doi:10.3969/j.issn.1673-9833.2012.04.021

基于蚁群算法的多Agent任务分配方法

文志强，何宇晨

（湖南工业大学 计算机与通信学院，湖南 株洲 412007）

摘 要：针对多Agent任务分配问题，结合蚁群算法的思想，设计了基于图的任务分配数学模型，提出了基于蚁群算法的多Agent任务分配方法，并通过实验与3个经典方法进行比较和分析，探讨了蚂蚁数对求解结果的影响。实验结果表明，所提出的算法是有效的。

关键词：多Agent；任务分配；蚁群算法；信息素

中图分类号：TP391 **文献标志码**：A **文章编号**：1673-9833(2012)04-0092-05

Multi-Agent Task Allocation Based on Ant Colony Algorithm

Wen Zhiqiang，He Yuchen

（School of Computer & Communication，Hunan University of Technology，Zhuzhou Hunan 412007，China）

Abstract：In view of multi-agent task allocation problems, a task allocation model based on graph is presented, and based on ant colony algorithm a multi-agent task allocation method is proposed. Through experiments, it is compared with three classic methods, and the influence of ants number on the solution is discussed. The experimental result shows that the proposed method is effective.

Keywords：multi-Agent；task allocation；ant colony algorithm；pheromone

0 引言

在计算机应用中，待处理任务的复杂程度越来越高，而由单个Agent难以快速处理一个大而复杂的任务，因而，由多个Agent共同合作去完成某项复杂任务是必然的趋势[1-2]。这就需要将一个任务分解为多个子任务，然后分配给不同的Agent协同处理，因此，合理的任务分配方案成为高效处理任务的前提。任务分配是将合适的任务分配给合适的Agent以实现整体执行效果最优，这是一个NP难问题。多Agent任务分配求解方法主要有：穷举搜索法，就是通过穷举所有的分配方案来求解问题。这种方法易实现且能找到最优分配方法，但该方法的时间复杂度高，不适合于实时处理。贪心算法，就是使每次所做的选择看起来都是当前最佳的，期望通过所做的局部最优选择产生出一个全局最优解。对大多数优化问题，该算法能产生最优解，但也不一定总是最优。Kuhn-Munkres算法，是通过给每一个顶点一个标号，将求最大权匹配的问题转换为求完全匹配的问题，该方法只适合于任务数和Agent数相等的情形。上述方法都是非智能化方法，不能完全解决任务分配的最优化问题。另一类方法是智能化任务分配求解方法，利用智能化算法如遗传算法[3]、蚁群算法[4]、粒子群算法[5]等来搜索最优任务

收稿日期：2012-06-08

基金项目：国家自然科学基金资助项目（61170102），湖南省自然科学基金资助项目（11JJ3070，10JJ3002，11JJ4050）

作者简介：文志强（1973-），男，湖南湘乡人，湖南工业大学副教授，博士，主要研究方向为图像处理和视觉跟踪，

E-mail：zhqwen20001@163.com

分配方案。这些方法利用了求解过程中的一些启发式信息，求解效果比非智能化方法好。如针对蚁群算法求解分配任务问题时易陷入局部最优解的缺陷，文献[6]提出了一个分组多态蚁群算法。为了高效获取最优解，文献[4]使用改进的蚁群优化算法来解决多目标资源分配问题。文献[7]将蚁群算法应用于自主车辆的任务分配和路径规划，以提高任务规划的性能。

任务分配的目的是求多个任务与多个 Agent 之间的最优匹配。而 TSP（travelling salesman problem）问题是指寻找从一个城市出发，经过所有城市后又回到原来出发城市的最短路径。最初的蚁群算法是针对 TSP 问题而设计的，用蚁群算法来求解任务分配问题，求解方法有 2 类：第一类是修改蚁群算法，以适应任务分配问题，如设计蚁群算法搜索最优组合[8]或最优匹配[9−10]；第二类是将任务分配模型改造为 TSP 问题，采用经典或改进的蚁群算法来求解。针对第二类方法，本文将任务分配问题改造为 TSP 问题，提出了基于图的任务分配模型。与经典 TSP 问题不同的是，该模型中有 2 种类型节点且节点间不完全连通。另外，针对该模型，提出了基于蚁群算法的任务分配方法，并用实验验证了所提出的方法是有效的。

1　问题的描述与模型

任务分配算法要解决的问题是如何分配多个 Agent 体去执行多个任务，使得总的收益最大。假设有 n 个任务需要分配给 m 个 Agent（满足 $n<m$），每个任务可分配给任意一个 Agent，每个 Agent 某一时刻只可执行 1 项任务，q_{ij}（包含执行时间、通信代价等）表示第 j 个任务分配给第 i 个 Agent 所获得的收益值，其组成的 $m \times n$ 阶矩阵用 $\boldsymbol{Q}$ 描述。则总的收益值可表示为

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\left(q_{ij}\times x_{ij}\right),$$

式中，x_{ij} 为 0−1 决策变量，x_{ij}=1 表示安排第 i 个 Agent 完成第 j 项任务，x_{ij}=0 表示不安排第 i 个 Agent 去完成第 j 项任务，由 x_{ij} 组成 $m \times n$ 阶矩阵表示为 $\boldsymbol{X}$。因此可得出如下数学模型：

$$\text{solution}=\max\left\{\sum_{j=1}^{n}\sum_{i=1}^{m}\left(q_{ij}\times x_{ij}\right)\right\},$$

$$\sum_{j=1}^{n}x_{ij}=1,$$

式中，$x_{ij}\in\{0,1\}$，$i\in\{1,2,\cdots,m\}$，$j\in\{1,2,\cdots,n\}$。

2　基于蚁群算法的任务分配求解

蚁群算法是一种用来在图中寻找优化路径的概率算法，其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。蚁群算法模拟多只蚂蚁的蚁群协作过程，每只蚂蚁在候选解的空间中独立地搜索解，并在所找到的解上留下一定量的信息。解的性能越好蚂蚁留在其上的信息量越大，信息量越大的解被选择的可能性也越大。在算法的最初阶段所有解上的信息量是相同的，随着算法的推进较优解上的信息量增加，算法渐渐收敛。蚁群算法已成功解决了一系列问题，如 TSP 问题、Job-Shop 问题，初步研究已显示出蚁群算法在求解复杂组合优化问题方面具有并行性、正反馈性、强鲁棒性等特性。

2.1　基于图的任务分配模型

根据第 1 章中的数学模型，可以建立一个基于图的任务分配模型，如图 1 所示。T_j 表示第 j 个任务节点，A_i 表示第 i 个 Agent；P 为起始节点，P' 为结束节点；任务节点之间没有边连接，同样 Agent 之间也没有边连接；P 与任意任务节点 T_j 之间存在边，且边上的代价为常量 0（代价为 0 不影响最终求解结果，后同），任意任务接点 T_j 与任意 Agent 节点 A_i 之间存在边，任意 A_i 节点与 P' 节点之间存在边，且边上的代价为常量 0。根据此模型，需要求解的问题是：寻找一条从 P 节点到达 P' 节点，且经过所有 T_j 节点的最短路径。由模型易知，求解的最短路径如果经过所有 n 个 T 节点，则同样经过 m 个 A 节点。从 T 节点到 A 节点的边具有的权值为收益值 q_{ij}（路径的长短），但从 A 节点到 T 节点的边具有 0 权值。因此，经过所有从 T 节点到 A 节点的边，则表示任务的分配边，即所求解，矩阵 $\boldsymbol{X}$。该模型类似于 TSP 问题，可用蚁群算法求解。所有蚂蚁的初始出发点都是 P，结束点是 P'（在某种意义上，可把 P 和 P' 看成是重合的），蚂蚁经过每条边时都会留下一定量的信息素（初始时每条边的信息素为常数）。在图 1 中，蚂蚁将经过的 2 种边：1）从任务节点 T 到 Agent 节点 A 的边；2）从 Agent 节点 A 到任务节点 T 的边。

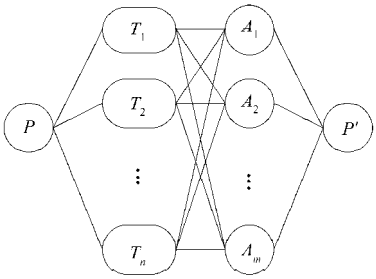

图1　基于图的任务分配模型

Fig. 1　Task allocation model based on graph

2.2　路径方向的选择

设时刻 t+1 蚂蚁 k 在第 i 节点上，则该蚂蚁可选

择与该节点相连的概率最大的边，蚂蚁从节点 i 转移到节点 j 的概率 $p_{ij}^k(t+1)$ 为：

$$p_{ij}^k(t+1)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}(t)\right]^{\beta}}{\sum\limits_{s\in J_k(i)}\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}(t)\right]^{\beta}}, & j\in J_k(i);\\ 0, & \text{otherwise}。\end{cases} \quad (1)$$

式中：$J_k(i)$ 表示蚂蚁 k 下一步允许选择的节点集合；

$\tau_{ij}(t)$ 表示 t 时刻在路径 (i, j) 连线上残留的信息量，由于在初始时刻各条路径上的信息量相等，可以设 $\tau_{ij}(0)$=const（const 为常数）；

η_{ij} 是一个启发式因子；

α 为信息启发式因子；

β 为期望启发式因子。

$J_k(i)$ 有 2 种情况：1）当蚂蚁从 T 节点到 A 节点时，$J_k(i)=\{A_1,A_2,\cdots,A_m\}-\text{Atabu}_k$，列表 Atabu_k 记录了蚂蚁 k 刚刚走过 A 类节点集；2）当蚂蚁从 A 节点到 T 节点时，$J_k(i)=\{T_1,T_2,\cdots,T_n\}-\text{Ttabu}_k$，列表 Ttabu_k 记录了蚂蚁 k 刚刚走过 T 类节点集。

η_{ij} 表示蚂蚁从节点 i 转移到下一节点 j 的期望程度，在蚂蚁系统中，η_{ij} 通常取城市 i 与城市 j 之间距离的倒数。由于节点之间的有权值，不妨取收益值 p_{ij}。蚂蚁选择路径的方法是：如果每一个可选择边的转移概率相等则随机选择一条边，否则根据式（1）选择概率最大的边，作为下一步的路径。

α 反映了蚁群在运动过程中所残留的信息量的相对重要程度，β 反映了期望值的相对重要程度，实验中常取 α=1，β=3。

2.3 信息素的更新

一只蚂蚁在图上沿某条边到达下一个节点，在该条边留有信息素，信息素根据式（2）和式（3）进行更新。

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau_{ij}(t), \quad (2)$$

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^k(t), \quad (3)$$

式（2）~（3）中：$\Delta\tau_{ij}$ 表示本次迭代节点 i 到节点 j 边上信息素的增量；

$\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次迭代中节点 i 到节点 j 边上信息素的增量；

ρ 表示在某条路径上信息素轨迹挥发后的剩余度，实验中 ρ 取 0.9。

如果蚂蚁 k 没有经过节点 i 到节点 j 边，则 $\Delta\tau_{ij}^k$ 的值为 0，$\Delta\tau_{ij}^k$ 表示为

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{L_k}, & \text{蚂蚁 } k \text{ 经 } ij \text{ 边};\\ 0, & \text{否则}。\end{cases} \quad (4)$$

式中：Q 为正常数；

L_k 表示第 k 只蚂蚁在本次周游中所走过路径的总收益值。

2.4 基于蚁群算法的任务分配算法

基于图 1 的模型，提出基于蚁群算法的任务分配方法如下：

第 1 步 初始化参数：设任务个数为 n，Agent 个数为 m，收益值矩阵 $\boldsymbol{Q}$，蚂蚁总数 ant_num，解初始值 x_{ij}=0，t=1，参数 $\tau_{ij}(0),\alpha,\beta,\eta_{ij}$。初始化每只蚂蚁的禁忌表 Atabu_k, Ttabu_k（k=1,2,…, ant_num）为空。

第 2 步 模拟第 k 只蚂蚁（k=1, 2,…,ant_num），初始化蚂蚁的出发位置 P，让蚂蚁根据 2.2 节的方法选择相应的边到达 T_j，然后选择相应的边到达 A_i，则相应 x_{ij}=1，同时累计获得的收益值。同样的方法选择边，让蚂蚁在图中爬行。

第 3 步 当蚂蚁经过所有的 T 节点且到达 A 节点后，蚂蚁没有可选择达到 T 节点的路径，这时蚂蚁则选择到达 P' 节点的路径，结束整个周游旅程。

第 4 步 蚂蚁到达 P' 节点后，计算蚂蚁经过路径所获得的总收益值，以此来更新走过路径上的信息素，更新方法见式（2）和（3），并更新获最大总收益值蚂蚁所经历的路径 $\boldsymbol{X}$。

第 5 步 如果 t 达到某个迭代次数或运行时间超过最大限度，则算法结束且蚂蚁获得的最大总收益值时所经历的路径即为所求解，否则 t=t+1，转至第 2 步。

3 实验数据及实验结果分析

实验环境为 Notepad++/Code::Blocks，编译环境为 GNU C/C++ 4.4.1，CPU 型号为 Intel（R）Core2 Duo CPU P8800，CPU 主频为 2.67 GHz，内存 4.00 GB，操作系统为 32 位 Windows 7。测试数据是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个方法路径或检验是否满足某个特定需求。测试数据应能较好地对比出各方法之间的优劣关系，或者能够发现某些方案中的不足。

实验中，T 节点与 A 节点边上的收益值矩阵 $\boldsymbol{Q}$ 为 2 种类型的数据：伪随机数和正态分布的数据，数据值限制在 0~99 范围内。伪随机数采用 C 语言库函数中的 rand() 随机产生，而正态分布的数据使用 MATLAB 中的正态随机数发生器产生。2 种类型的数据（包括参数 n 和 m）见矩阵 $\boldsymbol{Q}_1$~$\boldsymbol{Q}_4$，他们分别是 4 个任务和 12 个 Agent 的随机生成矩阵，4 个任务和 16 个 Agent 的随机生成矩阵，4 个任务和 12 个 Agent 的正态分布生成矩阵，4 个任务和 16 个 Agent 的正态分布生成矩阵。

$$\boldsymbol{Q}_1=\begin{bmatrix}6&9&3&6&7&6&9&4&9&8&8&7\\6&5&2&4&4&4&4&3&5&5&5&6\\4&3&7&3&6&2&8&6&3&8&7&7\\1&8&3&7&3&2&9&1&9&3&5&1\end{bmatrix},$$

$$\boldsymbol{Q}_2=\begin{bmatrix}41&2&64&88&12&29&40&67&62&50&40&61&53&48&60&92\\69&67&19&4&98&90&30&0&67&64&66&74&16&1&73&24\\36&17&23&75&75&69&61&19&58&31&71&4&45&91&46&1\\35&41&36&8&33&53&87&77&43&65&12&10&52&94&17&22\end{bmatrix},$$

$$\boldsymbol{Q}_3=\begin{bmatrix}72&26&35&28&81&29&46&22&78&34&54&64\\72&28&57&51&52&97&68&60&56&64&27&99\\33&50&45&61&20&38&35&46&54&67&27&37\\52&81&72&72&35&65&22&46&82&45&52&54\end{bmatrix},$$

$$\boldsymbol{Q}_4=\begin{bmatrix}57&5&45&26&91&43&23&50&55&18&54&56&29&35&70&36\\40&35&83&29&27&29&80&50&76&42&38&35&77&57&90&45\\43&25&54&49&78&35&34&22&43&53&28&43&90&39&56&43\\83&91&51&38&58&36&26&41&68&2&74&21&65&36&70&77\end{bmatrix}。$$

图 2 给出了本文方法（AC，ant_num=100）与 3 种经典任务分配求解方法在 4 个数据集上的实验结果比较。其中横坐标表示数据集，纵坐标表示求出解与真实值（通过穷举法获得）之间的偏差绝对值（error）。3 种经典任务分配方法分别是 Kuhn-Munkres 算法（KM），模拟退火算法（SA），贪心算法（GA）。

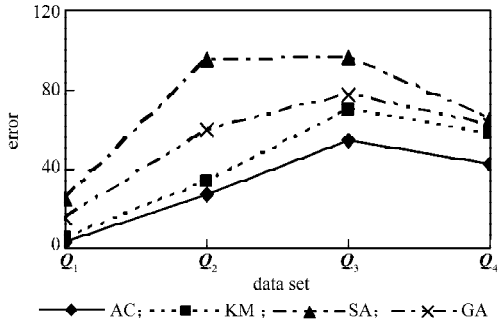

图2　不同数据集上的实验结果比较

Fig. 2　Comparison of experiment results for different data set

从图 2 可看出，本文所给方法的 error 在 4 个数据集上都是最小的，由此表明本文方法是有效的。另外，在本文方法中，蚂蚁数 ant_num 对寻优性能有较大的影响，太小则不能找到最优解，而太大又会影响方法的实时性能。

本实验中，对蚂蚁数 ant_num 分别取 10，20，50，100，200 和 500，可获得收益值 value 如图 3 所示。

从图 3 可看出，当蚂蚁数量小于 100 时，获得的收益值 value 随蚂蚁数的增加而增加；当数量超过 100 时，收益值基本没有发生变化，即蚂蚁数量超过一定的数量，对结果没有太大影响。因此，蚂蚁数 ant_num 取 100 能获得较好的结果。

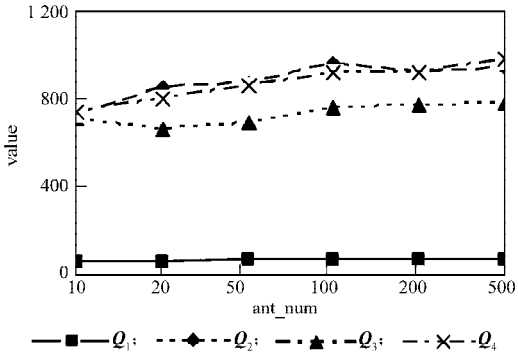

图3　不同蚂蚁数时的实验结果

Fig. 3　Experimental results for different number of ants

4　结语

针对多 Agent 任务分配问题，结合蚁群算法的思想，提出了基于图的任务分配模型，并设计了基于蚁群算法的任务分配算法。通过实验将本文的方法与经典方法进行了比较和分析，实验结果表明本文的方法是有效的。

虽然针对多 Agent 任务分配问题，本文提出了的一种有效的方法，但还有一些问题值得进一步探讨，如算法的效率和收敛停滞问题，或将蚁群算法与其它启发式算法，比如遗传算法、模拟退火算法等结合来求解任务分配问题，这些都是值得研究的内容。

参考文献：

[1] 李 勇. 多Agent系统联盟及任务分配的研究[D]. 合肥: 合肥工业大学，2008.
Li Yong. Research on Coalition and Task Allocation in Multi-Agent System[D]. Hefei： Hefei University of Technology，2008.

[2] 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]. 系统仿真学报，2010，22(5)：1266-1269.
Li Xiangqing，Sun Xiuxia，Xia Yan. Multi Agent Dynamic Group Model in Multi UAVs Cooperative Task Allocation [J]. Journal of System Simulation，2010，22(5)：1266-1269.

[3] 李湘清，孙秀霞，王 栋，等. 基于遗传算法的UCAV动态任务分配模型及研究[J]. 系统仿真学报，2008，20(16)：4387-4389.
Li Xiangqing，Sun Xiuxia，Wang Dong，et al. Dynamic UCAV Mission Assignment Using Genetic Algorithm[J]. Journal of System Simulation，2008，20(16)：4387-4389.

[4] Chaharsooghi S K，Meimand Kermani Amir H. An Effective Ant Colony Optimization Algorithm (ACO) for Multi-Objective Resource Allocation Problem (MORAP) [J]. Applied Mathematics and Computation，2008，200(1)：167-177.

[5] Yin P Y，Yu S S，Wang P P，et al. A Hybrid Particle Swarm Optimization Algorithm for Optimal Task Assignment in Distributed Systems[J]. Computer Standards & Interfaces，2006，28(4)：441-450.

[6] 张春艳，刘清林，孟 珂. 基于蚁群优化算法的云计算任务分配[J]. 计算机应用，2012，32(5)：1418-1420.
Zhang Chunyan，Liu Qinglin，Meng Ke. Task Allocation Based on Ant Colony Optimization in Cloud Computing[J]. Journal of Computer Applications，2012，32(5)：1418-1420.

[7] Kulatunga A K，Liu D K，Dissanayake G. Ant Colony Optimization Based Simultaneous Task Allocation and Path Planning of Autonomous Vehicles[C]// IEEE Conference on Cybernetics and Intelligent Systems. Sydney：Conference Publications，2006：1-6.

[8] 严建峰，李伟华，刘 明. 基于混合蚁群算法的MAS任务分配[J]. 计算机应用研究，2009，26(1)：68-70.
Yan Jianfeng，Li Weihua，Liu Ming. Task Allocation for MAS Based on Hybrid Ant Colony Algorithm[J]. Application Research of Computers，2009，26(1)：68-70.

[9] 王灵霞，张远平，吴佩莉. 蚁群算法求解分布式系统任务分配问题[J]. 计算机工程与设计，2008，29(6)：1472-1474.
Wang Lingxia，Zhang Yuanping，Wu Peili. Ant Colony Algorithm for Task Allocation Problem in Distributed System [J]. Computer Engineering and Design，2008，29(6)：1472-1474.

[10] 王灵霞. 分布式系统任务分配问题的蚁群优化算法研究[D]. 兰州：兰州理工大学，2008.
Wang Lingxia. Research on Ant Colony Optimization Algorithm for Task Allocation Problem in Distributed System [D]. Lanzhou：Lanzhou University of Teachnology，2008.

（责任编辑：邓光辉）