

# 一种改进的 Agent 学习迁移策略<sup>①</sup>

## An Improved Learning Migration Strategy of Mobile Agent

汪 勇 (武汉科技大学 管理学院 430081)

**摘要:**本文介绍一种改进的学习迁移策略能在能描述运行环境动态变化的有向图中寻找最短路径,根据探测 Agent 获得的不同时刻结点状态信息和以往的旅行计划,利用自适应遗传算法规划出一组最佳迁移路径。解决了学习迁移策略由于结点失效导致的迁移失败。该策略能适应复杂的动态网络环境,动态路由,及时自主地修改迁移路线。实例验证了该方法的有效性。

**关键词:**移动 Agent 迁移 遗传算法 最短路径

### 1 引言

Agent 的迁移机制是移动 Agent (MA) 的核心技术之一,受到来自学术界和工业界的广泛关注,其代表性的工作有 Dartmouth 学院的 Agent Tcl、General Magic 公司的 Telescript、IBM 公司的 Aglets 和 Mitsubishi 的 Concordia 等。

在 Agent 语句级迁移中,迁移条件和动作都是隐含在 Agent 的代码中的,对 Agent 的设计、测试和调试都有一定的难度,尤其在 Internet 环境中更是如此。结构化 Agent 迁移方法将 Agent 的迁移信息从 Agent 体中分离出来,用一种有足够能力的结构进行描述,并提供灵活的迁移信息处理手段,从而为 Agent 的移动提供有效的支持。

Agent 常使用中间件技术、CORBA 对象生命周期服务的 Move 方法和 Java 的序列化及类装载机制实现迁移。何时迁移一个 Agent 取决于系统的调度算法,选择哪个目的站点取决于调度算法的迁移策略。

### 2 迁移策略概述

迁移策略是根据 MA 的任务、网络的软硬件资源和其他约束条件为 MA 规划出一条迁移路线。迁移策略的优劣直接影响 MA 的性能及其任务的完成。现行的迁移策略大致有两种,分别是固定路径迁移和基于规则及目录服务的动态迁移<sup>[4]</sup>。

Concordia 系统中的旅行计划是一个完全独立于移动 Agent 的线性表,表中的每一个元素是一个二元

组  $E(\text{host}, \text{method})$ , 表示在地址为 host 的主机上执行名称为 method 的操作,二元组的次序指明了 MA 在这些主机之间移动的次序,旅行计划将控制 MA 移动的逻辑从任务体中分离出来,使得在不更改 MA 任务体代码的情况下,就可以修改 MA 的移动路线,增强了设计 MA 的灵活性。但 Concordia 的旅行计划实际上是一种固定路线的迁移,它只考虑了 MA 在主机之间的顺序移动,不能描述复杂的迁移情况。

Acharya 等人提出“信息监测”策略,他们意识到硬件资源及其使用状况对 MA 迁移路径的影响,为 Sumatra 系统设计了信息监测模块,该模块收集各种硬件的负载信息,作为 MA 选择迁移路径的重要依据<sup>[2]</sup>; Ashraf 等人提出了“最优决策图”(optimal decision graph, ODG)的概念并应用于 Mole 系统中,该策略从通信性能的角度研究了 MA 的迁移策略<sup>[3]</sup>; Moizumi 等人提出了“旅行 Agent 问题”(traveling agent problem, TAP),开发了基于 D'Agent 的分布式信息查询系统, TAP 求解模块根据当前网络负载和信息在主机上的分布概率,为 MA 规划出迁移路径,使 MA 以最快的速度找到待查信息<sup>[4]</sup>。

以上迁移策略的共同缺点是:①要求待访问的主机已知。然而在多数情况下,设计者事先无法决定 MA 应该访问哪些主机;②路径规划使用的负载信息在 MA 出发时采集,而环境的动态变化会使这些信息过时;③没有容错机制,网络断连或主机失效都会导

① 基金项目:湖北省自然科学基金(2004ABA004);湖北省教育厅科学研究计划青年项目(2004Q002)

致迁移失败;④选择最佳路径的标准只考虑了结点主机的服务速度,而没有考虑结点数据流量和网络质量等因素。

### 3 改进的学习迁移策略

#### 3.1 MA 迁移策略思想

改进的学习迁移策略思想是:利用状态探测 Agent 每隔一定的时间巡游各主机,并将各主机信息存入知识库,每次巡游结束更新知识库。实验证明,资源探测 Agent 使用的 CPU 时间和 I/O 资源很少,不会过多占用系统开销,导致 MA 迁移时间的明显增加。把实际的网络视为一个带权的旅行图,数据流量和带宽等作为权值保存在知识库中,根据 MA 的请求,利用遗传算法求出最短路径,生成某时刻的旅行计划。考虑到网络的动态特征,最终的旅行计划由不同时刻可能的旅行计划交集组成。MA 迁移思想如图 1 所示。

#### 3.2 SAGA 迁移算法

为实现上述思想,我们先给出以下定义。

定义 1:如果  $G = (V, A)$  是有向图, $V$  是  $n$  个结点集合, $A$  是结点连接弧集合,那么

$v_i = \{host_i, method_i\} \in V$   $host_i$  表示第  $i$  个主机地址, $method_i$  是在主机  $i$  上执行的操作

$a_{ij} = \{\vec{v}_i, \vec{v}_j\} \in A \quad i < n, j < i, i \neq j$

如果  $M = |a_{ij}|$  是连接矩阵,则  $a_{ij} = 1$  表示结点  $v_i$  到结点  $v_j$  是连通的,为 0 表示断开。

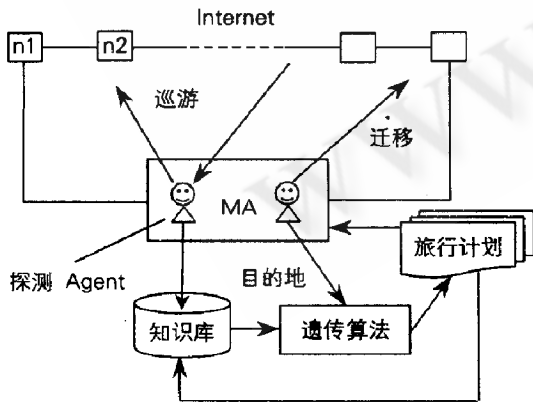


图 1 改进的学习迁移策略

定义 2:MA 从起始结点  $v_s$  出发,经过若干结点

$V'$  后,到达目的结点  $v_d$ ,再回到起点所经过的路线称为一条路径。如果用  $r$  表示一条路径,则  $r$  的递归定义为:

$$r = \{\vec{v}_s, \vec{V}, \vec{V}, \vec{v}_d, \vec{v}_d, \vec{v}_s\}$$

$$V' = \{v_{m-1}, v_{i_m}\} \quad t_m = t_{m-1} \cdot next, t_1 = s, v_1 = \Phi,$$

$m \geq 2$

例如,当  $m = 4$  时,表示由 4 个结点构成一条路径: $v_s \rightarrow v_{s.next} \rightarrow v_{(s.next).next} \rightarrow v_{((s.next).next).next} \rightarrow v_s$

定义 3:设结点  $v_i$  的状态值为  $f(x_i)$ ,则包含  $m$  个结点的路径  $r$  长度为: $L_r(m) = \sum_{i=1}^m f(x_i)$ 。最短路径为: $L_s = \text{Min}(L_r(m))$ 。如果  $\text{Lim}L_r(m) \rightarrow L_s$ ,则  $r$  称为一条迁移路径。

定义 4:状态函数  $\text{flow}(i)$ 、 $\text{price}(i)$ 、 $\text{bw}(i)$  和  $\text{cont}(i)$  分别表示结点  $v_i$  的数据流量、服务价格、带宽和连接状况, $C_i$  表示结点  $v_i$  与其前驱结点的连通性,连通性表示为一个邻接矩阵。令  $f_r = L_r(n)$ , $f_r$  是路径  $r$  的适应度,则

$$f_r = \sum_{i=1}^n f(v_i) = \lambda \cdot C_i \cdot (\sum_{i=1}^m (w_1/\text{flow}(i) + w_2/\text{price}(i) + w_3 \cdot \text{bw}(i)) \cdot \text{cont}(i))$$

$w_1$ 、 $w_2$  和  $w_3$  分别是流量、服务价格和带宽的权重。若只注重网络流量或带宽等影响因素,则相应的权重取值应该较大。 $\text{cont}(i) = 1$  时表示结点  $v_i$  连通, $\text{cont}(i) = 0$  时表示断开。

定义 5:旅行计划 TP 由不同时刻最佳迁移路径  $r$  组成,即

$$TP = r \in R_t \cap R_{t+\Delta t}, \quad r = \Phi \quad TP = R_{t+\Delta t}$$

$R_t$  是  $t$  时刻旅行计划集。

定义 6:设第  $g$  代的种群由个体  $x_1^g, x_2^g, \dots, x_N^g$  构成,适应度分别为  $f_1^g, f_2^g, \dots, f_N^g$ 。

则个体平均适应度为:

$$\bar{f}^g = \frac{1}{N} \sum_{i=1}^N f_i^g$$

种群的适应度方差:

$$D_g = \frac{1}{N} \sum_{i=1}^N (f_i^g - \bar{f}^g)^2$$

并记从第 1 代群体到第  $g$  代群体为止的最大适应

度方差  $D_{\max} = \max\{D_j | j = 1, 2, \dots, g\}$ , 设  $\lambda = 1 - \frac{D_g}{D_{\max}}$ ,

则指标  $\lambda$  可以用来表征第  $g$  代种群的多样性程度。

从以上定义可以看出, $\lambda$  的取值范围为  $[0, 1]$ 。

在进化初期,  $\lambda$  接近于 1, 有利于保持种群的多样性; 在进化后期,  $\lambda$  变小, 使种群越集中到某一局部域。  $\lambda$  对 GA 的遗传参数和适应度函数动态调整, 体现了算法的自适应性, 称为自适应遗传算法 (Self - Adaptive Genetic Algorithm, SAGA)。算法描述如下:

① 根据 MA 请求, 确定目标主机  $host_d$ , 这样路径  $\{\vec{v}_s, \vec{V}, \vec{V}\vec{v}_d\}$  的主机地址编码为:

$$host_s, host_1, host_1, \dots, host_d$$

② 随机确定一些初始路径作为原始种群, 种群中的个体是一些不等长的基因串;

③ 访问知识库和邻接矩阵, 计算  $t$  时刻第  $g$  代 ( $g=0$  表示原始种群) 路径适应度  $f^g$ , 平均适应度  $\bar{f}^g$ , 动态适应因子  $\lambda$ ; 如果路径  $r$  上存在  $cont(i) = 0$ , 则  $f^g = 0$ 。

由于 GA 没有利用目标函数的梯度信息, 所以无法判定算法收敛与否终止计算。常采用两种终止准则, 一是根据迭代次数终止计算, 在连续多少代后, 解的适应值没有明显变化时即终止; 二是根据客户要求终止, 当解满足用户要求时终止<sup>[9]</sup>。这里采用准则一终止计算, 所得到的迁移路径集为  $R_t$ 。

④ 在第  $g$  代路径集合中, 选择  $f^g > \bar{f}^g$  且  $f^g \neq 0$  路径; 对  $0.8 \cdot \lambda \cdot q_g$  条路径进行交叉 ( $q_g$  是第  $g$  代路径数量), 对  $0.15 \cdot \lambda \cdot q_g$  条径进行变异。选择、交叉和变异操作所得到的路径均作为第  $g+1$  代路径, 转③;

为保证遗传操作后个体的多样性, 交叉应在等长和不等长的个体间进行, 交叉和变异个体数不少于  $k$ ,  $k$  为编码的长度种类。同时为保证交叉和变异操作得到的路径是连通的, 可根据连接矩阵  $M$ , 删除路径结点序列中存在  $a_{ij} = 0$  的个体。

⑤ 重复③和④, 计算  $t + \Delta t$  时刻的  $R_{t+\Delta t}$ 。

⑥ 选择  $R_t \cap R_{t+\Delta t}$ , 作为 MA 的旅行计划。

### 3.3 仿真实例分析

假设有 10 个结点的网络如图 2 所示。MA 从结点  $n_1$  出发, 迁移至目标结点  $n_{10}$ 。邻接矩阵  $M$  如图 3 所示。

探测 MA 获得  $t$  时刻和  $t + \Delta t$  时刻各结点状态如表 1 所示。表 1 中的数据已经过处理,  $w_1$ 、 $w_2$  和  $w_3$  取值分别为 0.4、0.1 和 0.5。

不同于 TSP 问题, Agent 迁移不需要遍历所有结点, 因此, 迁移路径是一系列不同数量的基因串, 并且基因可以重复, 即 Agent 可以多次通过同一结点。如

何确定初始路径是一个必须解决的问题。文献<sup>[6]</sup>在求最短路径的遗传算法中没有具体介绍初始路径的产生方法。这里运用邻接矩阵  $M$ , 对 10 个结点组成的路径进行扫描, 确定适应度函数中的  $C_i$ , 得到一条连通的路径。

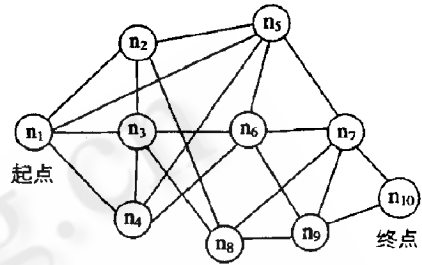


图 2 MA 迁移网络结构图

	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$
$n_1$	1	1	1	1	0	0	0	0	0	0
$n_2$	1	1	0	1	0	0	1	0	0	0
$n_3$	1	1	1	0	1	0	1	0	0	0
$n_4$	1	0	1	1	1	0	0	0	0	0
$n_5$	1	1	0	1	1	1	0	0	0	0
$n_6$	0	0	1	1	1	1	0	1	0	0
$n_7$	0	0	0	0	1	1	1	1	1	1
$n_8$	0	1	1	0	0	0	1	1	1	0
$n_9$	0	0	0	0	0	1	1	1	1	1
$n_{10}$	0	0	0	0	0	0	1	0	1	1

图 3 网络邻接矩阵  $M$

表 1 结点状态信息表

	t 时刻			t + Δt 时刻			带宽
	流量	价格	连接	流量	价格	连接	
$n_2$	0.90	0.45	1	0.40	0.45	1	8.0
$n_3$	0.15	0.85	1	0.15	0.85	1	2.7
$n_4$	0.50	0.24	1	0.50	0.16	1	6.5
$n_5$	0.75	0.60	1	0.75	0.60	1	8.2
$n_6$	0.13	0.71	1	0.13	0.71	0	9.8
$n_7$	0.27	0.43	1	0.27	0.52	1	5.4
$n_8$	0.61	0.77	1	0.26	0.77	1	7.6
$n_9$	0.36	0.82	1	0.36	0.82	1	6.0

对  $n_1$  至  $n_{10}$  10 个结点用下标序号进行实数编码, 1 表示起点, 10 表示终点, 则一条迁移路径可表示为:  $1, a_2, a_3, \dots, a_9, 10, a_i \in [2, 9]$ 。

最短迁移路径的搜索算法步骤如下:

① 随机化方法产生 10 条初始路径  $r_1 \sim r_{10}$ 。  $a_i = \text{Round}(7 * \text{Rand}(1)) + 2$ ;

② 按定义 5 计算每条路径的适应度函数值  $f_1 \sim f_{10}$ , 并按升序排序;

③ 进行遗传操作。选择适应度函数值大的 4 条路径复制到新的路径集合中; 在剩下的路径中按顺序选择 4 条两两交叉, 交叉后的路径加入到新的路径集合中; 适应度函数值较小的 2 条路径变异后也加入到新的路径集合中。交叉策略为偶数位交叉, 如有两条路径分别为  $1, a_2, a_3, \dots, a_9, 10$  和  $1, b_2, b_3, \dots, b_9, 10$ , 则  $a_2$  与  $b_2, a_4$  与  $b_4, \dots, a_8$  与  $b_8$  交叉; 变异策略为随机变异;

④ 重复步骤②、③, 直到执行规定的进化代数或出现满意的适应度函数值为止。

以上算法在 Matlab 中运行通过。图 4 是  $t$  时刻平均路径的迭代曲线, 纵轴为  $1/\bar{l}_t$ , 图 5 是  $t$  时刻最短迁移路径的迭代曲线, 在  $t$  时刻经过近 150 次迭代后, 最短路径为 43.41, 平均路径长度为 34.29, 迁移路径为  $1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 10$ 。在  $t + \Delta t$  时刻经过近 230 次迭代后, 最短路径为 43.46, 平均路径长度为 30.20, 迁移路径为  $1 \rightarrow 5 \rightarrow 7 \rightarrow 10$ 。因此, Agent 旅行计划:  $R_t = \{n_1, n_5, n_6, n_7, n_{10}, n_1, n_5, n_7, n_{10}\}$ 。

以上迁移路径不一定是真正最短路径, 他是一个 SAGA 解空间中接近最优的解, 随着迭代次数的增加, 搜索结果越接近最优解, 但算法的效率将越来越低, 所以最短路径是在效率与所期望的结果之间的折衷。

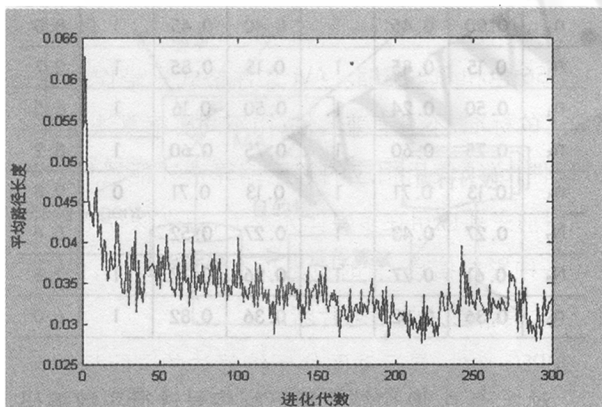


图 4  $t$  时刻平均路径迭代曲线

## 4 结论

对于规模较大的网络及较小的  $\Delta t$  选取, 学习迁移策略由于结点失效而使 MA 无法迁移到目标结点, 其环境适应性较差。SAGA 具有较强的全局搜索能力, 搜索遍及整个解空间, 容易得到全局最优解空间。改进的迁移算法能很好地适应网络动态变化的特征。

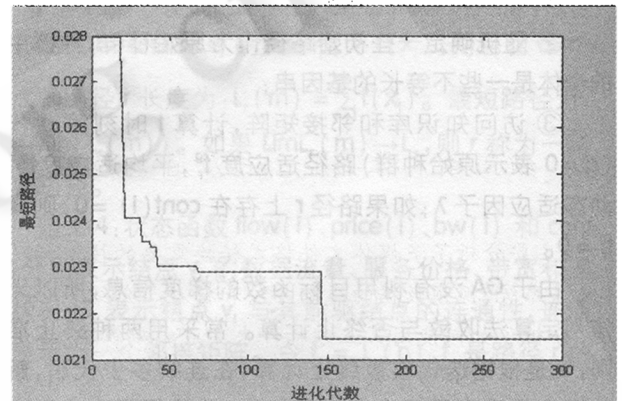


图 5  $t$  时刻最短路径迭代曲线

## 参考文献

- 1 张云勇、刘锦德, 移动 agent 技术[M], 北京 清华大学出版社, 2004(2): 2~4, 163~165。
- 2 A Acharya, M Ranganathan, J Saltz. A Language for Resource Aware Mobile Programs. J Vitek, C Tschudineds Proc. Of Mobile Object Systems: Towards the Programmable Internet Berlin: Springer, 1997, 111~130.
- 3 A Iqbal, J Baumann, M Straber. Efficient Algorithms to Find Optimal Agent Migration Strategies. Stuttgart University, Tech Rep: TR-1998-05, 1998.
- 4 B Brewington, R Gray, K Moizumi. Mobile Agents in Distributed Information Retrieval. M Klusch ed. Intelligence Information Agent. Berlin: Springer Verlag, 1999, 355~395.
- 5 刘大有、杨博, 基于旅行图的移动 Agent 迁移策略, 计算机研究与发展, 2003, 40(6): 838~845。
- 6 杨云、孙向军, 一种启发式遗传算法及其在最短路径求取中的应用, 计算机工程与应用, 2003(1): 12~14。