

个体速度差异的蚁群算法设计及仿真

印峰¹, 王耀南¹, 刘炜², 周良¹

(1. 湖南大学 电气与信息工程学院, 湖南 长沙 410082; 2. 湖南科技职业学院 软件学院, 湖南 长沙 410118)

摘要:针对如何提高蚁群算法搜索速度及防止算法停滞问题,提出一种改进的蚁群优化算法 VACO(ACO algorithm based on ant velocity),通过构造与局部路径和蚂蚁个体速度相关的时间函数,并建立与时间函数相关的动态信息素释放机制,加快信息素在较优路径上正反馈过程,从而提高了算法的收敛速度;采取一种连续小区间变异策略,在加快局部搜索过程的同时可有效防止算法陷入局部最优。对典型 TSP 问题的仿真研究结果表明,改进后的算法在收敛性和对较好解的探索性能得到一定程度的提高。

关键词:蚁群算法;旅行商问题;信息素;NP-难解

中图分类号:TP301.6 **文献标识码:**A **文章编号:**1673-4785(2009)06-0528-06

Design and simulation of an ant colony algorithm based on individual velocity differences

YIN Feng¹, WANG Yao-nan¹, LIU Wei², ZHOU Liang¹

(1. College of Electrical and Information Engineering of Hunan University, Changsha 410082, China; 2. School of Software, Hunan Vocational College of Science and Technology, Changsha 410118, China)

Abstract: A new implementation of the ant colony optimization (ACO) algorithm was primarily focused on improving search speed and preventing stagnation. To resolve these two issues, improvements based on velocity were proposed, producing a VACO algorithm. By constructing a time-function for local paths and ant velocity, and building a dynamic release mechanism for pheromones in the time-function, it accelerated positive feedback from the accumulation of pheromones, leading to better paths and improved convergence speed. A strategy of continuous inter-cell mutation sped up local searches and at the same time effectively prevented the algorithm being trapped in local optimums. The results showed that the proposed algorithm improves convergence and increases the possibility of finding optimal solutions.

Keywords: ant colony algorithm; TSP; pheromone; NP-hard

蚁群算法是模拟自然界中真实蚁群的觅食行为而形成的一种模拟进化算法^[1]。目前该算法已在求解组合优化问题、系统辨识、机器人路径规划、数据挖掘、网络路由等问题时取得了很好的效果^[2-3]。然而与其他算法相比,蚁群算法存在搜索速度较慢、容易出现停滞现象的缺陷。特别对于大规模优化问题,如何加快其收敛速度以及防止算法陷入局部最优一直是蚁群算法研究中有待解决的一个热点和难点问题。总体来说,解决该问题的关键在于如何处理好下述问题之间的平衡:即使算法的搜索空间尽可能大,

以寻找可能存在最优解的解区间,又同时充分利用当前群体的有效信息,使算法搜索的侧重点放在可能具有最优解的子区间内;使得算法在可以接受的时间内收敛到全局最优解。目前提出的各种改进的蚁群算法均围绕对上述问题的研究展开^[4-7]。

信息素更新策略是决定算法收敛速度的关键之一,针对如何提高算法收敛速度的问题,目前提出的几种有代表性的蚁群改进算法,比如最优保留蚂蚁系统^[4](ASelite)、蚁群系统^[5](ACS)、最大-最小蚂蚁系统^[6](MMAS)等,均采用针对全局最优解所属边进行信息素全局更新的策略,即通过加强对当前最优解的利用来缩小算法的搜索空间,进而提高算法的收敛速度。另外也有学者提出动态信息素更新策略^[8-9],其本质也是对蚂蚁所走较优路径上的信息加强的动态

收稿日期:2009-08-12.

基金项目:国家科技支撑计划资助项目(2008BAF36B01);国家“863”计划资助项目(2008AA04Z214)。

通信作者:印峰. E-mail: yinfeng83@126.com.

调整过程. 上述研究结果表明, 增加全局最优解的使用频率, 可以使算法获得较好的性能. 对较优解的强化有助于加快算法的收敛速度, 同时也容易引起停滞现象, 使算法陷入局部最优. 针对该问题, 目前提出的有效的解决策略可归结于: 1) 人为限制信息素的总量, 避免路径之间信息素差量过大^[6]; 2) 引入判决因子, 根据定性的分析判断蚁群当前的内部状态, 进而通过对相关参数的动态调整避免停滞现象^[8,10]; 3) 将蚁群算法与其他智能方法相结合, 在加快局部搜索的同时提高解的多样性^[11].

针对上述问题, 本文做了两方面的工作: 1) 提出一种改进的蚁群算法 VACO (ACO algorithm based on ant velocity), 通过构造与路径和蚂蚁个体速度相关的时间函数, 并建立与时间函数相关的动态信息素释放机制, 使得搜索过程兼顾局部和全局信息, 加快了在较优路径上信息素正反馈的过程, 提高了算法的收敛速度; 2) 采取一种连续变异策略, 在加快局部搜索过程的同时可有效防止算法陷入局部最优. 实验结果表明, 改进后的算法在收敛性和对较好解的探索方面得到一定程度的改善.

1 VACO 算法原理分析

1.1 考虑蚂蚁个体速度的信息素释放机制

以自然界真实蚂蚁路径寻优过程为对象, 考虑如下 2 种情况:

1) 2 组不同速度的蚁群路径寻优比较.

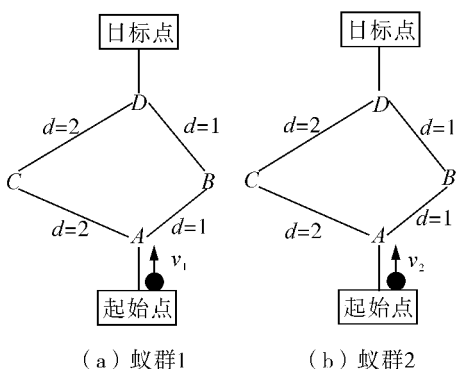


图1 不同蚁群寻优速度比较

Fig. 1 Comparison of the path optimization speed between different colonies

如图1所示, 在起始点和目标点之间有 2 条道路: ABD 和 ACD, 其长度分别为 4 和 6. 蚁群 1 和蚁群 2 分别以速度 v_1 和 v_2 匀速移动, 并且 $v_1 > v_2$, 假设这 2 种群采取完全相同的路径寻优机制. 显然, 蚁群 1 完成一次巡游的时间要快于蚁群 2, 在相同时间间隔内, 蚁群 1 在较优路径 ABD 上信息素的积累量要大于蚁群 2, 随着该过程的继续, 蚁群 1 的寻优

过程要快于蚁群 2.

2) 同一蚁群中蚂蚁个体路径寻优比较.

对于以相同速度移动的同一种群中的不同个体, 选择较短路径 ABD 移动的个体完成一次巡游的时间要快于选择较长路径 ACD 的个体; 因此, 在相同时间间隔内, 较短路线 ABD 上积累的信息素更多, 其被选择的机会相应更大, 随着该过程的继续, 2 条道路上的信息素数量的差距将越来越大, 直至绝大多数蚂蚁都选择了最短路径.

通过对于真实蚂蚁路径寻优过程的观察可知, 其寻优过程的快慢不仅与选择的移动路径有关, 而且与其移动速度相关. 即有如下关系成立:

$$\text{Search_time} \propto \frac{d(i,j)}{\text{vel}(k)}. \quad (1)$$

式中: $d(i,j)$ 为 2 节点连线距离, $\text{vel}(k)$ 表示第 k 只蚂蚁的移动速度. 根据式(1), 蚂蚁移动速度越快, 选择的路径越短, 搜索时间就越快. 不同于真实蚂蚁在连续时间内的路径寻优过程, 人工蚂蚁采取离散的搜索机制, 因此不能用蚂蚁速度直接衡量算法寻优的快慢.

由前面分析可知, 从整体上观察, 完成一次搜索时间较快的真实蚂蚁个体在移动路径上释放的信息素相对较多. 如果给人工蚁群设置一个初始“速度”, 并建立某种激励机制使得较优个体“移动速度”呈更快的增长趋势, 同时根据式(1)关系构造蚁群的路径信息素更新方式, 此时可以采用信息素释放量间接描述蚂蚁的寻优速度, 据此定义如下的算法构造规则:

规则 1 假设蚁群具有一定的初始速度 InitVel , 并且每移动一个步长, 第 k 只蚂蚁的速度增加 $\Delta \text{vel}(k)$, 其中, 蚂蚁速度的增量 $\Delta \text{vel}(k)$ 与路径残余信息素量成正比. 每完成一次迭代计算, 对蚂蚁的速度进行初始化.

规则 2 路径信息素释放量为一个与搜索时间相关的函数, 该函数值由移动路径和蚂蚁速度决定. 路径长度越短, 蚂蚁速度越快, 则在该路径上释放的信息素数量越多.

引入蚂蚁个体运动速度递增的机制, 可以加快信息素在较优路径上信息素正反馈的过程, 从而可以提高蚂蚁的搜索效率.

1.2 带变异机制的防停滞策略

根据文献[10,13], 对蚁群算法停滞及相关概念定义如下:

定义 1 在蚂蚁搜索过程中, 由于较好路径上的信息素远大于其他边时, 造成所有蚂蚁都选择同样的路

径,即系统不再搜索较好的解,称为停滞现象。

定义2 设 $\tau_{\min}(r,s)$ 、 $\tau_{\max}(r,s)$ 分别为与节点 r 相连的边上最大、最小信息素值,令

$$\delta(r) = \tau_{\max}(r,s) - \tau_{\min}(r,s).$$

对某个给定的 $\lambda(0 < \lambda < 1)$, 在所有与节点 r 相连的边中,信息素量大于 $\lambda\delta(r) + \tau_{\min}(r,s)$ 的边的数量即为节点 r 的节点分支数。

定义3 设 $\theta(r)$ 为节点 $r(r=1,2,\dots,n)$ 的节点分支数, N 为节点总数,则平均节点分支数定义为

$$\sum_{r=1}^n \frac{\theta(r)}{N}. \quad (2)$$

由于信息素蒸发的影响,较差解路径上的信息素量将越来越少,节点上连接边信息素最大值和最小值的差量将越来越大,根据定义2,节点分支数将随之减少.当平均节点分支数趋于2时,意味着所有的蚂蚁都选择了同样的路径,即算法出现停滞现象.如果此时当前解还未达到最优,则说明算法陷入了局部最优.利用变异算法局部寻优能力强的特点对当前最优个体进行变异,可加快局部寻优速度,并且有助于算法跳出局部最优。

在遗传算法中,变异处理过程一般是先随机地选择2个位置点,然后对区间之间的节点位置按所取适度函数值大小进行重排.为了表现变异过程的随机性,变异操作的区间长度是随机的.然而,区间长度过大会增加算法搜索的随机性和盲目性,特别地,对于算法已收敛到局部较优解的情况,变异区间较大时很难找到更优解,为此,提出一种效果较为理想的连续小区间变异策略:如图2所示,将各节点按排列顺序划分为若干个连续子区间,每个区间长度取3.以区间 i 为例,交换 b 、 c 2 节点的位置显然并不影响区间 i 与外区间节点间的连接关系;现在只需比较交换前后区间 i 内节点距离和大小,即比较 $D_1 = D(a,b) + D(c,d)$ 和 $D_2 = D(a,c) + D(b,d)$ 的大小,如果 $D_2 < D_1$,说明 $(a c b d)$ 序列组合要优于 $(a b c d)$,交换原节点序列中 b 、 c 的位置,否则保留区间 i 原节点顺序。

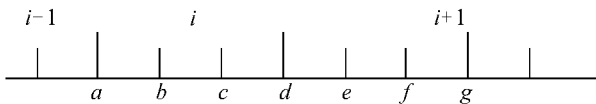


图2 变异区间划分规则

Fig. 2 Division rules of interval variation

这种局部变异调整后的搜索效率很高,在蚁群算法迭代前期有助于提高算法的搜索速度;在算法迭代后期,算法陷入局部最优,其宏观表现是蚂蚁的巡游线路出现局部路径交叉,显然,采取局部调整的策略对于这一情况的处理是非常有效地.综上,定义

如下变异操作规则:

规则3 采用对访问节点序列按排列组合的方法进行编码,即某个巡回路径的染色体是该巡回路径的节点序列.每完成一次迭代计算,只对当前最优解进行变异操作。

规则4 对最优个体重组分区进行,操作区间长度取3,变异适度函数取节点连线距离,交换区间内2节点的位置,并比较交换前后区间节点之间的距离和,如果交换后的区间节点之间的距离和小于交换前,则交换原节点序列相应2节点位置,否则保留区间内原节点序列顺序,进行下一个操作区间并重复相同的处理过程。

2 VACO 算法构造

1) 初始化,设置蚂蚁的初始速度 $vel = InitVel$, 用禁忌表 $tabuk$ 记录蚂蚁 k 当前所走过的节点, $allowed_k = \{C - tabuk\}$ 表示蚂蚁 k 下一步允许选择的节点,其中, C 为所有节点的集合;其余初始化过程与蚁群系统(ACS)算法^[3]相同。

2) 将 m 只蚂蚁随机分配到 n 个节点,并将出发节点置入禁忌表 $tabuk$ 。

3) 对处于第 i 个节点的第 k 只蚂蚁,依转移概率式(3)^[10]选择连接的下一个节点 j 。

$$p_k(i,j) = \begin{cases} \arg \max_{j \in allowed_k} \{ \tau(i,j) \eta^\beta(i,j) \}, & q \leq q(\lambda(t)); \\ S, & \text{其他.} \end{cases} \quad (3)$$

式中:

$$q(\lambda(t)) = \frac{\lambda(t)}{n}; \quad (4)$$

$\lambda(t) \in [2, n]$ 表示算法在第 t 次迭代时的平均节点分支数; n 表示节点数; q 为 $[0, 1]$ 上分布的随机数; $\tau(i,j)$ 表示节点 i 与节点 j 之间的信息素量; $\eta(i,j)$ 表示节点 i 与节点 j 之间的启发式因子,一般取2节点间距离的倒数; β 为权重系数。

蚂蚁在选择下一个节点之前先随机生成 q , 如果 q 值小于 $q(\lambda(t))$, 则从节点 i 到所有可行的节点中找出概率最大的节点,即为下一个要到达的节点;否则按如下概率公式选择下一个节点:

$$p_k(i,j) = \begin{cases} \frac{\tau(i,j) \eta^\beta(i,j)}{\sum_{j \in allowed_k} \tau(i,j) \eta^\beta(i,j)}, & j \in allowed_k; \\ 0, & \text{其他.} \end{cases}$$

4) 修改禁忌表:选择节点后将蚂蚁移动到新的

节点,并把该节点移动到该蚂蚁个体的禁忌表中。

5) 当第 k 只蚂蚁从节点 i 转移到节点 j 后,更新第 k 只蚂蚁的速度。根据规则 1 构造速度增量计算公式:

$$vel(k) = vel(k) + \frac{a_v \tau^\alpha(i, j)}{b_v \tau^\alpha(i, j) + c_v}. \quad (5)$$

式中: a_v 、 b_v 和 c_v 为与速度更新相关的常数,可用实验方法确定其值的最优组合;参数 α 表示路径信息素对蚂蚁速度的影响权重。由式(5)可知,蚂蚁速度增量与路径残余信息素成正比,并且每只蚂蚁的一次巡游速度是递增的,但选择较短路径的蚂蚁移动速度要快于选择较长路径的蚂蚁移动速度。

6) 计算动态路径信息素释放量:

$$\Delta\tau(i, j) = \frac{a_s}{b_s + c_s \text{time}^\gamma(d(i, j), vel(k))}. \quad (6)$$

式中: a_s 、 b_s 和 c_s 为与信息素释放量相关的常数;参数 γ 表示移动时间对路径信息素释放量的影响权重。时间函数根据前述的规则 2 构造如下:

$$\text{time}(d(i, j), vel(k)) = \frac{d(i, j)}{vel(k)}. \quad (7)$$

根据上式可知,蚂蚁移动速度越快,选择的路径越短,则移动所花的时间越短,在相应路径上释放的信息素越多。

7) 信息素局部更新。第 k 只蚂蚁从节点 i 转移到节点 j 后,按下式更新路径 ij 上的信息素:

$$\tau(i, j) = (1 - \rho)\tau(i, j) + \rho \cdot \Delta\tau(i, j). \quad (8)$$

式中: ρ ($0 < \rho < 1$) 表示路径信息素的蒸发系数, $1 - \rho$ 表示信息素的持久性系数, $\Delta\tau(i, j)$ 表示本次迭代路径 ij 上信息素的增量,根据式(6)计算。

8) 跳转到 3) 重新执行上述过程,直至所有的蚂蚁巡游一次完毕。

9) 计算当前最优路径 T^{ib} 的巡游距离 L_{best} , 记录当前最优路径节点编号。

10) 根据规则 3、规则 4 对最优个体进行变异操作,如果经变异产生新的更优解,则用该更优解取代当前最优解。

11) 信息素全局更新。对全局最优解所属的边进行信息素更新:

$$\tau(i, j) = 1 - \rho \cdot \tau(i, j) + \rho \cdot (L_{best})^{-1}, \quad \forall (i, j) \in T^{ib}. \quad (9)$$

12) 根据定义 1、定义 2 计算当前的平均节点分支数。

13) 判断是否已达到最大迭代步长 NC_{max} 。如果未达到最大迭代次数,则对禁忌表清零,并对蚂蚁

速度进行初始化,跳转到 2) 进行下一次迭代计算;否则,停止计算。

整个算法流程图如图 3 所示。

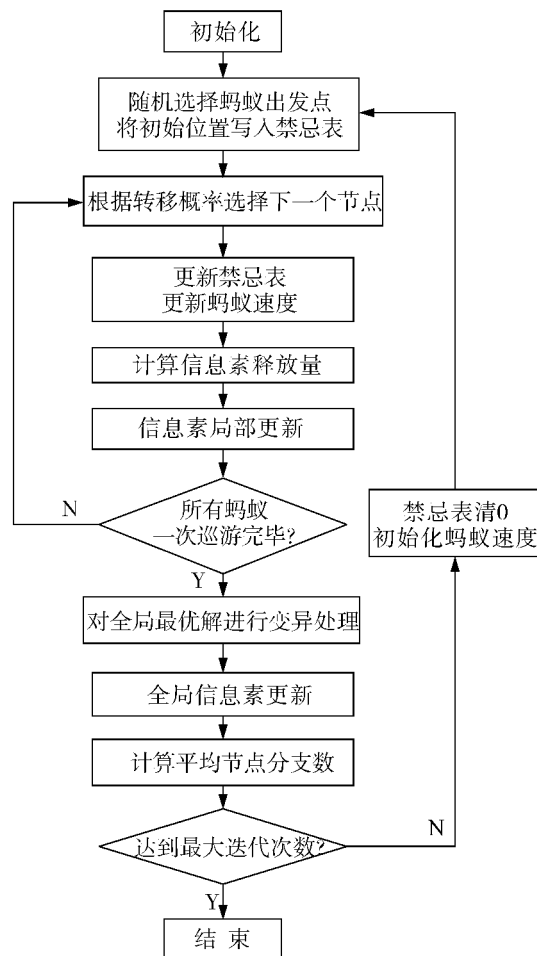


图3 VACO 算法流程图

Fig. 3 The process of VACO algorithm

3 仿真实例

为了检测提出的改进算法的性能,采用经典旅行商(TSP)问题进行了不同的实验,并将改进算法与几种有代表性的改进蚁群算法作了性能上的对比分析。本文算法中各参数经初步优化后的取值见表 1,蚁群初始速度 InitVel 取 200。

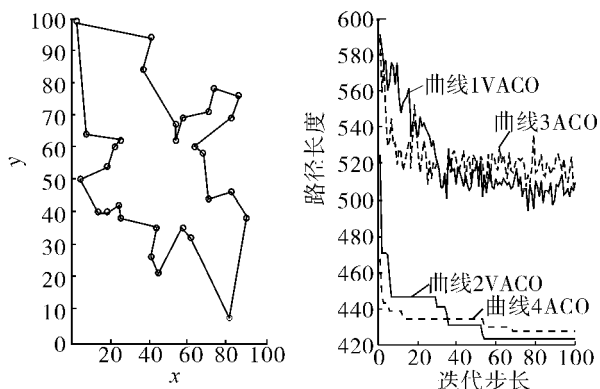
表 1 VACO 算法的参数设置

Table 1 VACO parameter setting

β	a_v	b_v	c_v	α	a_s	b_s	c_s	γ	ρ
4	1	0.01	1	1	1	0.01	1	1	0.01

首先分别采用提出的 VACO 算法和蚁群优化(ACO)算法对 Oliver30 问题进行求解。迭代次数取 100,计算结果见图 4,其中,曲线 1 和曲线 3 分别为 VACO 和 ACO 计算中每一代所有解取平均值的进化曲线,曲线 2 和曲线 4 分别为 VACO 和 ACO

计算中每一代最好解的进化曲线.



(a)30 城市的实际路径 (b)解的进化曲线

图4 30 城市的实际路径及解的进化曲线

Fig.4 The actual path of 30 cities and evolution of solution curves

由于蚁群中不同个体进化差异以及蚂蚁个体选择路径时具有一定的随机性,因此所有蚂蚁走过路径长度的平均值的进化曲线会产生一定的振荡.另外,由于蚁群算法属于一种增强型学习算法,因此平均路径值进化曲线整体将呈下降的趋势,而各代中最好解的进化曲线将逐渐收敛到最优解或近似最优解.比较图4中曲线1和曲线3可知,在有限迭代次

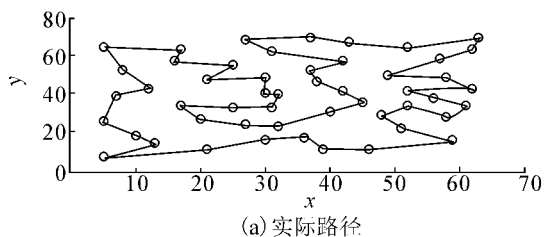
数内,采用 VACO 算法计算得到的平均路径值的进化曲线收敛性要好于 ACO 算法,并且呈现出更强的下降趋势,这说明引入速度机制的蚁群算法提高了个体的搜索能力,使得蚁群的整体寻优能力得到一定程度的提高.比较图4中曲线2和曲线4可知,在有限迭代次数内,采用 VACO 计算获得的最好解要优于 ACO 算法的计算结果,这说明改进后的蚁群算法在对最好解的探索能力方面优于 ACO 算法,同时也间接说明了 VACO 算法具有更快的寻优速度.另外,注意到虽然 VACO 算法搜索最优解的速度要快于 ACO 算法,但在算法搜索的初期,由于函数构造形式和参数选择等原因,其搜索速度优势并未体现出来.因此, VACO 算法本身的构造并不是最优形式,即便如此,由前面的分析可知, VACO 算法整体性能要优于 ACO 算法.

采用 VACO 算法对 Eil51 问题进行求解,并将实验结果与蚂蚁系统(AS)、蚁群系统(ACS)、自适应蚁群算法(AACS)计算结果^[13]进行比较.最大迭代步数为3 000,取连续10次计算结果,并计算其平均值(Avg)和标准方差(Std. Dev).为便于比较,路径长度取整数最优解,计算结果见表2.

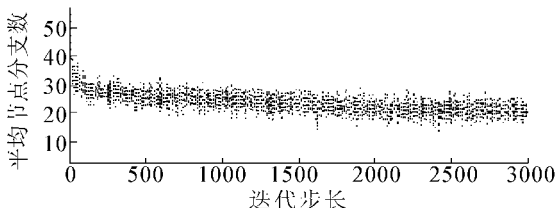
表2 不同改进蚁群算法求解 Eil51 问题的结果比较

Table 2 Comparison of the solving results between improved ant colony algorithms for the Eil51

算法	1	2	3	4	5	6	7	8	9	10	Avg	Std. Dev
AS	429	432	431	429	431	438	432	441	435	432	433.0	3.887
ACS	428	429	427	428	427	434	427	430	426	427	428.3	2.312
AACA	428	428	427	428	426	431	427	431	426	426	427.8	1.874
VACO	426	428	427	427	428	426	430	428	426	427	427.3	1.252



(a)实际路径



(b)平均节点分支数进化曲线

图5 50 城市的实际路径及平均节点分支数进化曲线

Fig.5 The actual path of 50 cities and evolution of average node branching

由表2可知, VACO 算法具有更强的发现较好解的能力,10次测试的平均结果也要优于其他算法;而最小标准方差的取得则说明 VACO 算法具有较好的计算稳定性.图5反映了 VACO 算法在一次典型计算过程中平均节点分支数的变化情况,显然, VACO 算法在搜索解的过程中保持了解的多样性,从而有助于防止算法陷入局部最优.

根据实验仿真结果可知,引入蚂蚁速度和变异机制的蚁群算法在寻优速度和对较好解的探索方面已呈现出一定的优势.但值得注意的是,算法中人工蚂蚁的“速度”与真实蚂蚁速度是有区别的,真实蚂蚁的移动速度越快,其搜索过程就越快;而人工蚂蚁搜索速度是用有效路径上信息素积累的快慢来衡量.因此,人工蚂蚁“速度”大小并不能直接决定算法搜索的快慢,其初始速度的设置也由算法相关函

数的具体构造形式决定. 而引入速度机制的蚁群算法其优点正在于加强了有效路径上信息素的正反馈过程.

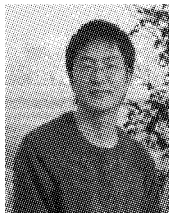
4 结束语

本文从新的角度提出了一种改进的蚁群算法. 通过构造与蚁群个体“速度”相关的信息素更新机制,并结合变异策略,在一定程度上提高了算法的求解效率和对较好解的探索性能. 在对典型 TSP 问题仿真研究中,应用本文方法求解取得了较好的效果,为蚁群算法研究及其在各种工程问题中的应用提供了一个新的思路. 目前研究工作还停留在实验阶段,有关算法本身的构造及相关参数的选取还值得进一步研究.

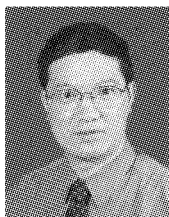
参考文献:

- [1] COLORNI A, DORIGO M, MANIEZZO V, et al. Distributed optimization by ant colonies[C]//Proceedings of European Conference on Artificial Life. Paris, 1991:134-142.
- [2] 高 玮. 新型智能仿生模型——蚁群模型[J]. 智能系统学报, 2008, 3(3): 271-278.
GAO Wei. The intelligent bionic model—ant colony[J]. CAAI Transactions on Intelligent Systems, 2008, 3(3): 271-278.
- [3] 段海滨. 蚁群算法及其应用[M]. 北京: 科学出版社, 2005: 98-101.
- [4] DORIGO M, MANIEZZO V, COLORINI A. The ant system: optimization by a colony of cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996, 26(1): 29-41.
- [5] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [6] BULLNHEIMER B, HARL R F, STRAUSS C. A new rank-based version of the ant system: a computational study[J]. Central European Journal for Operations Research and Economics, 1999, 7(1): 25-38.
- [7] 周建新, 杨卫东, 李 擎. 求解连续函数优化问题的改进蚁群算法及仿真[J]. 系统仿真学报, 2009, 21(6): 1685-1688.
ZHOU Jianxin, YANG Weidong, LI Qing. Improved ant colony algorithm and simulation for continuous function optimization[J]. Journal of System Simulation, 2009, 21(6): 1685-1688.
- [8] 覃刚力, 杨家本. 自适应调整信息素的蚁群算法[J]. 信息与控制, 2002, 31(3): 199-201.
TAN Gangli, YANG Jiaben. An improved ant colony algorithm based on adaptively adjusting pheromone[J]. Information and Control, 2002, 31(3): 199-201.
- [9] 马溪骏, 潘若愚, 杨善林. 基于信息素递减的蚁群算法[J]. 系统仿真学报, 2006, 18(11): 3297-3300.
MA Xijun, PAN Ruoyu, YANG Shanlin. Ant colony algorithm based on pheromone declining[J]. Journal of System Simulation, 2006, 18(11): 3297-3300.
- [10] DORIGO M, Di CARO G, GAMBARDELLA L M. Ant algorithms for discrete optimization[J]. Artificial Life, 1999, 5(2): 137-172.
- [11] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(19): 1241-1245.
WU Qinghong, ZHANG Jihui, XU Xinhe. An ant colony algorithm with mutation features[J]. Journal of Computer Research & Development, 1999, 36(19): 1241-1245.
- [12] 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): 186-192.
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.
- [13] 黄席樾, 张著洪, 何传江, 等. 现代智能算法理论及应用[M]. 北京: 科学出版社, 2005: 122-127.

作者简介:

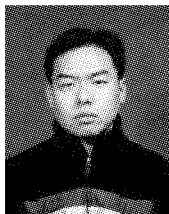


印 峰,男,1983 年生,博士研究生,主要研究方向为机器人控制及人工智能计算.



王耀南,男,1957 年生,教授,博士生导师,主要研究方向为智能机器人、智能信息处理和智能控制. 现任湖南大学电气与信息工程学院院长,国家高效磨削工程技术研究中心副主任,教育部输变电新技术工程研究中心主任. 国际

IEEE 高级会员,国际自动控制联 IFAC 会员,中国人工智能学会、自动化学会、电机工程学会理事. 在国内外发表学术论文 200 多篇,出版专著和教材 4 部.



刘 炜,男,1981 年生,助教,硕士,主要研究方向为人工智能.