

一种多 agent 网络中的分布式联盟形成算法

尹翔¹, 李斌¹, 于萌^{1,2}

(1. 扬州大学 信息工程学院, 江苏 扬州 225127; 2. 扬州广播电视传媒集团, 江苏 扬州 225009)

摘要: 针对多 agent 系统的具体网络结构, 提出一种分布式联盟形成算法. 首先, 每个 agent 向自己邻居任务中“信价比”最高的任务提出加入申请, 形成潜在联盟; 当联盟资源大于对应任务的资源需求时, 使相应 agent 退出当前联盟, 转而选择“信价比”次高的任务, 从而提高联盟形成的可能性. 仿真实验结果表明, 与其他方法相比, 所提出的算法能大幅提高系统效率和收益.

关键词: 多 agent 系统; agent 网络; 联盟形成

中图分类号: TP18

文献标志码: A

Distributed coalition formation algorithm in multi-agent network

YIN Xiang¹, LI Bin¹, YU Meng^{1,2}

(1. School of Information Engineering, Yangzhou University, Yangzhou 225127, China; 2. Yangzhou Broadcast & Television Media Group, Yangzhou 225009, China. Correspondent: YIN Xiang, E-mail: yinxiang@yzu.edu.cn)

Abstract: A distributed coalition formation algorithm is presented for the network structure in multi-agent systems. For agents, the task with the highest performance/cost ratio is firstly selected, which results in the potential coalition. If the resources of the coalition are greater than the required resources of the task, some agents quit to opt for tasks with the second highest performance/cost ratio, so that the possibility for coalition formation can be raised. The simulation results show that, compared with other methods, the proposed algorithm can highly improve the efficiency and utility of the systems.

Keywords: multi-agent system; agent network; coalition formation

0 引言

在面向任务的多 agent 系统(MAS)中, 通过完成任务, agent 可以获得收益. 单个 agent 由于资源和信息受限, 往往只能完成简单的任务; 而 agent 通过相互间的合作, 可实现信息与资源的共享, 从而提高求解问题的效率, 甚至能够完成单个 agent 不能完成的任务. 联盟作为 agent 协作的一种重要形式, 受到了学术界的广泛关注^[1-2].

MAS 是一个自组织、开放的分布式系统, 其中每个 agent 具有自主决策的能力. 为了提高收益, 资源受限的 agent 在遇到任务时, 会选择同系统中的其他 agent 进行交互和合作, 从而构成一个团队来共同求解这一任务, 这个团队就称之为联盟^[3]. 部分学者借用博弈论中 Shapley 值、核、核心等概念研究联盟内部收益分配问题, 通过合理的收益分配保证联盟的稳定性^[4-6]. 但在一般情况下, 这些解的计算都是 NP 完全问题, 与系统中 agent 的数目成指数关系, 计算量

过大; 同时, 基于博弈论的方法没有给出联盟形成的具体策略; 因此, 此类方法大都限于理论研究, 在实际系统中难以实现.

另一类方法从全局角度出发, 重点研究 MAS 中的任务分配问题. 文献[7]基于集合覆盖和集合分割理论, 设计了一种贪婪算法以求解该问题, 并对解的界进行了分析; 文献[8]发展和完善了 Shehory 的研究, 给出了该问题的理论结果, 同时在 agent 类型受限和联盟规模受限两个前提下得到次优解; 文献[9]基于快速最大和 (fast-max-sum) 算法研究任务分配问题, 并采用在线裁剪的方法缩减解的空间, 使得该问题能够在相对较大的规模内得到求解; 文献[10]研究了时间和空间受限情况下的联盟形成问题, 并设计了若干启发式规则用于求解该问题; 文献[11-12]探讨了在不同环境下的联盟结构生成问题, 证明了在保证解质量的前提下所需要的最小计算量.

上述研究主要集中于如何在 MAS 中形成最优、

收稿日期: 2013-12-22; 修回日期: 2014-04-02.

基金项目: 国家自然科学基金项目(61070133); 江苏省高校自然科学基金项目(11KJD520011).

作者简介: 尹翔(1980—), 男, 讲师, 博士, 从事分布式人工智能、多 agent 系统的研究; 李斌(1965—), 男, 教授, 博士生导师, 从事智能 agent 及应用、知识工程等研究.

稳定的联盟,并没有对MAS的拓扑结构给出具体限制,或者在通常情况下,这些方法都假设整个网络是全连通的,即一个agent可以同其余任意agent进行通信和合作.但是在实际环境中,尤其是大规模分布式网络中,这种假设是难以满足的.例如,在传感器网络中,由于单个传感器节点能力的限制(计算资源、通信能力和电源供给等),每个节点只能根据具体的网络结构,选出部分agent作为自己的潜在合作者.针对这类问题,近年来部分学者已经展开研究.文献[13]对两种网络拓扑结构(星型和环形)下的联盟形成进行了理论分析,并通过实验验证了网络结构对联盟的形成确实具有很大的影响;但其工作重点在于研究网络拓扑结构对联盟形成的影响,所给出的联盟形成机制过于简单,难以形成高效的联盟.文献[14]研究了分布式网络中的联盟问题,给出了一种基于协商的联盟形成策略,但由于协商本身具有复杂度高,结果不确定等问题,难以应用于实际系统.文献[15]研究了agent社会网络中的任务分配问题,给出了一种基于贪婪算法的分布式联盟形成策略,该策略结构简单,较容易实现,但解的质量难以保证.

基于上述背景,本文研究了当agent处于一个网络结构中时,MAS中的联盟问题,并提出了一种分布式联盟形成算法.首先,根据网络的具体结构,每个agent确定自己可能的合作者;然后,agent选择加入“性价比”最高的任务,形成潜在联盟.当潜在联盟所拥有的资源大于对应任务的资源需求时,部分agent退出当前联盟,转而选择次优任务.通过“退出-再选择”机制的设计,可有效避免当多个agent选择同一任务时产生的资源浪费,从而提高系统收益.实验结果表明,同其他方法相比,本文所提出的算法可以提高联盟形成效率,增加系统收益.

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 表示系统中资源的种类, $r_{A_i}^l \geq 0$ ($1 \leq l \leq k$) 表示agent A_i 拥有第 l 种资源的数量. $T = \{t_1, t_2, \dots, t_m\}$ 为系统中待求解的任务集,任务 $t_j = \langle \text{loc}(t_j), R_{t_j} \rangle$. 其中: $\text{loc}(t_j) \in N$ 表示任务 t_j 所在位置, $R_{t_j} = \langle r_{t_j}^1, r_{t_j}^2, \dots, r_{t_j}^k \rangle$ 表示任务 t_j 的资源需求. 本文主要研究agent网络中的联盟形成问题,下面给出相关定义.

定义1 定义联盟 C 为一个四元组 $C = \langle N_C, R_C, t_C, P(t_C) \rangle$. 其中: $N_C \subseteq N$; $R_C = \langle r_C^1, r_C^2, \dots, r_C^k \rangle$ 为联盟 C 所拥有的资源,等于联盟中各成员agent所拥有资源的总和,即 $r_C^l = \sum_{A_i \in N_C} r_{A_i}^l$; t_C 为联盟所

对应的任务; $P(t_C)$ 表示联盟值,为联盟完成对应任务能够获得的收益. 定义 Ω 为所有可行联盟集.

定义2 定义agent网络(AN)为一组具有“相互关系Re”的agent集合,可以用一个函数 $f: N \times N \rightarrow \{0, 1\}$ 表示关系Re,若 $f(A_i, A_j) = 1$,则agent A_i 与 A_j 间有关系Re,否则没有关系Re.

由以上定义可以看出,agent网络可由无向图来描述 $AN = (N, E)$. 其中: N 表示顶点,即agent的集合; E 表示边,即agent间的关系Re. 若 $(A_i, A_j) \in E$,则agent A_i 与 A_j 有关系Re. 很显然, $\forall A_i: (A_i, A_i) \in E, \forall A_i, A_j: (A_i, A_j) \in E \Rightarrow (A_j, A_i) \in E$. 为了不失一般性,本文用“邻居关系”代表Re.

定义3 对于 $\forall A_i \in N$, 将 A_i 的可达任务集 $T_i \subseteq T$ 定义为与 A_i 有关系Re的agent A_j 上的任务集,即 $T_i = \{t_{i1}, t_{i2}, \dots | f(A_i, A_j) = 1 \cap \text{loc}(t_{ij}) = A_j\}$. 其中: t_{ij} 表示 T_i 中的第 j 个任务, $\text{loc}(t_{ij}) \in N$ 表示 t_{ij} 所在的位置.

定义4 给定agent集合 N . 任务集 T 和agent网络AN, AN中的联盟形成(CF-AN)问题可以定义为一个映射 $o: \Omega \times T \rightarrow \mathbf{R}$, 满足以下条件.

1) 对于任务 t_j , 只有与 $\text{loc}(t_j)$ 有“相互关系Re”的agent才能形成联盟,即对于 $\forall A_i \in C_j$, 有 $f(A_i, \text{loc}(t_j)) = 1$.

2) 只有当联盟所拥有的资源大于对应任务的资源需求时,任务才能被完成,获得相应收益,即当且仅当 $R_C \geq R_{t_C} (\forall l, r_C^l \geq r_{t_C}^l)$ 时,任务 t_C 能够被完成,得到收益 $P(t_C)$.

由此可见,agent网络中的联盟形成与经典的联盟形成问题的最大不同在于:由于考虑了MAS的拓扑结构,使得个体agent不能够将系统中的任意agent作为自己的潜在合作者;同时,agent也不能选择加入任意任务,只能加入可达任务集中的任务.基于此,传统的联盟形成策略^[10-12]难以求解该问题.

2 Agent网络中的分布式联盟形成算法

2.1 数学模型

在agent网络的联盟形成问题(CF-AN)中,对于任务 t_j , 设定二元变量 $\delta_j \in \{0, 1\}$ 表示任务 t_j 是否被完成. 若 $\delta_j = 1$, 则 t_j 能被完成,系统可获得收益 $P(t_j)$, 否则不能获得收益. 对于agent A_i , 它有二元决策变量 $x_{ij} \in \{0, 1\}$, 如果 $x_{ij} = 1$, 则表示agent A_i 加入对应任务 t_j 的联盟 C_j , 否则不加入. 本文假设一个agent只能加入一个联盟,即 $\bigcap_j C_j = \emptyset$, 因此联盟 C 所拥有的资源为其成员agent所拥有资源之和. 对于CF-AN, 其目标是在满足agent网络结构的条件下,求解agent的决策变量 x_{ij} , 最大化系统总收益

$\sum_j P(t_j)$, 其数学模型描述如下:

$$\arg \max_{x_{ij} \in \{0,1\}} \sum_{t_j \in T} P(t_j) \cdot \delta_j, \quad (1)$$

其中

$$\delta_j = \begin{cases} 1, & \sum_{x_{ij}=1 \cap f(A_i, \text{loc}(t_j))=1} r_{A_i}^l \geq r_{t_j}^l, 1 \leq l \leq k; \\ 0, & \text{otherwise.} \end{cases}$$

2.2 联盟形成算法

由模型(1)可以看出,对于CF-AN问题,即使在agent数目和任务数不太大的情况下,也很难获得最优解.对此,文献[15]提出了一种基于贪心算法的联盟形成策略,其核心思想是每个agent都选择邻居任务中“性价比”最高的加入,从而求得次优解;但该算法在求解问题时,不从整体上考虑,而仅仅做出在当前看来是最好的选择.由此造成的后果是,那些“性价比”高的任务由于有过多的agent申请加入,对应联盟的资源远大于任务的需求,从而其中大部分agent会被拒绝;与此相反,“性价比”低的任务由于很少有agent申请加入,对应联盟的资源难以大于任务的需求,因而无法被完成.

由以上分析可以看出,采用贪心算法求解CF-AN问题,会造成系统资源的大量浪费.针对这个问题,本文设计了一种基于回退的分布式联盟形成策略,使得agent在选择联盟时能够更加“理性”,从而提高系统资源利用率.任务所在agent(管理者)向其邻居agent(契约者)发布任务相关信息;邻居agent收到信息后,按照任务的“性价比”对其进行排序,并告诉其管理者相关信息;管理者将信息转发给邻居agent.契约者向“性价比”最高的任务发送加入请求.管理者根据一定的原则选择部分agent形成联盟,其他agent进行回退,重新选择任务.在给出具体算法之前,首先定义任务“性价比”.

定义 5 (任务性价比) 对于任务 $t_j \in T$, 其性价比定义为完成 t_j 所能获得的收益与 t_j 资源需求总和的比值, 即 $e(t_j) = P(t_j) / \sum_{l=1}^k r_{t_j}^l$.

算法的具体步骤如下.

Step 1: 初始化. 任务管理 agent 向自己的邻居 agent 发送信息, 表示自己有任务需要完成; 邻居 agent 收到信息后, 将自己所拥有资源的种类与数量发送给管理者; 管理者将所有邻居 agent 的信息汇总后, 形成列表 N_{neig} , 并将此列表发送给每个邻居 agent.

Step 2: 任务管理者 agent 计算自己所管理任务

的“性价比”, 按“性价比”从高到低对任务进行排序形成任务列表 L_{mag} , 并将 L_{mag} 发送给邻居 agent.

Step 3: 每个 agent 在收到 L_{mag} 后, 对 L_{mag} 中的所有任务按“性价比”进行重新排序, 构成 L_{ctr} , L_{ctr} 包含了契约者 agent 对所有邻居任务的选择顺序. 注意, 因为一个 agent 可能和多个任务管理者为邻居, 所以可以收到多个 L_{mag} .

Step 4: 契约者 agent 将 L_{ctr} 发送给所有和自己相邻的任务管理者.

Step 5: 任务管理者 agent 将所收到的 L_{ctr} 转发给除发送者以外的其余邻居 agent, 这样每个 agent 便有了自己可以加入的潜在联盟的信息.

Step 6: 契约者 agent 将潜在联盟中成员的资源相加, 得到每个潜在联盟所拥有的总资源. 将联盟资源与任务资源需求相比较, 选择其中能完成的任务, 按“性价比”由高到低排序, 并向“性价比”最高的任务的管理者发送加入请求. 同时, 将能完成任务的潜在联盟的资源以及对应任务的资源需求发送给管理者.

Step 7: 管理者 agent 在收到加入请求后, 执行以下操作. 随机产生一个令牌, 该令牌对应一个契约者 agent, 若该 agent 满足以下两个条件, 则将令牌发送给该 agent, 同时将该 agent 从当前联盟中删除, 否则重新选择令牌.

条件 1 将该 agent 的资源从当前联盟资源中减去, 联盟剩余资源仍大于任务的资源需求;

条件 2 检查该 agent 的潜在联盟列表, 选择排在当前任务的下一个任务, 从其对应联盟资源中减去该 agent 的资源, 联盟剩余资源小于任务的资源需求.

Step 8: 收到令牌的 agent 进行回退, 按照潜在联盟中任务的顺序, 选择下一个任务, 向对应的任务管理者发送加入申请.

Step 9: 管理者 agent 重复执行 Step 7, 若剩余的任意 agent 都不能同时满足条件 1 和条件 2, 则检测是否有 agent 满足条件 1. 若存在, 则删除该 agent, 直到删除任意一个 agent, 联盟的剩余资源都无法完成任务. 此时管理者向剩余的 agent 发送确认消息, 形成联盟.

3 仿真实验与分析

为了验证本文算法的有效性, 下面进行一组仿真实验, 并将结果和文献[15]中的方法相比较. 在这里, 主要考虑在不同参数下, 两种联盟形成策略对系统资源利用率的影响, 其衡量指标为系统总收益. 实验环境和参数的设置如下.

设 MAS 中共有 n 个 agent 以及 m 个待求解的任

务, 每个 agent A_i 拥有一定资源, 资源的种类 $k = 5$, 每种资源的数量为 $[0, 50]$ 范围内的随机数, 若 $r_{A_i}^l = 0, 0 \leq l \leq k$, 则表示 A_i 没有类型为 l 的资源. 同样, 完成任务 t_j 需要一定资源, 种类也为 5. 完成任务 t_j 可获得收益 $P(t_j)$, 设 $P(t_j) = \alpha_j \cdot \sum_{l=1}^k r_{t_j}^l$, 其中 $\alpha_j \in (0, 3)$ 为一随机数. 为了模拟实际环境中任务出现的不确定性, 将 m 个任务随机分配到 agent 中 (某个 agent 可能是多个任务的管理者). 本文采用随机网络作为 MAS 的拓扑结构, 产生的随机网络是一个连通网络, 即任意两个 agent 间是连通的; 同时, 可通过控制“邻居”节点的数目来调整个网络的“度”. 在每个环境下, 两种算法各执行 50 次, 取平均值作为结果. 实验结果用置信区间为 95% 的误差条形图来表示. 在实验中, 用 G-CFAN 表示文献 [15] 的结果, 用 B-CFAN 表示本文算法的结果.

实验 1 考虑网络结构对两种算法性能的影响, 在这里网络结构主要是指网络的平均度, 即每个 agent 所具有的邻居节点的平均数. 实验中取度的变化范围为 $[4, 20]$, agent 数 $n = 60$, 任务数 $m = 40$. 在每次仿真中, agent 的资源为 $[0, 50]$ 内的随机数, 任务的资源需求为 $[35, 85]$ 内的随机数, 实验结果如图 1 所示. 由图 1 可以看出, 随着 agent 网络度的增加, 两种算法所获得的收益都会增加. 任务管理者 agent 度的增加, 意味着其有更多的邻居节点, 即可有更多的 agent 参与完成相应的任务, 因此每个任务被完成的概率会增加, 这一点很容易理解. 但是, 无论网络的度为多少, 利用本文算法所得到的系统收益都远高于文献 [15] 中的方法, 因此在不同网络结构下, 本文算法更具优越性.

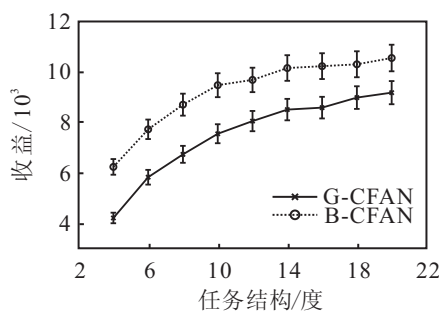


图 1 不同网络结构下系统收益比较

实验 2 考察系统资源对算法的影响, 这里系统资源包括每个 agent 所拥有的资源和每个任务所需求的资源, 用资源系数来表示, 定义为任务需求的资源总数与 agent 拥有资源总数的比值, 实验中取其变化范围为 $[1.2, 2]$. 网络的度固定为 10, agent 数 $n = 60$, 任务数 $m = 40$, 每个任务所需求的资源随机产生, 其总和等于 agent 的总资源乘以资源系数, 每个

任务的收益固定为 100, 结果如图 2 所示. 从平均意义上看, 随着资源系数的增加, 完成一个任务需要更多的 agent; 因此, 在网络结构 (度) 不变的前提下, 由于系统中能完成的任务更少, 总收益会下降. 与实验 1 相同, 在资源系数整个变化范围内, 本文算法所获得的收益都远高于文献 [15] 中算法所获得的收益, 因此在不同系统资源的情况下, 本文算法也更具优越性.

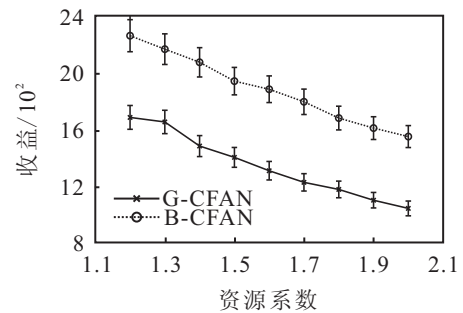


图 2 不同资源系数下系统收益比较

实验 3 研究系统中 agent 数目和任务数目对两种算法性能的影响. 两个实验中, 网络的度固定为 10, 资源系数固定为 1.4; 在研究 agent 数目对算法性能的影响时, 任务数为 30; 在研究任务数目对算法性能的影响时, agent 数为 100, 结果如图 3 和图 4 所示. 从图 3 中可以看出, 随着系统中 agent 数目的增加, 系统收益逐渐增加; 但是这种增长是比较缓慢的, 这是由于网络的度没有改变, 即网络中每个 agent 的平均邻居数不变, 能够为任务管理者提供资源的契约者平均数目没有发生变化; 因此虽然 agent 总数增加了, 但对系统总收益的影响有限, 这也反应了网络结构对联盟形成的重要影响. 由图 4 可见, 系统的收益随着任务数的增加而增加. 这可以解释为若新加入任务的“性价比”较高并能够被完成, 则 agent 会优先选择加入新的任务, 从而可提高资源的利用效率, 并增加系统收益. 与图 3 相似, 受限于 agent 数目和网络结构, 这种增长是缓慢的. 由此可以看出, 无论是 agent 数目还是任务数目的增加, 本文算法所获得的系统收益都大于文献 [15] 中算法所获得的收益, 因此在这两种环境下, 本文算法更具优越性.

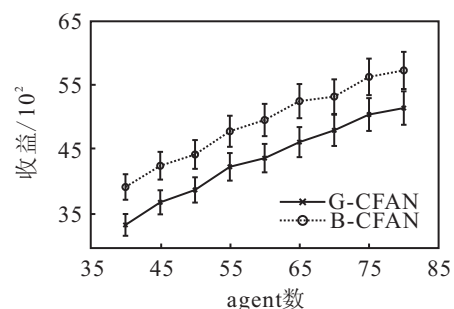


图 3 不同 agent 数目下系统收益比较

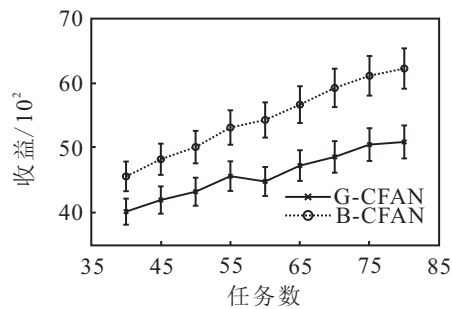


图4 不同任务数目下系统收益比较

4 结 论

Agent网络中的联盟形成问题是近年来兴起的一个问题,在电子商务、虚拟企业和传感器网络等领域有广泛应用.本文提出了一种基于“回退”思想的分布式联盟形成策略,agent首先选择“性价比”最高的任务组成联盟,当联盟资源大于任务需求时,部分agent退出该联盟,转而选择次优任务,这样可有效避免系统资源的浪费,提高总收益.仿真实验表明,与相关算法比较,本文算法具有更好的性能.下一步工作的重点将放在当一个agent可以加入多个联盟时,如何分配agent的资源以获得最大的收益.

参考文献(References)

- [1] Shehory O, Kraus S. Feasible formation of coalitions among autonomous agents in nonsuperadditive environments[J]. *Computational Intelligence*, 1999, 15(5): 218-251.
- [2] 程柏良, 曾国荪, 揭安全. 基于自组织演化的多agent可信联盟研究[J]. *计算机研究与发展*. 2010, 47(8): 1382-1391.
(Cheng B L, Zeng G S, Jie A Q. Study of multi-agent trust coalition based on self-organization evolution[J]. *J of Computer Research and Development*, 2010, 47(8): 1382-1391.)
- [3] 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.)
- [4] Sandholm T W, Lesser V R. Coalitions among computationally bounded agents[J]. *Artificial Intelligence*, 1997, 94(1-2): 99-137.
- [5] Blankenburg B, Dash R K, Ramchurn S D, et al. Trusted kernel-based coalition formation[C]. *Proc of 4th Int Joint Conf on Autonomous Agents and Multi-agent Systems*. Utrecht, 2005: 989-996.
- [6] Vytelingum P, Voice T D, Ramchurn S D, et al. Agent-based micro-storage management for the smart grid[C]. *Proc of 9th Int Joint Conf on Autonomous Agents and Multi-agent Systems*. Toronto, 2010: 39-46.
- [7] Shehory O, Kraus S. Methods for task allocation via agent coalition formation[J]. *Artificial Intelligence*, 1998, 101(12): 165-200.
- [8] Service T C, Adams J A. Coalition formation for task allocation: Theory and algorithms[J]. *Autonomous Agents and Multi-Agent Systems*, 2011, 22(2): 225-248.
- [9] Macarthur K S, Stranders R, Ramchurn S D, et al. A distributed anytime algorithm for dynamic task allocation in multi-agent systems[C]. *Proc of 25th Conf on Artificial Intelligence*. San Francisco: ACM, 2011: 701-706.
- [10] Ramchurn S D, Polukarov M, Farinelli A, et al. Coalition formation with spatial and temporal constraints[C]. *Proc of AAMAS*. Toronto, 2010: 1181-1188.
- [11] Rahwan T, Michalak T, Wooldridge M, et al. Anytime coalition structure generation in multi-agent systems with positive and negative externalities[J]. *Artificial Intelligence*, 2012, 186(1): 95-122.
- [12] Voice T, Polukarov, Jennings N R. Coalition structure generation over graphs[J]. *J of Artificial Intelligence Research*, 2012, 45(1): 165-196.
- [13] Gaston M E, Desjardins M. The effect of network structure on dynamic team formation in multi-agent systems[J]. *Computational Intelligence*, 2008, 24(2): 123-157.
- [14] Ye D Y, Zhang M J, Sutanto D. Self-adaptation-based dynamic coalition formation in a distributed agent network: a mechanism and a brief survey[J]. *IEEE Trans on Parallel and Distributed Systems*, 2013, 24(5): 1042-1051.
- [15] Weerd M M, Zhang Y Q, Klos T. Multiagent task allocation in social network[J]. *Autonomous Agents and Multi-agent Systems*, 2012, 25(1): 46-86.

(责任编辑: 滕 蓉)