

基于改进信息素的蚁群算法 在 QoS 组播路由中的研究^{*}

陈 暄¹, 万志平¹, 许方恒¹, 龙 丹²

(1. 浙江工业职业技术学院, 浙江 绍兴 312000; 2. 浙江大学, 杭州 310058)

摘 要: 针对传统的蚁群算法在求解大规模旅行商问题时容易导致搜索时间过长或陷入停滞的问题, 提出了一种基于改进信息素的蚁群算法。通过蚁群算法的改进, 使得每轮搜索之后的信息素都能更好地反映解的质量。实验仿真结果表明, 改进后的蚁群算法能获得比传统的蚁群算法更优的解, 同时具有更快的收敛速度和较好的稳定性。

关键词: 信息素; 蚁群算法; 服务质量; 收敛速度; 稳定性

中图分类号: TP393 文献标志码: A 文章编号: 1001-3695(2012)11-4296-04
doi:10.3969/j.issn.1001-3695.2012.11.076

Research based on improved ant colony algorithm in QoS multicast routing

CHEN Xuan¹, WAN Zhi-ping¹, XU Fang-heng¹, LONG Dan²

(1. Zhejiang Industry Polytechnic College, Shaoxing Zhejiang 312000, China; 2. Zhejiang University, Hangzhou 310058, China)

Abstract: QoS multicast routing problem has been widely used for solving complex optimization problems in various engineering and science fields. In order to solve the time or stagnant problems in large-scale traveling salesman problem by using ant colony algorithm, this paper proposed an ant colony algorithm based on the improved pheromone. The improved algorithm made the pheromone after searching reflect solution better and better. The results of simulation experiments show that, based on the pheromone adjustment improved ant colony optimization algorithm can obtain better solution than the basic ant colony algorithm, and increases the stability of the algorithm.

Key words: pheromone; ant colony algorithm; QoS; rate of convergence; stability

0 引言

随着 Internet 的发展, 用户的信息化程度逐渐增加, 对网络的要求逐渐增高。当前的 Internet 只能提供力所能及的服务, 而下一代网络要求能够支持多种不同的引用, 提供面向服务质量 (quality of service, QoS) 的服务, 这就引出了网络的 QoS 问题^[1]。当前, QoS 路由技术是网络支持 QoS 保证的关键技术之一, 在这方面已经有不少的研究成果。在 QoS 网络中, 不仅要保持网络的连通性, 还要满足用户对于网络的要求。在网络中的 Internet 路由算法如 OSPF、RIP 等, 都是基于 Bellman-Ford 或者 Dijkstra 算法, 它们都不能对 QoS 提供很好的支持。美国 Michigan 大学 Holland 教授提出的遗传算法, 以达尔文生物进化理论和孟德尔遗传变异理论为基础, 模拟生物进化过程^[2], 具有并行搜索、群体寻优的特点, 已广泛应用于各种具有 NP-Compute 复杂性的问题。蚁群算法是 1992 年由意大利学者 Dorigo 等人首先提出的, 它是对蚂蚁的觅食行为进行模拟而得出的一种模拟进化算法。该算法不依赖于具体问题的数学描述, 有很强的全局优化能力和本质上的并行性, 在许多组合优化问题上都取得了较好的效果, 如 TSP 问题等^[3]。然而对于 QoS 路由问

题, 上述算法均表现出了一定的缺陷, 主要表现在遗传算法对系统中的反馈信息利用不够, 在求解过程中, 当运算到一定范围时, 就会产生大量的冗余迭代, 增加了系统的负担, 精确解的效率低下; 蚁群算法初期信息素匮乏, 求解速度慢^[4-6]。

针对基本蚁群算法在求解路由问题时容易出现搜索时间过长或者陷入停滞等缺陷问题, 本文以基本蚁群算法作为基础, 提出了一种基于信息素的改进蚁群算法。

1 基本蚁群算法简介

1.1 蚁群算法简介

蚂蚁外出寻找食物时, 会在自己经过的路上释放一种叫做信息素的物质, 该物质能够引导后续的蚂蚁选择即将要走的路, 后续蚂蚁会选择信息素浓度高的路径。这样就造成了某条路径在短时间内通过的蚂蚁数目多, 造成信息素也比较多, 从而后续的蚂蚁走这条路径的数量就会多。这样就形成了一个正反馈机制, 最后导致所有的蚂蚁都会从蚁穴到食物源的最短路径。

1.2 状态转移机制

假设 m 表示蚂蚁的个数, d_{ij} 表示两个城市 i 与 j 之间的距

收稿日期: 2012-02-23; 修回日期: 2012-03-29 基金项目: 国家自然科学基金资助项目 (30900358/C100701); 浙江省教育科学规划课题 (SCG366); 浙江省科技厅计划资助项目 (2011R30008); 绍兴市教育科学规划课题 (SGJ12078)

作者简介: 陈暄 (1979-), 男, 系统分析师, 信息系统项目管理师, 硕士, CCF 会员, 主要研究方向为计算机体系结构、软件方法学、算法设计 (chenxuan1979@sina.com); 万志平 (1978-), 男, 副教授, 主要研究方向为控制工程、嵌入式与 EDA 技术; 许方恒 (1980-), 男, 讲师, 硕士研究生, CCF 会员, 主要研究方向为数据库与信息系统、信息安全; 龙丹 (1975-), 男, 博士, 主要研究方向为图像处理和图像分析、算法设计。

离, $\eta_{ij} = 1/d_{ij}$ 是启发式信息, τ_{ij} 表示 ij 边的信息素量, $p_{ij}^k(t)$ 是 t 时刻位于城市 i 的第 k 只蚂蚁选择城市 j 的概率, 则

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

其中: allowed_k 为当前可以选择的城市集合。

在式(1)中, 用禁忌表 tabu_k ($k = 1, 2, \dots, m$) 来记录蚂蚁 k 当前所走的城市, 集合会随着禁忌表的进化过程进行动态的调整, $\text{allowed} = \{C - \text{tabu}_k\}$ 表示蚂蚁 k 下一步允许选择的城市。 α, β 分别表示 τ 和 η 在计算中的权重。当 $\alpha = 0$ 时, 最靠近的城市最有可能被选择到, 其算法的本质就是贪心算法; 当 $\beta = 0$ 时, 只有当信息素起作用时, 会变成纯粹正反馈随机的启发式的搜索算法。当所有的蚂蚁布局完路线时, 信息素将被更新。虽然线路上所有边的信息素都将挥发一部分, 但当蚂蚁走过之后, 所走的边上增加信息素。

1.3 信息素更新规则

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (2)$$

其中: ρ 表示信息素挥发的因子, $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在 t 时刻通过 (i, j) 路径时释放出的信息素。

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁在本次循环中经过 } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

其中: L_k 表示信息素与第 k 只蚂蚁在 t 时刻走过的路线长度。从这里可知道, 蚂蚁在所走过的路线上释放过的信息素的量与路线长度成反比, 与线路的质量成正比, 信息素大的边被选择的机会大。通过正反馈机制, 可以发现最终收敛一个最优的或者近似最优的解, 但随着路径的增大, 那就容易出现搜索时间过长的现象。把信息素集中到最优路径上, 从而出现停滞的现象, 就容易出现进行到一定的过程后, 所有个体所发现的解完全一致, 停止对空间作进一步的搜索, 这样也许会丢失最好的解。

1.4 改进的信息素规则

基于状态转移机制的搜索算法中的问题发现蚁群算法的收敛速度慢, 不能快速地引导蚂蚁走向最优解, 主要的原因是在于基本蚁群算法中的信息素的更新不能准确地反映出路径解的优劣程序, 路径差的信息素与优质路径的信息素差别不大, 才造成了选择路径时的收敛速度, 影响最优解。本文在此基础上对信息素公式进行改动, 采用新的策略。

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{e^{l_k \times t \times 10^{-R}}} & ij \in l_k \quad t = \frac{\min \text{ length}}{\max \text{ length}} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

其中: l_k 表示第 k 只蚂蚁走过的路径; t 是系数, 在 $(\min \text{ length}, \max \text{ length})$ 之间; R 是一个 TSP 问题中的参数。

在针对 TSP eil51 问题中, 假设 Q 的值为 100, 在一次搜索路径的过程中, 五只蚂蚁找得到的路径长度分别为 100、120、150、180、200。按照信息素表达式(3), 得到表 1。

表 1 五只蚂蚁的信息素结果

(Q, l_i)	(100, 100)	(100, 120)	(100, 150)	(100, 180)	(100, 200)
$\Delta\tau_{ij}^k(t)$	1	0.83	0.67	0.56	0.5

从表 1 可看出, 虽然距离相差得比较大, 但是信息素的增量却是非常小。

2 QoS 路由问题的数学描述

2.1 网络模型

网络可以描述为一个无向赋权图 $G = (V, E)$ 。其中: 图的顶点表示网络节点, 图的边表示网络链路, V 为网络中所有交换节点集, E 为网络链路集 (对于有向图可以通过变换解决)。 $E \in (V, V)$ 表示单点投递情况下网络中 (源节点、目的节点) 的集合, $p(s, d)$ 表示从源节点 s 到目的节点 d 的所有路径 $p(s, d)$ 的集合。

2.2 QoS 的路由指标

一般的 QoS 路由问题是保证用户将数据从源点传送到终点, 来提供端到端的质量保证。其质量指标是时延 (delay)、费用 (cost)、带宽 (width)、时延抖动 (delay_jitter) 和丢包率 (pack_loss)。这些因素构成了 QoS 路由问题的约束条件。在满足时延、带宽、时延抖动、丢包率这四个条件下, 保证费用是最低的^[7]。

在网络模型中, 对于任意一条链路中, 存在四个属性, 分别是时延函数、费用函数、带宽函数和时延抖动函数; 对于任意一个节点, 也存在四个属性, 分别为时延函数、费用函数、时延抖动函数和丢包率函数。对于 $s \in V, d \in E$, 存在如下函数:

a) 时延

$$\text{delay}(p(s, d)) = \sum_{e \in p(s, d)} D(e) + \sum_{n \in p(s, d)} D(n)$$

$D(e): E \rightarrow R^+$, 链路 e 上的时延

$D(n): V \rightarrow R^+$, 节点 n 上的时延

b) 带宽

$$\text{width}(p(s, d)) = \min_{e \in p(s, d)} \{B(e)\}$$

$B(e): E \rightarrow R^+$, 链路 e 上的时延

c) 费用

$$\text{cost}(p(s, d)) = \sum_{e \in p(s, d)} C(e) + \sum_{n \in p(s, d)} C(n)$$

$C(e): E \rightarrow R^+$, 链路 e 上的费用

$C(n): V \rightarrow R^+$, 节点 n 上的费用

d) 时延抖动

$$\text{delay_jitter}(p(s, d)) = \sum_{e \in p(s, d)} Dj(e) + \sum_{n \in p(s, d)} Dj(n)$$

$Dj(e): E \rightarrow R^+$, 链路 e 上的时延

$Dj(n): V \rightarrow R^+$, 节点 n 上的时延

e) 丢包率

$$\text{packet_loss}(p(s, d_i)) = 1 - \prod_{i=1}^{n-1} (1 - \text{packet_loss}(p(s, d_{i-1})))$$

2.3 QoS 路由问题

QoS 采用基于源路由的路由选择机制, 即网络中每一个节点均维护一张全局的网络拓扑及其状态参数, 源节点利用这些信息计算并找出可行的路径^[8]。

QoS 路由问题的优化目标就是要选择一条从源节点 s 到目的节点 d 的路径 $p(s, d)$, 使得目标函数 F 最小化, 且满足以下各个约束条件:

a) 时延的约束, $\text{delay}(p(s, d)) \leq D_p$;

b) 带宽的约束, $\text{width}(p(s, d)) \geq B_p$;

c) 时延带宽的约束, $\text{delay_jitter}(p(s, d)) \leq DJ_p$;

d) 丢包率约束, $\text{packet_loss}(p(s, d_i)) \leq \text{packet_loss_rate}$;

e) 费用约束, 在所有满足时延和带宽瓶颈的路径下, 保证 $\text{cost}(p(s, d))$ 是最小的。

在满足以上四个约束条件下, $\text{cost}(p(s, d))$ 是最小的。由于路由的四个条件之间没有任何关系, 是一个 NP 完全问题。由于蚁群算法中各路径的信息量都是具有很大的不确定性, 从全局的角度出发, 在进行路由选择时, 路径也表现出了非常大的不确定性, 而信息素是作为不确定性的主要量度, 可以尝试把改进后的信息素的更新规则引入到蚁群算法中, 来消除这种不确定性^[9~12]。

3 基于改进型信息素规则在 QoS 中的应用

QoS 网络路由优化的问题就是解决在约束条件下的 TSP 问题。由于信息素更新的蚁群算法在解决 TSP 问题中具有很好的优越性, 所以可以将信息素的更新应用到 QoS 网络中。经过以上的分析, 可以得知 QoS 网络优化问题其实就是要满足以上四个约束的问题, 尽可能地取得最短的路径来作为最优路径。

以上面所举出的五只蚂蚁为例, 在路径迭代的过程中, 花费分别为 10、15、20、25、30, 设 $Q=10$, 得到的结果如表 2 所示。当采用新的信息素更新规则后, 设 $Q=10, R$ 为 0, 得到的结果如表 3 所示。

表 2 五只蚂蚁采用式(3)的信息素

(Q, L_i)	(10, 10)	(10, 15)	(10, 20)	(10, 25)	(10, 30)
$\Delta\tau_{ij}^k(t)$	1	0.67	0.5	0.4	0.33

表 3 五只蚂蚁采用式(4)的信息素

(Q, L_i)	(10, 10)	(10, 15)	(10, 20)	(10, 25)	(10, 30)
$\Delta\tau_{ij}^k(t)$	0.37	0.674	0.008 25	0.002 74	0.000 045

从表 2、3 可看出, 采用了新的信息素公式之后, 信息素的增量之间的改变是非常明显的, 这就在很大程度上降低了最差路径的选择方式, 降低了几率, 加快了算法的收敛。

3.1 目标函数

QoS 路由算法的关键是在全局信息素更新时, 限制条件以确定惩罚因子的数值, 从而调整目标函数的数值, 判断下一个时刻 $\Delta\tau_{ij}^k(t + \Delta t)$ 函数的大小, 在第 k 只蚂蚁完成一次寻径之后, 其目标函数的值为

$$L_k = \frac{a \times f_d + b \times f_{dj} + c \times f_{pl}}{\cos t[p(s, d)]}$$

$$f_d = \Phi_d \{ \text{delay}(p_k(s, d)) - D \}$$

$$f_{dj} = \Phi_{dj} \{ \text{delay_jitter}(p_k(s, d)) - DJ \}$$

$$f_{pl} = \Phi_{pl} \{ \text{delay_loss}(p_k(S, M)) - PL \}$$

$$\Phi_d(Z) = \begin{cases} 1 (Z \leq 0) \\ r_d (Z > 0) \end{cases}$$

$$\Phi_{dj}(Z) = \begin{cases} 1 (Z \leq 0) \\ r_{dj} (Z > 0) \end{cases}$$

$$\Phi_{pl}(Z) = \begin{cases} 1 (Z \leq 0) \\ r_{pl} (Z > 0) \end{cases}$$

其中: a, b, c 分别是 f_d, f_{dj}, f_{pl} 的加权系数, 表示时延、时延抖动、丢包率在目标函数 L_k 中所占的比例。根据不同的应用场合进行应用, 其值可根据具体应用来进行设定。

$\Phi_d(Z)$ 是延迟度量的控制函数, 当 $p_k(s, d) \leq D$ 时, 值为 1; 否则为 r_d 。 $\Phi_{dj}(Z)$ 是时延抖动控制函数, 当 $p_k(s, d) \leq DJ$ 时, 值为 1; 否则为 r_{dj} 。 $\Phi_{pl}(Z)$ 是丢包率控制函数, 当 $p_k(S, M) \leq PL$ 时, 值为 1; 否则为 r_{pl} 。 r_d, r_{dj}, r_{pl} 取值为 (0, 1)。通过比较, 取其值最大者为 L_k , 根据式(2)来调整信息素的大小。对于搜索到的任何一个路径, 此路径在同时满足以上四个约束条件时, 费

用最小, 路径最优。

3.2 算法设计

此算法采用了信息素更新的蚁群算法来解决网络路由问题。算法如下:

a) 初始化。初始化各个参数, 设置请求路由源节点 v_s 、目的节点 v_d 、带宽约束 width 、时延约束 delay , 删除不满足带宽约束的链路, 初始化信息素表, 将 m 只蚂蚁置于初始化城市 S 上。

b) 对于蚂蚁 k , 按照概率 p_{ij}^k 在初始节点领域内随机地选择下一个要访问城市 j , 并将节点置于禁忌表 Tabu_k 中, 按照更新方程更新信息素。

c) 如果蚂蚁遇到了不可越的障碍, 转向步骤 e); 否则, 转向步骤 b), 直到蚂蚁达到目的地节点。

d) 如果蚂蚁按照禁忌表 Tabu_k 中记录的节点回溯到起始节点, 则按照 b)d) 进行更新。如果蚂蚁达到目的节点, 计算目标函数 L_k 的值, 选择 r_d, r_{dj}, r_{pl} 取值为 0.5。将该路径加入最优路径集合 L_k^* 中, 转步骤 f)。

e) 蚂蚁按照禁忌表 Tabu_k 表中记录的城市回溯到初始的城市。

f) 算法执行结束, 输出最优路径集合 $L_{\max}$ 。

3.3 仿真实验与结果分析

通过程序实现本文提出的基于信息素改进的蚁群算法在解决网络模型中的应用。选择的网络模型与文献[13]一致, 如图 1 所示, 在各个节点的 (d, dj, pl, c) 和每条边的 (d, dj, b, c) 的描述中, d 表示时延, dj 表示时延抖动, pl 表示丢包率, b 表示带宽, c 表示代价。

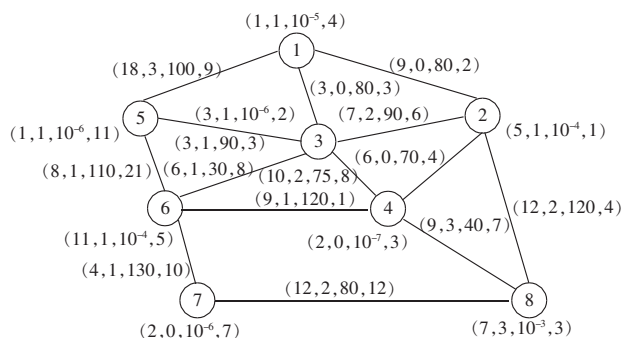


图 1 八个网络节点网络结果模型

在该仿真实验中, 以节点 1 作为源点, 以节点 2、5、7、8 为目的节点, 寻找满足约束条件的最优路径。约束条件是 $d=46, dj=8, b=60, pl=0.01$ ^[13]。

实验仿真是在 VC++ 6.0 语言编程设计的环境中进行的。在本实验中, 设定 a, b, c 的值均为 1, 即在满足 QoS 限制下, 考虑费用函数的制约作用。其步骤如下:

a) 选择 QoS 模型, 如图 1 所示。

b) 初始化不满足的约束条件。去掉图 1 中的节点 (3, 6) 之间的边和 (4, 8) 之间的边。

c) 选择蚂蚁的数目为 20, 迭代寻求次数 $N=200$ 。设置 α, β 的值。

d) 使用本文的算法, 让 20 只蚂蚁从源节点 1 出发, 经过了节点 2、4、5、7, 寻找满足约束条件的最优解的集合。

e) 在进行一次寻找之后, 重复步骤 c) 和 d), 并设置不同的参数, 进行多次的仿真实验。

根据上面的步骤,从实验中获得了最优组播路由结果:5-3-1-2-4-6-7。最大时延为 46,最大时延抖动为 5,最大丢包率为 0.002 99,最小总代价为 41。得到组播树的结果如图 2 所示。时延、时延抖动、带宽、丢包率和最小代价等多约束条件下的求解的时延、时延抖动、代价曲线如图 3、4 所示。图 5 为 QoS 路由响应代价。

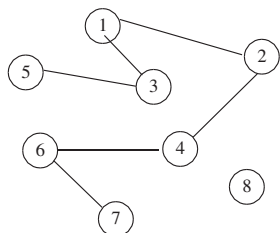


图 2 最佳组播树结果

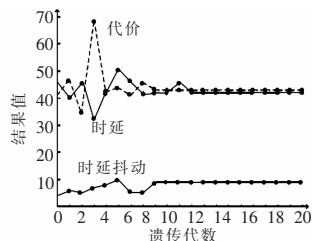


图 3 文献[13]的曲线

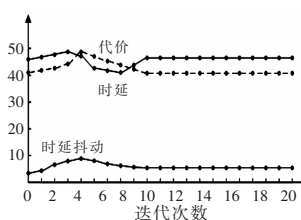


图 4 本文算法的曲线

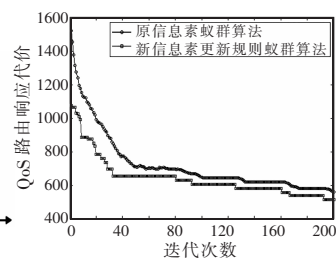


图 5 QoS 路由响应代价

从图 3、4 中可以看出,本文算法在进行信息素的更新之后,曲线图中代价的波动说明扩大了全局的搜索能力,渐渐地趋于平缓,加快了收敛的速度;但从时延抖动曲线、时延曲线以及代价上都要比一般的蚁群算法要平稳,也能够找到最优解。从图 5 中可以看出,在相同的环境下,采用信息素更新规则的蚁群算法与基本蚁群算法相比,只需更少的迭代次数就可以找到最优解,求解效率明显优于基本的蚁群算法。可以看到本文所提出的信息素改进的蚂蚁算法在解决问题上的平均解要优于基本蚁群算法,而在 200 次迭代次数重复实验中,说明信息素改进的蚂蚁算法具有很强的搜索特性,使其响应时间缩短了,在有效性方面有了实质性的提高。

4 结束语

通过上面的仿真可以看出,采用基于信息素更新策略的算法可以针对基本蚁群算法来解决 QoS 问题的两个主要缺点:

a) 初期信息素匮乏,蚂蚁搜索比较盲目,收敛速度慢;b) 算法容易停滞,陷入局部最优化。该算法有效地减少了蚂蚁在路由方面的相应代价,并通过仿真实验进行了验证。实验结果表明信息素的改进提高了基本蚁群算法在 QoS 问题上的求解能力。

多数成果都只是基于大量实验的数据分析,其中各种参数的选取也比较复杂,所以从算法的理论方面还有许多需要解决的问题。此外,蚁群算法在其他组合优化问题上的应用以及群集智能在其他研究领域的应用也都是值得进一步研究的问题。

参考文献:

- [1] WANG Zheng, CORWCROFT J. Quality-of-service routing for supporting multimedia applications[J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228-1234.
- [2] 张军英,王德峰,石美红. 基于点火耦合神经网络的多约束 QoS 路由选择算法[J]. 通信学报, 2002, 23(7): 40-46.
- [3] 吕国英,刘泽民,周正. 基于蚂蚁算法的分布式 QoS 路由选择算法[J]. 通信学报, 2001, 22(9): 34-42.
- [4] 王健,朱建启. 全局自适应蚁群优化算法[J]. 小型微型计算机系统, 2008, 6(6): 1083-1087.
- [5] 唐文渊,张洪伟,汪恒江. 蚁群算法在 QoS 中的应用[J]. 现代计算机: 专业版, 2008(1): 31-33.
- [6] 孟非,李静宜,朱人杰. 蚁群算法中蚂蚁更新方法之研究[J]. 计算机工程与应用, 2011, 47(25): 54-57.
- [7] LAWLER E L, LENSTRA J K, KAN A H G R, et al. The travelling salesman problem[M]. New York: John Wiley, 1985.
- [8] 陈峻,沈洁,秦玲. 基于分布均匀度的自适应蚁群算法[J]. 软件学报, 2003, 14(8): 1379-1387.
- [9] 刘永娟. 改进蚁群算法在 QoS 路由问题中的应用与研究[J]. 通信技术, 2008, 41(9): 128-129, 133.
- [10] 张纪会,徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240-1245.
- [11] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [12] 王菁,黄晓放. 基于改进蚁群算法的 QoS 单播路由优化[J]. 系统仿真学报, 2009, 21(19): 6081-6085.
- [13] 孙力娟,王汝传. 基于蚁群算法和遗传算法融合的 QoS 组播路由问题求解[J]. 电子学报, 2006, 34(8): 1391-1395.

(上接第 4295 页)有较大影响,本文改进的 PSO 通过延长迭代的开始和最后阶段的搜索时间,较好地调整了全局与局部搜索能力之间的平衡;同时针对混沌支持向量机预测模型在网络流量预测中存在参数选择困难的问题,将改进后的 PSO 算法与混沌支持向量机预测模型相结合,对支持向量机进行参数优化。通过实例证明,改进后的预测模型具有较高的寻优速度与预测精度,在网络流量预测中具有很好的应用前景。

参考文献:

- [1] 陆锦军,王执铨. 基于混沌特性的网络流量预测[J]. 南京航空航天大学学报, 2006, 38(2): 217-221.
- [2] LU Jin-jun, WANG Zhi-quan. Internet traffic data flow forecast by RBF neural network based on phase space reconstruction[J]. Transactions of Nanjing University of Aeronautics & Astronautics, 2006, 23(4): 316-322.
- [3] 张颖路. 基于遗传算法优化支持向量机的网络流量预测[J]. 计算机

机科学, 2008, 35(5): 177-179.

- [4] 罗赞骞,夏靖波,王涣彬. 混沌—支持向量机回归在流量预测中的应用研究[J]. 计算机科学, 2009, 6(7): 244-246.
- [5] 陈铿,韩伯棠. 混沌时间序列分析中的相空间重构技术综述[J]. 计算机科学, 2005, 32(4): 67-70.
- [6] 刘靖旭,蔡怀平,谭跃进. 支持向量回归参数调整的一种启发式算法[J]. 系统仿真学报, 2007, 19(7): 1540-1543.
- [7] CHERKASSKY V, MA Yun-qian. Practical selection of SVM parameters and noise estimation for SVM regression[J]. Neural Networks, 2004, 17(1): 113-126.
- [8] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proc of IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 1995: 1942-1948.
- [9] CHATTERJEE A, SIARRY P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers & Operations Research, 2006, 33(3): 859-871.

g) 计算每个粒子在新位置上的适应值,如果粒子的适应值优于 $pbest_i$ 的适应值,则 $pbest_i$ 更新为新位置;如果粒子的适应值优于 $gbest$,则 $gbest$ 更新为新位置。

h) 判断粒子群最优位置 $pbest_i$ 的适应值是否满足目标值或迭代次数是否达到最大值,如果满足停止条件,则输出最优参数组合,建立基于 IPSO 优化的混沌支持向量机模型;如果不满足,则返回 b)。

3 实例分析

3.1 训练样本的确定

本文数据源于流量文库: <http://newsfeed.ntcu.net/~news/2006>, 搜集了自 2006 年 7 月 21 日至 9 月 3 日共 45 天网络每小时访问流量,得到 $45 \times 24 = 1\,080$ 个数据;用前 40 天的每小时的流量数据训练模型、后 5 天的流量数据作为预测样本来检验模型的预测效果。

获得原始流量后,首先采用互信息法对前 40 天流量序列进行处理,得到延迟时间为 3 h。然后利用 Cao 方法求嵌入维数,得到流量时间序列相空间重构的最小嵌入维数 $m = 10$ 。利用 Sato 改进的小数据量法得到最大 Lyapunov 指数为 $\lambda = 0.0167$,说明该网络流量时间序列具有混沌特性。利用式(1)对流量序列进行相空间重构,得到 $960 - 3 \times (10 - 1) - 1 = 932$ 个训练样本,并利用支持向量机回归对这些样本进行训练。为了提高收敛速度、缩短训练时间,对生成的训练样本进行归一化处理,归一化式为 $x(t) = (x(t) - \min(x)) / (\max(x) - \min(x))$ 。

3.2 实例仿真

为了验证基于改进 PSO 优化的混沌支持向量机预测模型在网络流量预测中的有效性,本文共做了四组实验:实验 1,采用基于改进 PSO 优化的模型进行预测;实验 2,采用基于传统 PSO 优化的模型进行预测;实验 3,采用基于 GA 优化的模型进行预测;实验 4,采用基于网格搜索优化的模型进行预测。考虑到智能优化算法对模型的最优参数进行搜索时的随机性和网格搜索结果的唯一性,实验 1、2、3 各运行 10 次,实验 4 运行 1 次。

本文所有的实验都是在 MATLAB 平台上开发,利用 Libsvm 工具箱进行回归预测模型和交叉验证算法的实现,用 5-fold 交叉验证误差作为参数选择的准则。根据实验的初步验证,得到参数的搜索区间为 $C \in [0.1, 500]$, $\delta \in [0.001, 30]$, $\varepsilon \in [0.001, 0.1]$ 。在改进 PSO 算法中,设置种群规模为 $ps = 20$,最大迭代次数为 $me = 100$, $w_{\max} = 0.9$, $w_{\min} = 0.4$, $c_1 = c_2 = 2$, $V_{\max}$ 通常设置为参数变化范围的 10% ~ 20%,这里设置为 $V_{\max} = [50, 3, 0.05]$;在标准 PSO 算法中,采用线性递减惯性权重,其他参数与改进 PSO 一致;对于遗传算法,采用二进制编码的遗传算法的参数设置为:交叉概率 = 0.7,变异概率 = 0.1,种群个数和最大迭代次数同改进 PSO 算法;在网格搜索中,步长设置为 1,迭代截止误差 $\varepsilon = 10^{-5}$, C 、 ε 和 δ 的范围为: $C = \{2^{-4}, 2^{-3}, \dots, 2^8, 2^9\}$, $\delta = \{2^{-10}, 2^{-9}, \dots, 2^{-1}, 2^5\}$, $\varepsilon = \{2^{-10}, 2^{-9}, \dots, 2^{-3}\}$ 。在搜索过程中满足目标值或达到最大迭代次数则停止搜索。

3.3 结果分析

图 2、3 分别描绘了基于改进 PSO 算法和传统 PSO 算法优

化混沌支持向量机在 100 次迭代寻优过程中的适应值变化过程。

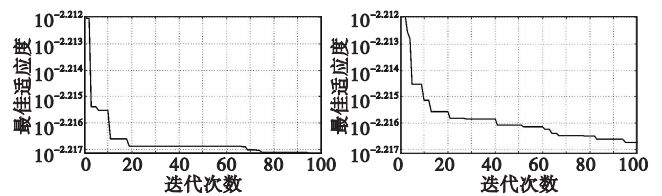


图2 基于改进PSO的迭代寻优过程

图3 基于传统PSO的迭代寻优过程

通过对比可以看出,在迭代初期,基于改进 PSO 的迭代寻优过程中具有较大的惯性权重,粒子全局搜索能力较强,寻优可以快速地收敛到最优解的大致位置;到了迭代的后期,基于改进 PSO 的迭代寻优过程中保持了较小的惯性权重,粒子速度减慢,局部搜索能力增强,算法具有更高的学习精度。

表 1 为基于改进 PSO 和传统 PSO 优化的预测模型分别运行 10 次的预测效果对比。其中 MAPE 为平均相对误差 $\frac{1}{N} \sum_{i=1}^N$

$\left| \frac{y_i - y'_i}{y_i} \right| \times 100\%$, MSE 为均方误差 $\frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2$, y_i 、 y'_i 分别为某时刻的实际和预测流量。

表 1 基于传统 PSO 和改进 PSO 的模型预测效果比较

实验次数	传统 PSO-混沌支持向量机		改进 PSO-混沌支持向量机	
	MAPE/100%	MSE	MAPE/100%	MSE
1	4.651 58	536.202	4.656 83	536.780
2	4.680 89	541.178	4.653 14	536.188
3	4.680 08	542.078	4.656 00	536.584
4	4.661 93	537.013	4.646 84	536.407
5	4.666 40	537.347	4.654 21	536.246
6	4.654 59	536.167	4.652 78	536.376
7	4.657 92	536.538	4.653 40	536.370
8	4.683 29	542.155	4.657 68	536.678
9	4.659 53	536.893	4.648 36	536.018
10	4.659 29	536.553	4.655 75	536.048
平均	4.665 55	538.212	4.653 50	536.370

从表 1 可以看出,基于 PSO 优化的模型得出的预测效果之间差异较大,基于改进 PSO 优化的模型得出的预测效果稳定,差异很小,且后者预测精度明显高于前者。

表 2 为基于改进 PSO、传统 PSO、GA 和网格搜索优化的混沌支持向量机预测模型进行预测时的性能对比。从表 2 中可以看出,基于改进 PSO 优化的模型预测效果略优于 GA 算法和网格搜索法,但是寻优时间快出很多,所以基于改进 PSO 优化的预测模型建立效率最高。

表 2 预测模型性能比较

项目	改进 PSO-混沌支持向量机	传统 PSO-混沌支持向量机	GA-混沌支持向量机	网格搜索-混沌支持向量机
寻优时间/s	912	942	1261	2932
MSE	536.370	538.212	537.892	537.871
MAPE/100%	4.65350	4.66555	4.65236	4.65055

因此,利用基于改进 PSO 优化的混沌支持向量机模型进行网络流量预测时,模型具有较高的寻优效率、预测精度和较好的稳态性能。

4 结束语

惯性权重对 PSO 算法的收敛速度和精度 (下转第 4299 页)