

# 基于分布均匀度的自适应蚁群算法\*

陈 峻<sup>1,2+</sup>, 沈 洁<sup>1</sup>, 秦 玲<sup>1</sup>, 陈宏建<sup>1</sup>

<sup>1</sup>(扬州大学 计算机科学与工程系, 江苏 扬州 225009)

<sup>2</sup>(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

## An Adaptive Ant Colony Algorithm Based on Equilibrium of Distribution

CHEN Ling<sup>1,2+</sup>, SHEN Jie<sup>1</sup>, QIN Ling<sup>1</sup>, CHEN Hong-Jian<sup>1</sup>

<sup>1</sup>(Department of Computer Science and Engineering, Yangzhou University, Yangzhou 225009, China)

<sup>2</sup>(National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

+ Corresponding author: Phn: 86-514-7978307, Fax: 86-514-7887937, E-mail: lchen@yzcn.net

<http://www.yzu.edu.cn>

Received 2002-11-12; Accepted 2003-01-20

Chen L, Shen J, Qin L, Chen HJ. An adaptive ant colony algorithm based on equilibrium of distribution.

*Journal of Software*, 2003,14(8):1379~1387.

<http://www.jos.org.cn/1000-9825/14/1379.htm>

**Abstract:** To settle the contradictory between convergence speed and precocity and stagnation in ant colony algorithm, an adaptive ant colony algorithm, which is based on the equilibrium of the ant distribution, is presented. By dynamically adjusting the influence of each ant to the trail information updating and the selected probabilities of the paths according to the equilibrium of the ant distribution, the algorithm can keep good balance between accelerating convergence and averting precocity and stagnation. Experimental results on symmetric and asymmetric TSP show that the method presented in this paper has much higher convergence speed and stability than that of classical ant colony algorithm, and is more suitable for solving large scale TSP.

**Key words:** ant colony algorithm; optimization; traveling salesman problem

**摘 要:** 针对蚁群算法加速收敛和早熟停滞现象的矛盾,提出一种基于分布均匀度的自适应蚁群算法,以求在加速收敛和防止早熟、停滞现象之间取得很好的平衡.该算法根据优化过程中解的分布均匀度,自适应地调整路径选择概率的确定策略和信息量更新策略.以数种对称和不对称 TSP(traveling salesman problem)问题为例所进行的计算结果表明,该方法比一般蚁群算法具有更好的收敛速度和稳定性,更适合于求解大规模的 TSP 问题.

**关键词:** 蚁群算法;优化;旅行商问题

**中图法分类号:** TP18 **文献标识码:** A

---

\* Supported by the National Natural Science Foundation of China under Grant No60074013(国家自然科学基金); the National Foundation of High Performance Computing of China under Grant No.0029(国家高性能计算基金); the Natural Science Foundation of Jiangsu Provincial Educational Commission of China under Grant No.02KJB520009(江苏省教育厅自然科学基金); the Foundation of National Key Laboratory for Novel Software Technology, Nanjing University(南京大学计算机软件新技术国家重点实验室开放基金)

第一作者简介: 陈峻(1951-),男,江苏宝应人,博士,教授,主要研究领域为并行算法,计算机体系结构,优化算法.

蚁群算法是近年来出现的一种新型的模拟进化算法,它是由意大利学者 M.Dorigo 等人首先提出来的<sup>[1]</sup>,他们充分利用蚁群搜索食物的过程与旅行商问题(TSP)之间的相似性,解决了 TSP 问题<sup>[2]</sup>,取得了很好的结果.随后,蚁群算法被用来求解分配问题、武器-目标分配(weapon-target assignment)问题<sup>[3]</sup>、指派问题<sup>[4]</sup>、频率分配问题<sup>[5]</sup>、电力系统故障诊断<sup>[6]</sup>等 NP 完全问题,显示出蚁群算法在求解复杂优化问题方面的优越性.

实验观察表明,蚂蚁在运动过程中会留下一种分泌物,其后面的蚂蚁可根据前边走过的蚂蚁所留下的分泌物选择其要走的路径.一条路径上的分泌物越多,蚂蚁选择这条路径的概率就越大.因此,蚂蚁群体的集体行为实际上构成一种学习信息的正反馈现象,蚂蚁之间通过这种信息交流寻求通向食物的最短路径.蚁群算法正是模拟了这样的优化机制,即通过个体之间的信息交流与相互协作最终找到最优解.

我们以 TSP 问题为例来说明基本蚁群算法的框架.设有  $n$  个城市,  $d_{ij}(i, j=1, 2, \dots, n)$  表示城市  $i$  和  $j$  间的距离,  $\tau_{ij}(t)$  表示在  $t$  时刻城市  $i$  和  $j$  之间的信息量,我们以此来模拟实际蚂蚁的分泌物.设共有  $m$  只蚂蚁,用  $p_{ij}^k(t)$  表示在  $t$  时刻蚂蚁  $k$  由城市  $i$  转移到城市  $j$  的概率,则

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

其中,  $allowed_k$  表示蚂蚁  $k$  下一步允许走过的城市的集合,  $\alpha$  表示路径上的信息量对蚂蚁选择路径所起的作用大小,  $\eta_{ij}$  为由城市  $i$  转移到城市  $j$  的期望程度,例如,可以取  $\eta_{ij} = 1/d_{ij}$ .  $\beta$  表示  $\eta_{ij}$  的作用.当  $\alpha=0$  时,算法就是传统的贪心算法;而当  $\beta=0$  时,就成了纯粹的正反馈的启发式算法.经过  $n$  个时刻,蚂蚁可走完所有的城市,完成一次循环.每只蚂蚁所走过的路径就是一个解.此时,要根据下面公式对各路径上的信息量作更新:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}, \quad (2)$$

其中  $\rho \in (0, 1)$ , 表示信息量  $\tau_{ij}(t)$  随时间的推移而衰减的程度.信息增量  $\Delta\tau_{ij}$  可表示为

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

$\Delta\tau_{ij}^k$  表示蚂蚁  $k$  在本次循环中在城市  $i$  和  $j$  之间留下的信息量,它的计算公式根据计算模型而定,例如在最常用的 ant circle system 模型中,

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{蚂蚁 } k \text{ 在本次循环中经过城市 } i \text{ 和 } j \text{ 之间} \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

其中,  $Q$  为常数,  $L_k$  为蚂蚁  $k$  在本次循环中所走路径的长度.在经过若干次循环以后,可以根据适当的停止条件来结束计算.

由上述可知,蚁群算法优化过程的本质在于:(1) 选择机制.信息量越大的路径,被选择的概率越大;(2) 更新机制.路径上面的信息量会随蚂蚁的经过而增长,同时也随着时间的推移逐渐减小;(3) 协调机制.蚂蚁之间实际上是通过信息量来互相通信、协同工作的.这样的机制使得蚁群算法具有很强的发现较好解的能力.但是,蚁群算法也有一些缺陷.例如,由于蚁群中多个个体的运动是随机的,当群体规模较大时,要找出一条较好的路径需要较长的搜索时间.为此,吴庆洪等人提出了具有变异特征的蚁群算法<sup>[7]</sup>,有机地结合了 2-opt 方法,提高了算法的性能.M.Dorigo 等人在基本的蚁群算法的基础上提出称为 Ant-Q System 的更一般的蚁群算法<sup>[8,9]</sup>,每次让信息量最大的路径以较大的概率被选中,以充分利用学习机制,强化最优信息的反馈.为了克服在 Ant-Q 中可能出现的停滞现象,T.Stutzle 等人提出了 MAX-MIN 蚁群系统<sup>[10]</sup>,允许各个路径上的信息量在一个限定的范围内变化.L.M.Gambardella 等人提出了一种混合型蚁群算法 HAS<sup>[11]</sup>,在每次循环中蚂蚁建立各自的解后,再以各自的解为起点用某种局部搜索算法求局部最优解,作为相应蚂蚁的解,这样可以迅速提高解的质量.H.M.Bottee 等人对参数  $m, \alpha, \beta, \rho$  的选择进行了深入的研究,用遗传算法求得这些参数最优组合<sup>[12]</sup>.W.J.Gutjahr 等人提出一种以图为基础构建的蚁群系统框架来解决组合优化问题,在一定的条件下,每次迭代所得到的解能以近似于 1 的概率向最优解收敛<sup>[13]</sup>.我们曾将蚁群算法改进,使其能够直接求解线性和非线性规划等连续空间的优化问题<sup>[14]</sup>.吴斌、史忠植提出了基于蚁群算法的分段求解算法<sup>[15]</sup>,提高了蚂蚁搜索的速度,为蚁群算法的并行化奠定了基础.

由于在蚁群算法构造解的过程中,选择策略一般是随机的,进化速度较慢.张纪会等人<sup>[16]</sup>提出了自适应的蚁群算法,采用确定性选择和随机选择相结合的策略,在搜索的过程中动态地调整选择的概率,实现了选择概率的自适应,提高了算法的速度和性能.

蚁群算法主要利用正反馈原理强化较优解,当进化到一定代数后就容易因为较优解的信息量不断强化使得蚂蚁大量聚集于较少的几条路径上,出现早熟、停滞现象,使得到的最优解是局部最优的.本文针对蚁群算法加速收敛和早熟、停滞现象的矛盾,提出一种基于分布均匀度的自适应蚁群算法.该算法根据优化过程中解的分布均匀度,动态地调整信息量更新策略和选择路径概率,这样,可以在加速收敛和防止早熟、停滞现象之间取得很好的平衡.我们以数种对称和不对称 TSP 问题为例进行了计算,结果表明,该方法比一般蚁群算法具有更好的收敛速度和稳定性.

## 1 聚度和信息权重

在蚁群算法中,加速收敛和防止早熟、停滞现象是一对矛盾.为了加速收敛,Ant-Q System 让信息量最大的路径对每次路径的选择和信息量的更新起主要的作用.由于强化了最优信息的反馈,会导致早熟、停滞现象.而 MAX-MIN 蚁群系统将各个路径上的信息量的更新限定在固定的范围内,虽然在一定程度上避免了早熟、停滞现象,但在解的分布较分散时收敛速度较慢.以上方法的共同缺点在于,它们都按一种固定不变的模式去更新信息量和确定每次路径的选择概率.本文提出的基于分布均匀度的自适应蚁群算法根据蚂蚁选路的分布均匀情况,动态地调整信息量更新策略和确定每次选择路径的概率,这样,可以在加速收敛和防止早熟、停滞现象之间取得很好的平衡.为此,我们基于常规数学模型引入“聚度”的概念来衡量解的均匀程度,从而决定每次选择路径的概率以及信息量更新的策略.当所有路径上蚂蚁的分布相对比较分散时,聚度就较小,此时难以强化最优信息,可能导致搜索速度较慢.因此,要强化正反馈信息,应该让较少的几个较优的路径有较大的概率被选取;在信息量更新时,应该仅让较少的几个较优的路径上的信息量得到较大程度的增强.反之,在所有路径上蚂蚁的分布相对比较集中时,聚度就较大,此时易引起早熟、停滞现象.因此,要使解趋于多样化,应该使较多的路径有可能被选取,在信息量更新时,应该让较多路径上的信息量得以增强.通过这样动态的自适应调节,可以在有效地改善蚂蚁搜索速度的同时避免局部优化.为此,我们在基于分布均匀度的自适应蚁群算法的迭代过程中,根据各条可能的路径上的聚度来确定下一次迭代中可供蚂蚁选择的路径的信息权重,并以此来确定它们被选中的概率.此外,算法也根据该分布范围内的各条路径所构成解的优劣及信息权重,对其信息量进行有差别的动态更新,从而实现信息量分布的自适应调节.我们仍然以求解 TSP 问题为例说明我们的方法.

**定义 1.** 设从城市  $i$  共有  $r$  条路径到达另外  $r$  个城市  $i_1, i_2, \dots, i_r$ , 另设上一次迭代中,经过这  $r$  条路径上的蚂蚁数分别为  $a_1, a_2, \dots, a_r$ , 如图 1 所示.记

$$sta(i) = \sqrt{\sum_{l=1}^r \left( \frac{m}{r} - a_l \right)^2} \quad (5)$$

为城市  $i$  的聚度.若在上一次迭代中,  $m$  只蚂蚁遍历时经过以城市  $i$  为起点的  $r$  条路径中的  $s$  条, 设经过它们的蚂蚁个数分别为  $a_1, a_2, \dots, a_s$ , 则这些值均不为 0, 而其余路径上的蚂蚁个数均为 0. 城市  $i$  的聚度随  $s$  的减小而增大, 在极端情况下:

(1) 当  $m$  只蚂蚁在以城市  $i$  为起始点的  $r$  条路径上均匀分布时, 每条路上有  $m/r$  只蚂蚁, 城市  $i$  的聚度为

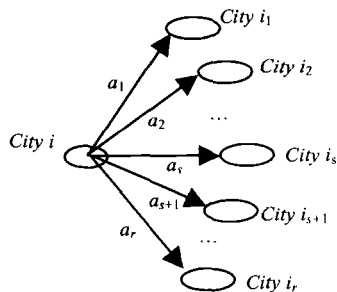


Fig.1 Distribution of the ants on each trail from city  $i$

图 1 源自城市  $i$  的各条路径上蚂蚁的分布状况

$$sta(i) = \sqrt{\sum_{j=1}^r \left( \frac{m}{r} - \frac{m}{r} \right)^2} = 0.$$

(2) 当  $m$  只蚂蚁集中在以城市  $i$  为起始点的  $r$  条路径中的某一条上时,城市  $i$  的聚度为

$$sta(i) = \sqrt{\sum_{j=1}^{r-1} \left( \frac{m}{r} \right)^2 + \left( m - \frac{m}{r} \right)^2} = m \sqrt{1 - \frac{1}{r}}.$$

我们记该值为  $\max sta(i)$ ,即最大可能聚度值.

若城市的聚度较大,说明蚂蚁上一次从这个城市到达另外城市的路径集中于少数几条上,也说明信息量集中于这少数几条路径,在以后的搜索最优解的过程中,蚂蚁在该城市选择下一城市的可选路径就会相对集中,因过度强化正反馈信息引起早熟、停滞现象的可能性就越大;相反,当城市的聚度越小时,这个城市到达另外城市的信息量分布相对比较分散,不利于强化最优信息,导致收敛速度较慢.为了在这两方面达到有效的平衡,以在改善蚂蚁搜索速度的同时避免局部优化,我们考虑根据城市  $i$  的聚度  $sta(i)$  来确定蚂蚁在该城市时下一步可供选择的的路径的条数  $w$ .我们取

$$w = \left\lfloor \frac{sta(i)}{\max sta} \cdot (r-1) + 0.5 \right\rfloor + 1, \quad (6)$$

在选择过程中,蚂蚁仅考虑信息量最高的  $w$  条路径.显然,当城市的聚度越大时, $w$  越大,蚂蚁下一步的分布范围越来越广;反之,聚度越小,蚂蚁搜索时分布范围越小.在极端情况下,当聚度最大时,即  $sta(i) = \max sta$  时,  $w=r$ ,  $m$  只蚂蚁可选择所有可选路径;当聚度最小时,即  $sta(i) = 0$  时,  $w=1$ ,蚂蚁只能选择一条最优路径.

虽然约定了蚂蚁下一次搜索的分布范围,但是一般蚁群算法中蚂蚁的选择策略主要依赖于其所在的当前城市  $i$  选择下一城市的期望程度  $\eta_{ij}$  (即可访问度,本文取一般值  $1/d_{ij}$ ) 和以  $i$  为起点的各条路径上的信息量强度  $\tau_{ij}(t)$ ,这会在一定程度上误导大量的蚂蚁聚集于当前信息量较大的几条局部距离较短的路径上.因此,我们引入“信息权重”来限制信息量和期望程度对蚂蚁选择概率的影响程度,从而调整各个路径被选中的概率.总体原则是要使信息量大的和当前局部距离小的路径被选择的概率大些;若信息量分布比较集中,则应当使各条路径上的信息量及期望程度对选择概率的影响的差别小些,各条路径被选中的概率相对均匀些;反之,应当使信息量和期望程度对选择概率的影响的差别大些,蚂蚁选择的路径相对更集中一些.

**定义 2.** 将以城市  $i$  为起点的  $r$  条路径按其信息量由高到低排序,序号依次存于数组  $rank$  中,即数组元素  $rank[j]$  的值为路径  $(i,j)$  的序号.设由式(6)计算得到的下一步可供选择的路径条数为  $w$ ,取  $q=w/r$ ,记

$$\xi_{ij} = \begin{cases} q^{rank[j]-1}, & \text{if } rank[j] \leq w \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

则  $\xi_{ij}$  为路径  $(i,j)$  的信息权重.应用信息权重  $\xi_{ij}$ ,蚂蚁由城市  $i$  按下式的概率选择城市  $j$

$$p_{ij}^k(t) = \begin{cases} \frac{\xi_{ij} \tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{u \in allowed_k} \xi_{iu} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t)}, & j \in allowed_k \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

对照式(1)可见,式(8)对各路径上的信息量及期望程度乘上了信息权重  $\xi_{ij}$ .路径  $(i,j)$  的信息权重  $\xi_{ij}$  反映了蚂蚁从城市  $i$  选择下一城市  $j$  时,路径  $(i,j)$  上的信息量  $\tau_{ij}(t)$  以及可访问度  $\eta_{ij}$  对蚂蚁选择概率的影响程度.对于信息量较大的路径,其信息权重较大,蚂蚁选取该路径的概率也较大.

由式(7)可知,各路径上的信息权重是公比为  $q(q \leq 1)$  的等比数列,而当上一次迭代中蚂蚁遍历经过城市  $i$  为起点的路径较集中时,城市  $i$  的聚度就较大,使得分布范围  $w$  以及  $q$  值较大,因而各条路径的信息权重  $\xi_{ij}$  差距就比较小,这样,在下次迭代中蚂蚁遍历经过城市  $i$  为起点的路径被选择的概率分布就比较均匀.特别地,当  $w=r$  时,  $q=1$ ,所有路径的信息权重的值均为 1,达到完全平均,式(8)即退化为式(1).相反地,当上一次迭代中蚂蚁遍历经过城市  $i$  为起点的路径较分散时,城市  $i$  的聚度就较小,使得分布范围  $w$  以及  $q$  值较小,因而各条路径的信息权重  $\xi_{ij}$  差距就比较大,城市  $i$  为起点的路径被选择的概率分布就比较集中.特别地,当  $w=1$  时,仅有信息量最大的一条路径上的信息权重为  $1/r$ ,其余所有路径的信息权重的值均为 0.此时,蚂蚁仅选择信息量最大的那条路径.该算

法框架描述如下:

**算法 1. Adaptive ant algorithm.**

(1) 初始化

随机产生  $m$  个初始解,设其中经过路径 $(i,j)$ 的初始解有  $s$  个,它们的总长度分别为  $L^1, L^2, \dots, L^s$ ,则路径 $(i,j)$ 上的初始信息量为

$$\tau_{ij}(0) = \sum_{k=1}^s Q/L^k,$$

其中  $Q$  为常数.

(2) 迭代过程

while not 结束条件 do

(2.1) for  $i=1$  to  $n$  do (对  $n$  个城市循环)

根据式(5)和式(6),计算城市  $i$  的聚度及本次迭代蚂蚁的分布范围  $w$ ;

for  $k=1$  to  $m$  do (对  $m$  个蚂蚁循环)

(2.1.1) 根据概率选择算法 2 选择下一城市  $j$ ;

(2.1.2) 根据式(10)局部更新路径 $(i,j)$ 上的信息量;

end for  $k$

$m$  只蚂蚁中若有当前已遍历的路径长度和已经超过上一次迭代得到的最优路径长度的,则终止该蚂蚁本次迭代的遍历;

end for  $i$

(2.2) 对所有城市的各条路径整体更新其信息量;

end while

**算法 2. Algorithm of selection.**

{

比较  $w$  与  $allowed_i$  中的城市数目  $num$ :

若  $num < w$ ,

$$j = \begin{cases} \arg \max \tau_{ij}(t) (j \in allowed_i), & q \leq q_0 \\ \text{按式(1)选取城市 } j, & \text{otherwise} \end{cases} \quad (9)$$

因为  $num < w$ ,此时可能是:(a)  $w$  比较大,即聚度较大,从而要求本次迭代中蚂蚁对路径的选择概率分布均匀;或者是:(b)  $num$  比较小,蚂蚁本次的搜索已接近尾声,所以我们参考选择概率自适应的方法<sup>[16]</sup>.设定一个阈值  $0.4r$  来限定  $q_0$  的取值,从而确定最优信息被强化的程度.当  $w < 0.4r$  时,即(b)的因素多些,则取  $q_0=0.8$ ,以较大的概率选择最优路径,使蚂蚁趋向于集中到最优路径上;当  $w \geq 0.4r$  时,即(a)的因素多些,则取  $q_0=0.2$ ,使蚂蚁对路径的选择比较均匀.

若  $num \geq w$ ,

将  $allowed_i$  集合中各条路径上的信息量由大到小排序,选择前  $w$  条路径并将其序号记入数组  $rank$  中.计算与城市  $i$  相连的各路径的信息权重,按式(8)选择城市  $j$ .

}

## 2 自适应的信息量更新策略

本文引言中所提及的一些典型算法在更新信息量时,要么对所有的蚂蚁所经过的路径增加其信息量<sup>[1,2]</sup>,要么只增加最优适应度的路径上信息量,其余信息量被削减<sup>[10]</sup>;或者就是基于固定等级的算法,让适应度相对较好的若干条路径根据其解的优劣程度决定信息量的增加程度<sup>[17]</sup>,这些做法都采用了固定的信息量增减的比例,而忽视了解的分布情况.在我们的算法中,总结了大量学者在研究中的实验经验,根据信息量的均匀度自适应地进行信息量的更新,以动态地调整各路径上的信息量的分布,使其不至于过分集中或者分散,以加速收敛同时避免

早熟.算法 1 的第 2.1.2 行的信息量局部更新可根据以下策略:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{ij}(t) - 10/d_{ij}, & \text{若本次迭代中已有 } m/3 \text{ 只蚂蚁选择同一路径 } (i, j) \\ & \text{或 } m/5 \text{ 只蚂蚁选择该路径后终止本次迭代的遍历.} \\ \tau_{ij}(t) + 1/d_{ij}, & \text{otherwise} \end{cases} \quad (10)$$

由于蚂蚁常常选择信息量较大的路径,当多只蚂蚁选中同一路径后,信息量增加的幅度太大就容易使多只蚂蚁集中到该路径,所以我们取  $1/d_{ij}$  为增加的信息量;若选择该路径的蚂蚁达到一定数量或多数蚂蚁选择该路径后因当前距离超过上一次的最优路径长度终止遍历(这里分别设定为  $m/3$  和  $m/5$ )信息量则取  $-10/d_{ij}$ ,大幅度削减其信息量使其趋于各条路径信息量的平均值,从而使蚂蚁选择其他路径的可能性增加,让搜索得到的解趋于多样化.

算法 1 的第 2.2 行按下式进行信息量的整体更新:

$$\tau_{ij}(i+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{l=1}^m \psi_l \cdot \Delta \tau_{ij}^l(t), \quad (11)$$

其中

$$\Delta \tau_{ij}^l(t) = \begin{cases} Q/L^l(t), & \text{第 } l \text{ 只蚂蚁经过路径 } (i, j) \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$L^l(t)$  为本次迭代中第  $l$  只蚂蚁遍历的路径全长,  $\psi_l$  为第  $l$  只蚂蚁所对应的解对该路径上信息量更新的影响程度.  $\psi_l$  的计算方法如下:设经过路径  $(i, j)$  的蚂蚁总数为  $k$ ,对它们在本次迭代中遍历的路径全长由小到大进行排序,所得序号存放于数组  $rank_l$  中,即  $rank_l[l]$  表示第  $l$  只蚂蚁对应的序号,则

$$\psi_l = (1 - \xi_{ij}) \times k - rank_l[l] + 1, \quad (13)$$

其中  $\xi_{ij}$  为路径  $(i, j)$  的信息权重.我们从以下 3 个方面来说明由上式决定的  $\psi_l$  的合理性.

① 从由城市  $i$  出发的各条路径来看,当路径  $(i, j)$  的信息权重较高时,说明该路径上的信息量相对较高,此时  $(1 - \xi_{ij}) \times k$  的值就比较小,对于较少的优秀解,由于它们的  $rank_l[l]$  值比较小,  $\psi_l$  为正值,因而在该路径上将增加信息量;而对较多的其他解,由于它们的  $rank_l[l]$  值比较大,因而它们的  $\psi_l$  为负值,在该路径上将减少信息量,这样会使路径  $(i, j)$  上的总体信息量减少.反之,当路径  $(i, j)$  的信息权重比较低时,会使该路径上的总体信息量增加.这样可以调节各路径上的信息量,防止过于集中.② 从各个蚂蚁本次所经过的路径来看,若第  $l$  只蚂蚁遍历路径总长度比较小,即  $rank_l[l]$  值比较小,则  $\psi_l$  就为正值,因而会增加路径上的信息量强度,而且  $rank_l[l]$  越小,即长度越短,路径上信息量增加的程度越大,有效地强化了短路径上的信息.反之,若第  $l$  只蚂蚁遍历路径总长度比较大,则将减少相应路径上的信息量强度,而且  $rank_l[l]$  越大,即长度越长,路径上信息量减少的程度越大.这样可在防止信息量过分集中的同时保持较好路径上的信息量.③ 从城市  $i$  本身的聚度来看,当城市  $i$  的聚度比较高时,与它相连的各条路径的信息权重  $\xi_{ij}$  差距就比较小,  $\psi_l$  的差异也就比较小,各个解对信息量的影响就比较均匀,以避免较优解对信息量的更新有较大的影响.反之,当城市  $i$  的聚度比较低时,  $\psi_l$  的差异就比较大,较优解会对信息量的更新有较大的影响,以使较优解的路径上集中较高的信息量.因此,这种自适应的信息量更新机制可以动态调节信息量,在蚂蚁的搜索速度和解的多样性之间取得较好的平衡.

### 3 实验结果及分析

我们从通用 TSPLIB 中选用数种对称和不对称 TSP 问题<sup>[18]</sup>,用上述算法和传统蚁群算法在 PC 机上用 C 语言编程进行了测试.根据多次实验所得结论<sup>[13]</sup>,我们取  $\rho=0.4$ ,  $Q=5$ .因为路径上  $t$  时刻的信息量  $\tau_{ij}(t)$  及期望程度  $\eta_{ij}(t)$  均由蚂蚁遍历所得的路径长短决定,因而在实验中取  $\alpha=1$  和  $\beta=0$ ,将这两个参数归结为由信息量影响蚂蚁的路径选择策略.另外,使蚂蚁的个数等于城市数目,运行 25 次,每次运行迭代 1 500 次,所测试的结果见表 1~表 3.在表中,“传统算法”是指 MMAS 算法,“平均时间”是指各次运行中找到最好解的平均时间的平均值,“允许时间”是指每次运行所允许执行的最长时间.各个问题名称中的数字即所包含的城市数(除了 kro124p 为 100 个城市).

**Table 1** The experimental results on symmetric TSP**表 1** 对称 TSP 问题的实验结果

| Instance | Algorithm   | Best solution | Average solution | The time required for the best solution | Average time (s) | The longest time allowed (s) |
|----------|-------------|---------------|------------------|---|------------------|------------------------------|
| d198     | Traditional | 15 780        | 15 780.7         | 18                                      | 108.6            | 300                          |
|          | Our         | 15 780        | 15 780.4         | 22                                      | 103.1            |                              |
| lin318   | Traditional | 42 029        | 42 029.0         | 25                                      | 193.5            | 500                          |
|          | Our         | 42 029        | 42 029.0         | 25                                      | 137.2            |                              |
| pcb442   | Traditional | 50 778        | 50 857.2         | 16                                      | 542.0            | 800                          |
|          | Our         | 50 778        | 50 801.3         | 22                                      | 395.6            |                              |
| att532   | Traditional | 27 686        | 22 702.1         | 22                                      | 578.2            | 1 000                        |
|          | Our         | 27 686        | 27 699.4         | 24                                      | 476.3            |                              |

**Table 2** The experimental results on asymmetric TSP**表 2** 不对称 TSP 问题的实验结果

| Instance | Algorithm   | Best solution | Average solution | The time required for the best solution | Average time (s) | The longest time allowed (s) |
|----------|-------------|---------------|------------------|---|------------------|------------------------------|
| ry48p    | Traditional | 14 422        | 14 423.9         | 18                                      | 61.4             | 250                          |
|          | Our         | 14 422        | 14 422.0         | 25                                      | 35.9             |                              |
| ft70     | Traditional | 38 673        | 38 675.2         | 21                                      | 79.2             | 300                          |
|          | Our         | 38 673        | 38 673.4         | 23                                      | 55.7             |                              |
| kro124p  | Traditional | 36 230        | 36 230.0         | 25                                      | 101.6            | 300                          |
|          | Our         | 36 230        | 36 230.0         | 25                                      | 69.2             |                              |
| ftv170   | Traditional | 2 755         | 2 755.8          | 22                                      | 118.4            | 500                          |
|          | Our         | 2 755         | 2 755.0          | 25                                      | 85.1             |                              |

**Table 3** Iteration numbers required for the best solution**表 3** 达到最优解所需的平均迭代次数

| Instance | Traditional algorithm | Our algorithm |
|----------|-----------------------|---------------|
| d198     | 1 633                 | 1 009         |
| lin318   | 2 718                 | 1 851         |
| ry48p    | 216                   | 173           |
| ft70     | 745                   | 542           |

由上述各表可见,本文算法具有较高的搜索较优解的能力,经较少的次数可以找到最优解,节约了计算时间.例如,由表 3 可以看出,使用传统蚁群算法,对问题 d198 平均需要 1 633 次迭代才能达到最优解,而本文的算法平均仅需 1 009 次迭代便可达到.通过以上实验结果还可以看出,使用本文提出的改进的蚁群算法命中最优的次数也相对较高,从而证明了本文算法具有较高的稳定性.

另外,我们还测试了本文算法和传统算法在同一机器上、各项参数设置同等的情况下对 d198、ry48p 进行计算时,各次迭代最佳结果的进化情况,结果如图 2 和图 3 所示.由此可以看出,使用本算法不但有较强的搜索能力,接近最优解的速度比传统算法要快,而且在达到最优解后可稳定在最优解的附近,说明本文算法对对称和不对称 TSP 问题都具有较好的收敛性.

我们让蚂蚁根据上一次的迭代情况自适应地决定选择概率和信息量的更新,与传统算法的搜索机制相比,特别是与 MAX-MIN 算法比较而言,不再因仅更新最优路径的信息量而过度强化最优信息;也不将信息量限定于特定的范围内,这样就使蚂蚁在搜索的过程中具有了“随机应变”的本领,可以使得大部分蚂蚁的搜索向适应度较好的方向进化,使得在解的各个方向具有较强的“爬坡”能力,因而可较快地构成较优解.又由于在信息量更新的同时也考虑到信息量的分布均匀性,从而使得解具有较好的多样性、全局性,避免了早熟现象.因此,本文所提算法在收敛速度和防止早熟之间取得了动态的平衡,具有很强的发现最优解的能力、更快的进化速度,对于求解规模较大的优化问题十分有利.

#### 4 结 论

本文提出了一种改进的基于分布均匀度的自适应蚁群算法,该方法通过对各个路径上信息量的分布均匀度的衡量,根据蚂蚁在搜索过程中所得到解的优劣和解的分布状况有效地调节蚂蚁的选择概率和各条路径上的信息量更新策略,可以有效地缓解基本蚁群算法容易早熟、停滞和收敛速度较慢的矛盾,使得蚁群算法在具

有更好的收敛性、稳定性和更快的速度的同时使解更具有多样性、全局性.我们以各种规模的 TSP 问题为例进行的实验结果表明,该方法尤其适用于规模比较大的优化问题.

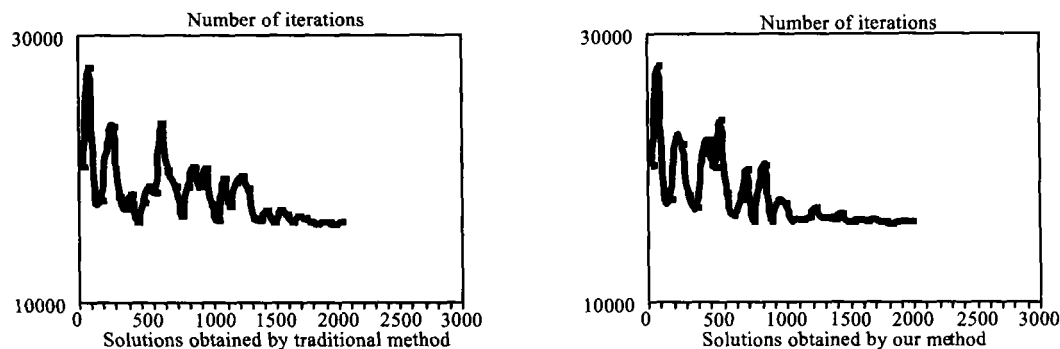


Fig.2 The evolutionary process of the best solution for the symmetric TSP of d198

图2 对称 TSP 问题 d198 的最优解进化过程

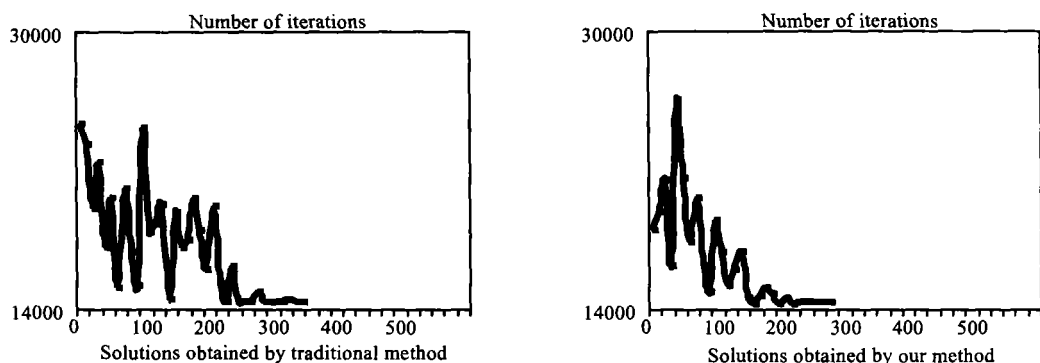


Fig.3 The evolutionary process of the best solution for the asymmetric TSP of ry48p

图3 不对称 TSP 问题 ry48p 的最优解进化过程

**致谢** 在此,我们向对本文的工作给予支持和建议的同行,尤其是扬州大学计算机科学与工程系并行与分布式计算研究室各位老师和同学表示感谢.

#### References:

- [1] Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996,26(1):29~41.
- [2] Dorigo M, Gambardella LM. Ant colonies for the traveling salesman problem. BioSystems, 1997,43(2):73~81.
- [3] Lee ZJ, Lee CY, Su SF. An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem. Applied Soft Computing, 2002,2(1):39~47.
- [4] Talbi E-G, Roux O, Fonlupt C, Robillard D. Parallel ant colonies for the quadratic assignment problem. Future Generation Computer Systems, 2001,17(4):441~449.
- [5] Maniezzo V, Carbonaro A. An ANTS heuristic for the frequency assignment problem. Future Generation Computer Systems, 2000,16(8):927~935.
- [6] Chang CS, Tian L, Wen FS. A new approach to fault section in power systems using ant system. Electric Power Systems Research, 1999,49(1):63~70.
- [7] Wu QH, Zhang JH, Xu XH. An ant colony algorithm with mutation features. Journal of Computer Research & Development, 1999,36(10):1240~1245 (in Chinese with English abstract).



- [8] Machado L, Schirru R. The ant-Q algorithm applied to the nuclear reload problem. *Annals of Nuclear Energy*, 2002,29(12):1455~1470.
- [9] Dorigo M, Gambardella LM. A study of some properties of ant-Q. In: Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-S, eds. *Proceedings of the PPSN 44th International Conference on Parallel Problem Solving from Nature*. Berlin: Springer-Verlag, 1996. 656~665.
- [10] Stützle T, Hoos HH. MAX-MIN ant system. *Future Generation Computer Systems*, 2000,16(8):889~914.
- [11] Gambardella LM, Dorigo M. An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 2000,12(3):237~255.
- [12] Botee HM, Bonabeau E. Evolving ant colony optimization. *Complex Systems*, 1998,1(2):149~159.
- [13] Gutjahr WJ. A graph-based ant system and its convergence. *Future Generation Computer Systems*, 2000,16(8):873~888.
- [14] Chen L, Shen J, Qin L. A method for solving optimization problem in continuous space by using ant colony algorithm. *Journal of Software*, 2002,13(12):2317~2322 (in Chinese with English abstract).
- [15] Wu B, Shi ZZ. An ant colony algorithm based partition algorithm for TSP. *Chinese Journal of Computers*, 2001,24(12):1328~1333 (in Chinese with English abstract).
- [16] Zhang JH, Gao QS, Xu XH. A self-adaptive ant colony algorithm. *Control Theory And Applications*, 2000,17(1):1~3 (in Chinese with English abstract).
- [17] Bullnheimer B, Hartl RF, Strauss C. A new rank based version of the ant system: A computational study. *Central European Journal for Operations Research and Economics*, 1999,7(1):25~38.
- [18] [www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95](http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95)

#### 附中文参考文献:

- [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):1240~1245.
- [14] 陈峻,沈洁,秦玲.蚁群算法求解连续型优化问题的一种新方法. *软件学报*, 2002,13(12):2317~2322.
- [15] 吴斌,史忠植.一种基于蚁群算法的 TSP 问题分段求解算法. *计算机学报*, 2001,24(12):1328~1333.
- [16] 张纪会,高齐圣,徐心和.自适应蚁群算法. *控制理论与应用*, 2000,17(1):1~3.