

## 带佳点交叉算子的非均匀窗口蚁群算法

张 然<sup>1,2</sup>, 贾瑞玉<sup>1</sup>, 钱光超<sup>1</sup>, 李龙澍<sup>1</sup>

(1. 安徽大学 计算机科学与技术学院, 安徽 合肥 230039;

2. 铜陵学院 计算机科学与技术系, 安徽 铜陵 244000)

**摘 要:**基本蚁群算法具有较强的鲁棒性,但收敛慢并容易陷入局部最优。针对这些缺陷,通过将蚂蚁的搜索空间缩减在非均匀的小窗口中,减少了蚂蚁的搜索时间。并将佳点集遗传算子引入到解的优化中来,提出了带佳点杂交算子的非均匀窗口蚁群算法,从本质上探索蚁群算法的寻优能力。实验结果表明:新提出的算法明显快于基本蚁群算法,佳点集杂交算子对解的优化有较好的作用。但需要继续探索避免陷入局部最优的方法,以及算法各部分所采用的方法的平衡问题。

**关键词:**蚁群算法;佳点集;交叉算子;窗口

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2007)12-0068-03

Ant Colony Algorithm with Good - Point Crossover Operator  
Based on Different Size WindowZHANG Ran<sup>1,2</sup>, JIA Rui-yu<sup>1</sup>, QIAN Guang-chao<sup>1</sup>, LI Long-shu<sup>1</sup>

(1. School of Computer Science and Technology, Anhui University, Hefei 230039, China;

2. Dept. of Computer Science and Technology, Tongling College, Tongling 244000, China)

**Abstract:** Basic ant colony algorithm has strong robustness, but has slow convergence and easily be trapped in a local optimum. Aiming at these disadvantages, by restricting the searching space of ants in a different size small window, has a big decrease of the searching time. By a good - point set genetic operator is introduced into the optimizing of solution, proposes an ant colony algorithm with good - point crossover operator based on different size window, exploring the ability of searching best solution of ACA in essential. Experiment shows that new algorithm is obviously fast than basic ant algorithm, and good point crossover operator is benefit to optimization of solution. But it need to further explore the method of avoiding trapping into local optimum, and the balance of method, which is used in every part of algorithm.

**Key words:** ant colony algorithm; good - point set; crossover operator; window

## 0 引言

蚁群算法(ant colony algorithm)是由意大利学者Dorigo等人<sup>[1,2]</sup>20世纪90年代初期通过模拟自然界中蚂蚁集体寻径的行为而提出的一种新的仿生类进化算法,是继模拟退火、遗传算法、禁忌搜索等之后的基于种群的启发式仿生进化系统。蚁群算法包含两个基本阶段:适应阶段和协作阶段。在适应阶段,各候选解根据积累的信息不断调整自身结构;在协作阶段,候选解之间通过信息交流,以期望产生性能更好的解,这类类似于学习自动机的学习机制。蚁群算法最早成功应用

于解决著名的旅行商问题(traveling salesman problem, TSP),随后又在二次分配、图着色、车辆调度、集成电路设计及通信网络负载等问题的解决上表现突出,吸引人们的很大关注<sup>[3]</sup>。

蚁群算法采用了分布式正反馈的并行计算机制,易于与其他方法结合,具有较强的鲁棒性<sup>[4]</sup>。但是,基本蚁群算法是在整个解空间中搜索,并通过启发式方法来寻找最优点。这种方法的缺点是搜索时间长、容易陷入局部最优点。现有的大部分改进的算法都是通过改进信息素的更新方式和引入某种变异机制来使算法的运行跳出局部最优点,但效果并不是很好,尤其是城市数目大于50的问题,随着问题维数城市数目的增大,其解空间急剧放大,使得在有限时间内很难找到满意的结果。而且通过调整信息素和加入变异的改进蚁群算法,对信息素强度 $Q$ 、 $\alpha$ 、 $\beta$ 权重参数是非常敏感

收稿日期:2007-02-03

基金项目:安徽省教育科研项目(2006KJ088B)

作者简介:张 然(1981-),男,安徽铜陵人,硕士研究生,研究方向为智能软件;贾瑞玉,副教授,研究方向为计算机图形学、数据挖掘、人工智能。

的,需要通过大量实验,才能测试出合适的参数组合。这就大大限制了该算法的使用。针对这些缺陷,文中提出了用非均匀小窗口来控制蚂蚁的搜索空间的思路,既有效减小了搜索空间,同时又考虑到城市分布密度的情形,在减少搜索时间的同时提高了效率。并且,将佳点集遗传算子引入到蚁群算法解的优化中,从新的角度改进解的质量。

## 1 基本蚁群算法

蚁群算法源于蚂蚁的觅食行为。蚂蚁在觅食的过程中,在走过的路径上留下一称之谓为信息素(pheromone)的物质。其它蚂蚁就是靠这种信息素的指引往返于食物源与巢穴之间。某条路径上经过的蚂蚁越多,路径上残留的信息素强度就越高,蚂蚁在运动的过程中总是倾向朝信息素强度高的方向运动。但各条路径上的信息素都会随着时间的流逝而不断蒸发减少。当几只蚂蚁分别沿着不同的路径回巢,长度越短的路径上信息素的强度越高,其它蚂蚁选择这样路径的概率也越大,选择的蚂蚁越多,路径上的信息素强度也会更高,这是一种典型的正反馈机制,使更多的蚂蚁集中到最短的路径上来。蚁群算法就是模拟上述蚂蚁的觅食行为,设计人工蚂蚁,使其随机搜索不同的路径,并留下会随时间变化而蒸发的信息素,根据信息素强度来寻找最短路径。

蚁群算法通常用于求解复杂的组合优化问题<sup>[5]</sup>。这里以平面上  $m$  个城市的 TSP 问题为例说明基本蚁群算法模型。

设  $n$  是蚁群中蚂蚁的数量,  $d_{ij}(i, j = 1, 2, \dots, m)$  表示城市  $i$  和城市  $j$  之间的距离,  $\tau_{ij}(t)$  表示  $t$  时刻在城市  $i$  与城市  $j$  连线上信息素的浓度,初始时刻,各条路径上信息素的浓度相同,设  $\tau_{ij}(0) = C$  ( $C$  为常数)。蚂蚁  $k(k = 1, 2, \dots, n)$  在运动过程中,根据各条路径上的信息素的浓度决定转移方向,  $P_{ij}^k(t)$  表示在  $t$  时刻蚂蚁  $k$  从城市  $i$  转移到城市  $j$  的概率,其计算公式如下:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{k \in \text{tabu}_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}(t)]^\beta}, & j \in \text{tabu}_k \\ 0, & j \notin \text{tabu}_k \end{cases} \quad (1)$$

其中,  $\text{tabu}_k(k = 1, 2, \dots, n)$  为蚂蚁  $k$  已走过城市的集合。开始时  $\text{tabu}_k$  中只有一个元素,即蚂蚁  $k$  的出发城市,随着算法的进行,  $\text{tabu}_k$  中的元素不断增加。随着时间的推移,以前留在各条路径上的信息素逐渐消失,用参数  $\rho$  表示信息素的挥发程度,所有蚂蚁完成一次循环,各路径上信息素浓度根据下式调整:

$$\left. \begin{aligned} \tau_{ij}^{\text{new}} &= (1 - \rho) \tau_{ij}^{\text{old}} + \rho \Delta \tau_{ij}, \rho \in (0, 1) \\ \Delta \tau_{ij} &= \sum_{k=1}^n \Delta \tau_{ij}^k \end{aligned} \right\} \quad (2)$$

$\Delta \tau_{ij}^k$  表示第  $k$  只蚂蚁的本次循环中留在路径  $ij$  上的信息素的浓度,  $\Delta \tau_{ij}$  表示本次循环所有蚂蚁在路径  $ij$  上所释放的信息素浓度之和。

Dorigo 曾给出 3 种不同的蚁群算法模型,分别称为蚁周算法、蚁量算法、蚁密算法。这 3 种模型中,它们的差别在于  $\Delta \tau_{ij}^k$  的计算表达式不同,后者利用的是局部信息,而前者利用的是整体信息。一般都采用蚁周算法模型作为基本模型:

$$\Delta \tau_{ij}^k = \begin{cases} Q/L_k, & \text{若第 } k \text{ 只蚂蚁在时刻 } t \text{ 和} \\ & t+1 \text{ 之间经过城市 } i, j \\ 0, & \text{否则} \end{cases} \quad (3)$$

其中  $L_k$  为第  $k$  只蚂蚁在本次循环中所走过路径长度。

算法中的参数  $\alpha, \beta, Q, \rho$  一般都采用实验的方法确定。由算法的复杂度分析理论可知,该算法的复杂度为  $O(nc \cdot m^3)$ , 其中  $nc$  表示循环次数,  $m$  表示城市的个数。

## 2 带佳点交叉算子非均匀小窗口蚁群算法

### 1) 采用小窗口减小搜索空间。

以往的基本蚁群算法都是在整个搜索空间上进行搜索,即完全搜索访问城市的禁忌表  $\text{tabu}$ 。当城市的规模达到一定程度的时候(如 50 个城市以上),搜索空间就急剧地被放大,完全搜索禁忌表需要大量的时间,非常影响算法的效率。根据小窗口搜索的思想,在一个  $N$  城市的 TSP 问题中,任何一个城市有  $N-1$  个从该城市出发的路径,而在这  $N-1$  条路径中,只有最短的若干条路径中的某一条才可能是组成最优解的路径之一,其它的较长的路径不可能是最优解中的一条路径。因此,将蚂蚁下一步要选择的城市限制在一个较小数量的范围内(最大不超过  $n/2$ ),从而极大地减小了搜索的空间,提高了效率。这里借鉴文献<sup>[6]</sup>提供的数,将每个蚂蚁的选择范围确定在  $\min\{n-x, \text{MAXPC}\}$  内,这里  $\text{MAXPC}$  为最大可选城市数,  $x$  为蚂蚁已经走过的城市。

窗口的大小对整个搜索过程有一定影响。窗口太小,容易陷入局部最优;窗口太大,设置窗口的作用就不能很好地体现出来。因此,考虑  $n$  个城市在二维网格上的分布情形,当蚂蚁所处的城市位于城市密集的区域时,邻近城市和可选的城市较多,因此,应该减小邻近城市对选择概率的影响,加大该蚂蚁搜索的窗口;反之,应减小蚂蚁搜索的窗口。

### 2) 用交叉操作优化解集。

对于交叉算子和变异算子在进化优化中的相对重要性,已有许多研究进行了评估。Holland 认为变异在适应中的基本作用并不是产生新的结构,而是为了保证重组算子可以开拓到各种等位基因值的全范围。Grefenstette, Goldberg, Davis 等也强调了交叉算子比变异算子更为重要<sup>[7]</sup>。Fogel 与 Atmer 的实验以及 Schaffer 和 Eshelman 的实验也至少证明了:仅有变异的进化不能一定优于具有交叉和变异的进化。

在引入变异思想的改进蚁群算法中,变异的策略大多带有很大的随机性。这使得变异的结果很多时候并不利于快速有效地改进解的质量。文中在解的优化上采用佳点交叉算子<sup>[8]</sup>。该算子是利用数论中的佳点集理论和方法重新设计的交叉算子。佳点理论能保证在所有取  $n$  个点的子集中,用佳点法求到的子集最能代表其家族性能。在佳点理论中,钟开莱定理表明,对一未知均匀分布的对象,用佳点集法取  $n$  点比用随机法取  $n$  点,其偏差要小得多(差平方倍),这是佳点集方法收敛速度快的理论依据。

用佳点交叉算子对本次迭代的最优解和次优解进行交叉操作,对所得的结果路径进行信息素全局更新<sup>[9]</sup>。因为佳点交叉操作要求进行交叉的两个染色体有部分等位基因是相同的,因此,蚂蚁寻得的路径在表示上有一定的敏感性。这里对次优解表示的路径进行循环表示,即将  $a_1 a_2 \cdots a_n$  变换为  $a_n a_1 a_2 \cdots a_{n-1}$ ,以此类推,以消除起点城市对交叉操作的影响。

根据上述思路,将带佳点杂交算子的非均匀窗口蚁群算法描述如下:

Step 1 初始化参数  $\alpha$ 、 $\beta$ 、 $Q$ 、 $\rho$ ,将  $n$  个城市的邻接城市按距离非递减排序,分别以  $n$  个城市为起点按距离排序,生成  $n$  个排序表。设置代数计数器  $NC = \max GEN$ 。

Step 2 确定每个城市  $i$  的搜索窗口。首先将窗口大小定为它的最临近城市 MAXPC,然后根据城市分布的密集程度对窗口大小进行调整。

Step 3 在每个城市  $i$  的窗口城市(允许访问的城市)中,按伪随机方式选择节点:

$$\begin{cases} j = \arg \max_{j \in \text{tabu}_k} [\tau_{ij}(t)] [\eta_{ij}(t)]^\beta, & \text{if } q \leq q_0 \\ P_{ij}^k = \frac{[\tau_{ij}(t)] [\eta_{ij}(t)]^\beta}{\sum_{j \in \text{tabu}_k} [\tau_{is}(t)] [\eta_{is}(t)]^\beta}, & j \notin \text{tabu}_k, \text{ else} \end{cases}$$

$q$  为随机数( $0 < q \leq 1$ ),将  $j$  加入禁忌表  $\text{tabu}_k$  中。

Step 4 每只蚂蚁走过一条边后,进行信息素局部更新:

$$\tau_{ij}^{\text{new}} = (1 - \rho) \tau_{ij}^{\text{old}} + \rho \Delta \tau_{ij}$$

Step 5 当  $m$  只蚂蚁完成一次周游后,计算  $L_k$ ,

$L_{k\min}, L_{k\min} = \min\{L_k\}$ ,  $L_k$  是蚂蚁  $k$  完成一次周游的路径长度,保留  $L_{k\min}$  和本次最优路径表。

Step 6 令  $NC = NC - 1$ 。若  $\max GEN - NC < M$ ,转 Step 7。 $M$  为设定的开始交叉限值。

交叉策略为:  $m$  只蚂蚁完成一次周游后,保留本次迭代的最优解、次优解,采用佳点交叉算子分别对最优解和次优解进行交叉操作,保留交叉结果。

Step 7 对全局最优解,即交叉后的结果路径,进行信息素全局更新:

$$\tau_{ij}^{\text{new}} = (1 - \alpha) \tau_{ij}^{\text{old}} + \alpha \Delta \tau_{ij}^k$$

$$\Delta \tau_{ij}^k = \begin{cases} \sum_{l=1}^n \frac{Q}{d_l}, & \text{if } l \in \text{best\_tour} \\ 0, & \text{otherwise} \end{cases}$$

$l \in \text{best\_tour}$  表示蚂蚁  $k$  所走的城市边  $l$  属于最佳路径。

Step 8 将本次周游得到的  $L_{k\min}$  与已经得到的最优长度  $L_d$  作比较,如果  $L_{k\min} < L_d$ ,则用  $L_{k\min}$  替换  $L_d$ ,同时替换最优路径表。

Step 9 设置的计数值  $NC - 1$  不等于 0,清空并初始化禁忌表,重复上述过程。

### 3 实验结果

采用国际通用的 TSPLIB 提供的标准问题 ei51 进行实验 (<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>)。各种参数的设置如下:  $m = n = 51$ ,  $\alpha = 0.2$ ,  $\beta = 5$ ,  $\rho = 0.2$ ,  $Q = 500$ 。算法迭代 1000 次,结果如下:

收敛到最优解的进化速度比较(见表 1)。

表 1 算法进化速度比较

基本蚁群算法(代)	文中算法(代)
137	71

全局最优解的优化结果比较(见表 2)。

表 2 算法优化结果比较

基本蚁群算法	文中算法	改进值	TSPLIB 提供值
492.96	465.24	27.72	429.98

实验的结果表明,文中提出的算法明显快于基本蚁群算法,经过 70 几代就收敛到最优解。全局最优解收敛到 465.24,虽然与 TSPLIB 提供的最优解比较,只改进了 27.72,但说明了佳点集交叉思想在蚁群优化改进中的可行性。但对于解的进一步优化,有以下两点思考:

1) 算法在寻优过程结束时陷入了局部最优,在跳出局部最优的处理上,需要寻找一种与佳点集方法相

(下转第 75 页)

表 2 针对图 2 模拟数据的实验中两种聚类的类间离散度和 J 的值

	第 1 次	第 2 次	第 3 次	第 4 次
K-均值	42082.3611	41204.2499	36016.7951	36026.7866
改进的遗传聚类	35939.3833	35938.5744	35939.3724	35938.2509

### 4 算法复杂度分析

本次实验中,K-均值算法消耗时间为 100ms,改进的遗传空间聚类算法消耗时间为 842ms(均采用 4 次实验消耗时间的平均值),说明改进的遗传空间聚类算法相对而言计算速度较慢,这是因为遗传算法是从全局最优的角度来进行空间聚类的。

在改进的遗传空间聚类算法中,估算每个染色体的类内离散度和的复杂度是  $O(nd)$ (其中, $d$  是维数,在对空间数据进行聚类时就是 2),交叉操作的复杂度是  $O(n^2d)$ ,K-均值优化操作是  $O(nKd)$ 。由于变异概率较低,等位基因中只有很少一部分做变异操作。而且,K-均值优化操作是一个逐步减少的操作,并在达到一个局部最优的时候不改变染色体编码(除非该染色体编码被变异操作改变)。同样,如果一个染色体编码在上一代的进化中没有变化的话则在当前无需做 K-均值优化操作。事实上,K-均值优化操作只是在最初的进化阶段运行的比较多,在进化的最后阶段,只有染色体编码被变异操作改变时才运行 K-均值优化

操作。因此,如果将染色体的变化情况记录下来的话则可以有效地减少计算复杂度。

### 5 结 论

提出了一个改进的遗传空间聚类算法,该算法结合了遗传算法全局搜索的优点和 K-均值方法局部收敛速度快的特点,克服了传统的 K-均值对初始选值敏感和易陷入局部最优的缺陷,提高了遗传算法的收敛速率,并取得较好的结果。但是该算法依然存在一些缺陷,例如在指定聚类结果数目的情况下,对孤立点数据和“噪音”的处理不是很科学,需要做进一步完善。

#### 参考文献:

[1] Han Jiawei, Kamber M. Data Mining: Concepts and Techniques[M]. 北京:机械工业出版社,2001.  
 [2] 徐宗本,张讲社,郑亚林. 计算智能中的仿生学:理论与算法[M]. 北京:科学出版社,2003.  
 [3] Maulik U, Bandyopadhyay S. Genetic Algorithm - based clustering technique[J]. Pattern Recognition, 2000, 33(9): 1455 - 1465.  
 [4] 杨善林,倪志伟. 机器学习与智能决策支持系统[M]. 北京:科学出版社,2004.  
 [5] Krishna K, Murty M N. Genetic K - means Algorithm[J]. IEEE Transactions on System, Man and Cybernetics: Part B, 1999, 29(3): 433 - 439.

(上接第 70 页)

适应的变异方法或其他方法来解决;

2)佳点集方法能够保证子代的解集保留父代最大特性,但在对解的优化过程中并未找到现有的最优解(429.98)。需要进一步探索佳点集方法的机理与蚁群算法信息素更新方式之间的匹配问题,研究其本质上是否存在的相互影响。

### 4 结 语

提出了带佳点杂交算子的非均匀窗口蚁群算法。通过缩小蚂蚁的搜索空间有效地改进了蚁群算法中搜索时间过长的缺点。同时,采用佳点交叉算子对蚁群算法的解进行优化,改进解的质量。实验表明佳点交叉算子对解的优化有较好的作用,但对算法各部分所采用的方法的平衡问题还有待于进一步研究。

#### 参考文献:

[1] Dorigo M, Maniezzo V, Colomi A. Ant system: optimization

by a colony of cooperating agents[J]. IEEE Transaction on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 28 - 41.  
 [2] Stutzlet, Hoos H H. MAX - MIN ant system[M]. Future Generation Computer Systems, 2000, 16(8): 889 - 914.  
 [3] 段海滨,王道波,朱家强,等. 蚁群算法理论及应用研究的进展[J]. 控制与决策, 2004, 19(12): 1321 - 1326.  
 [4] 段海滨,王道波. 一种快速全局优化的改进蚁群算法及仿真[J]. 信息与控制, 2004, 33(2): 241 - 244.  
 [5] 朱庆保,杨志军. 基于变异和动态信息素更新的蚁群优化算法[J]. 软件学报, 2004, 15(2): 185 - 192.  
 [6] 全惠云,文高进. 求解 TSP 的子空间遗传算法[J]. 数学理论与应用, 2002, 22(1): 36 - 39.  
 [7] 王正志,薄 涛. 进化计算[M]. 长沙:国防科技大学出版社, 2000: 287 - 296.  
 [8] 张 铃,张 钺. 佳点集遗传算法[J]. 计算机学报, 2001, 24(9): 1 - 6.  
 [9] 丁建立,陈增强,袁著祉. 遗传算法与蚂蚁算法的融合[J]. 计算机研究与发展, 2003, 40(9): 1351 - 1356.