, 各 agent 之间取长补短、亲密协作, 求解任务灵活性强, 解决了单个 agent 因资源不足而无法完成任务需求或勉强完成任务但效率低下的问题。为此, 蒋建国等提出了一种基于能力向量发挥率和拍卖的联盟形成策略, 在面向任务的领域中可以达到

全局优化解, 较好地满足了稳定性、时效性、分布等要求<sup>[3]</sup>; 张国富等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, 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<sup>[4]</sup>。

基于上述背景, 本文将二维二进制编码扩充至三维整数编码, 构建“任务”、“资源”、“agent”于一体的三维空间坐标系, 更直观、有效地完成智能资源体的快速分配, 并针对联盟形成过程中可能出现的资源冲突与联盟无效问题, 提出相应的三维编码修正方案。

### 1 OCF 数学模型

设 MAS 中的 agent 个数为  $n$ ,  $A = \{a_1, a_2, \dots, a_n\}$ , 需要求解的任务数为  $m$ ,  $T = \{t_1, t_2, \dots, t_m\}$ 。

(1) 对于  $\forall a_j \in A$ , 具有  $r$  种初始资源向量  $B_j = [b_1^j, b_2^j, \dots, b_r^j]$ , 其中  $b_k^j$  表示  $a_j$  具有第  $k$  种资源的数量;

(2) 对于  $\forall t_i \in T$ , 需  $r$  种资源  $D_i = [D_1^i, D_2^i, \dots,$

**作者简介:** 韩露 (1999-), 女, 本科生, 主要研究方向: 电子商务、大数据、人工智能; 聂艳艳 (1999-), 女, 本科生, 主要研究方向: 电商智能推荐、大数据下企业管理; 程文丽 (1997-), 女, 本科生, 主要研究方向: 电子商务、商务智能应用; 臧思雨 (1997-), 女, 本科生, 主要研究方向: 电子商务、企业发展与战略。

收稿日期: 2020-05-21

哈尔滨工业大学主办 ● 专题设计与应用

$D_r^i]$ , 其中  $D_k^i$  表示任务  $t_i$  需要第  $k$  种资源的数量, 且仅当  $b_k^{C_i} \geq d_k^i$ , 联盟  $C_i$  才能完成任务  $t_i$ , 否则, 联盟无效;

(3) 对于  $\forall a_j \in A$ , 参与任务  $t_i$  实际贡献的资源量  $W_{ji} = [W_1^j, W_2^j, \dots, W_r^j]$ , 其中  $W_k^j$  表示  $a_j$  求解任务  $t_i$  时实际贡献第  $k$  种资源的数量, 且为避免资源冲突, 需满足式(1)<sup>[5]</sup>:

$$\sum_{i=1}^m w_k^j \leq b_k^j. \quad (1)$$

(4) 用  $V(C_i)$  表示联盟  $C_i$  的值, 式(2)<sup>[6]</sup>:

$$V(C_i) = \sum_{i=1}^m [\varphi(t_i) - \theta(C_i) - \Pi(C_i)]. \quad (2)$$

其中,  $\varphi(t_i)$  为完成任务  $t_i$  获得的报酬, 一般为常数;  $\theta(C_i)$  为联盟  $C_i$  中所有 agent 成员的总资源成本, 即为联盟中各成员实际贡献的资源量;  $\Pi(C_i)$  为任务  $t_i$  的求解联盟  $C_i$  中各 agent 成员两两之间的通信成本之和,  $\pi_{i_1 i_2}$  为  $a_{i_1}$  与  $a_{i_2}$  之间的通信成本。重叠联盟形成问题即为在满足上述约束条件的基础上使  $V(C_i)$  值尽可能大。

## 2 三维整数编码

### 2.1 符号约定

为了方便描述与理解, 将方案中涉及的相关概念符号整理见表1。

表1 修正方案符号说明

Tab. 1 Symbol description of modification scheme

符号	意义
$t_k^i$	$t_i$ 的第 $k$ 种资源
$a_k^j$	$a_j$ 的第 $k$ 种资源
$X_k^i$	资源冗余量, 即联盟提供的资源量较任务需求的超出量
$Y_k^i$	$a_k^i$ 资源剩余量, 即 agent 参与联盟后第 $k$ 种资源剩余量
$F(t_k^i)$	$t_k^i$ 完成度, $F(t_k^i) = 0$ 表示未完成; $F(t_k^i) = 1$ 表示完成
$C(t_k^i)$	$t_k^i$ 冲突性, $C(t_k^i) = 0$ 表示未冲突; $C(t_k^i) = 1$ 表示资源冲突

### 2.2 编码修正方案

首先针对每个  $W_k^j$ , 以  $[0, B_k^j]$  为区间初始化任意值。此时可能会出现联盟无效与资源冲突现象, 因此进行以下步骤调整。

**Step 1** 计算此时每个任务的完成情况, 式(3)、式(4):

$$X_k^i = \sum_{j=1}^n w_k^j - D_k^i, \quad (3)$$

$$(k \in 1, 2, \dots, r; i \in 1, 2, \dots, m.)$$

$$X_k^i \geq 0 \rightarrow F(t_k^i) = 1. \quad (4)$$

其中,  $X_k^i > 0$  时具有资源冗余量, 式(5);

$$X_k^i < 0 \rightarrow F(t_k^i) = 0. \quad (5)$$

此时联盟提供的资源量少于任务需求量, 资源缺少量为  $-X_k^i$ ;

**Step 2** 计算此时每个 agent 的资源消耗情况, 式(6)、式(7)为:

$$Y_k^j = B_k^j - \sum_{i=1}^m w_k^j, \quad (6)$$

$$(k \in 1, 2, \dots, r; j \in 1, 2, \dots, n).$$

$$Y_k^j \geq 0 \rightarrow C(t_k^i) = 0. \quad (7)$$

其中,  $Y_k^j > 0$  时具有资源剩余量, 式(8);

$$Y_k^j < 0 \rightarrow C(t_k^i) = 1. \quad (8)$$

此时该 agent 该种资源拥有量不能够满足其消耗量, 资源冲突量为  $-Y_k^j$ ;

**Step 3** 根据 step1 中计算结果, 做出调整以保证所有任务皆可完成, 具体步骤如下:

取  $\max(-X_k^i)$  ( $k \in 1, 2, \dots, r; i \in 1, 2, \dots, n$ ). 对应的  $t_k^i$  为“调整  $t_k^i$ ”, 找出相同资源下  $\max Y_k^j$  对应的 agent, 在其能力范围内, 最大限度增加该 agent 在“调整  $t_k^i$ ”中的贡献量。若  $X_k^i \geq 0$ , 该  $t_k^i$  调整结束, 跳转至 step5; 若  $X_k^i < 0$ , 则跳转至 step4;

**Step 4** 找出  $\max(X_k^i)$  对应的  $t_k^i$  中  $\max(w_k^j)$  对应的 agent, 降低其贡献值, 相应增加该 agent 在“调整  $t_k^i$ ”中的贡献量。重复执行此步骤, 直到“调整  $t_k^i$ ” $X_k^i \geq 0$  为止。调整结束后, 更新各  $X_k^i, Y_k^j$  值;

**Step 5** 取更新后此时的  $\max(-X_k^i)$  对应  $t_k^i$  为“调整  $t_k^i$ ”, 继续 step 3 中步骤。直至所有  $X_k^i < 0$  对应的  $t_k^i$  调整完毕, 即所有任务皆可完成, 联盟有效;

**Step 6** 根据最终更新的结果, 做出调整以满足所有 agent 的资源贡献均在其能力范围内, 即避免资源冲突, 具体步骤如下:

取  $\max(-Y_k^j)$  对应的  $a_k^j$  为“调整  $a_k^j$ ”, 检查该 agent 在所有参与的  $t_k^i$  中是否有资源冗余的。若有, 则在保证任务完成的情况下, 最大限度减少它在该冗余  $t_k^i$  中的贡献量; 若无, 则直接跳转至 step7;

**Step 7** (1) 若经过调整后, “调整  $a_k^j$ ”仍存在资源冲突, 则找出与“调整  $a_k^j$ ”在同一  $t_k^i$  中  $\max(Y_k^j)$  对应的 agent, 增加其在  $t_k^i$  中的贡献量, 以相应减少“调整  $a_k^j$ ”在  $t_k^i$  中的资源贡献量, 直到“调整  $a_k^j$ ”无资源冲突现象。更新各  $X_k^i, Y_k^j$  值; (2) 若经过调整后, “调整  $a_k^j$ ”冲突消解, 则直接转至 step8;

**Step 8** 跳转至 step6 重复以上步骤, 直至所有  $Y_k^j < 0$  对应的 agent 调整完毕, 即所有冲突消解。更新各  $X_k^i, Y_k^j$  值, 调整结束。

### 3 实例分析

假设有 2 个 agent, 其所拥有的资源向量分别为  $B_1 = [2, 3]$ ,  $B_2 = [3, 2]$ , 需求解的任务数为 2, 其对应的资源需求向量分别为  $D_1 = [4, 3]$ ,  $D_2 = [1, 1]$ , 如图 1 所示。

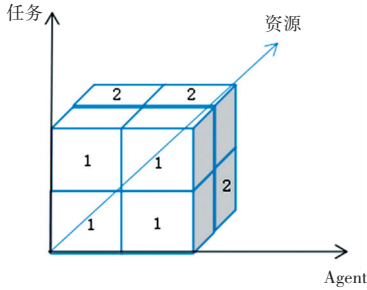


图 1 三维空间坐标系示意图

Fig. 1 Schematic diagram of three-dimensional spatial coordinate system

根据能力约束条件产生的初始联盟如下:

$$\begin{cases} W_{11} = [1, 2], \\ W_{12} = [1, 2], \\ W_{21} = [1, 2], \\ W_{22} = [1, 2]. \end{cases}$$

编码修正过程:

**Step 1** 首先计算此时每个任务对应每种资源的完成情况:

$$X_1^1 = W_1^{11} + W_1^{21} - D_1^1 = 1 + 1 - 4 = -2 \rightarrow F(t_1^1) = 0,$$

$$X_2^1 = W_2^{11} + W_2^{21} - D_2^1 = 2 + 2 - 4 = 1 \rightarrow F(t_2^1) = 1,$$

$$X_1^2 = W_1^{12} + W_1^{22} - D_1^2 = 1 + 1 - 1 = 1 \rightarrow F(t_1^2) = 1,$$

$$X_2^2 = W_2^{12} + W_2^{22} - D_2^2 = 2 + 2 - 1 = 3 \rightarrow F(t_2^2) = 1.$$

**Step 2** 计算此时每个 agent 的每种资源消耗情况:

$$Y_1^1 = B_1^1 - (W_1^{11} + W_1^{12}) = 2 - (1 + 1) = 0 \rightarrow C(t_1^1) = 0,$$

$$Y_2^1 = B_2^1 - (W_2^{11} + W_2^{12}) = 3 - (2 + 2) = -1 \rightarrow C(t_2^1) = 1,$$

$$Y_1^2 = B_1^2 - (W_1^{21} + W_1^{22}) = 3 - (1 + 1) = 1 \rightarrow C(t_1^2) = 0,$$

$$Y_2^2 = B_2^2 - (W_2^{21} + W_2^{22}) = 2 - (2 + 2) = -2 \rightarrow C(t_2^2) = 1.$$

**Step 3** 选中任务完成度最低的  $t_1^1$  为“调整  $t_1^1$ ”。该种资源剩余量最多的为 agent2, 剩余量为 1。增加 agent2 在  $t_1^1$  中的贡献量, 并更新:

$$Y_1^2 = B_1^2 - (W_1^{21} + W_1^{22}) = 3 - (2 + 1) = 0,$$

$$X_1^1 = W_1^{11} + W_1^{21} - D_1^1 = 1 + 2 - 4 = -1.$$

**Step 4** 此时  $t_1^1$  仍无法完成, 则选中同种资源条件下资源冗余最多的  $t_1^2$ 。其中 agent 贡献量相同, 则选中 agent1, 减少其在  $t_1^2$  中贡献量并增加在  $t_1^1$  中贡献量, 并更新:

$$X_1^1 = W_1^{11} + W_1^{21} - D_1^1 = 2 + 2 - 4 = 0,$$

$$X_1^2 = W_1^{12} + W_1^{22} - D_1^2 = 0 + 1 - 1 = 0.$$

**Step 5** 经检查, 此时所有  $X_k^i < 0$  对应的  $t_k^i$  调整完毕, 联盟有效。

**Step 6** 选中此时资源冲突量最多的  $Y_2^2$ , 冲突量为 2。且此时  $X_2^1$  与  $X_2^2$  皆冗余, 在保证  $X_2^2$  完成的情况下, 减少 agent2 在  $X_2^2$  中 2 个单位贡献量, 并更新:

$$X_2^2 = W_2^{12} + W_2^{22} - D_2^2 = 2 + 0 - 1 = 1,$$

$$Y_2^2 = B_2^2 - (W_2^{21} + W_2^{22}) = 2 - (2 + 0) = 0.$$

**Step 7** 接着选中此时资源冲突量最多的  $Y_2^1$ , 冲突量为 1。且此时  $X_2^1$  冗余, 减少 agent1 在  $X_2^1$  中 1 个单位贡献量, 并更新:

$$X_2^1 = W_2^{11} + W_2^{21} - D_2^1 = 1 + 2 - 3 = 0,$$

$$Y_2^1 = B_2^1 - (W_2^{11} + W_2^{12}) = 3 - (1 + 2) = 0.$$

**Step 8** 根据最终更新的结果, 此时所有  $Y_k^i \geq 0$  对应的 agent 调整完毕, 冲突消解。

### 4 结束语

为了更直观、有效的挖掘重叠联盟, 本文采用了三维整数编码的表示方式, 并提出一种新型的三维编码修正方案。传统的二维二进制编码只能表示各 Agent 成员是否参与联盟, 而不能显示各成员在联盟中贡献的资源量, 二维二进制编码和整数混合编码过于复杂和冗余。因此, 本文构建了“任务”、“资源”、“agent”于一体的三维空间坐标系, 并针对初始赋予以任意值可能产生的联盟无效与资源冲突问题, 设计了相应的编码修正策略, 以确保任何一个无效编码都能够被修正为一个合法编码。

### 参考文献

- [1] 牛娃. Agent 技术在企业动态联盟体系结构中的应用[J]. 黑龙江工程学院学报, 2003(2): 55-56.
- [2] 周鹏. 多 agent 系统中重叠联盟形成问题研究[D]. 合肥工业大学, 2013.
- [3] 蒋建国, 夏娜, 于春华. 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 (S1): 215-217.
- [4] 张国富, 周鹏, 蒋建国, 等. 基于虚拟联盟的重叠联盟形成算法[J]. 电子学报, 2012, 40(1): 121-127.
- [5] 张国富, 周鹏, 苏兆品, 等. 基于讨价还价的重叠联盟效用划分策略[J]. 模式识别与人工智能, 2014, 27(10): 930-938.
- [6] 桂海霞, 蒋建国, 张国富. 面向并发多任务的重叠联盟效用分配策略[J]. 模式识别与人工智能, 2016, 29(4): 332-340.