

基于信息素扩散模型解耦控制策略的蚁群算法

冀俊忠,刘椿年,黄 振

(北京工业大学 计算机学院,北京 100022)

摘 要:蚁群优化是一种元启发式的随机搜索技术.信息素是蚁群进行交流并实现群集智能的媒介,所以信息素的更新策略一直是蚁群算法中的一个研究热点.针对信息素扩散的耦合特征,提出一种基于信息素扩散模型解耦控制策略的蚁群算法.对信息素扩散模型进行改善,建立以蚂蚁经过的路径(直线段)为信源的信息素扩散模型,通过分析信息素扩散浓度场的耦合性,引入去耦控制策略来修正信息素的更新公式,大量 TSP (traveling salesman problem) 问题的实验表明:该算法不仅能获得更好的解,而且能加快算法的收敛速度.

关键词:蚁群算法;扩散模型;耦合性;解耦控制策略

中图分类号: TP18 **文献标识码:** A **文章编号:** 1673-4785(2007)04-0001-08

An ant colony optimization algorithm based on a decoupling control strategy of pheromone diffusion model

J I J un-zhong, L I U Chun-nian, H U A N G Zhen

(College of Computer Science and Technology, Beijing University of Technology, Beijing 100022, China)

Abstract: Ant colony optimization (ACO) is a meta-heuristic search technique. Pheromones are an important media ants use to communicate with each other and implement swarm intelligence. Thus research on pheromone updating strategies is a hotspot in ACO. A new decoupling control strategy model of pheromone diffusion is proposed based on the coupling characteristic of pheromone diffusion. First, the algorithm sets up a new pheromone diffusion model with the path that the ant travels as the pheromone source. Then, according to the coupling degree of the concentration field of pheromone diffusion, a decoupling control strategy is employed to revise the pheromone updating formula. Experimental results for many TSP problems demonstrate that the proposed algorithm can not only generate better solutions but also accelerate the speed of convergence.

Key words: ant colony optimization; diffusion model; coupling characteristic; decoupling control strategy

仿生学的研究已经为人类自然科学的发展和进步提供了许多有益的启迪.其中,群集智能 (swarm intelligence) 是人们对生物群落行为的观察和对生物社会性的研究中得到的一种演化计算技术,近年来受到学者们的广泛关注,并为解决复杂组合优化问题提供了一个理论框架.蚁群优化算法 (ant colony optimization, ACO) 是 Dorigo 等人根据蚂蚁群体在觅食过程中所体现出的智能行为提出的一种

元启发式的随机搜索技术^[1-4],目前已成为群集智能中解决组合优化问题最有效的算法之一,并已在生产过程、车辆管理、路由寻址、布局规划、资源调配和数据挖掘等领域中获得了成功的应用. ACO 进行优化的基本特性是:将可行解的先验结构信息与目前已得到的较好解的后验信息相融合,用以引导搜索过程,加速最优解的获取.在这个寻优过程中,信息素是蚁群进行相互交流、完成信息融合并实现群集智能的重要媒介.所以,近些年国内外的学者对蚁群算法的研究,都是围绕信息素的产生和更新策略进行的.例如, Gambardella 等^[5]提出的 ACS (ant colony system) 蚁群系统, Stutzle 等^[6]提出的

收稿日期:2006-12-28.

基金项目:国家自然科学基金资助项目(60496322);北京市教育委员会科技发展资助项目(KM200610005020).

MMAS(max-min ant system) 蚁群系统,朱庆保等^[7]提出的基于变异和动态信息素更新的 NDMA-CO 算法(ACO algorithm based on the nearest neighbor node choosing, dynamic pheromone updating and mutation),黄国锐等^[8]提出的基于信息素扩散模型的 PDACO 算法(ACO algorithm based on pheromone diffusion)等都是通过对信息素生成和更新策略的改进及优化来提高蚁群算法的寻优能力,并改善解的全局收敛性.虽然许多算法都具有发现较优解的能力,但迭代次数多、收敛速度慢仍是制约大多数算法在实际组合优化问题中应用的瓶颈.针对这一问题,本文结合信息素扩散的耦合特征,提出一种基于信息素扩散模型解耦控制策略的蚁群算法.首先,对信息素扩散模型进行改善,建立以蚂蚁经过的路径(直线段)为信源的信息素扩散模型,细化了信息素扩散机制;然后,通过分析信息素扩散浓度场的耦合性,引入去耦控制策略来修正信息素的更新公式,强化了蚂蚁间的协作和交流.大量 TSP 问题的实验结果表明:该算法不仅能获得更好的解,而且能加快算法的收敛速度.

1 蚁群算法及信息素扩散模型

1.1 ACS 蚁群算法

蚁群优化算法是一类基于模型的搜索算法^[9].该类算法实现的基本思想是通过模拟自然界中真实蚁群的觅食行为而获得所求问题的解,但不同的求解模型使用不同的状态转移方法和信息素的更新策略,从而形成了不同的蚁群算法.ACS 算法是其中最成熟且应用最广的蚁群算法之一.下面以 n 个城市的旅行商问题为例,介绍该算法的求解过程.设 m 是蚁群中蚂蚁的数量, d_{ij} 表示城市 i 与城市 j 之间的欧氏距离, a_{ij} 表示城市 i 与城市 j 之间的路径, $\tau_{ij}(t)$ 表示 t 时刻在 a_{ij} 上残留的信息素浓度,并设初始时刻各条路径上的信息素浓度均为 C (常数).蚂蚁 $k(1 \leq k \leq m)$ 在自己周游的过程中,将根据目前可行候选路径的结构信息(先验知识)与其上残留的信息素浓度(后验信息)来选择自己前进的方向.ACS 算法中,每只蚂蚁从城市 i 走向城市 j 的过程分 2 步进行:首先通过随机采样和比较,然后再确定状态变迁的规则,其实现的公式为

$$j = \begin{cases} \operatorname{argmax}_{u \in U_k} \{ [\tau_{iu}(t)] \cdot [I_{iu}] \}, & q < q_0 \text{ (利用)}, \\ J, & \text{otherwise (开发)}. \end{cases} \quad (1)$$

式中: U_k 表示蚂蚁 k 在本次周游中在当前位置允许

选择的城市列表, $U_k = \{ N - \text{禁忌列表 Tabu}_k \}$; τ_{ij} 为路径 a_{ij} 的能见度,可理解为该路径的启发信息(先验),一般取 $\tau_{ij} = 1/d_{ij}$; q_0 是初始设定的参数, q 是一个随机采样的数,且 $q_0, q \in [0, 1]$; J 是由式(2)确定的随机变量.若 $q < q_0$,将按式(1)选择下一城市,否则,按式(2)的转移概率确定下一城市:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)] \cdot [I_{ij}]}{\sum_{u \in U_k} [\tau_{iu}(t)] \cdot [I_{iu}]}, & j \in U_k, \\ 0, & \text{其他}. \end{cases} \quad (2)$$

式中: τ_{ij} 、 I_{ij} 分别表示路径 a_{ij} 上的信息素和先验结构信息对蚂蚁选择转移方向时的影响权重,在原始的 ACS 算法中 $\tau_{ij} = 1$.

蚁群在觅食过程中,一方面通过蚂蚁的行走会在路径上留下新的信息素,另一方面随着时间的推移已有的信息素会逐渐挥发,所以各条路径上的信息素会根据蚁群的行动和时间变化不断更新.信息素的更新体现了可行解空间的后验信息积累的变化.ACS 算法在每次迭代中,要进行局部和全局 2 次信息素的更新.

首先在构造解时,每只蚂蚁对其走过的路径用式(3)来进行局部信息素的更新.

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_{ij}^k. \quad (3)$$

式中:参数 ρ 表示信息素的挥发程度,且 $0 < \rho < 1$,表示 τ_{ij}^k 在本次周游中蚂蚁 k 留在路径 a_{ij} 上的信息素,如果采用 ant quantity system 模型(Q 为常数),有

$$\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}}, & \text{当第 } k \text{ 只蚂蚁在本次周游中} \\ & (t, t+1 \text{ 时段) 经过路径 } a_{ij} \text{ 时,} \\ 0, & \text{其他}. \end{cases} \quad (4)$$

其次,当每次循环所有的蚂蚁都完成了一次周游后,ACS 算法对最优解的每条路径上的信息素按式(4)进行全局更新:

$$\tau_{ij}^{\text{new}} = (1 - \alpha) \cdot \tau_{ij}^{\text{old}} + \alpha \cdot \tau_{ij}, \quad (5)$$

$$\tau_{ij} = \begin{cases} Q / (L_{\text{best}}), & \text{当全局最优解包含路径 } a_{ij} \text{ 时,} \\ 0, & \text{其他}. \end{cases}$$

同样, α 是挥发系数,且 $0 < \alpha < 1$; Q 为常数,且在原始的 ACS 算法中 Q 取 1; L_{best} 是到目前为止蚁群寻优得到的全局最优解的路线长度.ACS 蚁群算法一方面通过强化状态变迁规则,增强了解的多样性,可以避免早熟现象;另一方面,通过全局信息素的更新,强化正反馈的过程,加速收敛过程.所以在求解 TSP 问题时,ACS 算法具有良好的求解能力.

1.2 基于城市信源的信息素扩散模型

文献[8]在蚁群优化中提出了信息素扩散模型,其基本思想是:在蚂蚁进行路径选择时,适当考虑相近路径上信息素的相互作用.即一只蚂蚁在某条路径上留下的信息素,一方面会直接影响连接该路径的 2 个城市上的其他蚂蚁选择下一个城市的行为;另一方面,它会以这 2 个城市为中心向外扩散,影响该城市附近的其他城市上蚂蚁的选路行为.

该信息素扩散模型用于模拟以城市信源为中心,近似服从正态分布的扩散浓度场,模型较客观地反映了信息素浓度与到信源之间距离的反比关系,如城市 C 与信源 O 相邻,则位于城市 C 上的蚂蚁能够感受到信源 O 所扩散的信息素浓度 D_C :

$$D_C = D_{\max} \cdot ((h \cdot \tan \alpha - r) / (h \cdot \tan \alpha)). \quad (6)$$

式中: D_{\max} 为信源 O 处的信息素浓度, h 为简化的圆锥体模型高度, α 为圆锥体锥面与中心轴的夹角(为一设定的固定参数), $h \cdot \tan \alpha$ 为扩散范围的半径 r , $d_{C,O}$ 为位于扩散范围中的城市 C 与信源 O 的距离.

基于信息素扩散模型,蚁群算法进行相邻路径信息素的更新.假设蚂蚁 k 刚走过的 2 个城市 i 和 j 之间的距离为 d_{ij} ,该蚂蚁所留的信息素将以 i 和 j 为信源向周围扩散,即以 i 和 j 为中心形成扩散的浓度场,并按简化的扩散模型向周围扩散.这种扩散不仅影响城市 i 和 j 上的其他蚂蚁选择下一段路径,也会影响位于扩散范围内其他城市上蚂蚁的选择路径.若任一城市 l 满足 $d_{il} \leq r$ 或 $d_{jl} \leq r$,则该城市上的蚂蚁在进行下一城市的选择时将受到城市 i 或 j 信源的影响,即蚂蚁 k 的本次行动不仅会导致路径 a_{ij} 上信息素的变化: $\Delta_{ij}^k = Q / d_{ij}$ (Q 为常数),而且也会影响一些相邻路径的信息素浓度变化,如路径 a_{il} 或 a_{jl} 上信息素浓度的变化可表示为: $\Delta_{il}^k = D_{il}^k$, $\Delta_{jl}^k = D_{jl}^k$;令 $h = \bar{d}^{-1} (d_{ij})$, \bar{d} 为大于 1 的可调常数, \bar{d} 为各城市间的平均距离,则 $r = \tan \alpha \cdot \bar{d}^{-1} / (d_{ij})$;以 i 为中心扩散时 $d_{il} = d_{ij}$;以 j 为中心扩散时 $d_{jl} = d_{ij}$;并设 $D_{\max} = \Delta_{ij}^k$,为小于 1 的可调常数,则

$$D_{il}^k = \begin{cases} \frac{\Delta_{ij}^k \cdot Q}{d_{ij}} (1 - \frac{d_{il}}{r}), & d_{il} \leq r, \\ 0, & \text{其他.} \end{cases} \quad (7)$$

$$D_{jl}^k = \begin{cases} \frac{\Delta_{ij}^k \cdot Q}{d_{ij}} (1 - \frac{d_{jl}}{r}), & d_{jl} \leq r, \\ 0, & \text{其他.} \end{cases} \quad (8)$$

依据这种扩散模型,每只蚂蚁每走一步,不仅会改变其所经过的那段路径上的信息素浓度,而且可能会改变多条路径上的信息素浓度,这种改进提高

了蚂蚁群中个体之间的合作效果,增强蚁群算法的有效性,更凸现了群集智能的思想.

2 基于信息素扩散模型解耦控制的蚁群算法

2.1 基于路径信源的信息素扩散模型

从蚁群算法的状态变迁可以看出,蚂蚁选路总是偏向于长度短且信息素浓度高的边.由于路径长度是固定不变的先验信息,故信息素浓度就成为衡量某段路径优劣的一个重要的评价标准.在蚁群算法中,信息素是蚁群个体之间进行信息交流的载体,信息素浓度体现了蚁群群体后验信息的积累,所以对信息素浓度更新策略的合理设计是关系到能否产生高质量解的关键.

基于城市信源的信息素扩散模型就是考虑了信息素向周遍路径扩散的事实,对蚁群算法中相邻路径上信息素的更新进行修正来提高算法求解能力的.不过该算法中对信息素扩散浓度场的模拟过于简化,存在一定的缺陷.例如当 $d_{ij} > 2r$ 时,按照原扩散模型,信息素扩散的浓度场及扩散范围可如图 1(a) 中灰色区域所示,图中小圆点表示城市位置,黑色圆点表示受扩散影响的区域,圆圈表示不同位置浓度场强的等势线.从图可见,城市信源作为信息素扩散浓度场的中心,离信源越近,浓度场的场强越强(等势线越密).除所经过的城市 i 和 j 外,蚂蚁 k 在本次行走中还会影响位于城市 C_1, C_2, C_3, C_4 上蚂蚁选路的行为.但信息素是依附在可行路径上的,所以不失一般性,本文假设信息素随蚂蚁的行走均匀分布在路径上,那么信息素应以所依附的路径为中心向外扩散,所以图 1(a) 中的扩散模型存在一定的局限.为此,本文对其进行改进,将扩散模型改为以路径为信源向周边扩散,如图 1(b) 所示,此时,蚂蚁 k 在从城市 j 走到 i 的过程中留下的信息素将形成更大范围的浓度场,受其影响的区域集合也扩大为 $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8$.从图可见,处于扩散的范围内的城市,无论是到信源城市,还是到信源路径,只要处于信源扩散浓度场内同一等势线上的点都受到相同的信息素浓度的影响,这更符合自然界信源扩散的现象.所以这种变化能更真实地反映信息素扩散的浓度场,并扩大了信息素扩散的作用范围.

2.2 扩散模型的耦合性分析及解耦控制策略

信息素的扩散说明了相邻路径上信息素的浓度是非独立的,相互之间具有耦合作用,即扩散浓度场内相邻路径上信息素的浓度具有耦合性.在相邻路

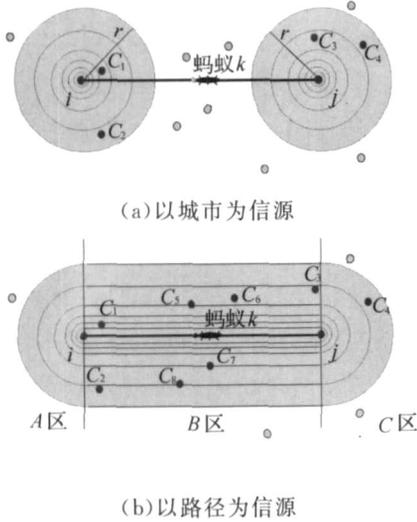


图1 信息素扩散的浓度场场强示意图

Fig. 1 The intensity fields of pheromone diffusion

径上信息素的这种耦合特性,使得一条路径上信息素的改变往往会引起其周边其他路径信息素的变化,从而间接地影响蚂蚁的选路行为.传统的蚁群算法没有考虑这一特性,将路径上信息素的更新独立地进行,这可能会误导蚂蚁选路的方向,增加算法的迭代次数.为解决这一问题,本文引入解耦控制策略,通过耦合补偿将彼此相互影响的多路径信息素浓度问题解耦为相对独立的单路径信息素浓度问题来处理,从而克服蚂蚁选路的干扰,增强算法的鲁棒性,扩散耦合补偿的原理示意图如图2所示.

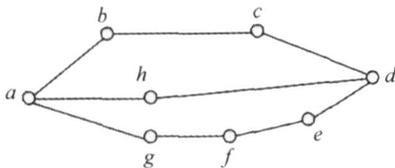


图2 扩散耦合补偿的原理示意图

Fig. 2 Principle map of coupled compensation for diffusion

如图2所示,蚂蚁从a到d存在3条可行通路,在3条通路中分别包含路径ab、ah和ag,其中,路径ab的长度稍小于ah和ag的长度,h到ab、ag的距离小于扩散半径.假设有3只蚂蚁刚刚从a出发,分别经过ab、ah和ag3条路径向d前进,并且在3条路径上留下了相同浓度的信息素.当第4只蚂蚁出发时,如果不考虑信息素的扩散,无疑它将以更大的概率选择b作为下一步要到达的位置.但如果考虑路径信息素的扩散影响,由于距离远近的不同,h处在ab和ag的扩散浓度场内,故ah路径上信息素的浓度为 $(ah) = (ab) + (ag)$;而b、g处在ah的扩散浓度场内,故ab路径上信息素的浓度

为 $(ab) = (ah) + (ag)$,ag路径上信息素的浓度为 $(ag) = (ah) - (ab)$,式中: (\cdot) 表示相应路径.对于其扩散浓度场内相邻路径的耦合补偿;可见,此时路径ah上信息素的浓度将大于其他2条路径ab、ag上信息素的浓度,当大到一定程度时将足以影响第4只蚂蚁的选路行为,即它将以更大的概率选择h作为下一步要到达的位置,从而使该蚂蚁选择了一条通向目标最短的道路,点到路径信源扩散距离求解的示意如图3.

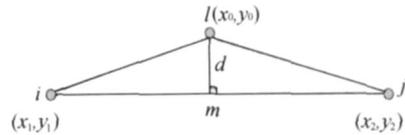


图3 点到路径信源扩散距离求解的示意图

Fig. 3 The sketch map of distance form a point to info fountain of path

针对这种耦合特性,本文的解耦控制策略是利用耦合补偿原理来消除耦合关系,使蚁群算法能够对每条路径上的信息素浓度进行单独的控制.下面推导耦合补偿的计算公式.如图3所示,假设蚂蚁k刚走过路径 a_{ij} ,为了求城市l到路径 a_{ij} 的垂直距离d,对此分2种情况讨论.

1)当TSP问题中的城市位置用坐标给出时,城市i、j和l的坐标分别为 (x_1, y_1) 、 (x_2, y_2) 和 (x_0, y_0) ,这时计算过程如下:

当 $x_1 = x_2$ 时,由i、j两点形成的直线L垂直于x轴,L的方程为: $x = x_1$;

当 $x_1 \neq x_2$ 时,由i、j两点形成的直线L的方程为 $y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$;将直线L的表达式化为

形如 $ax + by + c = 0$ 的标准形式,然后依据点到直线的距离公式,可以得到l点到直线L的距离: $d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$.

2)当TSP问题中城市间的距离已知时,即在图2中已知 a_{ij} 、 a_{jl} 、 a_{li} ,此时,可以利用海伦公式求垂直距离d,方法如下:令 $s = (a_{ij} + a_{jl} + a_{li}) / 2$,则三角形ijl的面积为 $S = \sqrt{s(s - a_{ij})(s - a_{jl})(s - a_{li})}$,又因为 $S = (d \cdot a_{ij}) / 2$,所以 $d = 2S / a_{ij}$.

得到城市l离路径信源的距离d后,就可判断城市l是否位于蚂蚁k在本次行走中所形成的信息素浓度场的扩散范围内,如果在扩散半径内,就根据它的值,计算扩散到相应路径上的信息素:

1)当城市l位于图1(b)中的A区,且 $d_{il} < r$ (扩散半径)时,城市i对路径 a_{ij} 的影响 D_{ij}^k 可按式(9)计

算得到:

$$D_{il}^k = \begin{cases} \frac{\cdot d_{ij} \cdot Q}{L_k} (1 - \frac{d_{il}}{r}), & d_{il} < r, \\ 0, & \text{其他.} \end{cases} \quad (9)$$

2) 当城市 l 位于图 1(b) 中的 C 区, 且 $d_{jl} < r$ (扩散半径) 时, 城市 j 对路径 a_{jl} 的影响 D_{jl}^k 可按式 (10) 计算得到:

$$D_{jl}^k = \begin{cases} \frac{\cdot d_{ij} \cdot Q}{L_k} (1 - \frac{d_{jl}}{r}), & d_{jl} < r, \\ 0, & \text{其他.} \end{cases} \quad (10)$$

3) 当城市 l 位于图 1(b) 中的 B 区, 且 $d_{li} < r$ 时, D_{li}^k 为直线段 a_{ij} 对路径 a_{il} 和路径 a_{jl} 上信息素的影响, 其计算公式为

$$D_{li}^k = \begin{cases} \frac{\cdot d_{ij} \cdot Q}{L_k} (1 - \frac{d_{li}}{r}), & d_{li} < r, \\ 0, & \text{其他.} \end{cases} \quad (11)$$

通过上面的推导, 就可计算出蚂蚁的每次行走对其所形成的扩散范围内相邻路径的影响, 从而对这些路径上的信息素做耦合补偿的修正更新.

2.3 算法描述

基于信息素扩散模型解耦控制策略的蚁群算法首先以路径代替城市作为扩散模型的信息源, 优化了信息素扩散模型, 扩大了浓度场的作用范围; 然后通过解耦控制策略, 在信息素的更新中引入了耦合补偿, 更真实、准确地反映了每条路径上信息素的浓度. 此外, 新算法采用 ACS 算法的基本框架, 即信息素得进行 2 次更新. 新算法的基本流程可以描述如下:

ETPDACO algorithm

{

1) 初始化阶段

初始化参数, 赋各条路径相同的信息浓度, 并将 m 只蚂蚁随机放到 n 个城市节点上;

为每只蚂蚁设置起点 s_k 和允许访问的城市列表 U_k ;

2) 蚁群的一次周游过程 (一次迭代)

FOR $p=1$ TO n 遍历所有城市, 最后回到出发点

FOR $k=1$ TO m 蚁群的一步转移

{ 设蚂蚁当前位置为 i

IF $p < n$ THEN 没有遍历完所有城市

{ 按状态转移公式 (1) 和 (2) 选择下一城市 j

完成一步行走 ij , 并变更允许访问的城市列表 $U_k(j)$;

ELSE 遍历完所有城市

{ 行走一步 ij , 回到出发的城市}

按式 (4) 计算路径 ij 上新产生的信息素浓度;

以 d_{ij} 为信源, 判断位于该扩散浓度场内的城市点, 并根据不同的情况, 依据式 (9)、(10) 或 (11) 计算该信源扩散到相邻其他路径上的信息素浓度;

综合上面各式计算结果, 利用式 (3) 进行局部信息素的更新;

} 信息素扩散、耦合补偿及局部更新

3) 本次迭代信息素的全局更新

FOR $k=1$ TO m

{ 计算 L_k ; } 计算每只蚂蚁在本次迭代中的旅行长度

FOR 每个 $d_{ij} = L_{best} - ittr$

{ 利用式 (5) 进行全局信息素的更新 ($i = 1$); }

全局信息素的更新

4) 判断算法是否结束

IF (结束条件满足) THEN

输出多次迭代后的最优解 $L_{best-all}$;

ELSE

{ 迭代次数增 1; 初始化始点各城市可访问的城市列表;

GO TO 2) 继续进行下一次迭代;

}

2.4 算法复杂度分析

在这一部分, 对基本流程给出算法进行计算复杂度分析. 在初始化阶段, 主要的花费是为每条路径指定初始的信息素浓度, 故 $O(C_n^1) = O(n^2)$; 在构造解的第 2 阶段, 在循环体内主要的计算花费包括城市转移: $O(C_{n-1}^1) = O(n)$, 以及判断其他城市是否落在本扩散浓度场内的计算花费: $3O(C_{n-2}^1) = O(n)$, 所以这阶段总的计算复杂度为: $O(n \cdot m \cdot (n + n)) = O(n^2 \cdot m)$; 在第 3 阶段, 计算蚂蚁的旅行长度复杂度为 $O(m)$, 对最佳路径进行全局信息素更新的复杂度为 $O(n)$, 由于 $m \ll n$, 所以该部分总的计算复杂度为 $O(n)$; 而第 4 阶段计算复杂度为 $O(1)$; 所以, 如果经过 NC 次迭代, 算法成功结束, 这时总的计算复杂度为 $NC \cdot (O(n^2)) + O(n^2 \cdot m) + O(n) + O(1) = O(NC \cdot n^2 \cdot m)$, 与基本的 ACS 蚁群算法具有相同的计算复杂度. 可见, 尽管在信息素扩散和耦合补偿过程中, 算法增加了一些计算量, 但并没有引起复杂度阶次的变化, 而且由于耦合补偿可以克服相邻路径间彼此的干扰, 更客观、正确地引导蚂蚁的选路, 减少算法运行的迭代次数, 所以总的计算性能会得到提高.

3 实验结果

本节对新算法的性能进行测试,并将测试结果与基本的 ACS 算法及 PDACO 算法进行了比较.实验中所使用的多个 TSP 问题的数据来源于 TSPLIB 标准库 (<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>). 算法用 Visual C++ 语言编程实现,实验的参数均通过实验确定,表 1 为获得下面实验结果所使用的参数表.

表 1 不同 TSP 范例所使用的参数配置

Table 1 Parameter deployments used in experiments for several TSP problems

TSP 问题的 典型范例	实验时使用的参数							
	1 -	Q	Q	Q	Q	Q	Q	Q
oliver30	0.8	5	600	1.5	3	0.55	1	0.4 / 8
bays29	0.8	5	600	2.0	3	0.55	1	0.4 / 8
dantzig42	0.8	5	450	2.0	4	0.5	1	0.4 / 8
berlin52	0.8	5	700	2.5	4	0.55	1	0.4 / 8

为了进一步验证本文算法的有效性,在 TSP 问题典型范例上进行了大量的实验.表 2 是不同算法运行得到的实验结果.在表 2 中,最优解是指获得的最短路径长度,进化代数是指当得到表 2 中的结果时,蚂蚁的周游代数(10 次实验的平均取整).从结果可见,对于 oliver30 问题,3 种算法都能得到最优解,但所需要的迭代次数和运行时间各不相同,ACS 算法的迭代次数和运行时间最大,PDACO 算法的迭代次数和运行时间较 ACS 算法有明显改进,而本文算法在迭代次数和运行时间上又有进一步的改进.类似地,在其他 3 个问题上本文算法在迭代次数和运行时间上都有显著的改进.而且,在较少的进化代数和较短的运行时间内,本文算法在 oliver30、bays29、dantzig42 问题上都得到了各自的最优解,在 berlin52 问题上也得到了更接近于最优解的结果.

表 3 为不同算法在各种 TSP 问题上的单次运行时间比较,从实验结果可以看出:正如在算法计算复杂度中所分析的那样,由于 PDACO 算法和本文算法都包含扩散模型的建立与计算,故单次迭代的运行时间都较 ACS 算法有所增加,而本文算法与 PDACO 算法相比,由于具有相当的单次计算复杂度,故在单次迭代的运行时间上非常接近.

表 2 各种 TSP 问题上不同算法的结果比较

Table 2 Comparison of results among different algorithms for several TSP problems

典型范例	问题的 最优解	ACS 算法			PDACO 算法			本文算法		
		得到的 最优解	所需进 化代数	运行时 间/s	得到的 最优解	所需进 化代数	运行时 间/s	得到的 最优解	所需进 化代数	运行时 间/s
oliver30	423.74	423.74	830	12.45	423.74	76.00	3.8	423.74	33	1.68
bays29	2 020.00	2 034.00	955	13.30	2 028.00	132.00	5.6	2 020.00	46	2.00
dantzig42	699.00	714.00	896	37.60	712.00	174.00	23.1	699.00	61	8.20
berlin52	7 542.00	7 681.45	1 754	147.30	7 663.59	316.00	90.2	7 544.37	82	23.80

表 3 不同算法在各种 TSP 问题上的单次运行时间比较

Table 3 Comparison of time consumption of a single cycle among different algorithms for several TSP problems

典型范例	问题的 最优解	ACS 算法		PDACO 算法		本文算法	
		得到的最优解	每次迭代的 运行时间/s	得到的最优解	每次迭代的 运行时间/s	得到的最优解	每次迭代的 运行时间/s
oliver30	423.74	423.74	0.015 00	423.74	0.050 00	423.74	0.051 00
bays29	2 020.00	2 034.00	0.013 93	2 028.00	0.042 42	2 020.00	0.043 48
dantzig42	699.00	714.00	0.041 96	712.00	0.132 76	699.00	0.134 43
berlin52	7 542.00	7 681.45	0.083 98	7 663.59	0.285 44	7 544.37	0.290 24

图 4 为不同算法在 dantzig42 问题上的解图,其中图 4(a) 为 ACS 算法得到的最优解,图 4(b) 为 PDACO 算法得到的最优解,而图 4(c) 为本文算法得到的最优解.从图可见,不同算法在局部连接的差异(虚线圆框内的连接部分)导致了最后所得到的最优解的不同.

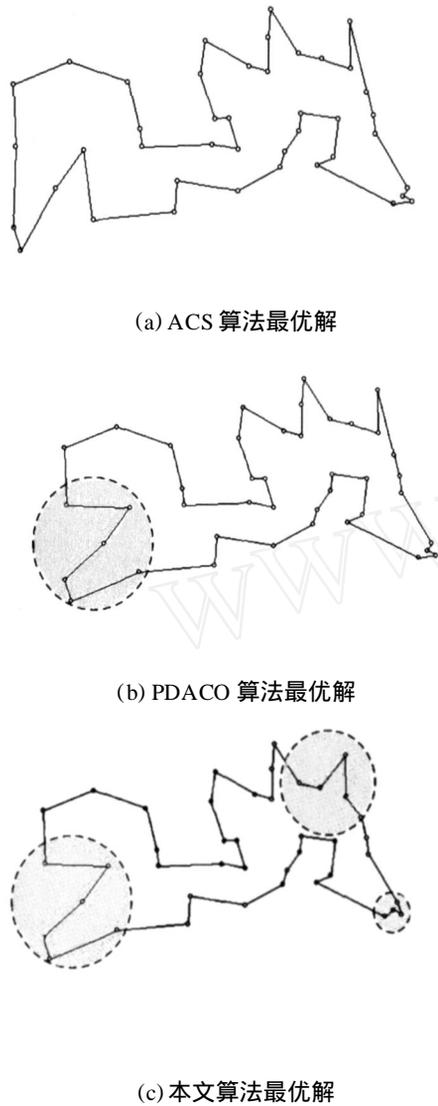


图 4 不同算法在 dantzig42 问题上的解图
Fig. 4 Tours of several algorithms for dantzig42

图 5 是不同算法在 dantzig42 问题的求解过程中得到的多样性曲线比较.其中,横坐标为蚁群旅行的迭代数,纵坐标为蚁群在每次迭代所得到的旅行长度的标准偏差,其公式见参考文献[8].从图 5 中曲线的对比(标准偏差的均值和振幅)可以看出,在整个随机搜索的过程中,PDACO 算法、本文算法的多样性性能明显优于 ACS 算法,而且本文算法在多样性性能方面也优于 PDACO 算法,即本文算器具

有更强的不断获得最优解的能力,因此本文算法更容易克服最优解停滞的现象(陷入局部最优),获得问题的全局最优解.

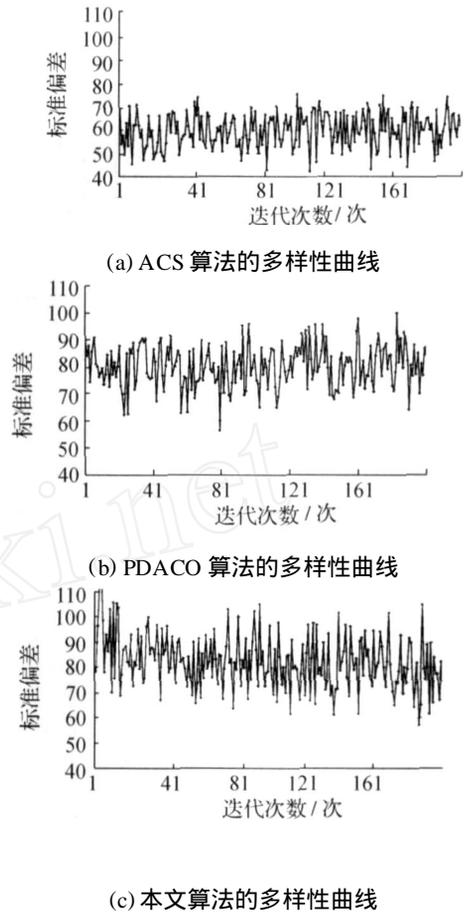


图 5 不同算法在 dantzig42 问题上的多样性曲线
Fig. 5 Diversity curves of several algorithms for dantzig42

总之,本文算法在解的质量、解的多样性、周游的迭代代数、算法运行时间等方面具有更好的结果.这说明了基于信息素扩散模型的解耦策略利于提高和改进蚁群算法的求解能力.

4 结束语

蚁群优化算法是目前群集智能理论研究领域中的一个研究热点,该算法已成功应用于许多复杂组合优化问题的求解.虽然目前的许多算法都具有发现较优解的能力,但迭代次数多、收敛速度慢仍是制约大多数算法在实际组合优化问题中应用的瓶颈.针对这一问题,本文结合信息素扩散的耦合特征,提出一种基于信息素扩散模型解耦控制策略的蚁群算法.大量 TSP 问题的实验结果表明:该算法不仅能获得更好的解,而且能加快算法的收敛速度.进一步

的研究包括借助于相关学科的知识对各种实验参数的确定进行理论分析,剖析参数的选择与问题最优解以及解的收敛性的关系。

参考文献:

- [1]彭喜元,彭宇,戴毓丰.群智能理论及应用[J].电子学报,2003,31(12):1982-1988.
PENG Xiyuan, PENG Yu, DAI Yufeng. Swarm intelligence theory and applications[J]. Journal of Electronics, 2003, 31(12):1982-1988.
- [2]DORIGO M, COLORNI A, MANIEZZO V. Distributed optimization by ant colonies [A]. Proc of the First European Conf of Artificial Life [C]. Paris: Elsevier, 1991.
- [3]DORIGO M, MANIEZZO V, COLORNI A. The ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1):29-41.
- [4]DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans Evolutionary Computation, 1997, 1(1):53-66.
- [5]GAMBARDELLA L M, DORIGO M. Solving asymmetric and asymmetric TSPs by ant colonies[A]. International Conference on Evolutionary Computation[C]. Nagoya, Japan, 1996.
- [6]STUTZLE T, HOOS H H. Max-min ant system and local search for the traveling salesman problem[A]. IEEE Intl Conf on Evolutionary Computation[C]. Indianapolis: IEEE Press, 1997.
- [7]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.
ZHU Qingbao, YANG Zhijun. An ant colony optimization algorithm based on mutation and dynamic pheromone updating[J]. Journal of Software, 2004, 15(2):185-192.

- [8]黄国锐,曹先彬,王煦法.基于信息素扩散的蚁群算法[J].电子学报,2004,32(5):865-868.

HUANG Guorui, CAO Xianbin, WANG Xufa. An ant colony optimization algorithm based on pheromone diffusion[J]. Chinese Journal of Electronics, 2004, 32(5):865-868.

- [9]BLUM C, DORIGO M. The hyper-cube framework for ant colony optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2004, 34(2):1161-1172.

作者简介:



冀俊忠,男,1969年生,博士,副研究员,中国计算机学会高级会员,主要研究方向为机器学习、Web智能、计算智能.在国内外学术刊物和重要国际学术会议上发表论文20余篇,其中10余篇被SCI、EI、ISTP三大检索收录.

Email: jjz01@bjut.edu.cn.



刘椿年,男,1944年生,教授,博士生导师,主要研究方向为数据挖掘、人工智能、约束逻辑程序设计.主持多项国家自然科学基金和863计划课题.在国内外学术刊物和重要国际学术会议上发表论文100余篇,出版专著及译著多部.



黄振,男,1981年生,硕士研究生,主要研究方向为机器学习、Web智能、计算智能.