

# 基于IPSO混沌支持向量机的网络流量预测研究\*

尹波<sup>1</sup>, 夏靖波<sup>1</sup>, 付凯<sup>1</sup>, 陈茂<sup>2</sup>

(1. 空军工程大学 电讯工程学院, 西安 710077; 2. 兰州军区通信网络技术管理中心, 兰州 730000)

**摘要:** 针对传统混沌支持向量机参数寻优算法的不足, 提出了一种改进的粒子群(IPSO)算法。该算法通过延长迭代的开始阶段和最后阶段的搜索时间, 实现了算法的全局搜索与局部搜索能力之间的平衡, 进而优化模型参数, 建立了基于IPSO优化的混沌支持向量机预测模型。应用实例结果表明, 该模型对网络流量预测是有效可行的, 并具有较高的寻优效率、预测精度和较好的稳态性能。

**关键词:** 网络流量预测; 混沌支持向量机; 改进粒子群算法; 遗传算法

**中图分类号:** TP393      **文献标志码:** A      **文章编号:** 1001-3695(2012)11-4293-03

doi:10.3969/j.issn.1001-3695.2012.11.075

## Study on network traffic prediction of C-SVM based on IPSO

YIN Bo<sup>1</sup>, XIA Jing-bo<sup>1</sup>, FU Kai<sup>1</sup>, CHEN Mao<sup>2</sup>

(1. Institute of Telecommunication Engineering, Air Force Engineering University, Xi'an 710077, China; 2. Communication Network Control Center of Lanzhou Military Area, Lanzhou 730000, China)

**Abstract:** Aiming the limitation of the conventional parameter optimized algorithm in C-SVM, this paper proposed an IPSO algorithm. The algorithm extended the search time in the beginning and final phase of the iterative course, in order to achieve the balance between global and local search capabilities. Then it optimized the parameters of C-SVM, and built the prediction model of C-SVM based on IPSO. The results of application example show that the model is efficient in network traffic prediction with higher optimized efficiency, prediction accuracy and better stability.

**Key words:** network traffic prediction; chaos support vector machine (C-SVM); improved particle swarm optimization (IPSO) algorithm; genetic algorithm (GA)

## 0 引言

随着互联网规模的不断扩大、业务量的急剧增长, 网络的可控性和可管性问题越来越突出。网络流量作为网络业务的直接载体, 其准确预测是实现科学规范的网络流量控制、网络安全管理和规划的重要基础, 因此对网络流量的预测是目前研究的一个热点问题。研究表明<sup>[1,2]</sup>, 网络流量存在一定的混沌特性, 可以通过一个非线性动力学系统近似, 这就启发了笔者通过将混沌理论中的相空间重构技术与机器学习相结合的方法来解决网络流量预测中复杂的非线性问题。文献[2]提出了基于相空间重构的网络流量 RBF 神经网络预测方法, 文献[3,4]提出了基于混沌支持向量机的网络流量预测方法。上述方法中, 由于神经网络基于经验风险最小化原则, 模型容易出现过学习现象, 从而导致泛化预测能力较差。支持向量机基于结构风险最小化原则, 学习能力强, 模型泛化能力好, 因此混沌支持向量机模型被广泛应用于网络流量预测中, 并取得了较好的效果<sup>[3,4]</sup>。

在采用混沌支持向量机模型进行网络流量预测时, 参数优化算法的选取十分重要。目前混沌支持向量机模型中常用的参数优化算法主要有网格搜索算法<sup>[4]</sup>和遗传算法<sup>[3]</sup>(GA)。

网格搜索算法在多个参数不同取值的所有可能组合上估计适应度的值, 当参数取值范围较大时, 计算费时。遗传算法操作比较复杂, 对于不同的优化问题需要设计不同的交叉和编译方式; 另外采用二进制编码, 会造成遗传操作时反复译码编码, 从而降低算法性能。针对上述寻优算法的不足, 本文提出了一种改进的粒子群算法。

## 1 混沌支持向量机模型

### 1.1 相空间重构

将混沌理论应用于网络流量预测时, 需要解决网络流量时间序列混沌性质的识别、系统相空间的重构和系统的混沌预测方法三个问题。时间序列性质的识别以及进一步的混沌分析, 都需要在相空间中进行, 所以相空间重构是网络流量时间序列分析的基础。

Takens 定理保证了可以从一维的混沌流量时间序列中重构一个与原动力系统拓扑意义下等价的相空间, 且不依赖重构过程的具体细节<sup>[5]</sup>。对于实测得一组网络流量时间序列  $x_1, x_2, \dots, x_{n-1}, x_n$ , 如果嵌入维数为  $m$ , 时间延迟为  $\tau$ , 采用坐标延迟法重构出的相空间向量为

$$X(i) = [x(i - (m-1)\tau), \dots, x(i - \tau), x(i)] \quad (1)$$

收稿日期: 2012-03-27; 修回日期: 2012-04-27      基金项目: 全军军事学研究生课题(2010XXXX-488); 陕西省自然科学基金资助项目(2009JM8001-1)

作者简介: 尹波(1988-), 男, 湖北汉川人, 硕士研究生, 主要研究方向为通信网管理、IP 网络流量分析(yinbo123\_cool@126.com); 夏靖波(1963-), 男, 河北秦皇岛人, 教授, 博导, 主要研究方向为军事通信网络管理与安全; 付凯(1987-), 男, 山东济宁人, 硕士研究生, 主要研究方向为容迟/容断网络; 陈茂, 男, 高工, 主要研究方向为通信网管理。

其中:  $i \in [(m-1)\tau, n]$ ,  $X(i)$  为重构后的相点。在重构的相空间中, 时间延迟和嵌入维数  $m$  的选取很重要, 本文采用互信息法计算时间延迟, 采用 CAO 方法对  $m$  进行选取, 采用 Sato 改进的小数据量法计算 Lyapunov 指数, 以检验网络流量时间序列是否具有混沌特性<sup>[4]</sup>。

## 1.2 模型的建立

根据式(1)对实际网络流量数据进行相空间重构, 得到新的数据空间,  $X(i)$  为重构后的相点。利用重构后的相点对流量数据进行单步预测, 可以构造映射(回归函数)  $f: R^m \rightarrow R$ , 使得  $y(i) = x(i+1) = f(X(i))$ 。设当前时刻为  $i$ , 训练数据数量为  $n$ , 那么训练样本可以表示为  $\{X(i), y(i), i = (m-1)\tau, \dots, n-1\}$ ,  $X_i \in R^m, y_i \in R, R^m$  代表  $m$  维输入空间。输入样本通过非线性映射函数  $\varphi(X)$  映射到高维的特征空间  $F$ , 并在  $F$  中求解最优回归函数, 这样就将低维空间中的非线性回归问题转换为高维空间中的线性回归问题<sup>[6]</sup>。在特征空间中对回归函数  $f$  定义为

$$f(X) = (w \cdot \varphi(X)) + b \quad X \in R^m, b \in R \quad (2)$$

其中:  $w$  和  $b$  分别为线性回归函数的法向量及偏移量。根据机构风险最小化原则, SVM 回归机可转换为如下优化问题, 即

$$\min \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

$$\text{s.t.} \begin{cases} y_i - (w \cdot \varphi(X_i)) - b \leq \varepsilon + \xi_i \\ (w \cdot \varphi(X_i)) + b - y_i \leq \varepsilon + \xi_i^* \quad i = 1, 2, \dots, l \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (3)$$

其中:  $\|w\|^2$  是与函数  $f$  复杂度相关的项;  $\varepsilon$  为不敏感损失系数; 常数  $C > 0$  为惩罚系数, 决定了大于  $\varepsilon$  的偏差容忍程度与函数扁平程度之间的均衡;  $\xi_i$  和  $\xi_i^*$  为松弛变量。

通过引入 Lagrange 函数, 并用核函数  $K(X_i \cdot X_j)$  代替高维空间中的向量内积  $(\varphi(X_i) \cdot \varphi(X_j))$ , 得到优化问题式(3)的最优解为

$$f(x) = \sum_{i=1}^l (a_i - a_i^*) K(X_i \cdot X) + b \quad (4)$$

研究表明, 当缺少过程的先验知识时, 径向基核函数相比其他核函数不仅具有较少的参数, 还具有良好的性能<sup>[7]</sup>。因此, 本文选择径向基核函数, 其具体形式为  $K(x, x') = \exp(-\|x - x'\|^2 / 2\delta^2)$ 。其中,  $\delta$  为核函数的参数。

## 2 IPSO 算法

### 2.1 标准 PSO 算法的介绍及 IPSO 算法的主要思想

在标准 PSO 算法中<sup>[8]</sup>, 假设在一个  $d$  维的目标搜索空间中有  $m$  个粒子组成一个群体, 其中第  $i$  个粒子的  $d$  维空间位置为  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ,  $i = 1, 2, \dots, m$ 。将  $X_i$  带入目标函数就可以算出其适宜度, 根据适宜度的大小就可以评价出该粒子的优劣, 进而求出最优解。第  $i$  个粒子的“飞翔”速度也是一个  $d$  维向量  $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ ,  $i = 1, 2, \dots, m$ , 其搜索到的历史最优位置为  $pbest_i = (p_{i1}, p_{i2}, \dots, p_{id})$ , 整个粒子群搜索到的最优位置为  $gbest = (p_{g1}, p_{g2}, \dots, p_{gd})$ 。找到这两个最优值时, 每个粒子根据式(7)和(8)来更新自己的速度和位置:

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot r_1 \cdot (pbest_i(t) - X_i(t)) + c_2 \cdot r_2 \cdot (gbest(t) - X_i(t)) \quad (5)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (6)$$

式中:  $t$  为当前迭代次数;  $V_i(t)$  为粒子第  $t$  代的飞行速度;  $X_i(t)$

为粒子第  $t$  代的当前位置;  $w$  为惯性权重, 它使粒子保持运动惯性;  $r_1$  和  $r_2$  为均匀分布在  $(0, 1)$  区间的随机数;  $c_1$  和  $c_2$  为加速因子, 分别调节向全局最优粒子和个体最优粒子方向飞行的最大步长, 通常取  $c_1 = c_2 = 2$ 。

惯性权重  $w$  代表粒子前一时刻自身运动状态对当前时刻的影响程度, 对 PSO 算法的收敛速度和精度有较大影响, 较大的  $w$  全局搜索能力较好, 收敛速度快; 而较小的  $w$  则局部搜索能力较强, 收敛精度高<sup>[9]</sup>。传统粒子群算法中一般采用的线性递减策略来动态地调整惯性权值  $w$ , 即  $w = w_{\max} - (w_{\max} - w_{\min}) \times t/t_{\max}$ 。但是线性递减策略存在一定的缺陷, 在迭代后期粒子接近全局最优解时, PSO 算法的局部搜索能力较弱, 可能会错过全局最优解, 造成算法精度不高。为此本文提出一种非线性的权重自适应更新方法, 该方法在保证惯性权重  $w$  随迭代递减的情况下, 通过延长迭代的开始阶段(全局搜索)和最后阶段(局部搜索)的搜索时间, 实现粒子的全局搜索能力与局部搜索能力之间的平衡, 提高了算法的快速收敛能力和精度。惯性权重  $w$  的计算公式为

$$w = w_{\max} - (w_{\max} - w_{\min}) \times \exp(-30 \times ((t_{\max} - t)/(t_{\max}))^n) \quad (7)$$

其中:  $w_{\max}$ 、 $w_{\min}$  分别取 0.9、0.4,  $t_{\max}$  为最大迭代次数, 这里分别取  $n = 3 \sim 7$ , 所得变化曲线如图 1 所示。

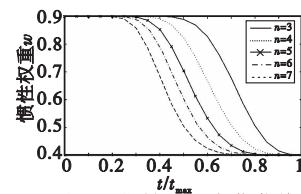


图1 惯性权重变化曲线

从图 1 中可知, 与其他值相比, 当  $n = 5$  时, 粒子的全局搜索时间和局部搜索时间都比较长, 因此公式中的  $n$  取 5。

### 2.2 基于 IPSO 算法优化的混沌支持向量机

对于混沌支持向量机模型, 惩罚系数  $C$ 、不敏感系数  $\varepsilon$ , 以及核函数中的系数  $\delta$  的选取与预测模型的学习精度和推广能力之间存在很重要的关系。因此, 选取合适的优化算法, 在一定的先验区间内搜索出最优的参数组合, 进而获得最佳性能的混沌支持向量机, 对网络流量预测具有重要的意义。基于此, 本文在利用改进 PSO 算法的全局搜索能力的基础上, 结合交叉验证算法对模型参数进行优化, 并在此基础上建立改进模型。具体步骤如下:

- 读入流量数据, 对数据进行相空间重构, 生成样本数据。
- 确定惩罚系数  $C$ 、不敏感系数  $\varepsilon$ , 以及核函数中的系数  $\delta$  在实际应用中可能的取值范围, 设置算法中种群最大数量、最大迭代次数、惯性权重  $w$  的计算式等。
- 根据各参数的取值范围, 确定粒子速度的变化范围为  $[-V_{\max}, V_{\max}]$ , 随机初始化粒子群  $(C, \delta, \varepsilon)$  中粒子的位置和速度。
- 读入样本数据, 以均方差作为适应度值, 采取交叉验证的方法训练模型得到每个粒子的适应度值, 取适应度最小的粒子所对应的个体极值最优为最初的全局极值  $gbest$ 。
- 按照式(5)(6)对粒子的位置和速度进行进化计算, (这里  $w$  按照式(7)进行计算), 更新粒子的位置和速度。
- 进行速度、位置限制。当  $V_i > V_{\max}$  或  $V_i < V_{\min}$  时, 令  $V_i = V_{\max}$  或  $V_i = V_{\min}$ ; 根据粒子反弹机制, 当  $X_i > X_{\max}$  或  $X_i < X_{\min}$  时, 令  $X_i = X_{\max}$ ,  $V_i = -V_i$  或  $X_i = X_{\min}$ ,  $V_i = -V_i$ 。

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  $\varepsilon = 10^{-5}$ ,  $C$ 、 $\varepsilon$  和  $\delta$  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。

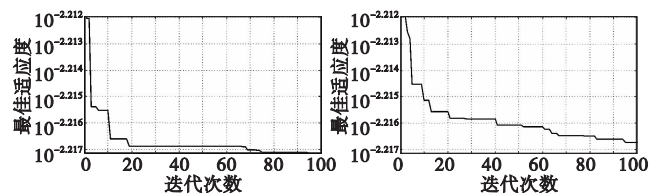


图2 基于改进PSO的迭代寻优过程

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)



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。

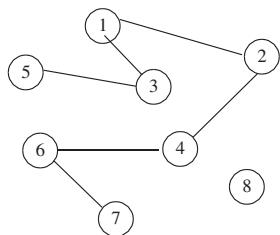


图 2 最佳组播树结果

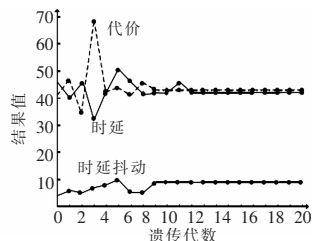


图 3 文献[13]的曲线

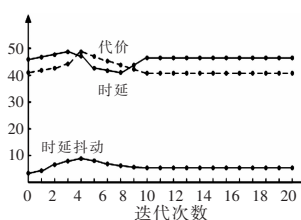


图 4 本文算法的曲线

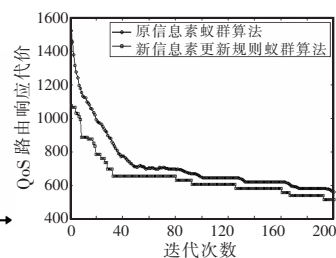


图 5 QoS 路由响应代价

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 Trans 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.