

基于适应度的多 Agent 层次任务分配方法

陶雪丽, 郑延斌

(河南师范大学 计算机与信息工程学院, 河南 新乡 453007)

摘 要: 为了解决动态环境中的任务分配问题, 提出了一种层次任务分配方法, 利用基于适应度的全局分配方法为 Agent 分配合适的任务, 当环境发生变换时, 通过基于能力和偏好的局部调整来解决任务和 Agent 之间的匹配问题, 使得每个 Agent 能够根据局部信息选择理想的任务来执行, 提高了分配算法的鲁棒性和多 Agent 整体效用. 仿真实验表明该方法是可行且有效的, 能够解决动态环境中的任务分配问题.

关键词: 多 Agent; 任务分配; 适应度; 能力; 调整

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2013)06-0113-04

A Leveled Task Allocation Method Based on Fitness for Multiple Agents

TAO Xue-li¹, ZHENG Yan-bin¹

(Department of Computer and Information Engineering, Henna Normal University, Xinxiang 453007, China)

Abstract: In order to solve the task allocation in dynamic environment, proposed a level task allocation method is proposed. Appropriate tasks are allocated to agents by using the global allocation method based on fitness. When ta environment changes, the matching issues among tasks and agents are solved through local adjustment based on ability and preferences, which makes each agent be able to choose desired task to be performed according to local information. So the robustness and multi-agent utility of the allocation algorithm are improved. Simulation results show that the method is feasible and effective to solve the task allocation problem in a dynamic environment.

Key words: multiple agent; task alloction; fitness; capability; adjustment

1 引言

任务分配是 MAS(Multi-Agent Systems)研究的热点问题之一, 研究者提出了许多任务分配方法^[1], 这些方法大体上分为两类: 集中式任务分配和分布式任务分配^[2-3]. 集中式方法简单, 可获得完全最优解, 但计算量大、动态性较差; 分布式方法灵活性高、适应性强, 但是系统结构复杂, 难以获得全局最优方案, 通常得到的是局部最优解; 随着任务和 Agent 数量的增加, 会出现通信量巨大、决策效率低等问题. 因此对于动态环境中任务及 Agent 的变化问题, 现有的分配方法很难适应. Kathryn^[4]给出了一种动态任务加入时的任务分配问题, Chapman^[5]等利用对策论解决灾难环境中的动态变化的任务分

配问题, 且假定 Agent 具有相同的、不变的能力, 他们的方法仅仅考虑任务的动态变化, 没有考虑 Agent 动态加入和退出对任务分配的影响. 李湘清^[6]等提出了一种分组模型用于解决动态环境中的任务分配问题, 通过动态分组来解决任务的动态调整问题, 每个小组成员需要频繁通信, 增加了通信量. 刘利枚等^[7]在结合集中式和分布式分配机制的基础上, 提出一种基于熟人联盟的结构化多 Agent 的任务分配机制, 该机制中每个 Agent 的能力值是由问题专家 Agent 控制的, 这不符合 Agent 自治的特性, 不符合成员特性随动态环境变化的要求.

为了能够把集中式和分布式两类方法结合起来, 提出一种层次任务分配方法, 来解决动态环境中任务的分配问题.

收稿日期: 2012-09-08; 修回日期: 2012-11-01

基金项目: 河南省重点科技攻关项目 (102102210179, 102102210176, 122102210086); 河南省教育厅自然基金项目 (2011B520022, 2011A520026, 2010A520027)

2 相关概念

每个任务都有一定需求, Agent 要完成这些任务必须满足任务需求, 为方便起见, 这里引用文献[8]中关于 Agent、任务的描述方法. 其中 $C = \langle c_1, c_2, \dots, c_s \rangle$ 为环境中基本的能力向量, $0 \leq \alpha_{ij}(t) \leq 1$, $0 \leq \beta_{ij}(t) \leq 1$, 分别表示 Agent 在 t 时刻所拥有的能力因子, t 时刻任务所需要的能力因子. $A = \{a_1, a_2, \dots, a_M\}$ 为 Agent 集合, 任务集合为 $W = \{w_1, w_2, \dots, w_N\}$.

定义 1 Agent 的能力, $C_i^a(t) = [\alpha_{i1}(t)c_1, \alpha_{i2}(t)c_2, \dots, \alpha_{is}(t)c_s]^T$, 表示 t 时刻 Agent a_i 所具有的能力, $\alpha_{ik}(t) \geq 0$ 对应于 Agent a_i 能力 c_k 的强弱程度. $\alpha_{ik}(t) = 0$, 表明 Agent a_i 不具备能力 c_k .

定义 2 任务需求, 完成任务所需要的能力, $C_j^w(t) = [\beta_{j1}(t)c_1, \beta_{j2}(t)c_2, \dots, \beta_{js}(t)c_s]^T$, 表示任务 w_j 在 t 时刻的能力需求, $\beta_{jk}(t) \geq 0$ 对应于任务 w_j 对能力 c_k 的强弱程度, 如果完成任务 w_j 不需要能力 c_k , 则相应的 $\beta_{jk}(t) = 0$.

定义 3 Agent 完成任务需要付出的代价:

$$\text{cost: resource} \rightarrow R^+ \cup \{0\}$$

环境的动态变化会导致完成任务的代价发生变化, 故代价函数是一个随时间变化的量, 记 Agent a_i 完成任务 w_j 的代价为: $\text{cost}(a_i, w_j)(t)$, 若 Agent 为空闲状态, 则 $\text{cost}(a_i, \Phi)(t) = 0$. 根据实际的实验环境, 代价函数可以是一个随时间变化的连续函数, 也可以是一个离散函数, 或者由专家组给出的经验值.

定义 4 任务报酬, 指 Agent 完成任务后获得的报酬. Agent a_i 完成任务 w_j 后得到的报酬为: $r(a_i, w_j) = U_{w_j} - \text{cost}(a_i, w_j)(t)$, 其中 U_{w_j} 表示成功完成任务 w_j 后所获得的奖赏. 若 Agent 没有承担任务, 则 $r(a_i, \Phi) = 0$.

定义 5 Agent 对任务的期望度, 表示 Agent 对任务的“偏好”. 在不同的时刻, 系统环境不同, Agent 对任务的偏好也会改变, 因此 Agent a_i 对任务 w_j 的期望度记为: $E_i^j(t) \in R^+ \cup \{0\}$, 满足 $0 \leq E_i^j \leq 1$. 当 $E_i^j = 0$ 时, 表示 Agent a_i 对任务 w_j 没有兴趣, 不会选择承担该任务.

基于以上的定义, M 个 Agent, N 个任务的分配问题可以描述如下:

(1) 任务分配行为描述

$$P_{ij} = \begin{cases} 1 & \text{任务 } w_j \text{ 分配给 Agent } a_i \\ 0 & \text{任务 } w_j \text{ 未分配给 Agent } a_i \end{cases} \quad (1)$$

(2) Agent 的效用函数, $u_i: S \rightarrow R$, 代表 agent 对

分配结果的满意度. $S = \{s_1, s_2, \dots, s_l\}$ 是任务分配结果的集合. 如果对于 Agent a_i , 如果两个分配结果 s_1 和 s_2 有如下关系: $u_i(s_1) \geq u_i(s_2)$, 则对 Agent a_i , 分配 s_1 不劣于分配 s_2 .

此处, 若任务 w_j 分配给了 Agent a_i , 则该 agent 对任务分配结果的满意度, 用下式衡量:

$$u(a_i, w_j) = \left(1 - \frac{\text{cost}(a_i, w_j)(t)}{u_{w_j}}\right) \times E_i^j(t) \quad (2)$$

式中, U_{w_j} 为 Agent a_i 完成任务 w_j 后得到的报酬.

(3) Agent 对任务的适应度, 表示任务与 Agent 的匹配程度. 为了描述任务与 Agent 间的匹配问题, 这里引入适应度的概念. 该算法, 将 t 时刻, Agent 对任务的适应度记为

$$M_i^j(t) = \frac{E_i^j(t)}{\text{cost}(a_i, w_j)(t)} \quad (3)$$

该值越大, 表示 Agent a_i 越适合承担任务 w_j .

(4) 全局效用函数, 表示任务分配后, 系统的总效用, 记为 $U: S \rightarrow R$

$$U = \sum_{i=1}^M \sum_{j=1}^N u(a_i, w_j) \quad (4)$$

多 Agent 任务分配的最优分配问题可以形式化为如下的线性规划问题:

$$U_{\max} = \text{MAX} \left(\sum_{i=1}^M \sum_{j=1}^N P_{ij} u(a_i, w_j) \right) \quad (5)$$

使得: $\sum_{i=1}^M P_{ij} = 1, j = 1, 2, \dots, N, \sum_{j=1}^N P_{ij} = 1, i = 1, 2, \dots, M.$

3 层次任务分配方法

动态环境中任务需求或者新任务的增加及 Agent 本身属性的变化, 导致 Agent 对目前承担的任务的适应度发生变化, 甚至不能完成目前承担的任务. 目前主要的解决方法^[7]是重新进行任务的分配, 但是这需要花费大量的代价. 因此为了适应动态的变换环境, 提高任务执行的效率和 Agent 的满意度, 采用一种层次任务分配方法, 如图 1 所示. 该方法把任务分配问题分为两个层次: 第一层为全局任务分配层. 多 Agent 中有一个成员负责任务的分配问题, 把每个任务分配给相应的 Agent; 第二层为动态任务调整层. 获得任务的 Agent, 当发现不能完成当前任务, 或环境中更好的任务可以来完成时, 通过调整的方法来解决, 使得更多的 Agent 能够得到满意的任务.

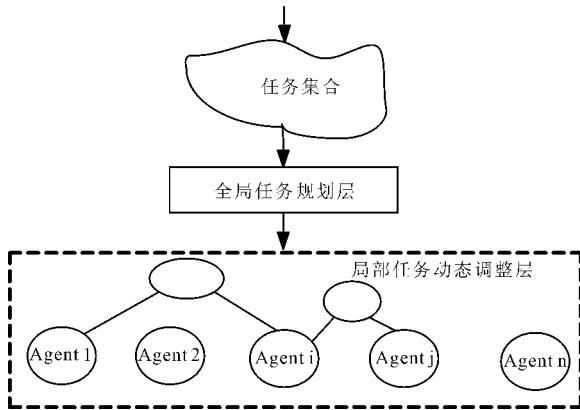


图1 层次任务分配方法

3.1 全局任务分配方法

在多 Agent 中选择一个管理 Agent 来完成全局分配任务,管理 Agent 掌握所有 Agent 的能力、偏好信息,并了解所有待分配的任务。但是鉴于 Agent 的自理性,了解所有 Agent 的能力和偏好是不现实的,本文采用拍卖的方法来实现第一层分配,根据 Agent 对任务的适应度 $M_i^t(t)$ 来作为中标的主要依据,算法如下:

Step1: 管理 Agent 发布所有任务 w_j 的信息,包括完成任务的能力需求 C_j^w ,以及任务完成后可以给出的报酬 u_{w_j} 等信息。

Step2: 等待所有满足能力需求的 Agent 提交投标信息 $M_i^t(t)$,若无投标信息,转向 step 8。

Step3: 管理 Agent 将每个任务的投标从低到高进行排序。设最高投标价格为 FP,共有 m 个 agent 给出了 FP。

Step4: 如果 $m=1$,管理 Agent 选择该 agent,转 step 6。

Step5: 如果 $m>1$,则管理 Agent 任意选择一个投标最高者。

Step6: 向选定的 Agent 发送“中标”信息,设置该 agent 的忙标志 $busy=1$,不允许再投标其他任务。

Step7: 向所有竞拍 Agent 发布分配结果。

Step8: 结束。

3.2 局部任务调整方法

经过第一层的任务分配过程,可以把任务分配给相应的 Agent,但是由于任务分配是一个 NP 问题,因此很难得到最优解,这就意味着可能存在 Agent 对目前承担的任务不满意的情况。另外随着任务的执行,Agent 的能力、任务的要求及完成任务的代价都可能发生变换,也可能出现新的任务,此时如果重新进行任务分配,需要花费很大的代价。由于以

上这些情况只出现在少数几个 Agent 或者任务中,因此采用局部调整的方法来解决这个问题。

3.2.1 问题描述

随着环境的变换,Agent 的能力、偏好或者任务的需求可能发生变化,这些变化导致原有的任务分配方法不能适应新的要求,Agent 对目前承担任务的适应度也发生变化,需要进行必要的调整。调整的条件描述如下:

设在 t 时刻,分配给 Agent a_i 的任务为 w_j 。对任务 w ,若 $\text{Allow}(w') = 1$,表示 w' 被分配给某个 Agent;若 $\text{Allow}(w') = 0$,表示没有 Agent 承担任务 w' ;若 $\text{Allow}(w') = 2$,表示 w' 目前被某个 Agent 承担,但是等待交换。某个 Agent a_k 没有承担任务,则 Agent a_k 的状态为空闲,即 $\text{state}(a_k) = \text{idle}$ 。

(1)if $\exists w' \in W, (\text{Allow}(w') = 0 \text{ or } \text{Allow}(w') = 2) \text{ and Agent } a_i \text{ 有能力完成该任务 and } r(a_i, w') > r(a_i, w_j) + T$, then $C1 = 1$, else $C1 = 0$. if $C1 = 1$,Agent 会放弃当前执行的任务,而选择任务 w' ,其中 T 为阈值;

(2)if $\exists k \in \{1, 2, \dots, l\}, \frac{\alpha_{ik}}{\beta_{jk}} < 1$, then $C2 = 1$, else $C2 = 0$. if $C2 = 1$,表明 Agent 能力发生变换或者任务需求发生变换,导致 Agent 不能完成当前任务。

(3)if $u(a_i, w_j) < T_d$, then $C3 = 1$, else $C3 = 0$. if $C3 = 1$ Agent 请求交换当前任务,其中 T_d 为阈值。

3.2.2 算法描述

设置一个黑板结构用于存放最新到达的尚未分配的任务、需求交换的任务信息,需要交换的 Agent 根据黑板中的信息,选择满足要求的任务。设当前 Agent 为 a_i ,其当前承担的任务为 w_j ,局部调整算法描述如下。

Step1: 如收到成功交换任务信息,则 $\text{state}(a_i) = \text{idle}$;

Step2: a_i 评估自身能力信息;检查黑板信息;

Step3: 计算 $C1, C2, C3$ 的值;

Step4: if $C1 + C2 + C3 < 1$ 则转 Step8;

Step5: if $C1 = 1$, 当前任务信息写入黑板, $\text{allow}(w_j) = 0$,把满足条件的任务设为 w' 作为当前任务, if $\text{allow}(w') = 2$,则 Agent a_i 通知目前承担 w' 的 Agent 愿意承担任务 w' ,转 Step8;

Step6: if $C2 = 1$, 当前任务写入黑板, $\text{allow}(w_j) = 0, \text{state}(a_i) = \text{idle}$,转 Step1;

Step7: 当前任务写入黑板, $\text{allow}(w_j) = 2$;

Step8: 执行当前任务;

Step9: 如果当前任务没有完成, 则转 Step1;

Step10: 结束.

4 仿真实验

战场环境和灾难环境都满足动态性和复杂性, 为了验证算法的有效性, 以战场环境中, 20 个 CGA (Computer Generated Actor) 为例进行仿真. 由于战场环境中, 道路会遭到破坏、安全区域会发生变换、CGA 本身可能被击毁或者出现故障等, 导致分配的任务不能执行或者执行代价变大. 经过 100 次的仿真实验, 图 2 中给出了全局分配未交换方法和该算法, 在完成任务总的效用上的差异. 从图 2 中可以看出, 随着任务的执行, 环境发生变换后, 如果没有调整, 则整体效应呈下降态势, 而且随着环境的变换, 有可能导致整体任务不能完成, 整体效用为零的情况. 这篇文章给出的算法可以保证 Agent 整体效用稳步上升, 可能会有小的上下波动, 但不会出现急速下降的情况, 从而提高了 MAS 的可靠性.

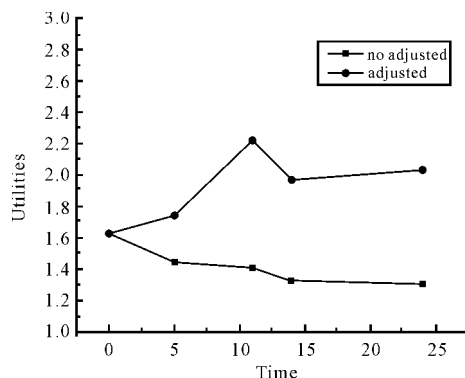


图 2 the Utilities Adjusted and No Adjusted Methods

在任务交换的过程中, 需进行必要的通信. 动态环境中通信是有代价的, 因此任务分配算法必须减少通信的数量, 才能有效地提高整体效用. 图 3 中给出了本文的方法与李湘清^[10]的方法和 Kathryn^[7]的方法在通信数量上的对比结果, 从图中可以看出, 文献^[7]的方法由于需要大量的计算, 需要的通信量最少. 当 Agent 数目较少时, 文献^[10]的方法需要的通信数量优于此算法, 当 Agent 数目较大时, 该算法需要的通信数量明显少于文献^[10]的方法.

5 结束语

合理的任务分配方法能够使 Agent 之间形成有效地合作, 提高 MAS 的整体性能. 本文针对动态环境

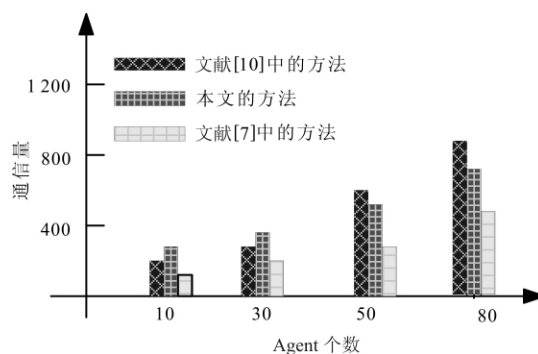


图 3 任务分配过程需要的通信次数

中的任务及 Agent 的动态变化, 提出了一种层次任务分配方法, 将 Agent 对任务的适应度作为中标依据, 提高任务分配时 Agent 与任务的匹配程度, 可以得到较优的分配结果; 而当 Agent 能力、任务需求发生变化及新任务出现时, Agent 能够自主地对当前执行的任务进行调整. 这种层次任务分配方法, 融合了集中任务分配和分布式任务分配的优点, 有效地提高了 MAS 的整体效用、MAS 系统的鲁棒性和可靠性, 仿真结果表明该方法是合理及有效的.

参考文献:

- [1] Pietro Baroni, Daniela Fogli. Modeling robot cognitive activity through active mental entities [J]. Robotics and Autonomous Systems, 2000, 30(4):325-349.
- [2] 唐苏妍, 朱一凡, 李群, 等. 多 Agent 系统任务分配方法综述[J]. 系统工程与电子技术, 2010, 32(10):2154-2161.
- [3] Ferber J. Multi-agent systems: an introduction to distributed artificial intelligence [M]. London: Addison Wesley, 1999.
- [4] Kathryn S Macarthur, Ruben Stranders, Sarvapali D Ramchurn, et al. A distributed anytime algorithm for dynamic task allocation in Multi-agent systems [C]// Proceedings of the 25th Association of Advancement on Artificial Intelligence Conference (AAAI 2011). San Francisco, USA: IEEE, 2011:701-706.
- [5] Chapman A C, Micillo R A, Kota R, et al. Decentralised dynamic task allocation: a practical game-theoretic approach [C]// Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-09), Budapest: Hungary, 2009: 915-922.
- [6] 李湘清, 孙秀霞, 夏岩. Multi Agent 动态分组模型用于多无人机协同任务分配 [J]. 系统仿真学, 2010, 22(5):1266-1269.

(下转第 121 页)

5 结束语

该文应用人工免疫系统的特性来提取均值抖动积 MDP、短期吞吐量 STT、瓶颈链路队列长队估计值 L 这三个参数并利用灰色关联度建立 MDP、STT、 L 与当前网络性能瓶颈链路队列长度真实值的关联性,通过分析可知瓶颈链路队列长队估计值 L 与瓶颈链路队列长度真实值贴进度最高,也即是最能反映网络性能。另一方面,在传统 TCP 协议的基础上,提出了一种新的适用于异构网络的网络传输控制协议 TCPL,该协议在传统 TCP 协议中加入上述描述的瓶颈链路队列长度估计值数学模型。该算法能够稳定发送速率和瓶颈链路队列长度,在不同负荷的背景流下都能取得较良好的性能,尤其是在负载较重时性能更为突出。

参考文献:

- [1] 王强,普杰信,刘伟. TCP 拥塞控制慢启动策略的研究[J]. 微电子学与计算机,2007,24(12):210-212.
- [2] 王国栋,任勇毛,李俊. 多重慢启动 TCP 协议研究[J]. 微电子学与计算机,2011,28(3):178-181.
- [3] 欧瑜枫,陈元琰,李鹏. TCPVegas-O:一种新的基于

延迟估计的 TCPVegas 改进算法[J]. 微电子学与计算机,2009,26(9):177-180.

- [4] 叶进,王建新. 异构网络中丢包识别研究综述[J]. 计算机科学,2006,33(12):19-33.
- [5] Osada S, Wang Hui, Yokohira T, et al. Throughput optimization in TCP with a performance enhancing proxy[J]. IEIC Technical Report,2006:145-150.
- [6] Jerne N K. Towards a network of the immune system[J]. Annual Immunolog, 1974(125):373-389.
- [7] 章森,吴建平,林闯. 互联网端到端拥塞控制研究综述[J]. 软件学报,2002,13(3):354-363.
- [8] Information Sciences Institute. The ns-2 simulator [EB/OL]. [2012-08-30][2011-12-1]. <http://www.isi.edu/nsnam/ns/>.

作者简介:

王伟男,(1986—),硕士研究生.研究方向为网络性能改善;

王辉女,(1966—),教授,硕士生导师.研究方向为网络性能改善与数据挖掘;

张潇男,(1986—),硕士研究生.研究方向为网络性能改善.

(上接第 116 页)

- [7] 刘利枚,石彪. 一种结构化的多 Agent 任务分配机制[J]. 信息技术与信息化,2009(1):27-29.
- [8] 刘淑华,张崧,吴洪岩,等. 基于群体智能的多机器人任务分配[J]. 吉林大学学报:工学版,2010,40(1):123-129.

作者简介:

陶雪丽女,(1974—),硕士,讲师.研究方向为虚拟现实、多 Agent 协作.

郑延斌男,(1964—),博士,教授.研究方向为人工智能、虚拟现实.