

编者按：一个高中生能写出如此论文实属不易，表明作者对蚁群算法有较深入的研究，反映在中学生中也有对计算机技术颇具研究、出类拔萃的人才，本刊为鼓励这样的人才脱颖而出，特发表此文，以引起社会的重视。

文章编号：1000—3428(2001)12—0074—03 文献标识码：A 中图分类号：TP301.6

## 带杂交算子的蚁群算法

陈 烨

(四川省内江市第六中学，内江 641000)

**摘 要：**蚁群算法是一种由意大利学者Macro Dorigo 等提出的新型模拟进化算法。它具有许多优良性质，因此被广泛用于求解组合优化问题。但基本蚁群算法有许多不足。特别是它搜索速度慢，且容易陷入局部最优。该文针对这个问题提出了一种改进算法。该算法通过引入遗传算法中用到的杂交算子来改善蚁群，使其对应的问题的解更加优良。用改进算法求解TSP问题的结果表明改进算法是有效的。

**关键词：**蚁群算法；杂交算子；遗传算法；组合优化；TSP问题

## An Ant Colony Algorithm with Crossover Operator

Chen Ye

(No 6 Middle School, Neijiang, Sichuan, Neijiang 641000)

**【Abstract】** Ant colony algorithm (ACA) is a new kind of simulated evolutionary algorithm. It is proposed by Italian scholar Macro Dorigo. Ant colony algorithm has many good features. So it was widely applied to complicate combinatorial optimization problems. But there is much deficiency. Specially, its searching speed is slow, and it is easy to fall in local best. An improved algorithm is presented to solve this problem. A crossover operator is contained in this algorithm. It is usually used in genetic algorithm. It can improve ant colony to make the corresponding solution better. The improved algorithm is applied to solve the traveling salesman problem(TSP). The result of the experiment suggests that the improved algorithm is effective.

**【Key words】** Ant colony algorithm(ACA) ; Crossover operator ; Genetic algorithm(GA) ; Combinatorial optimization ; TSP

蚂蚁在寻找食物时，总是能找到较短的路径。受此启发，意大利学者Macro Dorigo等于1991年提出了Ant System(AS)<sup>[1]</sup>，并用该算法求解TSP等问题，取得了较好的效果。该算法进化速度较慢是它的主要缺点，也容易陷入局部最优。因此，许多学者对蚁群算法提出了改进方法。如Stützle, Hoos提出的Max-Min AS(MMAS)、Dorigo, Gambardella提出的Ant Colony System(ACS)<sup>[2]</sup>等。这些改进算法对AS的性能有了很大的改进。这些改进使蚁群算法求解问题的能力可以与遗传算法、神经网络等求解组合优化问题的常用方法相媲美。对这种算法的研究结果表明，它具有很好的发展前景，也具有很好的实用性。本文将在ACS的基础上提出另一种改进方案。

### 1 蚁群算法

#### 1.1 蚁群算法的基本原理

蚁群算法通过模拟蚂蚁寻找食物并回到巢穴的方法来求解问题。

研究表明，蚂蚁在前进途中会留下一一种外激素(pheromone)。蚂蚁就是用这种物质来与其它蚂蚁交流、合作，以找到较短路径。经过某一地点的蚂蚁越多，这种外激素的强度就越大。蚂蚁选择路径时就偏向于选择外激素强度大的方向。这就是蚂蚁能找到回到巢穴或食物处的较短路径的原因。通过实验还表明，这种跟随外激素强度前进的行为会随着经过的蚂蚁的增多而加强。由于通过较短路径往返于食物和巢穴之间的蚂蚁能以更短的时间经过这条较短路径上的点，这些点上的外激素的强度就会因蚂蚁经过它的次数更多而更强<sup>[3]</sup>。这样就会有更多的蚂蚁选择这条路。这条路径上的外激素的强度就会越来越大，选择这条路径的蚂蚁也会越

来越多。直到最后，几乎所有的蚂蚁都选择这条较短路径。这是一种正反馈现象。蚁群算法就是利用正反馈来达到求解组合优化的目的。

#### 1.2 Ant System模型

Ant System 模型共有3种：ant-cycle system，ant-quantity system，ant-density system。由于在这3种模型中ant-cycle system的性能更好，因此下面只介绍这种模型。

Ant System最先用于求解TSP问题，因此，下面以TSP问题为例来说明Ant System模型。

假设有n个城市，m只蚂蚁，城市r与城市s之间的距离用 $d_{rs}$ 表示， $\tau_{rs}$ 表示城市r与城市s之间的残留信息量(对应于真实蚁群系统中各点的外激素强度)， $J_k$ 表示第k只蚂蚁还未访问过的城市， $p_{rs}^k$ 表示第k只蚂蚁由城市r转移到城市s的概率。

$$p_{rs}^k = \begin{cases} \frac{[\tau(r,s)] \cdot [J_k(r,s)]^b}{\sum_{u \in J_k} [\tau(r,u)] \cdot [J_k(r,u)]^b} & \text{如果 } s \in J_k \\ 0 & \text{否则} \end{cases} \quad (1)$$

其中， $\tau(r,s)=1/d_{rs}$ ，参数 $b$ 用于调节 $\tau(r,s)$ 与 $J_k(r,s)$ 之间的关系。

由公式(1)可知，蚂蚁在选择路径时尽量选择离自己较近且外激素浓度较大的方向。

当所有的蚂蚁完成一次旅行，外激素强度就将作以下调整

作者简介：陈 烨(1984~)，男，高二学生，曾发表论文两篇，对计算机技术很感兴趣，并有较深入的研究

收稿日期：2001-10-08

$$t(r,s) \leftarrow (1-a) \cdot t(r,s) + \sum_{k=1}^m \Delta t(r,s) \quad (2)$$

其中,  $0 < a < 1$ , 该参数确定各条路径上留下的信息的消逝速度,  $m$  是蚂蚁个数,

$$\Delta t(r,s) = \begin{cases} 1/L_k & \text{如果本次循环中蚂蚁 } k \text{ 经过 } (r,s) \\ 0 & \text{否则} \end{cases} \quad (3)$$

公式(3)中,  $L_k$  为第  $k$  只蚂蚁在本次循环中经过的路径的总长度。

上述更新各路径上的信息量的方法被称作全局更新规则(Global Updating Rule)。根据这个更新规则显然可知, 各路径上的信息量主要取决于该路径的长度。更新规则会让短路径对应的信息量逐渐增大。

Ant System 在求解 TSP 问题时, 当各参数已被赋予初始值后, 就会根据公式(1)求出蚂蚁从一个城市转移到另一个城市的概率。然后根据该概率, 随机生成各个蚂蚁经过的路径。接着, 按照公式(2)和公式(3)更新各条路径上的信息。然后又计算各蚂蚁从一个城市转移到另一个城市的概率……如此反复计算直到达到所指定的循环次数或计算结果在指定循环次数内无明显变化。

Ant-cycle system 虽是上述 3 种中最好的, 但其计算能力仍然有限, 因此在对蚁群算法的基本结构有所了解之后, 下面将在此基础上介绍蚁群系统(Ant Colony System)。

### 1.3 Ant Colony System

ACS 与前面的 AS 主要有 3 点不同: 1) 蚂蚁的转移规则不同; 2) 全局更新规则不同; 3) 新增了对各条路径信息量调整的局部更新规则(Local Updating Rule)。下面将作具体说明:

#### 1.3.1 ACS 的状态转移规则

在 ACS 中蚂蚁选择下一个城市时使用的公式为:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{ [t(r,u)] \cdot [h(r,u)]^q \} & \text{如果 } q \leq q_0 \\ S & \text{否则} \end{cases} \quad (4)$$

其中,  $q$  为一个在区间  $[0, 1]$  内的随机数,  $q_0$  是一个参数 ( $0 < q_0 < 1$ ),  $S$  是根据公式(1)确定的。

由上述公式确定蚂蚁转移到哪个城市的方法就是伪随机概率选择规则(pseudo-random proportional rule)。在这种规则下, 每当蚂蚁要选择向哪个城市转移时, 就产生一个在  $[0, 1]$  范围内的随机数。根据这个随机数按公式(4)确定用哪种方法产生蚂蚁转移的方向, 然后用对应的方法产生转移方向。

#### 1.3.2 ACS 的全局更新规则

在 ACS 中, 全局更新不再用于所有蚂蚁, 而是只对每一次循环中最优的蚂蚁使用。更新规则可用如下公式表示:

$$\text{且 } t(r,s) = \begin{cases} (1-a) \times t(r,s) + \frac{1}{L} & \text{如果路径 } (r,s) \text{ 是算法当前已求出的最优路径的一部分} \\ 0 & \text{否则} \end{cases} \quad (5)$$

其中,  $a$  为区间  $(0, 1)$  上的参数,  $L$  为算法已求出的最优路径长度。

#### 1.3.3 ACS 的局部更新规则

局部更新规则在所有蚂蚁完成每一次转移后执行。

$$t(r,s) = (1-a) \times t(r,s) + \frac{1}{L} \quad (6)$$

其中,  $a$  为区间  $(0, 1)$  上的参数。  $t(r,s)$  的取值有 3 种选择:

$$(1) \quad t(r,s) = 0$$

$$(2) \quad t(r,s) = t_0, \quad t_0 \text{ 为信息量的初始值。}$$

$$(3) \quad t(r,s) = \max_{k=1,2,\dots,m} J_k(s), \quad \text{其中, } J_k(s) \text{ 表示第 } k \text{ 只蚂蚁在}$$

城市  $s$  时还未经过的城市。采用这个公式的 ACS 后来被称为 Ant-Q。

由于采用第一种方案的蚁群算法运算结果较差, 第二、三种方案结果相近, 而第二种方案对计算量的要求较小, 在后面的实验中本文将采用第二种方案。

#### 1.3.4 ACS 算法描述

算法 1(ACS):

- (1) 初始化
- (2) 设置每只蚂蚁的位置为初始位置
- (3) 按公式(4)以及公式(1)计算每只蚂蚁将要转移到的位置
- (4) 按公式(6)执行局部更新规则
- (5) 若本次循环内每只蚂蚁都执行完步骤(3)、(4)则执行步骤(6), 否则转至步骤(3)
- (6) 若每只蚂蚁都已完成一个完整的路径, 则执行步骤(7), 否则转至步骤(3)
- (7) 按公式(5)执行全局更新规则
- (8) 重复进行步骤(2)-(7), 直到重复执行这些步骤的次数超过指定次数或所求得解在最近若干代内无明显改进
- (9) 输出结果

#### 1.3.5 蚁群算法的不足之处以及本文改进算法的思路

由前面对蚁群算法的介绍可知, 蚁群算法在运算过程中, 蚁群的转移是由各条路径上留下的信息量的强度和城市之间的距离来引导的。蚁群运动的路径总是趋近于信息量最强的路径。通过对蚁群以及蚁群算法的研究表明, 不论是真实蚁群系统还是人工蚁群系统, 通常情况下, 信息量最强的路径与所需要的最优路径比较接近。然而, 信息量最强的路径不是所需要的最优路径的情况仍然存在, 而且在人工蚁群系统中, 这种现象经常出现。这是由于在人工蚁群系统中, 各路径上的初始信息量是相同的, 蚁群创建的第一条路径所用到的信息就主要是城市之间的距离信息, 这时, 蚁群算法等价于贪心算法, 这一次蚁群在所经过的路径上留下的信息就不一定能反映出最优路径的方向, 特别是蚁群中个体数目较少或者所计算的路径的组合较多时, 就更不能保证蚁群创建的第一条路径能引导蚁群走向全局最优路径。这一次循环中, 蚁群留下的信息会因正反馈作用使这条不是最优, 而且可能是离最优解相差很远的路径上的信息得到不应有的增强而阻碍以后的蚂蚁发现更好的全局最优解。不仅是第一次循环所创建的路径可能对蚁群产生误导, 任何一次循环, 只要这次循环所利用的信息较平均地分布在各个方向上, 这次循环所产生的路径就可能会对以后蚁群的选择产生误导。因此, 蚁群所找出的解需要通过一定的方法来增强, 使蚁群所留下的信息尽可能地不对以后的蚁群产生误导。

本文将要介绍的改进算法就是增强蚁群所找出的解的一种方法。这种算法在 Ant Colony System 的基础上引入了杂交算子。当新的解产生时, 就对这些新的解进行杂交, 以产生更好的解, 从而对所求得解起了增强作用。由于杂交算子能对现有解进行重组, 很可能会发现更好的解(这也是遗传算法具有很好的搜索能力的原因)。文献[7]提出了用变异算子来增强解的方法。但是变异是随机的, 不能充分利用现有的信息, 而且变异的结果很可能比当前循环中发现的解更

差。本文介绍的杂交算子能够利用当前所搜索出的路径信息，能发现更好的解。

第4部分的实验将证明，ACS算法求解问题时确实会出现上述的误导现象。实验也将证明本文的改进算法能防止这种现象发生，并能在已有解的基础上产生更好的解。

## 2 杂交算子(Crossover Operator)

杂交算子主要用于演化规划(Evolutionary Programming)、遗传算法(Genetic Algorithm)等进化算法。根据所求解问题的不同、解的表示方式不同，杂交算子就会不同。这里仅介绍求解TSP问题、采用路径表示法来表示TSP的解时所用到的杂交算子中的次序杂交。

次序杂交是由Davis提出的。下面将举例说明这种杂交算子。

假设杂交的两个父体分别为：

$p1=(2\ 3\ 7\ 5\ 8\ 4\ 6\ 1)$   $p2=(5\ 2\ 3\ 1\ 8\ 6\ 7\ 4)$

这两个父体表示的参加杂交的两父体分别为：

路径1：由城市2出发，依次经过城市3,7,5,8,4,6,1

路径2：由城市5出发，依次经过城市2,3,1,8,6,7,4

杂交过程为：先随机选择两个杂交点。然后交换两父体中、在所选杂交点之间的部分(这个部分被称为杂交段)。将父体1的杂交段位置不变地复制到后代2中，同样也将父体2的杂交段位置不变地复制到后代1中。从父体1中删去与父体2的杂交段相同的内容。同样地，从父体2中删去与父体1的杂交段相同的内容。将父体1的剩下部分按原来的顺序依次填入后代2的第二个杂交点之后，当填到后代2的末尾时，转到后代2的首部继续填充，直到父体1中的剩余部分全部填入后代，对父体2以及后代1进行同样的操作。这样就完成了一次次序杂交。

例如，对上面所给出的两个父体，假设杂交点如下所示("I"表示杂交点)：

$p1=(2\ 3\ 7\ 5\ 8\ 4\ 6\ 1)$   $p2=(5\ 2\ 3\ 1\ 8\ 6\ 7\ 4)$

杂交产生的后代用 $o_1$ 和 $o_2$ 表示(其中，“#”表示暂时没有确定的城市)。将杂交段进行交叉复制的结果为：

$o_1=(\# \# 3\ 1\ 8\ \# \# \#)$   $o_2=(\# \# 7\ 5\ 8\ \# \# \#)$

将父体剩余部分填充到后代的结果(即杂交结果)为：

$o_1=(4\ 6\ 3\ 1\ 8\ 2\ 7\ 5)$   $o_2=(6\ 4\ 7\ 5\ 8\ 2\ 3\ 1)$

## 3 改进算法

改进算法主要在算法1的基础上增加了杂交算子。在杂交之前需要选择父体蚂蚁。本文算法采用的是转盘式选择(Roulette Wheel Selection)。采用这种选择策略需要先计算第k只蚂蚁本次循环所产生的路径的长度，并根据这个长度按式(7)计算各只蚂蚁被选择的概率：

第k只蚂蚁被选择的概率为

$$p_k = (1/L_k) / \sum_{i=1}^m (1/L_i) \quad (7)$$

然后根据选择概率 $p_k$ 把一个圆盘分成m份(共有m只蚂蚁)，其中第i个扇形的圆心角为 $2\pi p_i$ 。先选定一个参照点，在进行选择时，转动圆盘，当圆盘停止转动时，参照点落在哪个扇形中就选择哪个扇形对应的蚂蚁。用这种方式选择， $1/L_k$ 较大的蚂蚁(即 $L_k$ 较小的蚂蚁)被选中的几率更大。

选择蚂蚁杂交后，允许按一定的比例吸收略差的蚂蚁或只吸收比原解对应的路径长度小一定百分比的蚂蚁。对被吸收的蚂蚁执行全局更新规则(具体见下面所列算法)。

算法2：

- (1)初始化
- (2)设置每只蚂蚁的位置为初始位置
- (3)按公式(4)以及公式(1)计算每只蚂蚁将要转移到的位置
- (4)按公式(6)执行局部更新规则
- (5)若本次循环内每只蚂蚁都执行完步骤(3)、(4)则执行步骤(6)，否则转至步骤(3)
- (6)若每只蚂蚁都已完成一个完整的路径，则执行步骤(7)，否则转至步骤(3)
- (7)按公式(5)执行全局更新规则
- (8)根据公式(7)计算每只蚂蚁被选择的概率
- (9)根据每只蚂蚁被选择的概率，按照遗传算法中的转盘式选择从蚁群中选出两个父体
- (10)按第2部分所述杂交算子对选择的两个父体进行杂交
- (11)计算杂交生成的后代路径的长度 $L_1$ 、 $L_2$ ，若 $L_1 < \min\{L_{p1}, L_{p2}\}$ 或 $L_2 < \min\{L_{p1}, L_{p2}\}$ 则按公式(5)对 $L_1$ 或 $L_2$ 对应的路径执行全局更新，否则，不更新
- (12)重复进行步骤(2)-(11)，直到重复执行这些步骤的次数超过指定次数或所求得解在最近若干代内无明显改进
- (13)输出结果

## 4 实验结果

由于TSP问题曾被许多文献用于检验蚁群算法，而且TSP问题的应用范围广，因此本文也用TSP问题来检验本文的改进算法。本文用到的TSP问题来源于TSPLIB95。

以下实验的参数为： $\alpha=0.1$ ， $\beta=0.1$ ， $\rho=0.9$ ， $m=10$

表1是算法1(ACS)与算法2求解pr76(76城市的TSP问题)、d198(198城市的TSP问题)的平均结果。由该表可看出，本文介绍的算法具有比ACS更好的搜索解的能力，它能在更短的时间内发现更好的解。

表1 两种算法求解TSP问题的平均结果

所求解问题	取值	运行次数	每次运行的代数	路径平均长度	所使用的算法
pr76	0.9	20	100	140732.0084891	算法2
pr76	1	20	100	141846.118621812	算法2
pr76	1.1	20	100	142754.05865412	算法2
pr76	--	20	100	143786.65030045	算法1(ACS)
d198	1	20	500	19942.498489413	算法2
d198	0.9	20	500	19971.9725284186	算法2
d198	1.1	20	500	20038.3269025168	算法2
d198	--	20	500	20057.5074893379	算法1(ACS)

表2 较好解分布情况(以下实验，算法2中 $\alpha=1$ )

所求解问题	路径长度	所用算法
pr76	115278.652833249	算法2
pr76	118232.535651195	算法2
pr76	118313.462299959	算法2
pr76	119128.397600377	算法2
pr76	121969.832864618	算法2
pr76	127925.590527523	算法1(ACS)
d198	18674.9148154652	算法2
d198	18855.6604772039	算法2
d198	19097.4035088232	算法2
d198	19217.6588505707	算法1(ACS)
d198	19552.7866207975	算法2

(下转第176页)

### 2.3 第二层与第三层相结合的方案

上述第一种方案的缺陷在于不能使位于第三层的路由器享受到802.1P优先级带来的好处。因为802.1P优先级划分是在交换网络中实现的,所以只能在同一LAN或VLAN中实现。对于不同的LAN或VLAN之间的通信,由于要通过第三层的路由或交换设备转发,而第三层网络设备无法读取第二层帧头中的有关信息,所以关于帧优先级的相关信息会在位于第三层的网络互连设备中丢失。而根据目前网络通信的80/20规则,网络通信的大部分信息不再来自工作组内部,而是更多地来自工作组外部和对因特网的访问。也就是说,如果不能解决第三层上的QoS问题,校园网中的所有第三层网络设备(包括边界访问路由器,核心路由器等)就会成为校园网QoS的新瓶颈。而第二种方案则恰恰相反,只能提供基于第三层的QoS,不能为同一交换网络中的通信提供必要的QoS机制。所以将上述两种机制相结合,就得到了将差别服务和RSVP以及802.1P集成在一起的第三种解决方案。该方案可同时在OSI的第二层与第三层上提供QoS性能,但网络设计与管理的复杂度会较前面的两种方案有所增加。

(上接第76页)

表2是在大约50次实验(每次实验演化1000代)中,求解pr76和d198得到的较好解的分布情况。从表中可发现,求解pr76所得的解对应的路径长度在130000以下的,所用的算法大多数是本文的改进算法;求解d198所得的解对应的路径长度在19000以下的,所采用的算法全是本文改进算法。这也表明本文提出的算法具有较好的求解问题的能力。

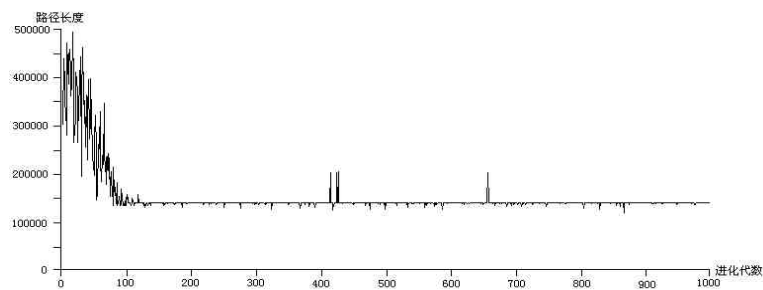


图1 本文改进算法最短路径演化图(=1)

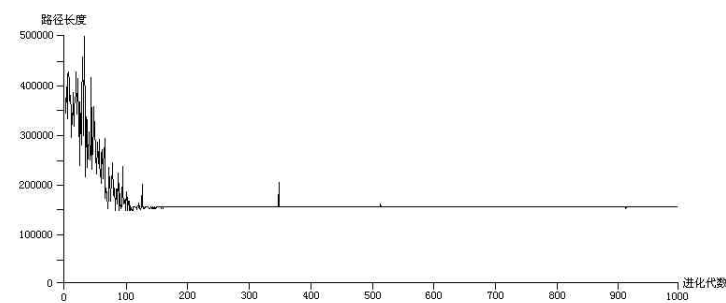


图2 算法1最短路径演化图

本次计算在第868代找到最短路径,其长度为118232.535651195

图1、图2是从所有实验中随机选出的两幅蚁群在每次循环中搜索出的最短路径的变化情况。不难发现,本文算法对应的曲线比算法1对应的曲线下落得更快,结合表1的数据可知,这绝不是偶然现象。这是由于在蚁群寻找路径的过程

### 3 结束语

通过对因特网中关于QoS第三层机制的有效利用,并结合基于交换的以太网技术所提供的通信优先级服务,本文所提出的若干解决方案对用交换以太网技术构建的其他园区网络实现QoS也有相当大的参考价值。但必须指出,对一个多层甚至是全交换网络来说,网络层以上尤其是传输层和应用层对QoS的实现也起着不可忽视的作用,如何在各层上寻求简单有效的QoS机制并将其恰当地结合起来以实现校园网QoS值得作更深入的研究。

#### 参考文献

- 1 Saunders S. Data Communications Gigabit Ethernet Handbook. McGraw-Hill, 1998
- 2 Jayant K, Lan C, Mohan K. Gigabit Ethernet Migrating to High-bandwidth LAN. Prentice Hall, 1998
- 3 <http://grouper.ieee.org/802/3>
- 4 赵慧玲, 胡琳. 宽带Internet网络技术. 北京: 电子工业出版社, 1999

中,变异算子在不断地对现有路径进行变异,从而产生许多较好的解,图中的曲线就表现为下降较快。这两幅图还有一个不同点:本文算法对应的曲线在演化代数较大时具有远远多于算法1的曲线的向下的突出部分。这表明杂交算子在演化代数较大时仍然能改进些,从而使本文算法求解问题时能够不断地发现更好的解。

本次计算在第111代找到最短路径,其长度为145637.766272631

### 5 结束语

本文在受遗传算法求解组合优化问题的方法的启示,提出了在蚁群算法中加入杂交算子来提高蚁群算法在现有基础上发现更好的解的能力,通过实验,证明了该算法的可行性。

#### 参考文献

- 1 Dorigo M, Maniezzo V, Colomi A. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics, Part-B, 1996, 26(1):1-13
- 2 Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1997, 1(1)
- 3 Dorigo M, Caro G D, Gambardella L M. Ant Algorithms for Discrete Optimization. Artificial Life, 1999, 5(3):137-172
- 4 Stützle T, Dorigo M. ACO Algorithms for the Traveling Salesman Problem. Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications. John Wiley & Sons, 1999
- 5 潘正君, 康立山, 陈毓屏. 演化计算. 北京: 清华大学出版社, 1998
- 6 Ansari N, Hou E. 李军, 边肇祺译. 用于最优化的计算智能. 北京: 清华大学出版社, 1999
- 7 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法. 计算机研究与发展, 1999, 36(10)