

# 动态蚁群算法求解 TSP 问题

李勇 段正澄

(华中科技大学国家数控系统技术研究中心, 武汉 430074)

E-mail:liyonghust@sina.com

**摘要** 蚂蚁群体能完成单个蚂蚁所无法完成的工作。它们通过称为信息素的物质交流信息而协同工作。蚂蚁在觅食活动中,在食物与巢穴之间的路径上留下信息素,较短路径信息素相对较浓,而蚂蚁倾向于沿信息素较浓的路径往返于巢穴与食物之间。经过一段时间后,就可发现从巢穴到食物的较短的路径。基于此原理,Marco Dorigo 提出了蚁群算法,并首先用于求解 TSP 问题。该文从更多方面模仿真实自然界中蚂蚁的行为,更为合理地制定信息素动态挥发规则,提出动态蚁群算法并用于解决 TSP 问题,实验表明了该算法有较好的性能。

**关键词** 蚁群算法 旅行商问题 组合优化

文章编号 1002-8331-(2003)17-0103-04 文献标识码 A 中图分类号 TP301.6

## A New Ant System for TSPs

Li Yong Duan Zhengcheng

(The National NC System Engineering Research Center, HUST, Wuhan 430074)

**Abstract:** Ant colony can perform tasks that cannot be carried out by individual ants. Ants work together through a chemical substance—pheromone. Ants look for food and lay the way back to their nest with pheromone, and the other ants can follow the pheromone to find the food efficiently. Basing on these, Marco Dorigo proposed the ant algorithm. This paper presents dynamic ant colony system, a improved version of ant system (AS). It shows better performance than the original algorithm.

**Keywords:** ant system, traveling salesman problems, combinatorial optimization

### 1 引言

蚂蚁系统是模仿蚁群的行为构建的人工系统。在自然界中蚂蚁几乎是瞎子,却能发现食物与蚁穴之间最短的距离。生态学家的研究表明,蚂蚁是借助信息素(pheromone)来实现这一点的。信息素是蚂蚁分泌的一种化学物质,蚂蚁在寻找食物过程中会分泌这种物质。蚁群在寻找食物时,会有一些蚂蚁四处游荡,当蚂蚁发现食物并返回巢穴过程中会在途中留下信息素。如果假定所有蚂蚁以相同的速度前进的话,则第一只返回巢穴的蚂蚁发现的路径是当前所有蚂蚁发现的路径中最好的。较短路径上信息素会比较浓一些,而较浓的信息素能吸引更多的蚂蚁走这条路径,从而发现比较好的路径。图 1 说明了蚂蚁如何发现最短路径的基本原理。图 1 中两只蚂蚁在同时离开蚁穴,分别走不同的路径并在各自的路径留下信息素。走较短路径(下面)的蚂蚁会首先返回巢穴。这条路径由于留下较多的信息素,它会吸引其它蚂蚁走这条路线。这种强化过程使蚁群发现最短路径。基于这种原理,Marco Dorigo 等人提出了称为蚂蚁算法一种新的启发式算法。蚁群算法已成功应用于求解 TSP 问题、调度问题、着色问题。并在大规模集成电路设计,电信路由控制方面表现出较好的性能。

蚁群算法是模拟蚂蚁的行为而提出一种启发式算法,该文从更多方面模拟蚁群行为提出了一种改进的蚁群算法,该文明称其为动态蚁群算法,并用于求解 TSP 问题(Traveling Salesman Problem)。TSP 问题,即旅行商问题,是一类已被证明具有 NPC 计算复杂性的组合优化难题,它的提法是:给定  $N$  个城市,有

作者简介:李勇(1974-),博士生。

一旅行商从某一城市出发,访问各城市一次且仅有一次后返回原出发城市,要求找出一条最短的巡回路径。该文对蚁群算法的改进主要在以下三个方面:(1)动态目标城市选择;(2)信息素挥发系数动态变化;(3)最差路线上的信息素弱化;实验表明改进后的算法比基本蚁群算法及改进的蚁群算法 MMAS (MAX\_MIN Ant system)总体性能有了一定的改善。该文首先介绍基本蚁群算法(AS)和改进的蚁群算法(MMAS),然后说明改进后的算法,最后进行实验结果分析比较及结论。

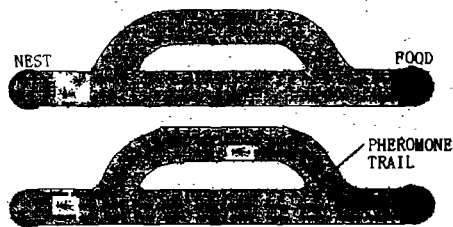


图 1 蚂蚁路径长短与信息素强度关系图

### 2 蚁群算法(AS)与 MMAS

#### 2.1 AS

蚁群算法(AS)是由 Dorigo 和 Gambardella 提出一种新的启发式算法,首先被应用于 TSP 问题。这一节也将以 TSP 为例说明蚁群算法。

在算法的开始,首先  $m$  只蚂蚁基于某种规则(如随机)置

于  $n$  个城市上,位于城市  $r$  上的蚂蚁  $k$  以  $P_k(r,s)$  为概率函数选择的下一个城市,这个公式给出蚂蚁  $k$  由城市  $r$  转移到城市  $s$  的概率。

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^\alpha \cdot [\eta(r,s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r,u)]^\alpha \cdot [\eta(r,u)]^\beta} & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$J_k(r)$ , 蚂蚁  $k$  还未访问的城市列表; $\tau(r,u)$ , 边  $(r,u)$  上的信息素浓度; $\eta(r,u)=1/d_{r,u}$ , 启发函数  $\alpha, \beta$ , 这两个参数反映了信息素和启发函数的相对重要性;

第二步,当所有蚂蚁完成环游,并按公式(3)进行信息素更新

$$\tau(r,s) \leftarrow (1-\rho) \cdot \tau(r,s) + \Delta\tau \quad (2)$$

$\Delta\tau = \sum_{k=1}^m \Delta\tau_k(r,s)$ ,  $\rho(0 < \rho < 1)$  是信息素挥发因子。

$$\Delta\tau_k(r,s) = \begin{cases} \frac{1}{L_k} & \text{当边}(r,s)\text{被蚂蚁}k\text{使用时} \\ 0 & \text{其它} \end{cases}$$

$L_k$  是第  $k$  只蚂蚁的环游长度。这种信息素更新方式称为局部信息素更新

最后看结束条件是否满足,如不满足则返回第一步。

## 2.2 MMAS

MMAS 主要在以下两方面区别于 AS。

(1) 当所有的蚂蚁完成环游,仅仅蚂蚁发现的最好路径上的信息素进行更新,这种信息更新方式称为全局信息素更新。

(2) 每条边上的信息素被限制在区间  $[\tau_{min}, \tau_{max}]$  内。

(3) 信息素初始化为  $\tau_{max}$ 。

对每条边上的信息素浓度的限制有利于减少早熟现象。MMAS 中每条边的信息素被初始化为  $\tau_{max}$ , 经过几次迭代之后,每条路径上的信息素因挥发而减小,而仅有每次迭代过程中的最好路径上的信息素被允许增加,因而只有最好路径上的信息素保持一个较高水准。MMAS 性能优于 AS。

MMAS 有一种改善性能的机制称之为 pheromone trail smoothing (PTS), 当 MMAS 计算结果收敛至一个值停滞不变时,按如下公式调整信息素:

$$\tau(r,s) \leftarrow \tau(r,s) + \delta \cdot (\tau_{max}(r,s) - \tau(r,s)), \text{其中 } 0 < \delta < 1 \quad (3)$$

这种机制是在进化接近停滞的情况下调总体上的信息素分布,其中参数  $\delta$  决定了对以前信息素的保留的多少。 $\delta=0$ , 则

是完全保留,PTS 不起作用; $\delta=1$  则完全去掉以前的信息素分布,重新开始计算。这种机制在长时间计算运算中有比较好的作用。

## 3 动态蚁群算法

动态蚁群算法相对 AS 和 MMAS 主要改进在于以下几点。AS 和 MMAS 算法依据信息素和启发函数选择目标城市。如何依据这两个因子选择城市对算法的性能是非常关键的。动态蚁群算法在迭代过程中,在选择目标城市的标准时,不是使用一个固定的标准,使之有利于减小进化停滞现象。在 AS 中,对信息素的强度没有限制,因而易于陷入局部最优点。MMAS 对信息素的强度给予一定的限制,从而大大改善了算法的性能。动态蚁群算法中挥发因子是动态变化,信息素浓度越高挥发因子越大,浓度越低挥发因子越小,这样实际对信息素浓度也进行限制。使信息素不可能无限增大,也不可能为零。MMAS 中仅有最好蚂蚁所走路径上的信息素进行全局更新,如果能对更多路径上的信息素进行更新则有可能加快演化的速度。为了比较算法的全局收敛能力,在随后的计算中没有采用如 r-opt(一般用 2-opt 或 3-opt), Lin & kerninggan 等局部优化方法。

### 3.1 权函数

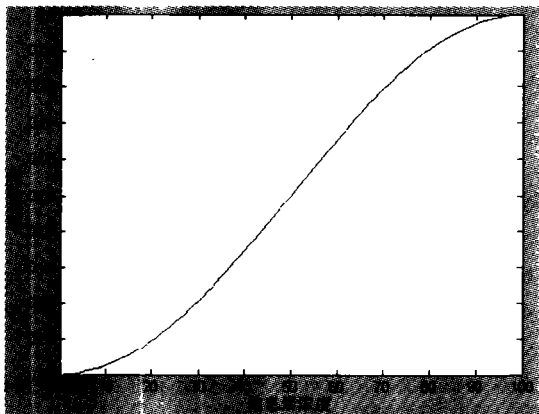
AS 和 MMAS 依据  $[\tau(r,s)]^\alpha [\eta(r,s)]^\beta$  来选择下一个转移城市, $\alpha, \beta$  是常数。在自然界中生物随着时间的推移对环境适应之后,不再对环境敏感。基于此原理,可认为蚂蚁随着时间的推移对信息素慢慢变得不敏感。在算法中体现为,  $[\tau(r,s)]^\alpha$  随着时间相对于  $[\eta(r,s)]^\beta$ , 减小,这样信息素的影响减小了,而启发函数的影响相对加大,这样  $\alpha, \beta$  由常数就变成与时间有关的函数。在 TSP 问题中  $\eta(r,s)=1/d_{r,s}$ ,  $d_{r,s}$  是城市  $r$  和  $s$  之间的距离。为简化计算,文章置  $\alpha=1$ , 而  $\beta$  按迭代次数进行如下变化

1~mn/3 次迭代  $\beta=5$ ; mn/3~mn/2 次迭代  $\beta=4$ ; mn/2~mn\*2/3 次迭代  $\beta=3$ ; mn\*2/3~mn 次  $\beta=2$ ;

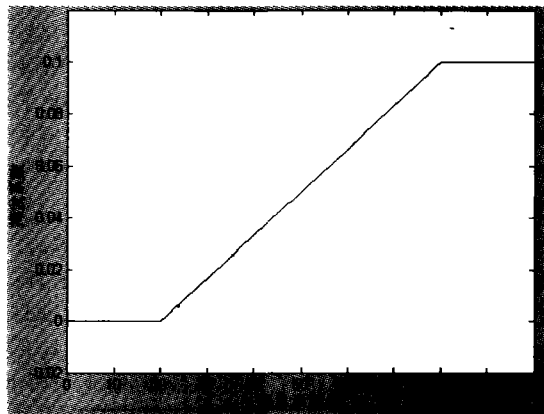
mn 是总的迭代次数。在该文中循环次数固定为 10000 次。

### 3.2 动态挥发因子

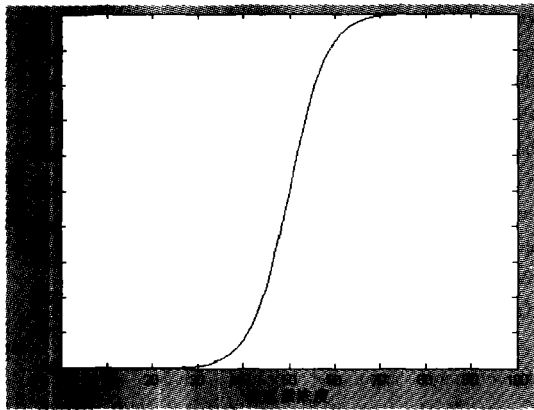
在 AS 中,挥发因子是一个常数。而在真实世界中,信息素浓度越高,挥发越快,信息素浓度越低,挥发越慢,这样可以防止信息浓度无限制地增长,或者变为零,减小陷入局部最优的可能。在这样情况下挥发因子由常数变成以  $\tau(r,u)$  为变量的函数。



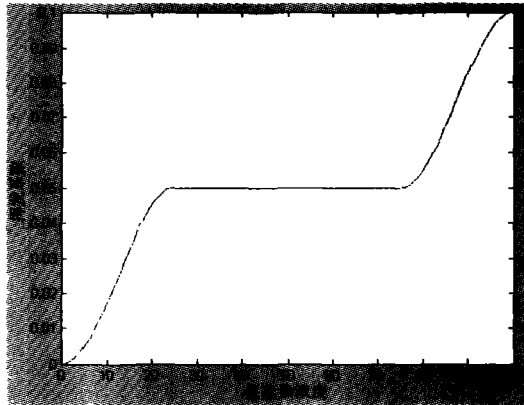
(a)



(b)



(c)



(d)

图2 挥发函数

局部信息素更新规则变为:

$$\tau(r,s) \leftarrow (1-\rho(\tau(r,s))) \cdot \tau(r,s) + \text{coe}(\tau(r,s)) \cdot \Delta\tau \quad (4)$$

挥发函数对算法的性能有直接的影响,在实验中曾试过如图 2所示 4 种类型函数(X 轴为浓度,Y 轴为挥发因子),实验表明图 2(d)所对应的挥发函数的效果比较好。事实上 MMAS 可以看作动态蚁群算法中的一个特例

### 3.3 最优、最差路径信息素全局更新

AS 和 MMAS 仅对最好路径上的信息素进行全局更新,仅有一只蚂蚁对全局信息素的更新产生影响。如果有多只的蚂蚁对全局信息素的更新产生影响,则可加速演化过程。更新规则如下:

$$\tau(r,s) \leftarrow (1-a(\tau(r,s))) \cdot \tau(r,s) + \text{coe}(r,s) \cdot \Delta\tau(r,s) \quad (5)$$

该文对两只蚂蚁所走路上的信息素进行更新,即对最好路径及最差路径上的信息素进行全局更新,则参数

$$\text{coe}(r,s) = \begin{cases} C & (r,s) \in \text{global\_best\_tour} \\ -C & (r,s) \in \text{global\_worst\_tour} \\ 0 & (r,s) \in \text{the\_other} \end{cases}$$

公式(5)变为:

$$\begin{aligned} \tau(r,s) &\leftarrow (1-a(\tau(r,s))) \cdot \tau(r,s) + C \cdot \Delta\tau(r,s) : \text{best} \\ \tau(r,s) &\leftarrow (1-a(\tau(r,s))) \cdot \tau(r,s) - C \cdot \Delta\tau(r,s) : \text{worst} \end{aligned}$$

其中

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1} & \text{if } (r,s) \in \text{global\_best\_tour} \\ (L_{gw})^{-1} & \text{if } (r,s) \in \text{global\_worst\_tour} \end{cases}$$

$L_{gb}$  是最好蚂蚁所走路径长度。 $L_{gw}$  是最差蚂蚁所走路径长度。 $a(\tau(r,s))$  是全局信息素挥发因子。

### 4 实验结果

从 TSPLIB (<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>) 中选取三个 TSP 和 ATSP 实例进行测试。值得注意的是从 TSPLIB 数据中取得的数据。对 TSP 问题提供是每个城市的坐标,需要转化成城市之间的距离,并且按四舍五入圆整为整数。否则会出现它提供最好环游长度,与所提供的环游城市之间距离之和不相等的情况出现。以 eil51 为例,公布的最好环游路径如图 3 所示,不进行圆整的结果为 429.983,而公布的对应的最短环游路径为 426,对城市之间的距离进行圆整后计算结果为 426。用动态蚁群算法计算并且不进行圆整,路径长度为 428.872,如图 4 所示。

在实验中选择的参数如下:

$m=10, \text{coe}(\tau(r,s))=0.1, C=0.1, \alpha=1, a(\tau(r,s))=0, \delta=0.5$  迭代次数  $mn=10000$ ,挥发函数选择如图(2)中(d)所示函数。计算结果如表 1 所示。

表 1 中“最好解”指目前所有算法发现的最好解。MMAS、MMAS+pts 和 AS 计算结果从文献[16]中得到。表中所有数据是 25 次独立运算的平均结果。对于 ACS 和 MMAS 迭代次= $n*10000$ 次, $n$ 为城市数目。MMAS 使用  $n$  蚂蚁。对于 TSP 问题 3 个实例的计算结果均比 MMAS 强,而对于 ATSP 问题,仅有

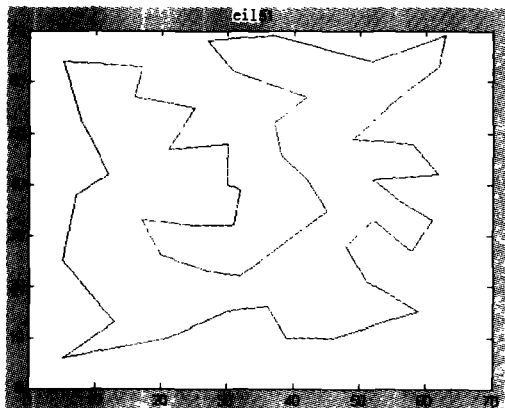


图3 TSPLIB 公布的最好环游路径

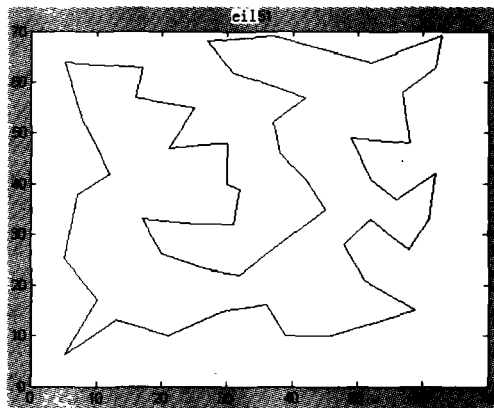


图4 用动态蚁群算法发现的最好路径

kro124 实例的计算结果比 MMAS 的结果差一些。

表 1 TSP 计算结果比较

实例	最好解	动态蚁群算法	动态蚁群算法+pts	MMAS+pts	MMAS	AS
eil51	426	426.24	<b>426.08</b>	427.1	427.6	43703
kroa100	21282	21299.32	<b>21289.346</b>	21291.6	21320.3	22471.4
d198	15780	15970.2	<b>15950.4</b>	15956.8	15972.5	16702.1
ry48p	14422	14472.84	<b>14455.2</b>	14523.4	14533.2	15296.4
ft70	38673	38854.4	<b>38823.2</b>	38992.7	39040.2	39596.3
kro124p	36230	36780.2	36578.346	<b>36573.6</b>	36773.5	38733.1

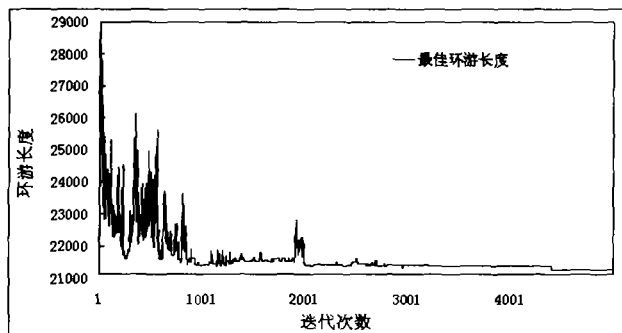


图 5 典型演化过程(kroa100)

## 5 讨论与结论

与 MMAS 相比,在动态蚁群算法中对更多蚂蚁所走路径上的信息素更新,以加速收敛,另一方面采用动态目标城市选择标准和动态挥发因子来减小进化停滞现象,防止早熟。从计算结果来看取得了比较好的结果。但在实验中发现参数选择对计算结果影响很大,而且参数选择没有特别的依据,主要依赖试验结果进行选择。在该算法中参数,动态权函数及挥发函数对算法的性能影响非常关键,如何更合理地选择这些函数,及参数将是下一步的主要工作。(收稿日期:2002年5月)

## 参考文献

1. Marco Dorigo, Gianni Di Caro. Ant Algorithms for Discrete Optimization[J]. *Artificial Life*, 1999; 5(3): 137-172
2. Luca M Gambardella, Marco Dorigo. Ant-Q: A reinforcement Learning approach to the traveling salesman problem[C]. In: *Proceeding of ML-95, Twelfth Intern Conf on Machine Learning*, Morgan Kaufmann, 1995: 252-260
3. Hidenori KAWAMURA, Masahito YAMAMOTO, Keiji SUZUKI et al. Multiple Ant Colonies Algorithm Based on Colony Level Interactions[J]. *IEICE TRANS Fundamentals*, 2000; E83-A(2)

(上接 76 页)

另一个重要参数是数据缓冲区大小,数据缓冲区太小,导致镜像线程同压缩线程之间的数据缓冲区传递频繁,系统的同步互斥开销也增加。而数据缓冲区太大,镜像线程和压缩线程的并行性就会降低,因为当缓冲区很大时,虽然镜像线程镜像了很多数据,当缓冲区未满载时,即使压缩线程空闲,也只能等待,从而降低系统性能。

对于这两个参数,采用动态调整技术,他们都和系统的硬件配置有关,服务器生产时设置一个初始值,主线程每次统计镜像压缩的信息,通过这些历史记录,主线程会逐步调整这两个参数,最终到达一个稳定的设置,当服务器硬件配置改变时,他也会逐步适应。(收稿日期:2002年7月)

4. Shyh-jier Huang. Enhancement of hydroelectric Generation Scheduling Using Ant Colony System Based Optimization Approaches[J]. *IEEE Transactions on Energy Conversion*, 2001; 16(3)
5. Christian Blum, Andrea Roli, Marco Dorigo. HC-ACO The Hyper-Cube Framework for Ant Colony Optimization[C]. In: *MIC 2001 - 4th Metaheuristics International Conference*, 2001: 16-20
6. Gianni di Caro, Marco Dorigo. AntNet: A Mobile Agents Approach to Adaptive Routing[R]. Technical Report, IRIDIA97-12, Universite Libre de Bruxelles, 1998
7. M Dorigo, L M Gambardella. Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem[J]. *IEEE Transactions on Evolutionary Computation*, 1997; (1): 53-56
8. A Colomi, M Dorigo, F Maffioli et al. HEURISTICS FROM NATURE FOR HARD COMBINATORIAL OPTIMIZATION PROBLEMS[M]. Published in *International Transactional in Operational Research*, 1-21
9. Marco Dorigo, Vittorio Maniezzo, Alberto Colomi. Ant System: Optimization by a Colony of Cooperating Agents[J]. *IEEE Transactions on System, Man, Cybernetics-Part B: Cybernetics*, 1996; 26(1)
10. Nicolas meuleau, Marco Dorigo. Ant Colony Optimization and Stochastic Gradient Descent[R]. Technical Report, N.TR/IRIDA/2000/2000-36, 2000
11. T Stutzle, H Hoos. The ant system and local search for traveling salesman problem[C]. In: *a Proceeding of ICEC '97 IEEE 4<sup>th</sup> International conference of evolutionary*, 1997
12. T Stutzle, Marco Dorigo. ACO Algorithms for the Traveling Salesman problem[C]. In: *K Miettinen, M Makla, P Neittaanmaki et al eds. Evolutionary Algorithms in Engineering and computer Science*, Wiley, 1999
13. Coello C A, Hernandez A, Zavala R L et al. Ant Colony System for the Design of Combinational Logic Circuits[C]. In: *Proceedings of the International Conference on Evolvable Systems, ICES2000, Edinburg, Scotland, LNCS, Springer Verlag*, 2000: 17-19
14. Liu J Tng Y Y. An Evolutionary Autonomous Agents Approach to Image Feature Extraction[J]. *IEEE Transactions on Evolutionary Computation*, 1997; 1(2): 141-158
15. Dorigo M, Di Caro G. Ant Colony Optimization: A New Meta-Heuristic[J]. *Proceedings of the Evolutionary Computation*, Washington DC, 1999; 1470-1477
16. Thimas stutzle. MAX-MIN Ant system. Preprint submitted to Elsevier Science, 1999-11-30
17. 吴斌, 史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法[J]. *计算机学报*, 2001; 24(12)
18. 马良, 项培军. 蚂蚁算法在组合优化中的应用[J]. *管理科学学报*, 2001; 4(2)

## 参考文献

1. 庞丽萍. 操作系统原理[M]. 第二版, 华中理工大学出版社, 1994-09
2. 林福宗, 陆达编著. 多媒体与 CD-ROM[M]. 清华大学出版社, 1995
3. [英] Phil Cornes 著. Linux 从入门到精通[M]. 电子工业出版社, 1998.7
4. Michael Barabanov. A Linux-based Real-Time Operating System[D]. M S Thesis. New Mexico Institute of Technology, 1997-06
5. [Watson95] Watson R, Coyne R. The Parallel I/O Architecture of the High-Performance Storage System (HPSS)[C]. In: *14th IEEE Symposium on Mass Storage Systems*, 1995-09
6. 林福宗编著. 多媒体技术基础[M]. 清华大学出版社, 2000-08
7. [美] John Goerzen 著. Linux 编程宝典[M]. 电子工业出版社, 2000-08