

具有自适应杂交特征的蚁群算法

朱海梅 朱庆保 胡 勇

(南京师范大学计算机系, 南京 210097)

E-mail: hm_zhu73@sina.com.cn

摘 要 蚁群算法是一种新型的具有许多优良特性的模拟进化算法,然而,在解决组合优化问题时,收敛速度比较慢,而且容易发生停滞。为此,论文提出了一种具有自适应杂交特征的蚁群算法,该算法在判断发生停滞的情况下进行杂交,而且杂交时利用现有信息,有方向地向着更好的结果进行杂交。仿真实验结果表明:该算法可以有效地克服停滞,提高搜索效率并改进优化结果。

关键词 蚁群算法 自适应杂交 旅行商问题

文章编号 1002-8331-(2004)22-0081-03 文献标识码 A 中图分类号 TP273

An Ant Colony Algorithm with Adaptive Crossover Features

Zhu Haimei Zhu Qingbao Hu Yong

(Computer Institute of Nanjing Normal University, Nanjing 210097)

Abstract: Ant colony algorithm is a novel simulated evolutionary algorithm which shows many promising characters. However, it needs much time and easily leads to stagnation in solving combinational optimization problem. So, this paper proposes an ant colony algorithm with adaptive crossover features which crossover is processed while stagnation is found and present information is considered and crossover is processed to better result. Numerical results demonstrate that stagnation can be effectively overcome and searching efficiency and result are also greatly improved.

Keywords: ant colony algorithm, adaptive crossover, Traveling Salesman Problem (TSP)

1 引言

蚁群算法(ant colony algorithm)是由意大利学者 M.Dorigo 最近几年才提出的一种新型的模拟进化算法^[1],是利用群集智能解决组合优化问题的典型例子^[2],并在求解旅行商问题(TSP: traveling salesman problem),资源二次分配问题(assignment problem)^[3]等优化问题取得了一系列较好的结果。然而,蚁群算法在解决优化问题时存在着两个主要缺陷:容易停滞,收敛速度较慢。为此,不少研究者提出了一些改进算法如:最近邻蚁群系统(ACS+NN: nearest neighbor to ant colony system)^[4],最近邻进化蚂蚁算法(EA+NN: nearest neighbor to evolutionary ant rules)^[5],最大最小蚁群系统(MMAS: Max-Min ant system)^[6],带杂交算子的蚁群算法^[7]等。这些改进的算法在一定程度上提高了收敛速度,但效果并不明显。其中,文献[7]提出了一种带杂交算子的蚁群算法,该算法在每一次蚁群搜索循环中都进行一次杂交,以期提高搜索的多样性。然而该算法中杂交点的选择是随机的,没有方向性,既耗时,又不能充分利用现有信息,因而改进的效果并不明显。为此,论文根据蚁群算法在开始时收敛快的特点,提出了一种利用已有信息进行自适应杂交的蚁群算法,该算法仅在停滞的情况下进行杂交,而且根据现有信息,向产生更优解的方向杂交。实验结果表明该算法在求解 TSP 问题时,进化较少的代数就能达到很好的优化结果,能有效地克服停滞。论文的组织结构如下:第 2 部分介绍基本的 ACS 算法,第 3 部分介绍具有自适应杂交特征的蚁群算法,第 4 部分给出算法的实验结果以及与其它算法的比较,第 5 部分为结论。

2 ACS 模型

人工蚁群算法是受到自然界中真实的蚂蚁集体行为的启发而提出的一种模拟进化算法。蚂蚁在寻找食物的运动过程中,能够在它所经过的路径上留下外激素(pheromone),而且蚂蚁在运动过程中能够感知这种物质的存在及其强度,蚂蚁倾向于朝着外激素浓度高的方向移动,因此,由大量蚂蚁组成的蚁群的集体行为便表现为一种信息正反馈现象:某路径长度越小,该路径上走过的蚂蚁越多,则后来该路径被选择的概率就越大。蚂蚁个体之间通过这种信息的交流达到搜索食物的目的。意大利学者 M.Dorigo 等人模拟蚂蚁这一觅食过程,于 90 年代提出了 AS(Ant System)优化算法,由于 AS 算法极易陷于停滞,经过不断改进,近几年又提出了 ACS(Ant Colony System)算法,ACS 算法的基本原理如下:

为了便于理解,作者以求解平面上 n 个城市的 TSP 问题为例进行说明。首先引入如下记号: m 为蚂蚁数量, $d_{ij}(i, j=1, 2, \dots, n)$ 表示城市 i 和城市 j 的距离, $\tau_{ij}(t)$ 表示 t 时刻在连线 ij 上残留的信息量。初始时刻,各条路径上信息量相等,设 $\tau_{ij}(0)=c$ (c 为常数)。蚂蚁 $k(k=1, 2, \dots, m)$ 在运动过程中根据各路径上的信息量决定转移方向, $p_{ij}^k(t)$ 表示 t 时刻蚂蚁 k 由位置 i 转移到位置 j 的概率。

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta}{\sum_u \tau_{iu}^\alpha(t) \eta_{iu}^\beta} & j, s \notin tabu_k \\ 0 & \text{else} \end{cases} \quad (1)$$

基金项目:江苏省教育厅自然科学基金项目(编号:01KJB520007)

作者简介:朱海梅(1973-),女,汉族,助教,南京师范大学计算机系硕士研究生,主要研究方向:智能控制。

其中, η_{ij} 为启发信息, $\eta_{ij} = \frac{1}{d_{ij}}$, α, β 分别反映第 k 只蚂蚁在

运动过程中所积累的信息素与启发因子在蚂蚁选择路径中的重要程度, $tabu_k (k=1, 2, \dots, m)$ 用以记录蚂蚁 k 所走过的城市, 并随时间做动态调整。蚂蚁的状态转移规则如下:

$$s = \begin{cases} \operatorname{argmax}_{j \in tabu_k} \{\tau_{ij}^\alpha(t) \eta_{ij}^\beta\} & q < q_0 \\ S & \text{else} \end{cases}$$

S 是由式(1)决定的状态变量, q_0 是一常数, $0 < q_0 < 1$, q 是一随机产生的数, $0 < q < 1$ 。

蚂蚁每走一步都要进行信息素的局部更新, 更新规则如下:

$$\tau_{ij} \leftarrow (1 - \rho_1) \tau_{ij} + \rho_1 \nabla \tau_{ij} \quad (2)$$

式中 $\nabla \tau_{ij} = (L_k)^{-1}$

其中 ρ_1 为局部信息素的挥发参数, $0 < \rho_1 < 1$;

L_k 为蚂蚁 k 到目前为止已走的路径长度。

当所有蚂蚁都完成一个闭合路径时, 只对最好蚂蚁的路径进行信息素的全局更新, 全局更新规则如下:

$$\tau_{ij} \leftarrow (1 - \rho_2) \cdot \tau_{ij} + \rho_2 \nabla \tau_{ij} \quad (3)$$

式中 ρ_2 为全局信息素的挥发参数, $0 < \rho_2 < 1$, $\nabla \tau_{ij} = \frac{Q}{L_k}$ (Q 为一常数), L_k 为蚂蚁走出的最佳路径长度。

3 具有自适应杂交特征的蚁群算法

ACS 算法及其各类改进算法虽然比 AS 算法有所改进, 但仍存在着引言中所述的缺陷。针对国内外已有研究的不足, 论文研究了自适应杂交蚁群算法, 该算法不是在每次循环都杂交一次, 而是判断最近的若干代(如 10 代)的搜索结果, 若处于停滞状态, 则进行杂交, 并充分利用现有信息, 以使杂交得到较好的结果。

3.1 杂交的基本思想

设 m 只蚂蚁所走出的 m 条路径构成一个遗传种群 A , 则有:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{11} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{21} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & a_{m1} \end{bmatrix}$$

式中, a_{ij} 为蚂蚁 i 所走过的第 j 个城市的城市号, 其中的一行则为蚂蚁 i 走过的一周的路径表, 在此表示为染色体个体。令适应度函数为:

$$L_i = \sum_{l=1}^n d_l \quad s=1, 2, \dots, m \quad (4)$$

式中, L_i 为蚂蚁所走环路的路径长度; d_l 为蚂蚁所走环路中第 l 个边长。用(4)式计算 m 个个体的适应度, 并按适应度进行升序排序, 设得到的两个最优个体(适应度最小的两个个体)为:

$$A_1 = (a_{11}, a_{12}, \dots, a_{1n}, a_{11})$$

$$A_2 = (a_{21}, a_{22}, \dots, a_{2n}, a_{21})$$

为了优化这两个最优个体中的最优路径, 将这两个个体进行杂交。若用通常的随机杂交方法有可能使杂交后得到新个体适应度变差, 如何通过杂交将个体中不太好的路径修改掉呢? 作者采用了依据杂交个体的路径信息进行杂交点选择的策略。

例如, 已得出了如图 1 所示的路径, 在图 1 中, 已知城市 4 到城市 6 的距离 d_1 为 29, 城市 4 到城市 5 的距离 d_2 为 15, 因为 d_1 远大于 d_2 , 则应该把 (4, 6) 这条边用 (4, 5) 替换, 相应地, 把 (5, 1) 用 (6, 1) 替换, 如图 2 所示。这种方法利用现有信息, 去掉染色体中质量较差的基因, 从而生成较好的个体。图 1 代表个体 (1, 2, 3, 4, 6, 5, 1), 其路径长度为: $15+10+12+30+12+29=108$, 图 2 代表个体 (1, 2, 3, 4, 5, 6, 1), 其路径长度仅为 78。

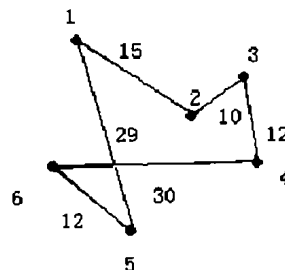


图 1

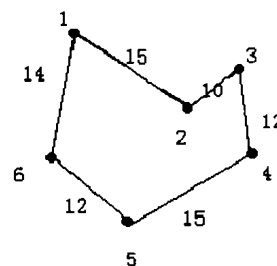


图 2

3.2 杂交方法

$$\text{令 } \gamma = L_{\min} - \frac{\sum_{i=1}^a L_{\min_i}}{a} \quad i \in a \quad (5)$$

式中, a —最近的周游代数, 例如可令 $a=10$; L_{\min_i} —在 a 次周游中第 i 次周游的最优路径。

当 $\gamma < 0.000001$ 时开始杂交。杂交按如下步骤进行:

步 1: 以 n 个城市分别为起点按它到其它城市的距离由小到大排序, 生成 n 个按距离升序排序表 E_1, E_2, \dots, E_n , 其中 $E_i = \{e_{i1}, e_{i2}, \dots, e_{in}\}$, e_{i1} 代表与第 i 个城市距离最小的城市号, e_{in} 代表与第 i 个城市距离最大的城市号。

步 2: 在最优个体 A_1 中和 A_2 中选择杂交点 C_1, C_2 ; 方法如下:

在个体 A_1 中, 若有 $d_{su} > d_{sk}$, 则令杂交点 $C_1 = j$, 式中, $s = a_{ij}, u = a_{i(j+1)}, j=1, 2, \dots, n; k$ —排序表 E_s 中位于排序号为 $\frac{2}{3}n$ 处的城市号; d_{su} — A_1 个体中城市 s 到城市 u 间的距离, d_{sk} —城市 s 到排序表 E_s 的 $\frac{2}{3}n$ 处的城市 k 的距离。

在个体 A_1 中, 设杂交点 C_1 前一个城市为 s , 对于个体 A_2 中的某城市 t , 若 $d_{st} < d_{sp}$, 则令杂交点 $C_2 = i$, 式中, $t = a_{2i}, i=1, 2, \dots, n; p$ —排序表 E_s 中位于排序号为 $\frac{1}{3}n$ 处的城市号。

步 3: 将两个个体从杂交点 C_1, C_2 处杂交, 同时进行合法性检查。方法如下:

将个体 A_1 中杂交点 C_1 以后的城市与个体 A_2 中杂交点 C_2 以后的城市互换, 若 A_1 中 C_1 以后城市号与 A_2 中 C_2 以前城市

号重复,则删除,再将后面城市依次前移。若长度不够,则将配偶个体的0号元素开始的不重复城市依次补充。

杂交策略用伪代码描述如下:

```
crossover()
{//参数初始化;
  计算最近10代最好路径长度的平均值 ave;
   $\gamma = L_{\min} - \text{ave}; // L_{\min}$  为10次周游中第1次周游的路径长度
  if( $\gamma < 0.000001$ )
    for( $j=0; j < N; j++$ )
    {
      if( $d_{a_j a_{j+1}} > d_{a_j k}$ )  $C_1=j$ ;
      for( $i=0; i < N; i++$ )
      {
        if( $d_{a_i a_j} < d_{a_i p}$ )  $C_2=i$ ;
        把  $A_1$  中从  $C_1$  开始的后面部分与  $A_2$  中从  $C_2$  开始的后面部分互
        换,并删去与前面相同的城市,长度不够则循环补充;
      }
    }
}
```

3.3 具有自适应杂交特征的蚁群算法

步1: 初始化 $\alpha, \beta, N_c=0$ (N_c 为进化代数计数器), N_{cmax} (最大进化代数), 以 n 个城市分别为起点按距离排序, 生成 n 个排序表。其余初始化同 ACS 算法。

步2: m 只蚂蚁随机分配到 n 个城市, 并将 m 个蚂蚁的出发点城市加入到禁忌表 $tabu_k$ 中 ($tabu_k$ 用于记录蚂蚁已走过的城市, 对已走过的城市, 以后不可再走)。

步3: 对于每一只蚂蚁, 从当前城市 i 选择下一个城市 j , 并将 j 加入禁忌表 $tabu_k$ 。选择下一城市的策略如下:

如果 $q < q_0$, 蚂蚁选择 $\max_{j \in tabu_k} \{\tau_{ij}^\alpha(t) \eta_{ij}^\beta\}$ 对应的城市, 否则, 根据公式(1)按赌轮盘规则选择城市 j 。式中, q 为随机数 ($0 < q \leq 1$); q_0 为预先设定的大于0小于1的数。

步4: 局部信息素动态更新

每只蚂蚁走完一个边后, 即按(6)式进行局部信息更新。

$$\tau_{ij} \leftarrow (1 - \rho_1) \tau_{ij} + \rho_1 \nabla \tau_{ij} \quad (6)$$

$$\text{式中 } \nabla \tau_{ij} = \frac{Q_1}{L_{gb}}$$

其中 ρ_1 为局部信息素的挥发参数, $0 < \rho_1 < 1$; Q_1 为常数。

L_{gb} 为蚂蚁 k 到目前为止已走的路径长度。

步5: 如所有蚂蚁都完成一个闭合路径, 则计算最优路径长度:

$$L_i = \sum_{l=1}^n d_l \quad i=1, 2, \dots, m$$

$$L_{\min} = \min\{L_1, L_2, \dots, L_i, \dots, L_m\}$$

步6: 进行全局信息素更新:

$$\tau_{ij}^{new} \leftarrow (1 - \rho_2) \tau_{ij}^{old} + \rho_2 \nabla \tau_{ij}^k$$

$$\nabla \tau_{ij}^k = \begin{cases} \frac{Q_2}{L_{\min}} & \text{if } (i, j) \in \text{global-best-tour} \\ 0 & \text{otherwise} \end{cases}$$

$(i, j) \in \text{global-best-tour}$ 表示蚂蚁 k 所走的城市边 (i, j) 属于最佳路径; Q_2 是个较大常数。 L_{\min} 为该次周游的最优路径长度, ρ_2 为全局信息素挥发参数, $0 < \rho_2 < 1$ 。

步7: 令 $N_c = N_c + 1$, 若 $N_c \bmod a = 0$, 用(5)式计算 γ , 若有

$\gamma < 0.000001$ 则为停滞状态, 此时按 3.1, 3.2 方法进行杂交, 生成新的子个体 B_1 和 B_2 。

步8: 对 B_1, B_2 进行适应度评价, 也就是计算两个子个体的路径长度 L_s , 若有 $L_s < L_k$, 则记忆 L_s 及其路径, $s=1, 2$, 式中, L_k 一至今为止得到的最优路径; 并按(7)式对该路径上的信息素进行全局更新:

$$\tau_{ij} \leftarrow (1 - \rho_3) \tau_{ij} + \rho_3 \nabla \tau_{ij} \quad (7)$$

$$\text{式中 } \nabla \tau_{ij} = \frac{Q_3}{L_s}$$

其中, ρ_3 为全局信息素挥发参数(其值不一定与 ρ_2 相同), $0 < \rho_3 < 1$; Q_3 为常数。

步9: 如 $N_c = N_{cmax}$, 则输出结果, 否则转步3。

4 实验结果

作者用 VC 语言编程实现了算法, 并求解了从 TSPLib 下载的若干 TSP 问题, 取得了较好的结果。在求解 eil51 问题时, 运行到 200 代就能达到最优路径长度 426, 在求解 st70 问题时, 运行 300 代就能达到最优路径长度 675, 在求解 eil76 问题时, 运行 149 次就能达到最优路径长度 538, 在求解 kroa100 问题时, 运行 577 次就能达到最优路径长度 21652, 比 ACS 算法到达同样的结果需要的进化代数大大减少。为了和最新改进算法比较, 选择部分城市的实验结果列入表 1~表 3。

表1 求解 eil51 问题结果比较

所用算法	进化代数	达到的最优结果	说明
ACS	2500 代以上	438	文献[5]的结果
EA+NN	2500 代以上	429	文献[5]的结果
文中算法	200 代	426	参数见表下括号
TSPLib 优化结果		426	

$$(\rho_1=0.9, \rho_2=0.2, \rho_3=0.1, Q_1=100, Q_2=1000, Q_3=1000, q_0=0.98, \alpha=1, \beta=4)$$

表2 求解 eil76 问题结果比较

所用算法	进化代数	达到的优化结果	说明
ACS	2500 代以上	591	文献[5]的结果
EA+NN	2500 代以上	573	文献[5]的结果
文中算法	149 代	538	参数见表下括号
TSPLib 优化结果		538	

$$(\rho_1=0.9, \rho_2=0.2, \rho_3=0.1, Q_1=100, Q_2=1000, Q_3=1000, q_0=0.98, \alpha=1, \beta=3)$$

表3 求解 kroa100 问题结果比较

所用算法	进化代数	达到的最优结果	说明
ACS	2944 代以上	22468	文献[5]的结果
EA+NN	2944 代以上	21698	文献[5]的结果
文中算法	577 代	21652	参数见表下括号
TSPLib 优化结果		21282	

$$(\rho_1=0.8, \rho_2=0.8, \rho_3=0.97, Q_1=800, Q_2=11000, Q_3=11000, q_0=0.99, \alpha=3, \beta=2.6)$$

从表中结果可知: 在求解 TSP 问题时, 文中算法比最新改进算法需要的进化代数减少了数十倍, 优化结果也比最新改进算法减小 3~46。

5 结论

论文提出的自适应杂交蚁群算法在发生停滞的情况下, 根据现有信息, 向着产生更优解的方向杂交, 从而提高了搜索的多样性, 较好地克服了停滞, 使达到最优解所需要的进化代数大大减少, 并能达到更好的优化结果。实验说明, 自适应杂交蚁

(下转 96 页)

表3 Schaffer 函数的计算结果

运行 序号	x_1	x_2	最大值	迭代 次数
1	-0.00006893064117	0.02359602363216	0.99944277008649	100
2	0.00266897727737	-0.00201653832817	0.99998879898559	200
3	0.64755837490020 $\times 10^{-3}$	0.09539878203412 $\times 10^{-3}$	0.99999957113885	300
4	-0.02103773691690 $\times 10^{-3}$	-0.22898495745093 $\times 10^{-3}$	0.99999994707043	400
5	0.29745094521479 $\times 10^{-3}$	-0.25218472712814 $\times 10^{-3}$	0.99999984777373	500

$$\begin{cases} \min f_4(x, y) = (4 - 2.1x^2 + \frac{1}{3}x^4)y^2 + xy + (-4 + 4y^2)y^2 \\ -3 \leq x \leq 3, -2 \leq y \leq 2 \end{cases} \quad (9)$$

该函数有 6 个局部极小点, 全局极小点为 (-0.0898, 0.7126) 和 (0.0898, -0.7126) 最小值为 -1.031628。文[2]中的启发式遗传算法在种群规模 $N=24$, 最大代数 200, 交叉概率 $P_c=0.6$ 和变异概率 $P_m=0.05$ 条件下, 连续运行 20 次后得到的最优值中最小的为 -1.031624, 最大的为 -0.972058, 平均值为 -1.022866。而按照文中的方法在相同参数条件下连续运行 20 次所得最小值如下:

-1.03162823794619 -1.03162845341285 -1.03162845054808
-1.03162841662132
-1.03162845347831 -1.03162753963073 -1.03162819560760
-1.03162844068802
-1.03162845139744 -1.03162844041091 -1.03162679580771
-1.03162749436873
-1.03162842381386 -1.03162845267178 -1.03162829679812
-1.03162723655362
-1.03162840685138 -1.03162844666815 -1.03162709360706

(上接 80 页)

明, 采用基于访问概率的固定扇出的非平衡索引树和可变扇出的索引树都可以降低平均访问代价。另外, 由于 VF 算法采用了可变扇出, 因而其性能优越于 CF 算法。

(收稿日期: 2003 年 11 月)

参考文献

1. Ming-Syan Chen, Kun-Lung Wu, Philip S Yu. Optimizing Index Allocation for Sequential Data Broadcasting in Wireless Mobile Computing[J]. IEEE Transactions on Knowledge and Data Engineering, 2003; 15(1): 161~173
2. T Lmielinski, S Viswanathan, B R Badrinath. Data on Air: Organization and Access[J]. IEEE Transactions on Knowledge and Data Engi-

(上接 83 页)

群算法在一定程度上改善了周游的质量, 从而能在较小的循环次数下, 达到较好的优化结果。但是, 到目前为止, 国际上各种改进的蚁群算法都是基于实验分析, 还没有形成坚实的数学基础, 各种参数的选取也没有理论指导, 所以, 从理论上还有许多要解决的问题, 这都值得进一步研究。(收稿日期: 2003 年 11 月)

参考文献

1. Marco Dorigo, Vittorio Maniezzo, Alberto Colomi. Ant system: Optimization by a colony of cooperating agents[J]. IEEE TRANSACTIONS ON SYSTEM, MAN AND CYBERNETICS - PART B CYBERNETICS, 1996; 26(1): 29~41

-1.03162836778009

其中最小的为 -1.03162845347831, 最大的为 -1.03162679580771, 平均值 -1.0316281047330975。20 个最优解的平均值为 (0.0898260819911145, -0.712590946516917) 或 (-0.0898260819911145, 0.712590946516917)。文中的混合遗传算法优势是明显的。

4 结论

论文提出的新型混合遗传算法是在标准遗传算法的基础上引入新的交叉运算和变异运算, 将通常的选择算子改为择优交叉, 在搜索过程中保留每代中的最优个体, 充分利用已有的个体适应值信息, 使得算法在交叉过程中有效地探测出最优区域, 变异运算又对最优区域随机进行了更为精细的搜索, 从而使该混合算法可以有效防止早熟现象的发生, 同时又具有较高的求解精度和较快的收敛速度。针对常用测试函数的算例进一步验证了这种新型混合遗传算法的有效性和优越性。

(收稿日期: 2003 年 11 月)

参考文献

1. 袁礼海, 宋建社等. 混合遗传算法及与标准遗传算法对比研究[J]. 计算机工程与应用, 2003; 39(12): 124~125
2. 金聪. 启发式遗传算法及其应用[J]. 数值计算与计算机应用, 2003; 24(1): 30~35
3. 周明, 孙树栋. 遗传算法原理及应用[M]. 国防工业出版社, 1999
4. 黄樟灿, 陈思多, 李亮. 演化计算中的种群隔离与自聚集[J]. 软件学报, 2002; 13(4): 827~832
- neering, 1997; 9(3): 353~372
3. Chi-Hao Hsu, Guanling Lee, Arbee L P Chen. Index and Data Allocation on Multiple Broadcast Channels Considering Data Access Frequencies[C]. In: Mobile Data Management, 2002 Proceedings, Third International Conference on, 2002: 87~93
4. Shou-Chih Lo, Abree L P Chen. Optimal Index and Data Allocation in Multiple Broadcast Channels[C]. In: Data Engineering, 2000, Proceedings, 16th International Conference on, 2000: 293~302
5. 李霖, 周兴铭. 非对称网络环境中数据广播的带索引多盘调度算法[J]. 计算机研究与发展, 1999; 36(2): 219~224
6. 林怀忠, 陈纯等. 移动环境下事务一致性的动态实现[J]. 计算机研究与发展, 2002; 39(1): 100~104
7. 林怀忠, 陈纯等. 数据复制与一致性[J]. 计算机工程与应用, 2001; 37(20): 107~108

2. Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behavior[J]. Nature, 2000; 406(6)
3. Thomas Stutzle, Marco Dorigo. ACO Algorithms for the Quadratic Assignment Problem[J]. New Ideas in optimization, McGraw-Hill, 1999
4. Cheng-Fa Tsai, Chun-Wei Tsai, Ching-Chang Tseng. A New Approach for Solving Large Traveling Problem[J]. IEEE, 2002
5. Cheng-Fa Tsai, Chun-Wei Tsai. A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules[J]. IEEE, 2002
6. Stutzl T, Hoos H H. The MAX-MIN ant system and local search for the traveling salesman problem[C]. In: Proc IEEE International Conference on Evolutionary Computation (ICEC, 97), 1997: 309~314
7. 陈烨. 带杂交算子的蚁群算法[J]. 计算机工程, 2001; 27(12)