

# 一种动态环境下的移动 Agent 智能迁移算法

党 辰, 王嘉祯, 刘爱珍, 赵新青

(军械工程学院计算机工程系, 石家庄 050003)

**摘 要:** 在对移动 Agent 服务系统进行建模的基础上, 提出一种基于蚁群算法的移动 Agent 智能迁移算法。为减少探测软件带来的额外开销, 利用 Agent 探测全局系统状态。当 Agent 移动到服务节点时, 能根据信息素自适应地调整旅行路线。实验结果表明, 该算法能适应动态环境的变化, 具有容错性。在对蚁群算法引入自适应调整的信息素增加强度系数后, Agent 能以更高的效率完成迁移。

**关键词:** 移动 Agent; 蚁群算法; 迁移; 迁移策略; 旅行 Agent 问题

## Mobile Agent Intelligent Migration Algorithm in Dynamic Environment

DANG Chen, WANG Jia-zhen, LIU Ai-zhen, ZHAO Xin-qing

(Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003)

**【Abstract】** By modeling Mobile Agent(MA) server system model, this paper presents a MA intelligent migration algorithm founded on ant colony algorithm. It uses Mobile Agents to detect the global system status, which can minimize costs by detecting software. When Mobile Agents arrive at servers' nodes, they can feel the pheromone by others laid. According to the pheromone, they choose a shorter route to move intelligently. Experimental results show that algorithm can adapt to environmental changes and has fault tolerant. It introduces the self-adjust pheromone to add intensity coefficient, so that MA can move more efficiently.

**【Key words】** Mobile Agent(MA); ant colony algorithm; migration; migration strategy; travelling Agent problem

### 1 概述

移动性是移动 Agent(Mobile Agent, MA)最重要的特性。MA 在迁移过程中, 能自主地选择服务节点主机并移动到服务节点上完成用户交给的任务。如何根据任务的需求及动态变化的系统状况智能规划自己的移动路线, 是移动 Agent 研究的核心问题之一。

文献[1]利用求解模块求得当前网络负载和信息在主机上的分布概率情况, 从而为 MA 规划一条查询信息的最快迁移路径。文献[2]通过感知模块和“资源监测 Agent”获取动态变化的环境, 并在旅行图的基础上减少 MA 在迁移过程中的开销。文献[3]利用计划图、文献[4]利用“最优决策图”(Optimal Decision Graph, ODG)的概念为 MA 规划一条最佳迁移路径。文献[5]利用遗传、蚁群算法对 MA 旅行问题进行了初步研究。这些算法从各方面对 MA 的旅行迁移进行了研究, 并取得了一定的成果, 但仍有以下不足:

(1)没有对系统进行建模, 动态环境参数无法得知, 所采用的动态环境他人很难重建, 算法间不易作实验对比。

(2)需要探测软件对动态环境进行探测, 增加了网络与主机的额外开销。

(3)蚁群算法需要全局信息作为算法的依据, 文献[5]虽然利用蚁群算法取得了较好的性能, 但没有提及全局信息同步问题, 即 MA 在移动时如何获知全局信息。

(4)节点规模扩大时, 由于 MA 要携带服务节点地址信息, 因此使 MA 的体积增大。

本文利用蚁群算法的基本思想, 在服务节点环境中释放信息素, 通过 MA 相互合作来感知动态的环境变化, 最终降

低了后继的 Agent 的迁移开销。

### 2 移动 Agent 服务系统模型

#### 2.1 移动 Agent 在服务节点中的迁移问题

移动 Agent 的首要目标是找到有用的服务。在一个移动 Agent 服务系统中, 除了移动 Agent, 还必须有 2 个要素: (1)注册节点, 即注册服务并接收来自 MA 查询的主机节点; (2)服务节点, 即为 MA 提供运行环境及某项服务的主机节点。MA 在注册节点上查询到服务节点信息后, 即可在服务节点间旅行以完成用户交给的任务。这里定义 MA 在服务节点中的迁移问题(ASMP)如下:

设共有  $n$  个节点, 除了注册服务节点外, 其他服务节点要么失效, 要么为 MA 提供服务。MA 在遍历服务节点时, 可以绕过失效节点, 并在有效节点上获得服务。设  $d(i, j)$  为 MA 从服务节点  $i$  移动到节点  $j$  所用的开销。ASMP 问题就是要找一条 MA 遍历所有有效服务节点时所需开销最小的移动路径。

为使 MA 易于测得系统开销, 定义节点间的开销  $d(i, j)$  为

$$d(i, j) = \sum_i w_i c_i + w_0 c_{i \rightarrow j} \quad (1)$$

其中,  $c_i$  为 MA 在节点  $i$  上获得服务所造成的各项开销(如服

**基金项目:** 国家“863”计划基金资助项目(2004AA1Z2450); 河北省科技攻关计划基金资助项目(052435179D)

**作者简介:** 党 辰(1971—), 男, 博士研究生, 主研方向: 分布式计算, 智能算法, 软件工程; 王嘉祯, 教授; 刘爱珍, 博士; 赵新青, 硕士

**收稿日期:** 2008-10-10 **E-mail:** director\_2001@126.com

务时间、服务质量、服务价格);  $c_{i \rightarrow j}$  为从服务节点  $i$  移动到服务节点  $j$  的网络时延;  $w$  为各开销指标的权重。

例: 若只考虑 2 个开销指标: MA 服务时间和 MA 被传送时间, 并设权重为 1, 则服务节点间的开销为

$$d(i, j) = c_i + c_{i \rightarrow j}$$

其中,  $c_i$  为 MA 在节点  $i$  上获得服务的时间开销;  $c_{i \rightarrow j}$  为 MA 从节点  $i$  传送到节点  $j$  的时间开销。显然有  $c_{i \rightarrow j} = c_{j \rightarrow i}$ ,  $d(i, j) \neq d(j, i)$ 。注册节点和服务节点之间的连接可以看作全联通图。如图 1 所示, 节点 1 为注册节点( $c_1 = 0$ ), 其他节点为服务节点, 单位取 ms。

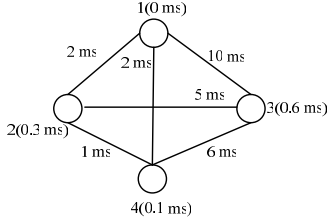


图 1 移动 Agent 服务系统状态

## 2.2 模型的定义

注册节点定义如下:

$$Register = \langle \{s_1, s_2, \dots, s_n\}, C, P, Mp \rangle$$

其中,  $\{s_1, s_2, \dots, s_n\}$  为记录注册服务节点的信息;  $C$  为系统开销域, 记录节点间的系统开销信息;  $P$  为信息素域, 记录 MA 在节点间释放的信息素信息;  $Mp = \langle Tabu, Tabu\_d, Er \rangle$ : 记录一个或多个 MA 迁移完后所带回的系统开销信息;  $Tabu$  表示所走过的路径序列(禁忌表);  $Tabu\_d$  表示在路径序列上各节点间的开销;  $Er$  表示失效节点信息。

服务节点定义如下:

$$Server = \langle \{s_1, s_2, \dots, s_n\}, C, P \rangle$$

其中,  $\{s_1, s_2, \dots, s_n\}$  记录注册服务节点的信息, 当有服务节点注册时各服务节点间要进行同步;  $C, P$  的含义与注册节点相同, 代表了各节点所感知的系统环境状态。

Agent 定义如下:

$$Agent = \langle \{1, 2, \dots, n\}, Mc, Mp \rangle$$

其中,  $\{1, 2, \dots, n\}$  为了减少 Agent 的体积, 仅记录各服务节点的编号;  $Mc = \langle Tabu, Tabu\_d, Er \rangle$ : 记忆信息域, 当前漫游中的 MA 记录的系统开销信息;  $Tabu$  表示所走的路径序列;  $Tabu\_d$  表示在路径序列上各节点间的开销;  $Er$  表示失效节点信息;  $Mp = \langle Tabu, Tabu\_d, Er \rangle$ : 携带信息域, Agent 携带的最近其他 MA 记录的系统开销信息, 其中的元素含义同前。MA 利用此信息修改各节点的系统开销域  $C$  和信息素域  $P$ 。

## 3 算法描述

### 3.1 蚁群算法

本文的算法根据蚁群算法的基本思想<sup>[6]</sup>, 每个 MA 在迁移过程中都具有以下特征:

(1) 以某种概率选择下一个要去的节点, 这个概率是 2 个节点之间距离及其路径上所含有的信息素数量的函数。定义  $1/d(i, j)$  为启发因子  $\eta_{ij}$ 。  $p_{ij}^k(t)$  为第  $k$  个 Agent 由节点  $i$  向节点  $j$  转移的概率, 由下式计算:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

其中,  $allowed_k = \{n - tabu_k\}$ ,  $tabu$  为禁忌表; 参数  $\alpha, \beta$  分别表示信息素的浓度和启发因子在转移概率  $p_{ij}^k(t)$  中的相对重要性;  $\tau_{ij}(t)$  为信息素浓度。

(2) 在旅行过程中, 各节点信息素表中的信息素浓度按下式修正:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}^k \quad (3)$$

其中,  $\rho$  是一个系数;  $(1-\rho)$  表示时间  $t$  和  $t+n$  之间信息素的挥发程度;  $\Delta\tau_{ij}^k$  是第  $k$  个 MA 在时间  $t$  和  $t+n$  之间、释放在路径  $(i, j)$  上每单位长度的信息素的数量(即第  $k$  个 MA 释放在路径  $edge(i, j)$  上的信息素浓度), 由以下公式计算:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{如果第 } k \text{ 个 MA 在时间和之间走过路径 } (i, j) \\ 0 & \text{否则} \end{cases} \quad (4)$$

其中,  $Q$  是一个信息素增加强度系数;  $L_k$  是第  $k$  个 MA 完成一次旅行时所走过的路径总开销。

(3) 对信息素增加强度系数  $Q$  值进行自适应调整。经过一段时间后, 由于某些弧段上的信息量越来越大, 会限制算法搜索的全局性, 因此随着搜索状态的变化,  $Q$  值应不断调整, 其调整原则如下:

$$Q = Q(1 - \frac{f_{ij}}{F_i - f_{ij}}) \quad (5)$$

其中,  $F_i$  为经过节点  $i$  的 MA 数目;  $f_{ij}$  为经过路径  $edge(i, j)$  之间的 MA 数量。

### 3.2 算法流程

该算法的主要步骤如下:

(1) 初始化

令 Register.D 和 Server.D 域中各节点间的开销值为一较小的常数  $c$  且 Register.P 和 Server.P 域中各节点间的信息素浓度值为 1

Register.Mp 域中记录一条或几条随机的路径, 并记路径间的开销均为  $c$

(2) MA 开始移动

1) 若 MA 在注册节点上

令 Agent.Mp = Register.Mp

按 4) 选择下一个节点移动

2) 若 MA 在其他节点上时

用 Agent.Mp 的信息修改服务节点上的 D 和 P 域

If Agent.Mp.Er <>  $\emptyset$

从 Server.  $\{s_1, s_2, \dots, s_n\}$  删除 Agent.Mp.Er 记录的节点

End

3) 获得服务, 并记录服务开销到 Agent.Mc

4) 按式(2)计算概率, 并按转盘法选择一个节点

发送 MA 到这个节点

如果发送失败, 则将失效节点存入 Agent.Mp.Er; 继续 4) 直到成功

在下一个节点记录网络传输的开销到 Agent.Mc, 并把节点存入禁忌表

5) 重复 2)~4) 直到 MA 回到注册节点

用 Agent.Mc 的信息, 按式(5)计算信息素浓度, 并修改 Register.D 和 Register.P 中的信息。将 Agent.Mc 存入 Register.Mp 域中。

其后的 MA 按(2)进行; MA 间可在节点间并行请求服务。

## 4 仿真实验

实验环境取 20 个主机节点(1 个注册节点; 19 个服务节点), 仅考虑服务时间和网络传送时间开销, 并设所占的权重都为 1, 初始网络状态为常数(单位按实际系统环境取 ms)。蚁群算法中用到的参数设置如下:  $\alpha = 1; \beta = 5; \rho = 0.1; Q = 100$ 。

这里在注册节点上每 20 个 MA 进行一次统计, 以 20 个 MA 的平均迁移时间和最短的迁移时间作图。

**实验 1** 保持系统状态不变, 在第 4 次统计时, 从注册节点和各服务节点观察域 Register.D 中的值, 其中保存了 MA 探测到的系统状态, 如图 2 的 *a* 点, 域 Register.D 中的值和网络初始状态比较, 结果发现两者的值完全相同。在第 50 次统计后, 对网络传送时间进行一次随机扰动, 如图 2 的 *b* 点。在第 100 次统计后, 令节点 20 失效, 如图 2 的 *c* 点。实验结果表明, MA 能通过相互合作来探测系统状态(*a* 点); 当系统状态发生变化或服务节点失效时, MA 能重新探测系统状态并调整路径, 以沿着较小开销的路径移动(*b* 点、*c* 点)。

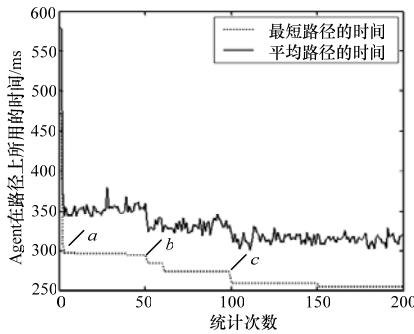


图 2 实验 1

**实验 2** 对系统状态做持续的较小的扰动, 观察收敛状况, 如图 3 所示。可以看到, MA 能随着环境的变化在收敛解附近波动, 进一步说明 MA 能适应动态环境的变化。

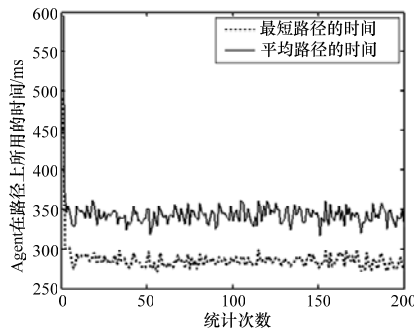


图 3 实验 2

**实验 3** 对比实验。自适应调整  $Q$  值参数后, 与不做调整时进行收敛性比较, 见图 4。从图中可知, 利用式(5)所求的最好解 270.9 优于式(4)的最好解 285.7。

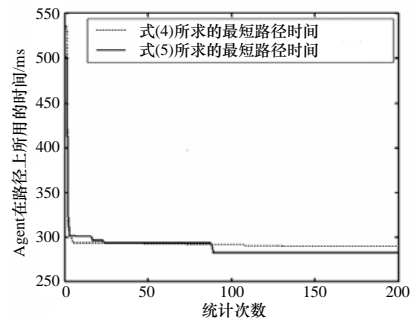


图 4 实验 3

## 5 结束语

与其他相关算法相比, 本文算法具有如下优点: (1)支持移动 Agent 在动态环境下的迁移。(2)实现简单。算法本身就是针对移动 Agent 迁移问题的一个可行的解决方案; 并且算法无需其他探测软件对系统状态进行探测。(3)引入自适应的信息素增加强度系数来求解移动 Agent 迁移问题。实验结果表明, 相对于传统蚁群算法, 本文的改进算法具有更强的全局最优解搜索能力。(4)由于 MA 仅携带服务节点的编号, 因此算法不会使 MA 体积明显增大。(5)算法具有容错性。在节点失效或网络断连时也具有良好的性能。

## 参考文献

- [1] Brewington B, Gray R, Moizumi K. Mobile Agents in Distributed Information Retrieval[M]//Klusch M. Intelligence Information Agent. Berlin, Germany: Springer-Verlag, 1999: 355-395.
- [2] 刘大有, 杨博, 杨鲲. 基于旅行图的移动Agent迁移策略[J]. 计算机研究与发展, 2003, 40(6): 838-845.
- [3] 张正球, 蔡声镇, 余敏. 一种改进的基于迁移计划图的移动Agent迁移策略[J]. 计算机应用研究, 2007, 23(1): 40-45.
- [4] Iqbal A, Baumann J, Strafer M. Efficient Algorithms to Find Optimal Agent Migration Strategies[R]. Stuttgart University, Tech. Rep.: TR21998205, 1998.
- [5] 骆正虎. 移动 Agent 系统若干关键技术问题研究[D]. 合肥: 合肥工业大学, 2002.
- [6] Colorni A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies[C]//Proceedings of the European Conference on Artificial Life. Paris, France: Elsevier Publishing, 1991.

编辑 张正兴

(上接第 18 页)

运动物体的轨迹测量, 该方法成功地扩大了摄像机的拍摄视角, 提高了拍摄精度, 将通常单目摄像机  $45^\circ$  视角扩展到最大  $360^\circ$  视角。方法中引入全景图像合成和图像对齐技术, 很好地利用旋转摄像机的图像合成出完整的背景, 将旋转的摄像机转变为“静止”的摄像机拍摄, 从而可以利用减背景等静止摄像机的相应技术完成物体的定位和轨迹测量。现有的基于计算机视觉的物体运动分析系统通常是利用固定摄像机对物体进行拍摄, 因此, 要适应大场地需要多台摄像机和许多复杂的设备。新的方法则采用软件模拟多摄像机(每个角度可看作一个摄像机), 不仅减少了设备简化了操作, 而且可适应不同的场地, 对拍摄位置约束少。唯一的不足是在测量过程中需要进行图像对齐和图像合成, 从而计算时间较长。该方法可以很好地完成大场地中人与物的轨迹测量, 可应用于

体育运动分析、车辆等物体运动分析中。

## 参考文献

- [1] 王亮, 胡卫明, 谭铁牛. 人运动的视觉分析综述[J]. 计算机学报, 2002, 25(3): 1-16.
- [2] Schmid C, Mohr R, Bauckhage C. Evaluation of Interest Point Detectors[J]. International Journal of Computer Vision, 2000, 37(2): 151.
- [3] Lowe G. Distinctive Image Features from Scale-invariant Key Points[J]. International Journal of Computer Vision, 2004, 60(2): 91.
- [4] Perez P, Hue C, Vermaak J, et al. Color-based Probabilistic Tracking [C]//Proc. of Eur. Conference on Computer Vision. Copenhagen, Denmark: [s. n.], 2002.
- [5] Hartley R I, Zisserman A. Multiple View Geometry in Computer Vision[M]. [S. l.]: Cambridge University Press, 2002.

编辑 张正兴

