

# 一种改进的多 agent 分布式联盟形成算法

卢少磊<sup>†</sup>, 方浩

(北京理工大学自动化学院, 北京 100081)

**摘要:** 仅采用任务性价比作为多智能体任务分配过程中的任务选择标准, 会产生时间消耗大、资源利用低等问题. 为此, 综合任务性价比和智能体资源的特点, 提出了多任务准备度的概念. 根据多智能体任务分配过程的收敛性和时效性, 采用 Learning Automata 算法动态调整任务准备度各项的权重; 进而利用该方法模拟解决了低、中、高 3 种任务需求下多智能体任务分配问题. 仿真实验结果验证了所提出方法的有效性, 资源冗余可至少减少 20%.

**关键词:** 多 agent 系统; 联盟形成; 任务准备度; 强化学习

**中图分类号:** TP273      **文献标志码:** A

## An improved distributed coalition formation algorithm in MAS

LU Shao-lei<sup>†</sup>, FANG Hao

(School of Automation, Beijing Institute of Technology, Beijing 100081, China)

**Abstract:** Cost-effective is usually used as a unique standard in multi-agent task allocation process, which leads to large time consumption and low resource utilization. Therefore, the concept of task readiness is proposed, which combines cost-effective and the resource feature of the agent. According to the astringency and timeliness in the task allocation process, the learning automata algorithm is used to dynamically adjust the weights of each item of task readiness. Simulations of task allocation are done under low, medium and high task demands by using the proposed. The results show the effectiveness of the method, and the resource redundancy is reduced at least by 20%.

**Keywords:** MAS; coalition formation; task readiness; reinforce learning

## 0 引言

多智能体系统中, 当单个智能体无法完成某一任务时, 会与网络内其他智能体通过合作的方式形成联盟共同完成该任务. 智能体通过形成联盟来执行个体无法完成的任务, 提高了任务成功完成的概率和执行的效率. 然而, 联盟形成方式的不同直接影响着任务的执行效率. 因此, 联盟形成机制的设计是联盟形成的一个本质问题, 受到了学术界的广泛关注<sup>[1-4]</sup>.

在联盟形成机制研究方面, 部分学者通过采用博弈论方法来形成不同的多智能体联盟<sup>[5-8]</sup>, 主要是根据博弈论中的 shaply 值、核、核心等概念, 通过收益分配的稳定和合理来达到联盟形成的目的. 然而, 通常情况下这种方式的计算过程是 NP 完全的, 且计算量会随着智能体的数量呈现指数增长. 同时, 基于博弈论的研究文献中, 并没有给出具体的联盟形成策略, 大多只停留于理论研究, 很难直接应用于实际.

还有一些学者通过任务分配这一手段进行联盟形成的研究, 包括多智能体多任务、多智能体单任务分配、单智能体多任务分配、单智能体单任务分配. 文献 [9] 通过一致性任务分配方法进行联盟形成研究, 该方法只考虑了一个智能体执行多个任务, 并没有对多个智能体执行一个任务进行研究. 文献 [10-12] 利用仿生学方法, 模拟海豚捕食等行为的联盟形成过程建立数学模型. 文献 [13] 提出了一种基于贪婪算法的任务分配策略, 该算法结构简单、较容易实现, 但解的质量难以保证. 文献 [14] 提出了一种基于任务性价比的潜在联盟形成策略, 同时, 通过随机删除联盟内多余智能体进行任务选择, 依次选择性价比最优、次优等任务. 该算法避免了多个智能体选择同一个任务的资源浪费, 但是任务选择的过程是串行形式, 前一个任务联盟成员的随机选取而造成的资源浪费会影响后续任务的资源利用.

**收稿日期:** 2016-03-04; **修回日期:** 2016-05-18.

**基金项目:** 国家基金委创新研究群体项目 (61321002); 国家基金委重大国际合作项目 (61120106010); 国家自然科学基金项目 (61573062).

**作者简介:** 卢少磊 (1987-), 男, 博士生, 从事多智能体任务分配、模式识别的研究; 方浩 (1973-), 男, 教授, 博士生导师, 从事多智能体系统、模式识别等研究.

<sup>†</sup>通讯作者. E-mail: lu\_shaolei@126.com

针对文献[14]中仅依据任务性价比进行任务选择所带来的任务分配时间消耗较大以及资源利用率不足等问题,本文从任务本身特点(任务性价比)和智能体本身能力的特点两个方面出发,提出智能体准备度的概念,该概念表示该智能体执行该任务的准备程度。

## 1 问题描述

设  $N = \{A_1, A_2, \dots, A_n\}$  表示系统中 agent 的集合,每个 agent  $A_i$  拥有一定的资源  $R_{A_i} = \langle r_{A_i}^1, r_{A_i}^2, \dots, r_{A_i}^k \rangle$ . 其中:  $k$  表示系统资源的种类,  $r_{A_i}^l \geq 0$  ( $1 \leq l \leq k$ ) 表示 agent  $A_i$  拥有第  $l$  种资源的数量.  $T = x\{t_1, t_2, \dots, t_m\}$  为系统中待分配的任务集,  $R_{t_j} = \langle r_{t_j}^1, r_{t_j}^2, \dots, r_{t_j}^k \rangle$  表示任务  $t_j$  分别对  $k$  种资源的需求. 在多 agent 联盟形成过程中,联盟  $C$  的成员 agent 拥有的第  $l$  项资源的总和为  $r_C^l = \sum_{A_i \in N_C} r_{A_i}^l$ , 若对于任意  $l \in [1, k]$  满足  $r^l \geq r_{t_j}^l$ , 则联盟形成完成. 文献[14]中令每个智能体选择加入任务性价比最高的任务. 假设任务为  $t$ , 其任务性价比的定义为完成任务  $t$  所能获得的收益与完成任务  $t$  需要的资源总和的比值,即

$$e(t) = \frac{P(t)}{\sum_{l=1}^k r_{t_j}^l}. \quad (1)$$

从式(1)可以看出,任务性价比是一个绝对指标,当系统内智能体按照这一指标进行任务选择时,是按照一个绝对的顺序来选择的,即智能体选择任务的过程是个单线程。

文献[14]中 agent 优先申请性价比最高的任务,任务管理 agent 判断所有申请 agent 的资源总和是否满足任务需求. 若满足需求,则对所有申请采用回退机制,产生随机令牌,令多余 agent 退出联盟选择性价比次高的任务;依次完成所有任务的选择,形成最终联盟. 由该文献中仿真实验可以看出,该方法比单纯采用贪心算法提高了联盟形成效率,增加了系统收益. 但依然存在如下问题:没有从智能体出发,没有考虑智能体本身的能力资源特征,容易造成 agent 能力资源的浪费,尤其在系统资源有限的情况下,可能出现因资源分配不合理而造成有些任务无法完成。

## 2 改进的分布式联盟形成算法

### 2.1 智能体与任务契合度

在智能体采用性价比作为任务选择的唯一原则时,忽略了智能体自身的资源或能力特征. 由短板理论(木桶原理)可知,若木桶中一块木板很短,则盛水量就被短板所限,这块短板就成了木桶盛水量的“限

制因素”. 若在形成联盟的智能体中存在智能体的资源或能力与任务的需求差异性过大,可认为智能体的资源或能力的“短板”和任务需求的“短板”位置不同,则多智能体联盟形成的结果可能会造成智能体的资源或能力的浪费。

例如:设智能体  $A, B, C$  的资源向量分别为

$$\begin{aligned} r_A &= \langle r_A^1, r_A^2, r_A^3 \rangle, \\ r_B &= \langle r_B^1, r_B^2, r_B^3 \rangle, \\ r_C &= \langle r_C^1, r_C^2, r_C^3 \rangle, \end{aligned}$$

任务  $t$  的资源需求向量为  $r_t = \langle r_t^1, r_t^2, r_t^3 \rangle$ . 若智能体  $A, B, C$  的资源向量中,最小值均为第 2 项资源,而任务  $t$  的需求向量中,最小值为第 1 项资源,则当  $r_A^2 + r_B^2 + r_C^2 \geq r_t^2$  时,即智能体第 2 项(最小值项)资源的总和满足任务需求相应项时,智能体其余各单项的资源总和与任务需求向量中相应单项资源需求的差值  $\delta$ ,会随着智能体资源向量与任务需求向量差异的增大而增大. 即智能体  $A, B, C$  的资源向量与任务需求向量的方向越接近,资源的利用率越高。

这里引入巴氏系数作为智能体能力资源与任务需求的契合度,即智能体与该任务的适合度. 契合度值越大,执行该任务时智能体自身资源的利用率越高. 智能体  $r$  的能力值与任务  $t$  的契合度值为

$$Q(g, t_r) = \sum_{l=1}^k \sqrt{g(l)t_r(l)}. \quad (2)$$

其中:  $k$  为任务需求的资源种类;  $g(l)$  和  $t_r(l)$  分别表示智能体  $g$  和任务  $t$  的第  $l$  项资源在各自总的资源中所占的比例,即

$$\sum_{l=1}^k g(l) = 1, \quad \sum_{l=1}^k t_r(l) = 1.$$

### 2.2 基于强化学习的任务性价比与契合度权重分配

融合任务性价比与契合度可以使智能体从任务角度和自身角度同时考虑,从而更加合理地选择任务. 假设智能体  $A$  收到  $p$  项任务,按照任务性价比从高到低排序得到列表  $E_p$ ,以及按照智能体和任务的契合度从高到低排序得到列表  $\theta_p$ . 两个列表分别以列表中最大的值为分母进行归一化处理,得到新的列表  $E_{\text{new}}$  和  $\theta_{\text{new}}$ . 则对于任意任务  $t_i \in T$  的选取值为

$$Ch_{t_i} = \omega_1(t_i) \cdot E_{\text{new}}(t_i) + \omega_2(t_i) \cdot \theta_{\text{new}}(t_i). \quad (3)$$

其中:  $E_{\text{new}}(t_i)$  和  $\theta_{\text{new}}(t_i)$  分别为任务  $t_i$  在列表  $E_{\text{new}}$  和  $\theta_{\text{new}}$  中的性价比值和契合度值;  $\omega_1(t_i)$  和  $\omega_2(t_i)$  分别为任务  $t_i$  关于  $E_{\text{new}}(t_i)$  和  $\theta_{\text{new}}(t_i)$  的权重,且  $\omega_1(t_i) + \omega_2(t_i) = 1$ . 权值  $\omega_1(t_i)$  和  $\omega_2(t_i)$  初始化均为 0.5.

Learning Automata(LA)算法是一种简单且性能良好的强化学习类型的学习算法. LA算法包含了个体的历史数据,个体根据其经验和回报进行下一步动作的选择. 这里,采用LA强化学习算法来调节 $\omega_1$ 、 $\omega_2$ ,使得智能体通过一步步的学习,令智能体的选择更加符合系统的特性. 采用的LA学习策略如下:

$$\begin{aligned}\omega_1(t_i) &= \omega_1(t_{i-1}) + \alpha(1 - \omega_1(t_{i-1})), \\ \omega_2(t_i) &= \omega_2(t_{i-1})(1 - \alpha),\end{aligned}\quad (4)$$

其中 $\alpha$ 是一个数值较小的学习因子. 每次联盟形成失败(即联盟内agent资源总和不能满足任务需求)时,智能体利用式(4)不断进行学习,动态调整权值 $\omega_1(t_i)$ 和 $\omega_2(t_i)$ . 通过不断学习,智能体对任务的选择重心逐渐趋向于任务性价比这个绝对指标,从而保证了算法最终收敛的效果. 联盟形成后 $\omega_1(t_i)$ 和 $\omega_2(t_i)$ 恢复初始值,即0.5.

### 2.3 回退机制中智能体的回退顺序

文献[14]中随机产生一个令牌,并且该令牌对应一个agent,判断该agent是否满足被删除条件,若满足条件,则agent退出联盟选择任务性价比次高级的任务. 然而,通过随机性删除agent不能保证agent删除顺序的合理性. 本文agent在提出任务申请的同时,将任务契合度信息一并提交给任务agent. 任务agent统计契合度降序列表中各agent能力资源,进行agent的删除.

假设该联盟执行的任务为 $t$ ,任务agent接收到含有任务契合度 $\theta$ 的申请信息. 任务agent将收到的契合度值按降序排列,得到列表 $L_Q$ ,并对列表 $L_Q$ 中agent的能力或资源依次累加. 当agent的资源满足要求,即

$$\sum_{i=1}^l r_i^k \geq r_t^k \quad (5)$$

时,任务agent便认为联盟形成完成,删除列表 $L_Q$ 中位置在 $l$ 以下的agent. 式(5)中: $l$ 为任务agent累加的成员agent的个数, $k$ 为资源的第 $k$ 项, $r_t^k$ 为完成任务 $t$ 所需第 $k$ 项的资源需求.

为了验证本文改进算法的有效性,设立3组仿真模拟实验,分别是任务需求低、任务需求一般以及任务需求高情况下的任务分配过程,对应地表示多智能体总能力充裕、一般、不足3种情况下的任务分配过程. 进行大量仿真实验,将实验结果与文献[14]的结果进行比较,观察任务完成情况,包括两种联盟形成策略的分配效果和对资源的利用率. 实验结果的衡量指标是系统总收益、时间成本开销、资源冗余以及任务完成率.

## 3 仿真实验及结果分析

### 3.1 仿真实验

设计火力打击任务,验证两种任务分配策略的完成情况,只需保证任务执行采用相同方法即可,这里只做简单的任务执行方法设计,不对战术进行重点研究. 首先,敌我双方坦克类型分为3类,简单定义为微型坦克、轻型坦克和中重型坦克. 坦克涉及火力打击任务的参数如表1所示,各参数范围为[0,1].

表1 智能体能力值

坦克	火力	防御
微型	0.3	0.4
轻型	0.5	0.5
中重型	0.6	0.5

这里定义3种共30个智能体,每种智能体的个数随机给出,能力值按表1中给出的数值赋值. 共进行3组实验,每组实验中进行100次分配任务. 将3个任务分配完表示一次任务分配,3个任务的任务收益如表2所示. 第1组实验100次分配中,3个任务的任务需求随机给出. 3组实验的任务需求按照0.6:1:1.5进行配置,分别模拟任务需求低、中、高3种情况. LA算法的学习因子取值0.2.

表2 3种任务的任务收益

任务	任务收益
1	0.6
2	0.5
3	0.4

#### 实验1 任务需求比例为0.6.

该组实验模拟任务需求低、多智能体总能力处于充裕状态的任务分配过程. 100次实验统计结果如表3、表4所示,统计数据包括100次实验中文献[14]

表3 文献[14]算法(实验1)

任务	完成任务	收益	火力总冗余	防御总冗余	平均火力冗余	平均防御冗余
1	100	60	726	227.6	—	—
2	100	50	147.2	816.4	—	—
3	100	40	1008	265.4	—	—
总和	300	150	1881.2	1309.4	6.270	4.364

注:任务分配平均消耗时间为0.2s.

表4 本文算法(实验1)

任务	完成任务	收益	火力总冗余	防御总冗余	平均火力冗余	平均防御冗余
1	100	60	478	278.6	—	—
2	100	50	347.2	413.4	—	—
3	100	40	501	438.4	—	—
总和	300	150	1326.2	1130.4	4.420	3.768

注:任务分配平均消耗时间0.19s.

算法和本文算法的总的任务收益、完成任务分配时间、平均火力冗余和平均防御冗余.其中冗余值的计算是任务联盟的能力减去任务需求值的差.

实验2 任务需求比例为1.

该组实验模拟任务需求一般、智能体总体能力相当的情况下任务分配过程.文献[14]算法和本文算法运行100次的实验数据分别如表5和表6所示.

表5 文献[14]算法(实验2)

Table with 7 columns: 任务, 完成任务, 收益, 火力总冗余, 防御总冗余, 平均火力冗余, 平均防御冗余. Rows include tasks 1, 2, 3 and a total row.

注:任务分配平均消耗时间0.2s.

表6 本文算法(实验2)

Table with 7 columns: 任务, 完成任务, 收益, 火力总冗余, 防御总冗余, 平均火力冗余, 平均防御冗余. Rows include tasks 1, 2, 3 and a total row.

注:任务分配平均消耗时间0.24s.

实验3 任务需求比例为1.5.

该组实验模拟任务需求高、智能体总体能力不足或匮乏情况下的任务分配过程.文献[14]算法和本文算法运行100次的实验数据分别如表7和表8所示.

表7 文献[14]算法(实验3)

Table with 7 columns: 任务, 完成任务, 收益, 火力总冗余, 防御总冗余, 平均火力冗余, 平均防御冗余. Rows include tasks 1, 2, 3 and a total row.

注:任务分配平均消耗时间0.17s.

表8 本文算法(实验3)

Table with 7 columns: 任务, 完成任务, 收益, 火力总冗余, 防御总冗余, 平均火力冗余, 平均防御冗余. Rows include tasks 1, 2, 3 and a total row.

注:任务分配平均消耗时间0.41s.

3.2 实验结果分析

首先从任务分配完成后的智能体能力资源利用率方面进行分析.由实验数据可知,文献[14]算法和本文算法在智能体能力处于能力资源充裕、能力资源一般、能力资源不足3种情况下,能力资源的冗余情况如表9所示.

表9 智能体能力资源冗余情况

Table with 5 columns: 实验, 平均火力冗余, 平均防御冗余, 文献[14]算法, 本文算法. Rows include experiments 1, 2, 3.

由表9可以看出,无论在哪种情况下,本文算法在能力资源节省方面都优秀,资源冗余减少了至少20%.说明当智能体选择任务时同时考虑自身的能力特点,有利于自身能力的充分利用,提高智能体能力资源的利用率,在智能体资源不足的情况下尤为重要.另外,智能体系统在任务需求低、中、高3种情形下的任务收益如表10所示.

表10 任务总收益情况

Table with 3 columns: 实验, 文献[14]算法, 本文算法. Rows include experiments 1, 2, 3.

由表10可以看出,两种方法在任务收益方面基本一致,可以得到本文算法在节省智能体能力的同时保证了任务收益的结果.

下面从完成任务分配的时间方面来分析.文献[14]算法和本文算法在智能体能力处于能力资源充裕、能力资源一般、能力资源不足3种情况下,平均每个任务消耗时间情况如表11所示.

表11 任务分配平均时间消耗情况

Table with 3 columns: 实验, 文献[14]算法消耗时间/s, 本文算法消耗时间/s. Rows include experiments 1, 2, 3.

由表11数据可以看出,随着智能体能力的不足,本文算法消耗的时间逐渐变长,这是因为本文算法在进行任务选择的过程中基本是并行选择任务,但当智能体能力不足时,智能体并行选择任务分散了智能体的能力值,使得智能体需要进行多轮的任务选择才能

最终完成任务的分配.同时,由于在LA算法中学习因子使用的是0.2,考虑可能是因为学习因子值太小,导致智能体任务选择的循环次数变多.这里给出一组补充实验,将实验3中的学习因子设为 $\alpha = 0.5$ ,并将获取的实验结果与实验3进行比较,得到表12.

表12 不同 $\alpha$ 的对比实验结果

指标	$\alpha$	
	0.2	0.5
完成任务个数	173	172
任务总收益	95.6	95.1
任务平均时间/s	0.312	0.190
平均火力冗余	6.127	6.075
平均防御冗余	4.289	4.526

由表12的对比结果可以看出,学习因子的增大,加快了任务分配的时间,这是因为学习因子的增大,加速了本文算法向基于任务收益算法的收敛速度.在时间消耗降低的情况下,其余各项指标基本能保持 $\alpha = 0.2$ 时的水平.

## 4 结论

本文提出了融合任务收益和智能体能力的智能体准备度概念算法,并进行了仿真实验.实验结果表明该算法不仅能保证任务收益,而且有利于提高智能体能力资源的利用率. Learning Automata算法的引入保证了算法的收敛性,即极限情况下本文算法将收敛到文献[14]的算法,保证了算法的稳定性.同时,可根据有任务时间约束的某些情况增大学习因子,加快任务分配时间.至于如何进行学习因子的自动调节,是本文接下来将要研究的内容.

### 参考文献(References)

- [1] Arib S, Aknine S. A plan based coalition formation model for multi-agent systems[C]. Proc of the 2011 IEEE/WIC/ACM Int Conf on Web Intelligence and Intelligent Agent Technology. IEEE Computer Society, 2011: 365-368.
- [2] 田敬北, 蒋建国, 张国富, 等. 基于云模型的agent联盟评价[J]. 控制与决策, 2013, 28(1): 152-156. (Tian J B, Jiang J G, Zhang G F, et al. Evaluation of agent coalition based on cloud model[J]. Control and Decision, 2013, 28(1): 152-156.)
- [3] El-Ashmawi W, Jun H, Renfa L. Stability coalition formation with cost sharing in multi-agent systems based on volume discount[J]. Int Arab J of Information Technology (IAJIT), 2015, 12(3): 296-303.
- [4] Souidi M, Piao S, Li G, et al. Coalition formation algorithm based on organization and Markov decision process for multi-player pursuit evasion[J]. Multiagent and Grid Systems, 2015, 11(1): 1-13.
- [5] Ghazikhani A, Mashadi H R, Monsefi R. A novel algorithm for coalition formation in multi-agent systems using cooperative game theory[C]. The 18th Iranian Conf on Electrical Engineering (ICEE). Isfahan: IEEE, 2010: 512-516.
- [6] Contreras J, Wu F F. A kernel-oriented algorithm for transmission expansion planning[J]. IEEE Trans on Power Systems, 2000, 15(4): 1434-1440.
- [7] Sandhlo T W, Lesser V R T. Coalitions among computationally bounded agents[J]. Artificial Intelligence, 1997, 94(1): 99-137.
- [8] Contreras J, Klusch M, Yen J. Multi-agent coalition formation in power transmission planning: A bilateral shapley value approach[C]. Artificial Intelligence Planning Systems. Menlo Park, 1998: 19-26.
- [9] Choi H L, Brunet L, How J P. Consensus-based decentralized auctions for robust task allocation[J]. IEEE Trans on Robotics, 2009, 25(4): 912-926.
- [10] Haque M, Egerstedt M. Coalition formation in multi-agent systems based on bottleneck dolphin alliances[C]. American Control Conf. St Louis: IEEE, 2009: 3280-3285.
- [11] Haque M, Rahmani A, Egerstedt M. Biologically inspired coalition formation of multi-agent systems[C]. Proc of the 9th Int Conf on Autonomous Agents and Multiagent Systems: Int Foundation for Autonomous Agents and Multiagent Systems. Toronto, 2010: 1427-1428.
- [12] Haque M, Egerstedt M, Rahmani A. Multilevel coalition formation strategy for suppression of enemy air defenses missions[J]. J of Aerospace Information Systems, 2013, 10(6): 287-296.
- [13] Shehory O, Kraus S. Methods for task allocation via agent coalition formation[J]. Artificial Intelligence, 1998, 101(1): 165-200.
- [14] 尹翔, 李斌, 于萌. 一种多agent网络中的分布式联盟形成算法[J]. 控制与决策, 2015, 30(3): 536-540. (Yin X, Li B, Yu M. Distributed coalition formation algorithm in multi-agent network[J]. Control and Decision, 2015, 30(3): 536-540.)

(责任编辑: 李君玲)