

# 基于分布式遗传算法的移动 Agent 迁移策略<sup>\*</sup>)

朱翠涛 杨宗凯 程文青 吴 砥 严 鹤

(华中科技大学电子与信息工程系 武汉 430074)

**摘 要** 迁移策略是移动 Agent(Mobile Agent, MA)的核心技术之一,MA 的效率很大程度上取决于迁移策略的优化。本文提出了一种改进的分布式遗传算法(EDGA),用于对多约束条件下 MA 迁移策略最优问题进行求解。EDGA 将分布式遗传算法和 Cascade 模型相结合,在迁移算子部分设计一个中心监控器,观察每个子种群的进化,并对迁移个体的选择以及相应子种群的大小做出调整,使进化能力好的子种群得到更大的空间来搜索最优值。实验结果表明:本文所提出的 EDGA 算法在求解速度和质量上取得了较大的改善。

**关键词** 移动代理,迁移策略,分布式遗传算法

## Migration Strategy for Mobile Agent Based on Distributed Genetic Algorithm

ZHU Cui-Tao YANG Zong-Kai CHENG Wen-Qing WU Di YAN He

(Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** Migration strategy is one of the most critical problems in the system based on mobile Agent, the efficiency of mobile Agent depends on the optimal migration strategy. This paper proposes a new genetic Algorithm, called Extended Distributed Genetic Algorithms (EDGA), for multiple constrained migration strategy of mobile Agent. EDGA combines the advantages of Distributed Genetic Algorithms with Cascade Model. We design a central monitor dynamically allotting the size of sub-populations according to their performance, directing the migration, therefore making the searching of routes converge to the global optimization faster and better. The result of the experiments shows that the method is more effective than others.

**Keywords** Mobile Agent, Migration strategy, Distributed genetic algorithm

## 1 引言

移动 Agent 为了完成用户指定的任务,通常需要迁移到多台主机上,与这些主机交互,使用这些主机提供的服务和资源。因此,迁移是 MA 最显著的特性,迁移在 MA 程序的整体性能表现中起非常关键的作用<sup>[1]</sup>。

迁移策略是移动 Agent 的核心技术之一,迁移策略根据 MA 的任务、网络的软硬件环境和其他约束条件为 MA 规划出最佳迁移路径<sup>[2]</sup>。迁移策略的优劣直接影响 MA 的性能乃至其任务的完成。因而,移动 Agent 的迁移策略受到学术界和工业界的广泛关注,并进行了有益的探索,取得了可喜的成绩。这些迁移策略主要分为三类:(1)语句级迁移。即迁移条件和动作序列都隐含在 Agent 的代码中,如 IBM 公司的 Aglets、Dartmouth 学院的 Agent Tcl 和 GeneralMagic 公司的 Telescript 等。(2)结构化迁移。即将 Agent 的迁移信息从 Agent 体中分离出来,如 Concordia 提出的旅行计划<sup>[3]</sup>、M. Ashraf 等人提出的“最优决策图”(optimal decision graph, ODG)<sup>[4]</sup>、刘大有等提出的旅行图<sup>[5]</sup>、郭忠文等提出的基于相关分析与神经网络方法<sup>[6]</sup>。(3)部分代码迁移。对 Agent 的代码按功能进行模块化设计,有计划地调度代码中的各个模块,如杨公平等提出的远程代码动态装配策略、武成岗等提出的模块化移动 Agent 及其调度算法<sup>[7]</sup>。

这些工作在移动 Agent 选择最佳迁移路径的标准时只考虑了某一方面的因素,如服务速度。在实际应用中,根据移动 Agent 任务特性,往往需要同时制定多个标准,如服务质量、服务价格和网络传输时延等其它约束条件,使移动 Agent 迁移的代价到达最小。

我们知道,受限最优路径如果涉及到两个或两个以上的度量约束问题时,人们只能试图通过启发式算法或递归等其他更有效的方法。现有的路由选择算法如 Dijkstra 方法、Bellman-ford 方法、模拟退火法等用于求解多约束最优路径选择问题,已取得了不少成果。但多数这类算法通常都难以避免局部最优问题。遗传算法(GA)相对于其它判决和寻优算法,具有并行搜索、快速评估的优点,并适合在复杂而庞大的搜索空间中寻找最优解。在算法过程中,算法本身自动获得和积累有关搜索空间知识,自适应地控制搜索过程,不断缩小搜索空间,得到优化解直至最优解,这样得到的最优解往往是全局最优<sup>[8,9]</sup>。

本文结合并行遗传算法和 Cascade 模型的优点提出了一种改进的分布式遗传算法用于解决在多约束条件下的移动 Agent 路径寻优问题。实验结果表明提出的算法是可行的,提高了寻优速度和质量。

## 2 问题定义及模型建立

为实现上述思想,我们先给出以下定义。

<sup>\*</sup> 本课题得到高等学校科技创新工程重大项目培育资金项目(编号:705038-03),澳大利亚研究院国际合作基金(LX0240468)资助。朱翠涛 博士研究生,研究方向为移动 Agent 及网络教育关键技术。杨宗凯 教授,博士生导师,研究方向为多媒体网络通信,网络教育等。程文青 教授,博士生导师,研究方向为无线传感网络。吴 砥 博士,研究方向为网络教育。严 鹤 博士研究生,研究方向为移动 Agent 及网络教育关键技术。

**定义 1**  $G=(V,E)$ 称为迁移图或旅行图,集合  $V$  是结点的有限集合,每一个结点表示一个二元组(host, method), host 表示主机地址,method 表示在 host 上执行操作的名称,  $E$  是连接弧的有限集合,每个连接弧从一个前驱结点指向后继结点。连接弧的前驱结点和后继结点称为相邻。

**定义 2** 设  $G$  为旅行图,如果  $G$  中的每一个结点和每一个连接弧都附有权值,则称  $G$  为加权旅行图。

设  $S$  为源结点,  $D=\{d_1, d_2, \dots, d_m\}$  为目的结点,  $\delta_{ij}$  表示从结点  $i$  到结点  $j$  的传输时延,  $\Delta=(\delta_{ij})$  表示连接弧的加权时延矩阵。

同样可以指定旅行图中结点的权值。设  $i$  为旅行图中的一个结点,  $\tau_i, q_i$  和  $c_i$  分别表示结点  $i$  的服务时间、服务结果系数和服务价格。服务结果系数取值为  $0 \leq q_i \leq 1$ , 用于刻画主机提供服务的质量,系数越大表示服务质量越高,系数为 0 时表示主机不提供相应的服务。服务价格刻画了主机提供某项服务的收费标准。对于有  $n(n \geq m+1)$  个节点的网络拓扑图,结点的权值可用向量表示成  $T=(\tau_i)_n, Q=(q_i)_n, C=(c_i)_n$ 。

MA 在出发时刻  $t$  由其感知模块感知网络的软硬件负载信息,建立加权旅行图,在其中找到给定标准下的最佳迁移路由,同时满足服务时间最少、服务费用最小和服务质量最好, Agent 按照该路由进行迁移。其目标函数为:

$$\text{Min: } \sum_{i=j}^D \sum_{j \neq i} \delta_{ij} f_{ij} + \sum_{i=1}^D \tau_i I_i + \sum_{i=1}^D (1-q_i) I_i + \sum_{i=1}^D C_i I_i \quad (1)$$

对于一个 MA 路由请求  $R$ ,如果能够找到一条具有最小代价的路径  $L$ ,同时路径  $L$  要满足以下条件,则  $L$  可被接受。这些条件是:

$$\sum_{i=1}^d \delta_{ij} f_{ij} \leq \Delta_s, (t=1, 2, \dots, n) \quad (2)$$

$$\sum_{i=1}^d \tau_i I_i \leq T_s, (t=1, 2, \dots, n) \quad (3)$$

$$\prod_{i=1}^d q_i I_i \geq Q_s, (t=1, 2, \dots, n) \quad (4)$$

$$\sum_{i=1}^d c_i I_i \leq C_s, (t=1, 2, \dots, n) \quad (5)$$

其中  $T_s, \Delta_s, Q_s, C_s$  分别服务时间约束、传输时延约束、服务结果系数约束和服务价格约束。

$f_{ij}$  为链路连接指示函数:

$$f_{ij} = \begin{cases} 1 & i, j \text{ 间存在路由} \\ 0 & i, j \text{ 间不存在路由} \end{cases} \quad (6)$$

$I_i$  为节点指示函数:

$$I_i = \begin{cases} 1 & i \text{ 节点存在于路由中} \\ 0 & i \text{ 节点不存在于路由中} \end{cases} \quad (7)$$

为了阻止环路产生,约束条件为:

$$\sum_{j \neq i}^D f_{ij} - \sum_{j \neq i}^D f_{ji} = \begin{cases} 1, & i=D \\ -1 & i=1 \\ 0 & i \neq D \end{cases} \quad (8)$$

$$\sum_{j \neq i}^D f_{ij} = \begin{cases} \leq 1, & i \neq D \\ 0 & i=0 \end{cases} \quad (9)$$

### 3 染色体及遗传算子的设计

遗传算法是从一组初始解出发,除了目标函数值,在不需要其它信息的情况下就可以实现对可行域的全局高效搜索。遗传算法操作的是一群编码化的可行解,称为种群。它通过种群的更新与迭代来搜索全局最优解,而迭代是通过选择、交叉和变异等具有生物意义的遗传算子来实现。一般称包含选择、交叉和变异算子的遗传算法为传统遗传算法,记为 SGA。为了提高遗传算法的求解速度和质量,本文在实现 SGA 的基

础上引入了迁移算子,提出了一种改进的分布式遗传算法,记为 EDGA。

#### 3.1 染色体的设计

在遗传算法的应用中,首要工作是设计问题解的编码方式,即染色体表示,可以采用二进制、整数和实数变量等表示问题的编码。本文采用矩阵编码方式,即用结点表示矩阵的行和列,矩阵中元素表示对应两结点间的连接弧。如果存在连接,那么对应的基因为 1,不存在,则为 0。

网络结点的服务价格、服务时间和服务质量使用矩阵表示成  $C=(c_i)_n, T=(\tau_i)_n, Q=(q_i)_n$ ,链路的时延加权矩阵  $\Delta=(\delta_{ij})_{n \times n}$ 。为了真实地反映网络拓扑结构,约束条件通过初始化来完成,特别地,如果网络结点与下一跳路由没有直接相连,设它们之间的时延为无穷大。我们设无穷大为相对原始数据的 1000 倍,那样如果选择不存在的链路,由于它远远超过了初始值,通路在自然选择中会被淘汰掉。

#### 3.2 适应度评估函数

由于遗传算法在进化搜索中以适应度函数为依据,利用种群中每个个体的适应度值进行搜索,因此适应度函数直接影响到遗传算法的收敛速度以及能否找到全局最优解。本设计的适应度函数为:

$$F(f_{ij}, I_i) = \frac{1}{\sum_{i=1}^n c_i * \tau_i I_i / \prod_{i=1}^n q_i I_i + \sum_{i=1}^n \sum_{j \neq i} \delta_{ij} f_{ij}} \quad (10)$$

我们考虑了 4 个约束条件,服务价格  $C$  和服务时间  $T$  的乘积与服务结果系数  $Q$  的比值,以及与结点间传输时延  $\Delta$  之和越小,通路的适应度就越大,网络选择的路由越佳。

根据个体编码与适应度函数解空间的对应关系来计算它的适应度。在矩阵编码方式下,适应度函数要求的是所有含有 1 通路适应度的总和。

#### 3.3 遗传算子的实现

选择算子实现的方法很多,通常人们利用一个随机函数决定一个个体是否被保存,此方法的选择步骤简单,但是它带有盲目性,因为它的选择与个体优劣无关。本文采用排序法,用一个适应度线性相关的选择函数,把适应度高、性能好的个体保存下来。

在遗传算法中,交叉算子是主要算子,它的作用是全局寻优,它与编码方式密切相关。在矩阵编码方式下,我们首先假设两个父矩阵  $P_1, P_2$ ,临时矩阵  $T_1, T_2$ ,然后交叉产生两个新子矩阵  $N_1, N_2$ 。

$$\text{父矩阵: } P_1 = (x_{ij}^1)_{m \times n}, P_2 = (x_{ij}^2)_{m \times n}$$

$$\text{临时矩阵: } T_1 = \frac{P_1 + P_2}{2}, T_2 = (P_1 + P_2) * 2$$

$$\text{且 } T_2 \text{ 分解为 } T_2 = T_2^1 + T_2^2$$

$$\text{子矩阵: } N_1 = T_1 + T_2^1, N_2 = T_1 + T_2^2$$

引入变异算子的目的:一是使在选择和交叉过程中某些丢失的遗传基因进行补充和恢复,二是避免算法陷入局部最优,因此是必不可少的辅助算子。本文采用的方法为:假设父矩阵为  $P_{m \times m}$ ,随机在其中选择行和列组成  $T_{p \times q} \subset P_{m \times n}, p < m, q < n$ 。在满足约束条件下把  $T_{p \times q}$  中的元素进行随机变化,产生新的矩阵  $T'$ ,把  $T_{p \times q}$  在  $P_{m \times n}$  中元素用  $T'$  替换,产生新的变异矩阵  $P'$ 。

### 4 改进的分布式遗传算法

为了提高遗传算法的求解速度和质量,在 SGA 算法基础上,人们提出了并行遗传算法(PGA)。目前,并行遗传算法主要有主从式、粗粒度和细粒度的并行化模型。基于粗粒度型

的遗传算法也称分布式遗传算法(DGA),是适应性最强和应用最广的并行化模型。在并行遗传算法中引入了迁移算子, Grosso 是最早专门从事多群体 PGA 研究的人之一。他发现子种群个体平均适应值的提高速度远远大于 SGA 的单一大群体,说明小群体中优良基因的传播速度快于大群体。然而,小群体的收敛速度或进化停止要早于大群体,当子种群之间不进行个体迁移时,子种群所获得的最优解劣于大群体,说明在多群体 PGA 中,迁移是必要的。

基于以上思想,本文提出了一种改进的分布式遗传算法(EDGA),EDGA 算法流程如图 1 所示。实现过程为:首先将整个种群分解成几个子种群,每个子种群单独的执行 SGA,同时周期性地每隔 Migint 代在每个子种群中选取特定的个体迁移到别的子种群中去。DGA 可以使种群的进化跳出局部最优,使搜索更灵活,但是由于进行 SGA 的单个种群变小,个体的多样性减少,进化能力也随之削弱。为了克服此问题,本文在迁移算子部分设计了一个中心监控器,观察每个子种群的进化,并对迁移个体的选择以及相应子种群的大小做出调整,使进化能力好的子种群得到更大的空间来搜索最优值,同时尽可能把进化能力弱的子种群中的精英个体吸收到进化能力更好的子种群中去,满足其进化个体的多样性,使其更好地搜索出最优解。

总的来说,我们从以下两个方面对分布式遗传算法进行改进:

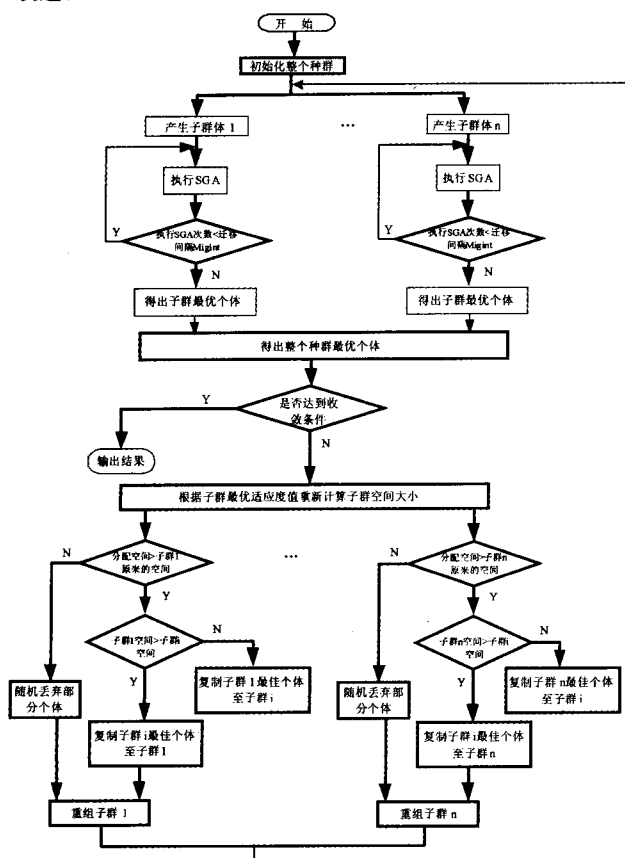


图 1 改进的分布式遗传算法流程图

1)子群体空间再分配

迁移过程中,中心监控器对每个子种群的进化程度进行询问,整个子种群空间将依据各个子种群进化的优劣被重新瓜分。中心监控子程序计算出每个子种群中最优个体的适应度,并根据适应度大小对各个子种群进行排队。子种群下一次进化得到的空间大小与其当前的适应度成线性关系。

2)个体迁移

每个子种群都有一个局部最优解,适应度函数较低子种群将把自己的局部最优解迁移到适应度函数较高的子种群,如下式所示, $M_{p_i}$  表示可以接受第  $i$  个子种群迁移个体的子种群集。

$$M_{p_i} = \{p_j | f(p_i) \leq f(p_j)\} \quad (11)$$

当子种群规模减小时,其中适应度最小的那一部分个体被丢弃;当子种群规模增大时,除了吸收外部来的最优个体外,还要随机复制自己来填充空间。适应度最高的子种群不会迁移任何个体到其它子种群中,而适应度最低的子种群将没有个体迁移过来,并且还要丢失一部分个体。

5 仿真实验与分析

我们利用 Matlab 仿真工具对本文提出的算法进行了实现。在 20 个结点的拓扑下,分别对 SGA、EDGA 以及两种迁移策略进行了性能比较,为了便于观察,以  $1/\text{fitness}$  作为参照观察算法的收敛特性。

1)SGA 收敛图分析

本文首先利用 SGA 对问题进行了求解,SGA 收敛图如图 2 所示。图中横轴为收敛的代数,进行了 1000 代,根据观察,它在 400~450 代收斂,最优路径和收敛时间为:short-pathlength=3803,elapsed\_time=25.5310

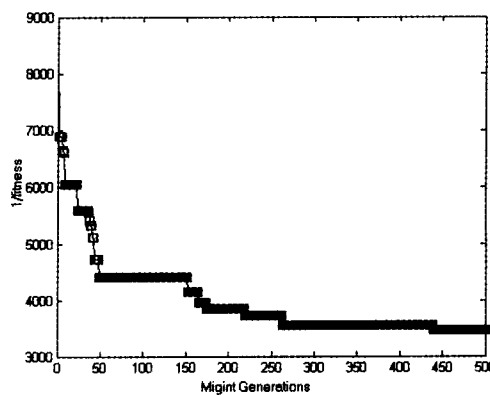


图 2 SGA 收敛图

2)EDGA 收敛分析

我们把整个种群分成 4 个子种群 SUBP1、SUBP2、SUBP3 和 SUBP4,同时执行本文实现的 SGA,当执行了  $q$  代后,比较各个子种群的最大适应度值,重新分配空间,然后再分别对种群 SUBP1, SUBP2, SUBP3, SUBP4 进行迁移操作。为了方便比较,SGA 运行了 500 代,在这里每隔 10 代迁移一次。EDGA 收敛图如图 3 所示。

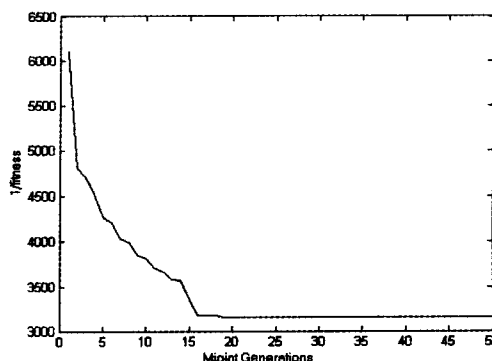


图 3 EDGA 收敛图

- rithms. In: Fonseca C M, et al. eds. Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, Springer. Lecture Notes in Computer Science. Volume 2632, April 2003. 494~508
- 35 Purshouse R C. On the Evolutionary Optimisation of Many Objectives:[PhD thesis]. Sheffield, U K;Department of Automatic Control and Systems Engineering, The University of Sheffield, September 2003
- 36 Kennedy J, Eberhart R C. Swarm Intelligence. San Francisco, California, Morgan Kaufmann Publishers, 2001
- 37 Sefrioui M, Periaux J. Nash Genetic Algorithms; examples and applications. In: 2000 Congress on Evolutionary Computation,

vol1. San Diego, California, IEEE Service Center, July 2000, 509~516

- 38 Van Veldhuizen D A, Lamont G B. Multi-objective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation, 2000, 8(2):125~147
- 39 Knowles J, Corne D. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. IEEE Transactions on Evolutionary Computation, 2003, 7(2):100~116
- 40 Laumanns M, Thiele L, Deb K, et al. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation, 2002, 10(3):263~282

(上接第 112 页)

略改善数据的局部性,尽量变远程访问为本地访问,从而提高存储系统乃至整个系统的性能。为了对共享存储管理进行优化,本文针对动态数据局部性优化技术——页迁移策略进行了研究,并提出了一种基于遗传算法的动态页迁移策略。

### 参 考 文 献

- 1 Amza C, et al. TreadMaks: Shared Memory Computing on Networks of Workstations. IEEE Computer, February 1996
- 2 Agarwal A, et al. The MIT Alewife Machine: Architecture and Performance. In: the Proceedings of the 22<sup>nd</sup> Annual International Symposium on Computer Architecture, June 1995

- 3 Bircsak J, et al. Extending OpenMP for NUMA machines. Supercomputing 2000, Dallas, Texas, 2000
- 4 Culler D E, et al. Parallel Computer Architecture (Second Edition). Morgan Kaufmann Publishers Inc, 1996
- 5 Gharachorloo K. The Plight of Software Distributed Shared Memory. In: Proceedings of 1<sup>st</sup> Workshop on Software DSMs, 1999
- 6 Bligh M J, Hansen D. Linux memory management on larger machines. The linux Symposium, Ottawa, Canada, 2003
- 7 Nikolopoulos D, et al. Scheduler-activated dynamic page migration for multiprogrammed DSM multiprocessors. Journal of Parallel and Distributed Computing, 2002, 62960:1069~1103

(上接第 180 页)

可以看出, EDGA 在运行到第 16 代迁移时就收敛了,  $shortpathlength_4 = 3158$ ,  $elapsed\_time = 0.3440$ 。由于迁移间隔为 10 代,那么达到总体收敛的时间是 160 代,而 SGA 收敛代数达到了 450 左右。同时,最优路径综合代价改进前为 3803,改进后是 3158。由此可以看出 EDGA 的优越性。

#### 3) SGA、DGA、EDGA 性能比较

我们将本文提出的 EDGA 算法中的迁移策略与一般基于网络拓扑(DGA)的迁移策略进行了对比,从实验结果可以看出 EDGA 的性能优越于 DGA。三种方法的性能对比如图 4 所示。

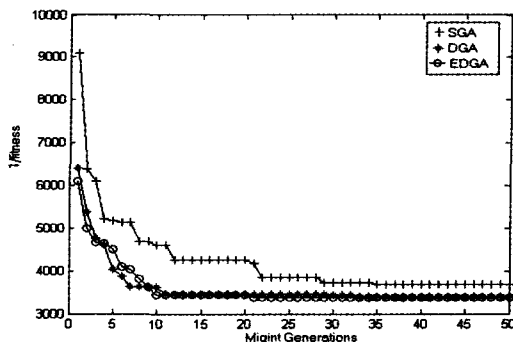


图 4 SGA、DGA、EDGA 性能比较

结论 本文提出了一种改进的分布式遗传算法,对多约束条件下移动 Agent 的迁移策略最优问题进行了求解,通过仿真实现证明了所提算法的优越性。由于本文所讨论的属于

静态迁移路由,而网络环境是动态变化的,因此下一步工作将进一步研究动态环境下移动 Agent 迁移策略的全局最优问题。

### 参 考 文 献

- 1 Kotz D, Gray R S. Mobile agents and the future of the Internet: [rep. Dartmouth College]. 1999. 7~13
- 2 Acharva A, Ranganathan M, Saltz J. Sumatra: A language for resource-aware mobile programs. In: Proc. of Mobile Object Systems: Towards the Programmable Internet. Berlin: Springer, 1997. 111~130
- 3 Glitho R H, et al. Mobile Agents and Their Use for Information Retrieval. IEEE Network, 2002. 34~41
- 4 Iqbal A, Baumann J, Straber M. Efficient algorithms to find optimal agent migration strategies. Stuttgart university: [Tech Rep: TR-1998-05]. 1998
- 5 刘大有,杨博,杨巍,王生生. 基于履行图的移动 Agent 迁移策略. 计算机研究与发展, 2003(6): 838~845
- 6 郭忠文,等. 基于相关分析与神经网络的 Agent 迁移策略. 见:第五届全球智能控制与自动化大会, 2004, 6: 1958~1962
- 7 武成岗,史忠植. 基于模块化移动 Agent 及其调度算法. 软件学报, 2002(8): 1628~1636
- 8 Yuan X, Liu X. Heuristic Algorithms for Multi-Constrained Quality of Service Routing. In: Proceedings of IEEE INFOCOM, April 1997. 58~62
- 9 Korkmaz T, Krunz M. Multi-constrained Optimal Path Selection. In: Proc. of IEEE INFOCOM, 2000. 113~118