

改进的最短路径算法在多点路由上的应用

张毅^{1,2} 张猛¹ 梁艳春¹

(吉林大学计算机科学与技术学院国家教育部符号计算与知识工程重点实验室 长春 130012)¹

(吉林工商学院计算机系 长春 130062)²

摘要 Dijkstra算法是目前公认的较好的最短路径算法。由于多点路由问题最终归结为最短路径问题,因此将算法改进后应用于多点路由问题。提出的改进主要有以下3点:(1)改变选路策略,基于蚁群算法实现Dijkstra算法的选路操作,使选路更加灵活。(2)结合网络模型的特点,减少了对两顶点之间最短路径以外的大量顶点的计算,提高了算法的速度。(3)考虑到网络路由问题中的阻塞问题,对阻塞顶点进行标识,防止算法选择无用顶点。模拟实验结果表明改进算法较之Dijkstra算法在运算速度上有明显提高。

关键词 Dijkstra算法,蚁群算法,多点路由问题,选路策略,并行策略

中图分类号 TP301.6 **文献标识码** A

Application of an Improved Dijkstra Algorithm in Multicast Routing Problem

ZHANG Yi^{1,2} ZHANG Meng¹ LIANG Yan-chun¹

(Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China)¹

(Department of Computer Science, Jilin Business and Technology College, Changchun 130062, China)²

Abstract Three improvements on Dijkstra algorithm were presented. The improvements were given as follows: (1) A novel optimized base on ACO implementing approach is designed to reduce the processing costs involved with routing of ants in the conventional Dijkstra algorithm. (2) Based on the model of network routing, in order to reduce the counting of other points, the set of candidates is limited to the nearest c points. (3) Marking the flags on the blocked points in order to prevent selecting these points. By this way simulations show that the speed of convergence of the improved algorithm can be enhanced greatly compared with the traditional algorithm.

Keywords Dijkstra algorithm, ACO, Traveling salesman problem, Parallel strategy, Route strategy

1 引言

随着计算机的普及以及计算机网络的快速发展,网络路由问题日益得到相关学者的重视。而网络路由分析中最基本最关键的问题是最短路径问题。最短路径不仅仅指一般意义上两点之间的距离最短,还可以引申到其他的度量,如时间、费用、阻塞等。相应地,最短路径问题就成为最快路径问题、最低费用问题等。其实,无论是距离最短、时间最快还是费用最低,它们的核心算法都是最短路径算法。经典的最短路径算法——Dijkstra算法是目前多数系统解决最短路径问题采用的理论基础,只是不同系统对Dijkstra算法采用了不同的实现方法。

蚁群优化算法(Ant colony optimization ACO)是一种随机搜索算法,由若干个蚂蚁共同构造解路径,通过在解路径上遗留并交换信息素提高解的质量,进而达到优化的目的。该算法最早由意大利的M. Dorigo等人在1991年提出^[1-3]。

ACO的主要特征是正反馈和隐并行性。正反馈机制可以快速发现优化解。隐并行性通过多个个体之间的并行交换信息素可防止算法陷入局部最优解,并可使算法收敛于解空间的一个子集,有利于对解空间进行进一步搜索,发现较好解。在过去的10多年,蚁群算法(ACO)的研究和应用取得了很大的进展,大量结果证明了算法的有效性和在某些领域的优势^[5,6]。同时,我们也注意到该算法存在一定的缺陷,即搜索时间过长和容易陷入局部最优解^[14]。国内外的学者针对这些问题,提出了一些有效的方法^[7-13]。但问题并未完全消除,算法的执行效率问题仍是制约蚁群算法在大规模问题中有效应用的关键问题。

本文针对原始的最短路径算法在网络路由问题上的实现进行了改进。主要是将最短路径算法与蚁群算法相结合,利用蚁群算法的隐并行性提出了一种基于最短路径算法的改进算法——优化选路算法的最短路径算法。该算法通过减少基本蚁群算法中的选路次数,减小选路开销,提高算法的执行效

到稿日期:2008-09-28 返修日期:2009-04-03 本文受高等学校博士学科点专项科研基金(批准号:20030183060),吉林省教育厅科研项目(批准号:2007248,2007251,2008411)资助。

张毅(1972-),女,博士,副教授,主要研究领域为计算智能、计算机应用,E-mail: whdzy2000@vip.sina.com;梁艳春(1953-),男,博士,教授,博士生导师,主要研究领域为计算智能、参数识别、智能仿真建模。

率,并将这种策略实现在最短路径算法中。模拟实验结果表明在处理大规模路由问题上该算法较之基本蚁群算法和原始的最短路径算法在执行效率上有明显优势。

2 多点路由问题

多点通信路由问题的实质就是寻找连接一组点的基于某种代价的最小连接树^[1]。为简单起见,在分析多点路由问题时,可将网络看成无向带权连通图,用 $G(N, E, C)$ 代表,其中 N 表示网络节点集合, E 表示网络的链路,每一条边 $e \in E$ 的代价函数为 $C(e): C \in R^+ (R^+ \geq 0)$,典型的代价函数包括:链路上的延迟,可用资源,带宽和价格。多点通信路由问题为:给定顶点集合 DAN ,寻找 G 中一棵覆盖 D 的子树 $T(VT, VE)$,使 T 的代价最小。求解多点路由问题也可以看作求解 Steiner 树。

定义 1 给定无向带权连通图 $G(N, E, C)$ 和 N 的一个子集 D ,在 G 中寻找一个连通 D 的全部顶点的最小树,这个树被称为 Steiner 最短树。当 $D=N$ 时,所求得的 Steiner 树为最小生成树(MST)。当 $D < N$ 时,引入点 $SD \in N/D$ 可能使所求得的树长度更短。求解 Steiner 树的实质是寻找图中的 Steiner 点(SD)。Steiner 树属于图论中的 NP 完全问题,一般地说,最优算法无法在多项式时间内完成。而有效的启发式算法,虽算法难度降低,但只能得到次优解。由于构造最小生成树相对简单,因此许多启发式算法是基于 MST 进行的。

3 Dijkstra 算法的主要思想

Dijkstra 算法的基本思路是:假设每个点都有一对标号 (d_j, p_j) ,其中 d_j 是从起源点 s 到点 j 的最短路径的长度(从顶点到其本身的最短路径是零路(没有弧的路),其长度等于零); p_j 则是从 s 到 j 的最短路径中 j 点的前一点。求解从起源点 s 到点 j 的最短路径算法的基本过程如下:

(1) 初始化。起源点设置为: $d_s=0, p_s$ 为空;标记起源点 s ,记 $k=s$,其他所有点设为未标记的。

(2) 检验从所有已标记的点 k 到其直接连接的未标记的点 j 的距离,并设置: $d_j = \min[d_j, d_k + l_{kj}]$,式中, l_{kj} 是从点 k 到 j 的直接连接距离。

(3) 选取下一个点。从所有未标记的结点中选取 d_j 中最小的一个 $i: d_i = \min[d_j, \text{所有未标记的点 } j]$,点 i 就被选为最短路径中的一点,并设为已标记的。

(4) 找到点 i 的前一点。从已标记的点中找到直接连接到点 i 的点 j^* ,作为前一点,设置: $i=j^*$

(5) 标记点 i 。如果所有点已标记,则算法完全推出,否则,记 $k=i$,转到(2)再继续。

4 基本蚁群算法的原理

为更清楚地理解蚁群算法的基本原理,一般多借助经典的对称 TSP 问题来进行说明。TSP 问题的目标函数是 TSP 问题的目的是从有向图 G 中寻出长度最短的 Hamilton 圈。

基本蚁群算法可以简单表述如下:初始化,将 m 个蚂蚁随机放在 n 个城市上,城市间的每一条边都有初始化信息素 $\tau_{ij}(0)$,每个蚂蚁的禁忌表 $\text{Tabu}_k(s)$ 的第一个元素设置为其初始城市。然后每只蚂蚁开始选路,即选择下一步要去的城市。在选路中蚂蚁依据概率函数

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

选择将要去的城市,这个概率取决于城市间的距离和信息素的强度。其中 $\tau_{ij}(t)$ 表示边弧 (i, j) 上信息素的强度(i 为出发城市, j 为到达城市); η_{ij} 表示城市间距离因子,通常取值为 $1/d_{ij}$; α 表示信息素在选择概率上的作用; β 是指路径长度在选择概率上的作用。在 n 次循环后,所有蚂蚁的禁忌表都已填满,此时计算每个蚂蚁走过的路径的长度,并找到最短路径保存,记录此路径并更改信息素。重复这一过程直至达到最大周游值结束。

信息素更新的公式是:

$$\tau_{ij}(t+n) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad (2)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3)$$

其中 $\Delta \tau_{ij}$ 表示在某条边上的累加新增信息素的和, ρ 表示信息素消散的等级, $\Delta \tau_{ij}^k$ 表示 t 和 $t+n$ 之间第 k 个蚂蚁在此边上留下的信息素的数量。 $\Delta \tau_{ij}^k$ 的计算公式为:

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } k\text{-th ant use edge}(i, j) \text{ inits tour} \\ 0 & \text{others} \end{cases} \quad (4)$$

其中 Q 为常量, L_k 为第 k 个蚂蚁周游的路径长度。

由于该算法是典型的概率算法,算法中的参数通常由实验的方法确定^[1],其中对解有重大影响参数主要是 α 和 β ,它们主要控制信息素和距离两个因子对解的影响,另外,信息素的消散系数 ρ 对解的影响也很大。在基本蚁群算法中信息素更新方式对算法的运行效率和求解能力有很大的影响, M. Dorigo 曾给出 3 种不同的算法模型,分别称为 ant-cycle system, ant-quantity system 和 ant-density system^[2],它们的差别在于表达式(4)的不同。其中 ant-quantity system 和 ant-density system 是利用局部信息素的更新;而 ant-cycle system 是利用整体信息,在解决 TSP 问题上效果最好。因此我们采用 ant-cycle system 为基本蚁群算法(ACO)的模型,该算法的复杂度为 $O(NC \cdot n^2)$,其中 NC 为算法中的最大循环次数。

5 改进的基于最短路径算法思想的蚁群算法求解多点路由问题

国内外学者在最短路径问题已经取得了许多成果^[15-19],设计了很多算法,其中 Dijkstra 算法是解决这个问题的最有效的算法之一。在串行情况下,该算法的时间复杂度为 $O(m + n \log n)$ 。在改进算法时,主要是将蚁群算法的选路方法的改进应用于最短路径算法,经最短路径算法的选路方法用蚁群算法的选路策略所代替。

在改进的算法(ORSR)中从以下 3 个方面对 ACO 进行了优化并在最短路径算法中实现:

(1) 减少选路次数 在基本 ACO 中一轮求解需要 $m * n * (n-1)/2$ 次选路计算,算法的计算开销随着节点数目 n 和蚂蚁个数 m 的增加而急剧增加。根据蚁群算法的本质并行性,提出了具有较少选路次数的优化选路算法。在这个算法中,假设蚂蚁的行为都是独立并行的,在每一次选路结束后,找到 m 只蚂蚁中走过路程最短的那只蚂蚁,只让它进行选路操作,在它到达下一个节点并计算路径长度后,再将 m 个蚂蚁的路径进行比较(其它蚂蚁的路径长度未发生变化),

依次进行以上操作,直到有一只蚂蚁走完全部节点算法结束。

(2) 减少选路计算量 在选路操作中,只根据当前节点的前 c 个距离最近的且未经过节点为候选节点计算选择概率。这种方式在应用于节点规模较大的路由问题的求解时,有更好的执行效率(如表 1 所列)。

(3) 已知最短路径方法 为了进一步提高算法的执行效率,在算法中引入变量 $L_{\min}$,来记录每轮求解后的路径最小值。该变量初始化为可能的最大值。在求得解路径后更新该变量的值。并在以后各轮中,把每次蚂蚁选路后求得的当前解路径与该值进行比较,如果大于该值,则本轮循环终止,进入下一轮循环,蚂蚁进行重新选路。

(4) 结合网络模型的特点,减少了对两顶点之间最短路径以外的大量顶点的计算,提高了算法的速度。考虑到网络路由问题中的阻塞问题,对阻塞顶点进行标识,防止算法选择无用顶点。在算法中增加变量 $FLAG=1$ 用于标志两点之间没有阻塞,如果通过流量的增减使两点出现阻塞时连通图的代价函数 $C(e)$ 的值会小于等于 0,这时 $FLAG$ 则会由 1 变为 0,有效地防止算法选择阻塞节点。

从一系列模拟实验结果中发现,改进的蚁群算法虽然有较强的发现较好解的能力,但随路由节点规模的扩大,使问题求解更为复杂,最终找到的可能只是可行的较好解。所以在算法中加入了局部优化算法 2-opt^[4] 对所求的解做进一步优化,模拟实验表明这种改进加快了收敛速度。

6 仿真实验结果和分析

我们进行了一系列模拟实验,比较了 Dijkstra 算法与 ORSR 算法的性能,以及收敛速度和求解能力。实验的硬件平台是赛扬 2.4G 的 PC 机,内存为 512M,操作系统为 Windows 2000 Server,开发工具为 VC++6.0。选取的实验对象是文献[15]中的实验算粒(如图 1 所示)。图例为无向带权连通图,用 $G(N, E, C)$ 代表,其中 N 表示网络节点集合, $E = \{e_1, e_2, \dots, e_{n-1}\}$ 表示边的集合,每一条边 $e \in E$ 的代价函数为 $C(e); C \in R+(R+ \geq 0)$,表示链路上的流量限制。

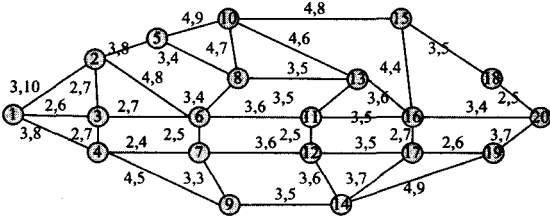


图 1 无向带权连通图

首先比较两个算法的求解能力。设定了源节点和目标节点,分别用原始的 Dijkstra 算法与 ORSR 算法寻找解,ORSR 算法中的参数设置如下: $m=20, \alpha=1, \beta=5, \rho=0.67, Q=1000, NC=5$ 。运算结果如表 1 所列。

表 1 Dijkstra 算法与 ORSR 算法求解能力比较

源节点	目标节点	Dijkstra 算法		ORSR 算法	
		路径长度	时间(us)	路径长度	时间(us)
1	20	16	30.9	13	25.7
2	19	16	27.4	14	23.3
10	14	15	15.3	12	14.1

由表 1 可以发现,改进的算法总能得到最优解,而且求解的速度更快,说明该优化实现是有效的。

还测试了两种算法在网络拥塞的情况下的求解能力。参照文献[15]将找到最短路径上的最小流量,并用最短路径上的每条边上的流量减去最小容量,即网络上必有一个网段的 $C(e)$ 的值为 0,则网络出现阻塞。

由表 2 可以看出,改进的算法在阻塞情况下得到的优化解优于或等于原始 Dijkstra 算法所得到的解,因为蚁群算法的优点就是能够在有障碍的情况下迅速得到较优解,改进算法恰恰证明了蚁群算法和最短路径算法相结合的改进算法是有优势的,而且求解的速度更快,说明该优化实现是有效的。

表 2 Dijkstra 算法与 ORSR 在阻塞情况下求解能力比较

源节点	目标节点	Dijkstra 算法		ORSR 算法	
		阻塞点	路径长度	阻塞点	路径长度
1	20	{16,20}	21	{16,20}	16
2	19	{7,9}	14	{6,7}	14
10	14	{7,9}	15	{13,11}{11,12}	12

结束语 本文用蚁群算法对最短路径算法进行优化,在提高算法的收敛速度以及优化算法实现效率等方面得到了一些有意义的结果。模拟实验表明,引入蚁群算法后的最短路径算法,其运行效率有很大提高,收敛速度以及解的质量都明显优于原始的最短路径算法。

参考文献

[1] Dorigo M, Maniezzo V, Colormi A. The Ant System; optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 1-13

[2] Dorigo M, Maniezzo V, Colormi A. Ant system; Optimization by a colony of cooperating agents[J]. IEEE Trans. on SMC, 1996, 26(1): 28-41

[3] Colormi A, Dorigo M. Heuristics from nature for hard combinatorial optimization problems[J]. International Trans. Operational Research, 1996, 3(1): 1-21

[4] Huang L, Zhou CG, Wang KP. Hybrid ant colony algorithm for traveling salesman problem[J]. Progress in Natural Science, 2003, 13(4): 295-299

[5] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutional Comutation, 1997, 1(1): 53-66

[6] Dorigo M, Gambardella LM. A study of some properties of ant-Q[C]//Voigt H-M, Ebeling W, Rechenberg I, et al., eds. Proceedings of the PPSN 44th International Conference on Parallel Problem Solving from Nature. Berlin: Springer-Verlag, 1996: 656-665

[7] Gambardella LM, Dorigo M. An ant colony system hybridized with a new local search for the sequential ordering problem[J]. INFORMS Journal on Computing, 2000, 1(3): 237-255

[8] Parpinelli RS, Lopes HS, Freitas AA. Data mining with an ant colony optimization algorithm[J]. IEEE Trans. on Evolutionary Computation, 2002, 6(4): 321-328

[9] Wu QH, Zhang JH, Xu XH. An ant colony algorithm with mutation features[J]. Journal of Computer Research & Development, 1999, 36(10): 1240-1245 (in Chinese)

[10] Chen L, Shen J, Qin L. An Adaptive Ant Colony Algorithm Based on Equilibrium of Distribution[J]. Journal of Software, 2003, 14(8): 1379-1387 (in Chinese)

Sigmoid 核函数 $k(x, x') = \tan h(p_c \cdot x \cdot x' + p_d)$

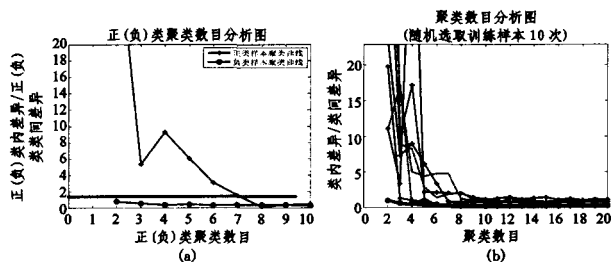


图 1 (a)聚类数目不同时正负类样本评价对比图;(b)聚类数目不同时评价值的稳定性图

在实验过程中采用基于主动反馈学习的样本选择方法^[13]。设 P 是训练样本集, S 是当前迭代的训练样本集, 新被选择的样本满足以下条件:

$$\min_{x_i \in Q \setminus S} ((1 - \lambda_1 - \lambda_2) |d(x_i)| + \lambda_1 \max_{x_j \in S} |k(x_j, x_i)| + \lambda_2 \max_{x_j \in SV} |k(x_j, x_i)|) \quad (12)$$

其中 $d(x_i) = \alpha^* * \text{diag}(y) * K(x_j, x_i)$, θ 是权衡因子且 $0 \leq \theta \leq 1$, $K = \sum_{d=1}^D \mu_d K_d$ 。当分类器的分类性能稳定的时候, 反馈学习终止。实验结果表明, KM-MKSVM 可以提高乳腺微钙化簇检对率, 如表 2 所列。

表 2 不同 SVM 的性能对比表

分类器	核函数	核参数	检对率
SVM	高斯核	$b=0.5$	76.51%
CSVM	高斯核	$b=0.5$	78.80%
K SVM	高斯核	$b=0.5$	79.40%
MKSVM	线性核+多项式	$a=1.7$	77.16%
MKSVM	线性核+高斯	$b=0.5$	79.09%
MKSVM	线性核+sigmoid	$c=4.1; d=10$	79.53%
MKSVM	多项式+高斯核	$b=0.5; a=1.7$	77.59%
MKSVM	多项式+sigmoid	$a=1.7; c=4.1; d=10$	76.51%
MKSVM	高斯核+sigmoid	$b=0.5; c=4.1; d=10$	79.31%
KM-MKSVM	线性核+多项式	$b=0.5$	81.47%
KM-MKSVM	线性核+高斯	$b=0.5$	79.09%
KM-MKSVM	线性核 sigmoid	$c=4.1; d=10$	81.25%
KM-MKSVM	多项式+高斯	$a=1.7; b=0.5$	80.18%
KM-MKSVM	多项式+sigmoid	$a=1.7; c=4.1; d=10$	81.68%
KM-MKSVM	高斯核+sigmoid	$b=0.5; c=4.1; d=10$	82.11%

结束语 为提高微钙化簇的检对率, 考虑到乳腺微钙化簇正负类样本分布不平衡以及特征多样性的特点, 提出基于 K 均值聚类的多核 SVM。本方法将正类和负类样本分别聚类, 对不同类样本加不同的惩罚因子, 用主动反馈学习的方法选择训练样本, 最后用多核 SVM 训练空间模型。实验结果

表明, 该方法可以有效地提高乳腺微钙化簇的检对率。

参考文献

- [1] Thangavel K, Karnan M, Sivakumar R, et al. Automatic Detection of Microcalcification in Mammograms-A Review[J]. International Journal on Graphics Vision and Image. Processing, 2005, 5(5): 31-61
- [2] Bazzani A, et al. A SVM classifier to separate false signals from microcalcifications in digital mammograms[J]. Phys. Med. Biol., 2001, 46(6): 1651-1663
- [3] Papadopoulos A, Fotiadis D I, Likas A. Characterization of clustered microcalcifications in digitized mammograms using neural networks and support vector machines[J]. Artificial Intelligence in Medicine, 2004, 34(2): 141-150
- [4] Wei Liyang, Yang Yongyi, et al. A Study on Several Machine-learning Methods for Classification of Malignant and Benign Clustered Microcalcifications[J]. March. IEEE Transaction on Medical Imaging, 2005, 4(3): 371-380
- [5] El-Naqa I, et al. A support vector machine aproach for detection of microcalcifications[J]. IEEE Transactions on Medical Imaging, 2002, 21(12): 1552-1563
- [6] Li Ying, Jiang Jianmin. Combination of SVM Knowledge for Microcalcification Detection in Digital Mammograms[C]//IDEAL, 2004, LNCS 3177. 2004: 359-365
- [7] 邓乃扬, 田英杰. 数据挖掘中的新方法-支持向量机[M]. 北京: 科学出版社, 2004: 369-370
- [8] 边肇祺, 张学工. 模式识别(第二版)[M]. 北京: 清华大学出版社, 2000: 280-281
- [9] Rose C, Turi D, Williams A, et al. Digital Database for Screening Mammography[DB/OL]. <http://marathon.csee.usf.edu/Mammography/Database.html>, 1998
- [10] Wang Jiaqi, Wu Xindong, Zhang Chengqi. Support Vector Machines Based on K-means Clustering for Real-time Business Intelligence Systems[J]. International Journal of Business Intelligence and Data Mining, 2005, 1(1): 54-64
- [11] Li Maokuan, Cheng Yusheng, Zhao Honghai. Unlabeled Data Classification via Support Vector Machines and k-means Clustering[C]//Proceedings of the International Conference on Computer Graphics, Imaging and Visualization. 2004: 183-186
- [12] Lanckriet G, et al. Learning the Kernel Matrix with Semi-definite Programming[J]. Journal of Machine Learning Research, 2004: 27-72
- [13] Jiang J, Ip H H S. Active Learning with SVM[M]. Rabuñal J R, Dorado J, Pazos A, eds. Encyclopedia of Artificial Intelligence; Information Science Reference, 2008

(上接第 207 页)

- [11] Chen L, Shen J, Qin L. A method for solving optimization problem in continuous space by using ant colony algorithm[J]. Journal of Software, 2002, 13(12): 2317-2322 (in Chinese)
- [12] Wu B, Shi ZZ. An ant colony algorithm based partition algorithm for TSP[J]. Chinese Journal of Computers, 2001, 24(12): 1328-1333 (in Chinese)
- [13] Zhang J H, Gao Q S, Xu X H. A self-adaptive ant colony algorithm[J]. Control Theory And Applications, 2000, 17(1): 1-3 (in Chinese)
- [14] Dorigo M, Maniezzo V, Colomi A. Ant System; An autocatalytic optimizing process[R]. 91-106. 1991
- [15] Ding Jian-li, Chen Zeng-qiang, Yuan Zhu-zhi. Dynamain optimiza-

tion routing method base on ant adaptive algorithm[J]. Control and Decision, 2003, 18(6): 751-753 (in Chinese)

- [16] Zhang Su-bing, Lu Guo-ying, LIU Ze-min, et al. QoS Routing Based on Ant-Algorithm[J]. Journal of Circuits and Systems, 2000, 5(1)
- [17] Xiang F, Junzhou L, Jieyi W, et al. QoS Routing Based on Genetic Algorithm[J]. Computer Communications, 1999, 22: 1392-1399
- [18] Wang Z, Crowcroft J. Quality of Service Routing for Supporting Multimedia Applications[J]. JSAC, 1996, 14(7): 1228-1234
- [19] Schoonderwoerd R, Holland O, Bruten J. Ant-based Load Balancing in Telecommunications Networks[J]. Adaptive Behavior, 1996, 5(2): 169-207