

一种快速求解旅行商问题的蚁群算法

冀俊忠 黄 振 刘椿年

(北京工业大学计算机学院多媒体与智能软件技术北京市重点实验室 北京 100124)

(jjz01@bjut.edu.cn)

A Fast Ant Colony Optimization Algorithm for Traveling Salesman Problems

Ji Junzhong, Huang Zhen, and Liu Chunnian

(Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, College of Computer Science and Technology, Beijing University of Technology, Beijing 100124)

Abstract Ant colony optimization (ACO) is a population-based metaheuristic technique to solve combination optimization problems effectively, such as traveling salesman problem (TSP), multidimensional knapsack problem (MKP), and so on. However, how to improve the performance of ACO algorithms is still an active research topic. Though there are many algorithms solving TSPs effectively, there is an application bottleneck that the ACO algorithm costs too much time in order to get an optimal solution. Combining the pheromone updating with an improvement of the stochastic search strategy, a fast ACO algorithm for solving TSPs is presented in this paper. Firstly, a new pheromone increment model called ant constant, which keeps energy conversation of ants, is introduced to embody the pheromone difference of different candidate paths. Meanwhile, a pheromone diffusion model, which is based on info fountain of a path, is established to reflect the strength field of the pheromone diffusion faithfully, and it strengthens the collaboration among ants. Finally, a simple mutation strategy with lower computational complexity is adopted to optimize the evolution result. Experimental results on different benchmark data sets show that the proposed algorithm can not only get better optimal solutions but also enhance greatly the convergence speed.

Key words traveling salesman problem; ant colony optimization; increment model; diffusion model; mutation strategy

摘 要 蚁群优化是一种元启发式的随机搜索技术,是目前解决组合优化问题最有效的工具之一。将信息素更新和随机搜索机制的改进相结合,提出一种快速求解旅行商问题的蚁群算法。首先给出了一种新的信息素增量模型,以体现蚂蚁在不同路径上行走时所产生的信息素差异;然后以蚂蚁经过的路径(直线段)作为信息素扩散浓度场的信源,改进了信息素扩散模型,强化了蚂蚁间的协作和交流;最后采用较低复杂度的变异策略对迭代的结果进行优化。在大量通用数据集上的实验表明,该算法不仅能获得更好的最优解,而且收敛速度有显著的提高。

关键词 旅行商问题;蚁群优化;增量模型;扩散模型;变异策略

中图法分类号 TP18

收稿日期:2007-12-10;修回日期:2008-11-12

基金项目:国家自然科学基金重大项目(60496322);北京市自然科学基金项目(4083034);北京市教育委员会科技发展基金项目(KM200610005020)

©1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

蚁群优化算法 ACO (ant colony optimization) 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^[1-3]. 经过十几年的发展, 蚁群算法已成为目前群集智能 (swarm intelligence) 中解决组合优化问题最有效的算法之一, 并在生产过程、车辆管理、路由寻址、布局规划、资源调配乃至数据挖掘等领域得到许多典型的应用^[4]. 然而, 迭代次数多、收敛速度慢仍是制约 ACO 算法广泛应用的主要瓶颈. 为此, 人们提出了许多种改进的蚁群算法^[5-11]. 概括地讲, 改进方法包含 2 种基本途径: 1) 信息素更新机制的改进, 即通过对信息素生成和更新策略的优化来提高蚁群算法的寻优能力, 改善解的全局收敛性. 比如, Gambardella 等人提出的 ACS (ant colony system) 蚁群系统^[5]对信息素进行局部和全局两次更新以提高全局收敛能力; Stutzle 等人提出的 MMAS (max-min ant system) 蚁群系统^[6]对路径上的信息素进行上下界的限制以克服解的停滞问题; 黄国锐等人提出的 PDACO 算法^[7]利用信息素扩散模型对信息素更新公式进行修正以增强蚂蚁间的通信. 2) 搜索机制的改进, 对随机搜索过程的控制策略进行调整, 以减少迭代次数, 加速收敛. 如吴庆洪等人提出一种具有变异特征的蚁群算法^[8], 通过变异策略以加快局部搜索; 吴斌等人提出的相遇算法^[9], 用 2 只蚂蚁共同构造一条解路径以加快构造解的过程; 丁建立等人提出的 GAAA 算法^[10], 将遗传算法与蚁群算法相融合也加快了求解速度. 文献 [11] 将这两种机制的改进相结合, 提出了一种基于变异和动态信息素更新的 NDMACO 算法, 该算法在时间性能上取得了较大的提高. 基于融合类似思想, 本文提出一种快速的求解旅行商问题的蚁群算法 PIPDMACO (ACO algorithm based on new pheromone increment and path diffusion models). 首先, 在分析已有信息素模型的基础上, 给出了一种新的信息素的增量模型, 以体现蚂蚁在不同路径上行走时所产生的信息量差异; 同时, 以蚂蚁经过的路径 (直线段) 作为信息素扩散浓度场的信源, 改善了信息素扩散模型, 强化了蚂蚁间的协作和交流; 最后采用较低复杂度的变异策略对迭代的结果进行优化, 以加快可行解的搜索. 在大量通用数据集上的实验表明: 该算法不仅能获得最优解, 而且收敛速度有显著的提高.

1 蚁群算法及其信息素模型

生物学家的研究发现, 自然界中的蚂蚁能够分

泌一种叫信息素 (pheromone) 的化学物质, 蚁群个体间信息的交流主要是通过信息素这种载体完成的. 蚁群算法就是通过模拟蚂蚁在觅食过程中进行信息交流的机制来求解实际问题的.

1.1 蚁群算法及其信息素增量模型

AS (ant system)^[12] 算法和 ACS (ant colony system)^[13] 算法是蚁群优化 ACO 算法中最基本的两种算法, 最初都被用于求解 TSP 问题. 这两种算法的主要区别在于城市的选择策略和信息素更新策略的不同. 下面以 ACS 算法为例介绍 ACO 算法. 设 m 是蚁群中蚂蚁的数量, a_{ij} 表示城市 i 与城市 j 之间的路径, $d_{ij} (i, j \in N = (1, 2, \dots, n))$ 表示相应的欧氏距离, $\tau_{ij}(t)$ 表示 t 时刻在 a_{ij} 上所留信息素的大小. 蚂蚁 $k (1 \leq k \leq m)$ 在周游的过程中, 将根据可行的候选路径上滞留的信息素随机地选择前进的方向. 在 ACS 算法中, 每只蚂蚁从城市 i 走向城市 j 的过程分两步进行: 首先通过式 (1) 进行随机采样和比较, 然后再结合式 (2) 确定状态变迁的规则.

$$j = \begin{cases} \arg \max_{j \in U_k} \{ [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta \}, & \text{if } q < q_0 (\text{利用}), \\ J, & \text{otherwise (开发)}, \end{cases} \quad (1)$$

其中, U_k 表示蚂蚁 k 在本次周游中在当前位置允许选择的的城市列表, η_{ij} 为路径 a_{ij} 的能见度 (启发信息), 一般取 $\eta_{ij} = 1/d_{ij}$; 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 是一个随机变量, 其值按式 (2) 计算得到的概率经随机采样来确定.

蚂蚁 k 从城市 i 走向城市 j 的转移概率为

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}{\sum_{j \in U_k} [\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}, & j \in U_k, \\ 0, & \text{其他}, \end{cases} \quad (2)$$

式 (2) 中 α, β 分别表示路径 a_{ij} 上的残留信息和启发信息对蚂蚁选择转移方向时的影响权重, ACS 算法中令 $\alpha = 1$.

在蚁群的每次周游中, ACS 算法要对信息素进行局部的和全局的两次更新. 首先在构造解时, 每只蚂蚁对其经过的路径用式 (3) 来进行信息素的局部更新:

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \rho \times \Delta \tau_{ij}^k, \quad \rho \in (0, 1), \quad (3)$$

式 (3) 中参数 ρ 表示信息素的挥发程度 ($1-\rho$ 可视为残留系数), $\Delta \tau_{ij}^k$ 表示在本次周游中蚂蚁 k 留在路径 a_{ij} 上的信息素增量 (可取常数 ω). 其次, 当每次

迭代所有的蚂蚁都完成本次周游后, ACS 算法要对本次周游中最优解所包含的每条路径进行信息素的全局更新:

$$\tau_{ij}^{nw} = (1 - \rho_1) \times \tau_{ij}^{old} + \rho_1 \times \Delta\tau_{ij}, \quad \rho_1 \in (0, 1), \tag{4}$$

式中 ρ_1 也是挥发系数, 而 $\Delta\tau_{ij}$ 表示这次迭代中最优解所含路径 a_{ij} 上的信息素增量, 为了例化信息素增量 $\Delta\tau_{ij}^k$ 和 $\Delta\tau_{ij}$, Dorigo 等人先后定义了蚁量 (ant-quantity)、蚁密 (ant-density) 和蚁周 (ant-cycle) 3 种信息素增量模型^[1-2], 如表 1 所示:

Table 1 Three Sorts of Increment Models of the Pheromone
表 1 三种信息素增量模型

$\Delta\tau_{ij}^k$	Ant-Quantity	Ant-Density	Ant-Cycle
If the k -th ant passes the a_{ij} in $[t, t+1]$ updating at the every step	Q/d_{ij}	Q	
If the k -th ant passes the a_{ij} in $[t, t+n]$ updating at the every tour			Q/L_k

表中 Q 为常数, L_k 为蚂蚁 k 在本次周游中所经过的旅行长度. 从表 1 可见, 蚁密模型的形式最简单, 其信息素增量是一个与其他信息无关的常量 Q ; 蚁量模型的信息素增量与所经过的路径长度成反比, 故该模型体现了解路径对信息素增量的影响, 解路径长则增量小, 解路径短则增量. 而在蚁周模型中, 信息素的增量则与蚂蚁本次周游的旅行长度成反比, 即该模型体现了可行解对信息素增量的影响, 周游的旅行长则增量小, 周游的旅行短则增量. 尽管这 3 种信息素增量模型的定义不同, 但都包含了这样的一个假设:

假设 1. 信息素分布假设. 随蚂蚁的行走信息素分布在所经过的路径上.

1.2 基于城市信源的信息素扩散模型

文献[7]在 AS 蚁群优化中提出了信息素扩散模型, 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. 其基本思想是在蚂蚁进行路径选择时, 适当地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所留下的信息素, 一方面会直接影响连接该路径的两个城市上的其他蚂蚁选择下一个城市的行为, 另一方面, 它会以这两个城市为中心向外扩散, 影响附近其他城市上蚂蚁的选路行为. 而且扩散范围内某点的信息素浓度与该点到信源之间距离呈反比关系, 即如果城市 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 所扩散来的信息素浓度 D_C , 其计算如式(5)所示:

$$D_C = D_{\max} \times ((h \times \text{tg}\theta - \sigma) / (h \times \text{tg}\theta)), \tag{5}$$

其中, 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, θ 为圆锥体锥面与中心轴的夹角, $h \times \text{tg}\theta$ 为扩散范围的半径 r , σ 为位于扩散范围中的城市 C 与信源 O 的距离.

假设蚂蚁 k 刚经过的两个城市 i 和 j 之间的距离为 d_{ij} , 那么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. 若有一城市 l 位于扩散范围内 ($d_{il} \leq r$ 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 $\Delta\tau_{ij}^k = Q/d_{ij}$, 而且也会影响相邻一些路径上信息素浓度的变化, 如路径 a_{il} 或 a_{jl} 上信息素浓度的变化可表示为 $\Delta\tau_{il}^k = D_{il}^k$, $\Delta\tau_{jl}^k = D_{jl}^k$. 令 $h = \bar{d}^{\omega+1} / (d_{ij})^\omega$, 其中 ω 为大于 1 的可调常数, \bar{d} 为城市间的平均距离, 则 $r = \text{tg}\theta \times \bar{d}^{\omega+1} / (d_{ij})^\omega$; 且以 i 为中心扩散时 $\sigma_i = d_{il}$, 以 j 为中心扩散时 $\sigma_j = d_{jl}$, 并设 $D_{\max} = \gamma \times \Delta\tau_{ij}^k$, γ 为小于 1 的可调常数, 则:

$$D_{il}^k = \begin{cases} \frac{\gamma \times Q}{d_{ij}} \left[1 - \frac{d_{il}}{r} \right], & \text{如果 } d_{il} \leq r, \\ 0, & \text{否则,} \end{cases} \tag{6}$$

$$D_{jl}^k = \begin{cases} \frac{\gamma \times Q}{d_{ij}} \left[1 - \frac{d_{jl}}{r} \right], & \text{如果 } d_{jl} \leq r, \\ 0, & \text{否则.} \end{cases} \tag{7}$$

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大小, 而且可能会改变相邻多条路径上的信息素, 这种改进提高了蚂蚁群中个体之间的合作效果, 更凸现了群集智能的思想.

2 基于信息素增量和扩散新模型的蚁群算法 PIPDMACO

2.1 新增量模型——蚁恒模型

能量守恒定律是自然界最普遍、最基本的规律, 能量守恒定律指出: 自然界的一切物质都具有能量,

能量既不能创造也不能消灭, 而只能从一种形式转换成另一种形式, 从一个物体传递到另一个物体, 在能量转换和传递过程中能量的总量恒定不变. 不失一般性, 在蚂蚁觅食的过程中, 我们基于能量守恒定律给出如下假设, 并定义蚁恒模型.

假设 2. 信息素能量转换假设. 蚂蚁分泌信息素需要消耗能量, 即信息素这种化学物质的产生是由蚂蚁的某种能量转换而来的.

根据上述假设, 蚁恒模型的定义如下:

定义 1. 蚁恒模型 (ant-constant). 如果把常数 Q 看做是每只蚂蚁从蚁穴出发觅食时具有的可以在一次周游中消耗的能量, 那么出发时蚂蚁具有全部的能量 Q , 随着蚂蚁的行走, 该能量被逐渐消耗, 并转化为等量的信息素留在相应的路径上. 依该思想定义的增量模型为

$$\Delta\tau_{ij}^k = \begin{cases} \frac{d_{ij}Q}{L_k}, & \text{当第 } k \text{ 只蚂蚁在本次周游中} \\ & (t \rightarrow t+n \text{ 时段}) \text{ 经过路径 } a_{ij} \text{ 时,} \\ 0, & \text{否则.} \end{cases} \quad (8)$$

该模型具有如下两个特点: 1) 当蚂蚁周游的路线 L_k 短时, 周游路线单位长度上所留下的信息素浓度就大, 所以, 即使是同一段路径, 其上信息素的浓度也会随周游行程的不同而有所不同. 这个特点反映了可行解的全局信息对信息素增量的影响. 2) 由于蚂蚁的一次周游所包含的各段路径的长度不尽相同, 故一只蚂蚁在经过长度不同的路径时所消耗的能量也各不相同. 在本次周游单位长度上信息素浓度确定的前提下 (Q/L_k), 蚂蚁经过长路径时所消耗的能量大, 故在该路径上所积累的信息素增量也大, 反之亦然. 这个特点反映了解路径的局部信息对信息素增量的影响. 可见, 蚁恒模型能体现在一次周游中蚂蚁所消耗能量 Q 在各段路径上的分配. 不难证明, 该模型具有能量守恒的性质:

$$\sum_{a_{ij} \in g_k} \Delta\tau_{ij}^k = \frac{\sum_{a_{ij} \in g_k} d_{ij}Q}{L_k} = Q \quad (9)$$

式中 g_k 为蚂蚁 k 这次旅行的连通图. 可见, 每只蚂蚁在各自周游中所产生的信息素总量为 Q , 服从能量守恒与转换定律. 经典的蚁周模型和蚁量模型分别从可行解、解路径两个方面考虑了路线长度对信息素增量的影响, 其实用效果各有利弊. 本文从能量转换的角度给出了新的蚁恒模型, 它可将可行解的整体信息和解路径的局部信息相结合, 共同完成对信

息素增量的调控, 更好地体现了蚂蚁在不同路径上行走时所产生的信息素的分布差异.

2.2 基于路径信源的信息素扩散模型

如上所述, 蚁群算法中信息素具有挥发性, 即信息素会随着时间的推移向周边路径扩散. 文献[7]中的信息素扩散模型就是考虑相邻城市间信息素的相互影响而建立的, 即基于相邻扩散浓度场之间存在的耦合作用, 对相近路径上的信息素进行一定的耦合补偿, 目的是更准确地更新每条路径上的信息素, 更客观地引导蚂蚁的选路行为. 该算法是以城市为信源向周边扩散的, 如果此时 $d_{ij} > 2r$, 则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 \circ 表示城市位置, 黑色圆点 \bullet 表示受扩散影响的城市, 以 i, j 为中心的同心圆表示扩散场内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.

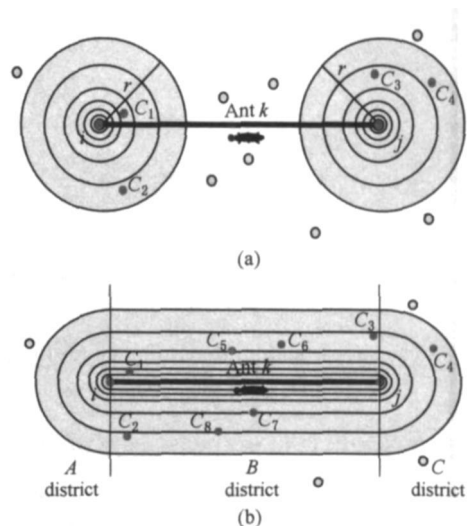


Fig. 1 The pheromone diffusion using a city as an info fountain. (a) The sketch map of the pheromone diffusion and (b) A reduction pheromone diffusion model.

图 1 信息素扩散浓度场的场强示意图. (a) 以城市为信源; (b) 以路径为信源

然而, 根据以上的信息素分布假设, 信息素是滞留在蚂蚁所经过的整条路径上的, 所以信息素应以所依附的路径 a_{ij} 为信源向外扩散. 可见, 图 1(a) 中的扩散模型存在一定的缺陷. 为此, 我们将原扩散模型修正为以路径为信源向周边扩散, 如图 1(b) 所示. 在扩散半径相同的情况下, 蚂蚁 k 在从城市 j 走到 i 的过程中所留下的信息素将形成更大范围的浓

度场, 影响的集合也扩大为 $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8$. 本文依据信息素的实际分布对扩散模型的信源进行合理的调整, 促使了信息素扩散浓度场的变化, 这种变化更客观地模拟了自然界中直线段信源扩散的场强变化, 并扩大了信息素扩散的作用范围.

假设蚂蚁 k 刚经过路径 a_{ij} , 下面我们基于路径为信源的信息素扩散模型, 针对欧氏距离描述的 TSP 问题, 推导蚂蚁 k 的此次行走所导致的相邻路径上信息素浓度的变化.

我们首先介绍判定一个城市位于扩散区域的方法. 给定一城市 l , 该城市到路径信源 a_{ij} 的垂直距离 d 可以分两种情况计算: 1) 当 TSP 问题中城市间的距离已知时, 这时 3 条边 a_{ij}, a_{jl}, a_{li} (对应的距离分别为 d_{ij}, d_{jl}, d_{li}) 构成一个三角形, 可利用海伦公式求出城市 l 到路径信源 a_{ij} 的垂直距离 d , 方法如下: 令 $s = (d_{jl} + d_{li} + d_{ij}) / 2$, 则该三角形的面积为 $\Delta = \sqrt{s(s-d_{ij})(s-d_{jl})(s-d_{li})}$, 又因为 $\Delta = (d \times d_{ij}) / 2$, 所以 $d = 2\Delta / d_{ij}$; 2) 当 TSP 问题中的城市位置用坐标给出时, 先利用距离公式求得 3 条边距离, 再采

用上面方法得到 d .

给定一个待求解的 TSP 问题, 采用文献[7]的计算方法就可计算出圆锥体的模型高度, 然后设定圆锥体锥面与中心轴的夹角 θ 后, 便可得到扩散半径 $r = h \times \tan \theta$. 当 $d > r$ 时, 城市 l 不在路径信源 a_{ij} 所形成的信息素扩散浓度场内, 否则, 城市 l 落在该信源形成的扩散浓度场内, 它所处的具体扩散区域 (如图 2 所示的灰色区域) 可利用余弦定理按下面方法来确定:

- 1) 当 $\cos \theta_1 = \frac{d_{li}^2 + d_{ij}^2 - d_{jl}^2}{2 \times d_{li} \times d_{ij}} \geq 0$, 且 $\cos \theta_2 = \frac{d_{jl}^2 + d_{ij}^2 - d_{li}^2}{2 \times d_{jl} \times d_{ij}} \geq 0$, 即当 θ_1 和 θ_2 均位于 $(0, \pi/2]$ 区间时, 城市 l 落在扩散浓度场的 B 区, 如图 2(b) 所示;
- 2) 当 $\cos \theta_1 < 0$, 而 $\cos \theta_2 > 0$, 即 θ_1 为钝角, θ_2 为锐角时, 城市 l 落在扩散浓度场的 A 区, 如图 2(a) 所示;
- 3) 当 $\cos \theta_2 < 0$, 而 $\cos \theta_1 > 0$, 即 θ_2 为钝角, θ_1 为锐角时, 城市 l 落在扩散浓度场的 C 区, 如图 2(c) 所示.

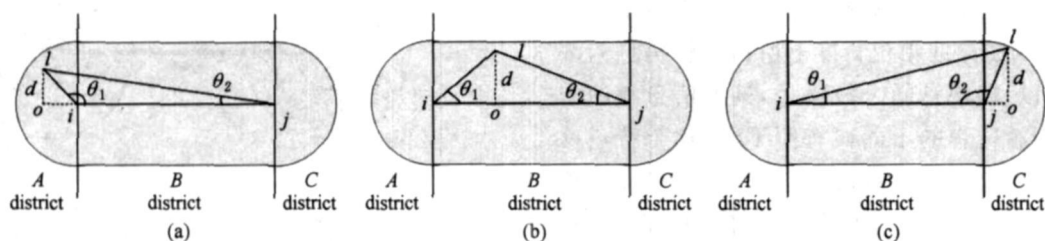


Fig. 2 The sketch map of determining a diffusion district for a city. (a) A district; (b) B district; and (c) C district.

图 2 判定城市位于不同扩散区域的示意图. (a) A 区; (b) B 区; (c) C 区

然后在各扩散区域内分别计算路径信源扩散到相应路径上的信息素:

1) 当城市 l 位于图 2(a) 中的 A 区时, 城市 i 对路径 a_{il} 的影响 D_{il}^k 可按式 (10) 计算得到:

$$D_{il}^k = \begin{cases} \frac{\gamma \times d_{ij} \times Q}{L_k} (1 - \frac{d_{li}}{r}), & \text{如果 } d_{li} \leq r, \\ 0, & \text{否则;} \end{cases} \quad (10)$$

2) 当城市 l 位于图 2(c) 中的 C 区时, 城市 j 对路径 a_{jl} 的影响 D_{jl}^k 可按式 (11) 计算得到:

$$D_{jl}^k = \begin{cases} \frac{\gamma \times d_{ij} \times Q}{L_k} (1 - \frac{d_{jl}}{r}), & \text{如果 } d_{jl} \leq r, \\ 0, & \text{否则;} \end{cases} \quad (11)$$

3) 当城市 l 位于图 2(b) 中的 B 区时, D_{il}^k 为直

线段 a_{ij} 对路径 a_{li} 和路径 a_{jl} 上信息素的影响, 其计算公式为

$$D_{a_{ij}l}^k = \begin{cases} \frac{\gamma \times d_{ij} \times Q}{L_k} (1 - \frac{d}{r}), & \text{如果 } d \leq r, \\ 0, & \text{否则.} \end{cases} \quad (12)$$

从上述计算过程可见, 无论是到城市还是到所行走的直线段, 只要与信源等距的点都受等势的场强浓度的影响, 这种扩散机制更能真实地再现自然界中直线段信源扩散的过程.

2.3 变异策略

为了提高蚁群算法的寻优能力, Dorigo 等学者在后来的论文中^[12-13]均提到使用局部优化的方法 (如 2-opt) 来改进原算法的性能. 而吴庆洪、朱庆保等人提出了利用变异算法局部寻优能力强的特点^[8, 11], 对所得到的解进行局部变异, 加快最优解的

获得. 本质上这两种方法是一致的, 它们都是对当前解的有限邻域进行快速搜索以得到本次迭代的更好解, 从而减少蚁群算法的迭代次数. 不同之处在于搜索空间及计算复杂度的差异, 2-opt 局部搜索空间是当前解的某一有限邻域, 其计算复杂度为 $O(C_n^2) = O(n^2)$, 而变异策略搜索的仅是该有限邻域的一个子域, 如文献[11]中变异算法的计算复杂度为 $O(n-2) + O(n-3) = O(n)$, 可见, 变异策略的计算复杂度更小, 搜索效率可能更高. 为此, 在本文算法中也采用类似的变异策略以加快算法的求解.

2.4 算法描述

综合以上几点, 本文研究了一种将信息素生成、扩散和变异策略相融合的快速求解 TSP 问题的新算法. 新算法的基本框架如下:

PIPDMACO 算法

```
{
  1) 初始化阶段
  初始化参数, 赋给各条路径相同的信息素浓度, 并将  $m$  只蚂蚁随机放到  $n$  个城市节点上;
  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 */
      { IF  $p < n$  THEN /* 没有遍历完所有城市 */
        { 按式(1)(2)选择下一个城市完成一步行走, 并变更允许访问的城市列表 }
        THEN /* 遍历完所有城市 */
          { 行走一步, 回到出发的城市 }
      }
  3) 本次迭代的解及其优化
  FOR  $k=1$  TO  $m$ 
    { 计算  $L_k$ ; /* 构造每只蚂蚁各自的解, 并计算旅行长度 */
    FOR  $q=2$  TO  $n$ 
      { 在解序列中, 将城市  $C_q$  与  $C_{q+1}$  交换, 并计算新的旅行长度  $L_z$ 
        IF ( $L_z < L_k$ ) THEN  $L_k = L_z$ , 发生变异并更新解序列;
      } /* 相邻城市的变异优化 */
    IF ( $q \geq n$ )
      { FOR  $q=2$  TO  $n-1$ 
        { 在解序列中, 将城市  $C_q$  与  $C_{q+2}$  交换, 并计算新的旅行长度  $L_z$ ;
          IF ( $L_z < L_k$ ) THEN  $L_k = L_z$ , 发生变异并更新解序列;
        }
      }
}
```

```
} /* 相间(隔一)城市的变异优化 */
}
4) 本次迭代的信息素的更新
FOR 每条边  $a_{ij} \in L_1 \cup L_2 \cup \dots \cup L_m$ 
  { 按式(8)计算路径  $ij$  上新产生的信息素增量;
    以  $a_{ij}$  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, 依式(10)(11)或(12)计算该信源扩散到相邻其他路径上的信息素增量;
    综合上面各式计算结果, 利用式(3)进行信息素的更新;
  } /* 信息素的局部更新 */
FOR  $k=1$  TO  $m$ 
  {  $L_{\text{best-iter}} = \min\{L_k\}$ ; /* 计算蚁群在本次迭代中得到的最短旅行长度 */
    IF ( $L_{\text{best-iter}} < L_{\text{best-all}}$ ) THEN  $L_{\text{best-all}} = L_{\text{best-iter}}$  /* 更新当前的最优解 */
  }
FOR 每个  $a_{ij} \in L_{\text{best-iter}}$ 
  { 利用式(4)进行全局信息素的更新 ( $\rho = \rho_1$ ); /* 全局信息素的更新 */
}
5) 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  $L_{\text{best-all}}$ ;
THEN
  { 迭代次数增 1; 初始化始点各城市可访问的城市列表;
    GOTO 2). /* 继续进行下次迭代 */
}
```

2.5 算法复杂度分析

本节按基本流程给出算法的计算复杂度分析. 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 $O(C_n^2) = O(n^2)$; 在蚁群周游的第 2 阶段, 在循环体内的计算花费包括城市转移: $O(C_{n-1}^1) = O(n)$, 所以计算复杂度为 $O(n \times m \times n) = O(m \times n^2)$; 在构造及优化的第 3 阶段, 两轮变异优化的花费为 $O(n-2) + O(n-3) = O(n)$, 故该阶段的计算复杂度为 $O(m \times n)$; 第 4 阶段完成信息素的更新, 局部更新循环体内的主要花费是判断其他城市是否落在本扩散浓度场内的计算: $3O(C_{n-2}^1) = O(n)$, 求最短旅行的复杂度为 $O(m)$, 全局更新的复杂度为 $O(n)$, 所以这阶段总的计算复杂度为 $O(m \times n \times n + m + n) = O(m \times n^2)$; 而第 5 阶段计算复杂度为 $O(1)$; 如果经过 NC 次迭代算法成功结束, 则总的计算复杂度为 $O(n^2) + NC \times (O(m \times n^2) + O(m \times$

$n) + O(m \times n^2) + O(1)) = O(NC \times m \times n^2)$, 与基本的ACO 蚁群算法^[3]具有相同的计算复杂度. 可见, 尽管在信息素扩散和解的变异优化中, 算法增加了一些计算量, 但并没有引起复杂度阶次的增加, 而且如果通过信息素更新和搜索机制的改进, 能减少迭代次数 NC 就有可能提高算法总体的计算性能.

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ACS 算法、改进的ACS 算法(包含 2-opt 优化)以及最新文献中的一些算法进行了比较. 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>). 实验的运行环境为: 操作系统 Windows XP, CPU 为 Pentium IV 1.8G Hz, 内存为 512MB, 算法用 Visual C++ 语言编程实现, 实验的参数均通过实验确定.

3.1 信息素增量模型的收敛性能比较

为比较不同信息素增量模型的收敛特性, 我们以 Oliver30 为例, 利用基本的 AS 算法, 在各种模型最佳的参数配置下(蚁量、蚁密、蚁周模型的最佳参数配置见文献[2], 蚁恒模型的参数取值为 $\alpha=1$, $\beta=6$, $\rho=0.2$, $Q=50$), 可以得到基于不同信息素增量模型的算法的收敛性能曲线. 图 3 用柱状图给出了各种算法在一些关键迭代次数上的收敛情况:

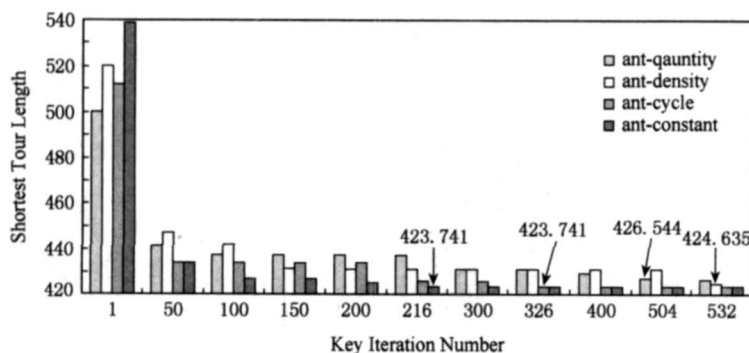


Fig. 3 The Comparison of the convergence performance on Oliver30 for different AS algorithms.

图3 不同 AS 算法在 Oliver30 上的收敛性能比较

实验发现: 基于蚁量模型的算法经过 504 次的迭代收敛到 426.544, 基于蚁密模型的算法经过 532 次的迭代收敛到 424.635, 基于蚁周模型的算法经过 326 次的迭代收敛到 423.741, 而基于我们提出的蚁恒模型的算法只经过 216 次的迭代便能收敛到 423.741. 可见, 在所有信息素增量模型中, 蚁恒模型最利于传递信息, 具有最好的收敛性能. 而在原有 3 种模型中, 蚁周模型的收敛效果最佳, 蚁密模型收敛最缓慢. 究其原因, 主要在于蚁密模型对蚂蚁选路缺乏引导信息, 而蚁量模型从解的局部信息(候选的解路径), 蚁周模型从解的整体信息(一个可行解)分别引入了蚂蚁选择较短路径的偏爱, 所以都有助于解的收敛. 不过, 正如文献[2]所指出的: 蚁量模型从局部出发, 容易陷入局部最优, 导致解的停滞现象, 故所得到的解最差. 本文提出的蚁恒模型由于融合了可行解的整体信息和解路径的局部信息, 所以既有效地避免了早熟现象, 又提高了算法的收敛速度.

3.2 信息素扩散模型的实验比较

为评价信息素更新机制和扩散模型对蚁群算法

优化结果的影响, 我们分别对基本 AS 算法(无扩散模型)、基本 ACS 算法(无扩散模型)、基于城市信源扩散模型^[7]的 AS 算法(AS-1)和本文提出的基于路径信源扩散模型的 AS 算法(AS-2)进行了大量的实验.

图 4 是不同算法在 dantzig42 问题的求解过程中得到的解的多样性曲线. 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. 从图中曲线的对比(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, 一方面, ACS 算法由于拥有随机采样的状态变迁过程, 增强了解的多样性, 所以多样性曲线优于 AS 算法; 另一方面, AS-1 算法、AS-2 算法的多样性性能也明显优于 AS 算法, 而且 AS-2 算法在多样性性能方面也略优于 AS-1 算法. 这说明, 扩散机制的引入能够增强蚁群算法不断获得最优解的能力, 而基于路径信源扩散模型的蚁群算法更容易克服最优解停滞的现象, 获得问题的全局最优解.

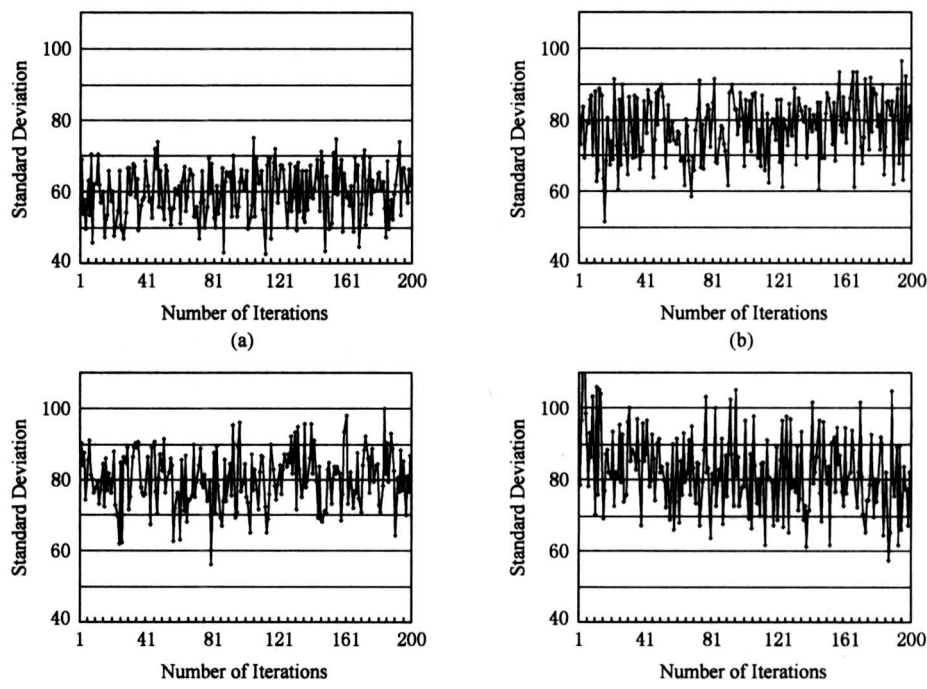


Fig. 4 The diversity curves of different AS algorithms on dantzig42. (a) The diversity curves of AS; (b) The diversity curves of ACS; (c) The diversity curves of AS-1; and (d) The diversity curves of AS-2.

图 4 不同算法在 dantzig42 问题上的多样性曲线. (a) AS 算法得到的多样性曲线; (b) ACS 算法得到的多样性曲线; (c) AS-1 算法得到的多样性曲线; (d) AS-2 算法得到的多样性曲线

图 5 为不同 ACO 算法在各自最佳的参数配置下得到的 dantzig42 问题的解图, 其中 (a) 为基本 AS

算法经过 1500 次迭代得到的最优解 714 的解图, (b) 为基本 ACS 算法经过 1300 次迭代得到的最优

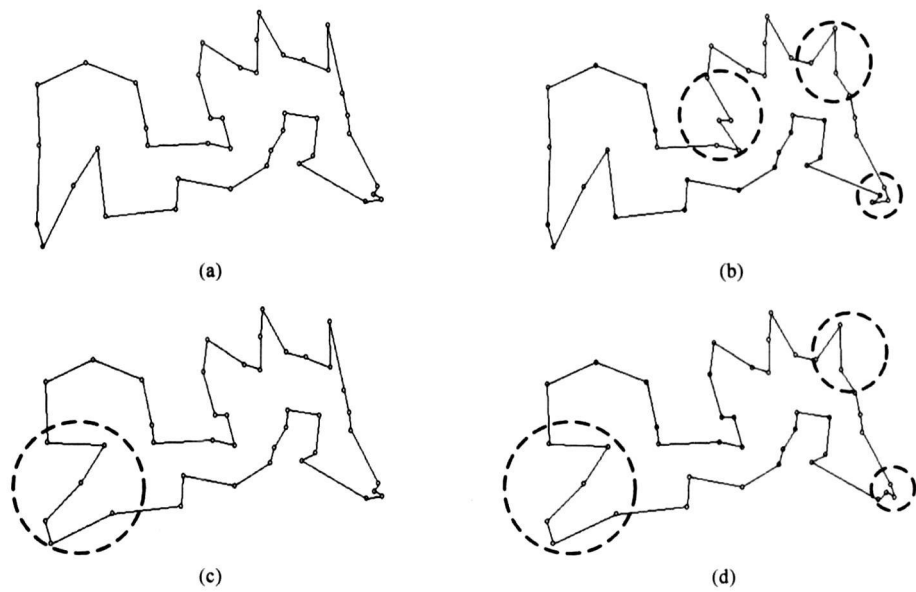


Fig. 5 The best tours obtained by different AS algorithms on dantzig42. (a) The best tour obtained by AS; (b) The best tour obtained by ACS; (c) The best tour obtained by AS-1; and (d) The best tour obtained by AS-2.

图 5 不同算法在 dantzig42 问题上的最优解图. (a) AS 算法得到的最优解图; (b) ACS 算法得到的最优解图; (c) AS-1 算法得到的最优解图; (d) AS-2 算法得到的最优解图

解 707 的解图, (c) 为 AS-1 算法经过 180 次迭代得到的最优解 712 的解图, 而(d) 为 AS-2 算法经过 85 次迭代得到的最优解 699 的解图. 从图 5 可见, 由于信息素更新和扩散机制的不同, 使得 ACO 算法在局部连接上产生了差异(虚线圆框内的连接部分), 最后导致了所得到的最优解的不同. 与基本 AS 算法相比, ACS 算法由于采用了局部和全局两次信息素的更新, 强化了正反馈的过程, 因此解的质量得到了提高. 而扩散模型的引入同样也能提高解的质量, 并且由于以路径为信源的扩散模型更能强化蚁群的信息交流, 故比基于城市信源的扩散模型更加有效.

综合解的质量和多样性性能的比较, 不难发现 ACS 算法的运行机制和本文提出的路径信源扩散模型都有利于蚁群的优化. 为获得更好的结果, 本文基于 ACS 算法框架, 并引入蚁恒增量模型、路径信源扩散模型和变异策略来实现新的算法.

3.3 算法整体性能的实验比较

为验证、比较算法效果, 我们在不同的 TSP 问

题上对基本 ACS 算法^[3]、改进的 ACS 算法(+ 2-opt 局部优化)、PDACO 算法^[7]、本文算法(PIPDMACO)进行了大量实验. 为方便与最新文献的比较, 我们选择了几种与文献[11] 相同的 TSP 问题来进行求解. 表 2 为几种不同算法在各自最佳的参数配置下得到的一些 TSP 问题的最佳结果, 其中, 收敛代数是指得到表中结果时蚁群周游的次数. 从表中可知, 与原 ACS 相比, 2-opt 局部优化能够改善解的精度, 并提高收敛速度(约加快 1 ~ 4 倍); PDACO 算法在收敛速度方面能得到较大提高(5 ~ 10 倍), 但除文献[7] 给出 Oliver30 事例外, 其他事例上精度欠佳; 而本文算法在精度和收敛代数方面都得到了显著的提高: 在所有列出的这些问题上都得到了标准数据库中所提供的最优解(而在 pr107 上也得到目前文献中所能得到的最好解 44302), 收敛代数与原 ACS 相比降低了数百倍, 与改进的 ACS 算法相比也降低数十倍.

Table2 The Results Comparison of Different Algorithms on Various TSP Problems
表 2 一些 TSP 问题上不同算法的结果比较

Public Data Sets		ACS		ACS+ 2opt		PDACO		PIPDMACO	
Instances of TSP	Optimum Values	Best Result	Num. of Cycles	Best Result	Num. of Cycles	Best Result	Num. of Cycles	Best Result	Num. of Cycles
Oliver30	423. 741	423. 74	830	423. 74	210	423. 74	76	423. 74	9
dantzig42	699	707	1300	700	450	712	180	699	8
eil51	426	441	2100	431	900	445	290	426	10
berlin52	7542	7598	2500	7573	1200	7664	320	7542	13
st70	675	690	3900	681	2200	696	550	675	28
pr107	44303	44401	4700	44352	2600	44619	850	44302	47
tsp225	3919	3989	7600	3951	4500	3997	1400	3919	160

而且, 本文算法与目前我们所了解的最佳实验结果^[11] 相比, 在 pr107 问题上收敛代数更少, 而在 eil51, berlin52, st70 等问题上, 收敛代数则大致相当. 另外, 在 dantzig42 问题上仅需要 8 次就可得到最佳值, 而对于有 225 个城市的较大规模的 TSP 问题也仅需 160 次迭代就可找到最优解, 可见, 本文算法是一种有效的快速求解 TSP 问题的蚁群算法.

表 3 是本文算法在不同 TSP 范例上在获得表 2 结果时所使用的参数配置, 其他算法的参数配置方法与文献[3, 7] 一致. 在所有范例的实验中, 另外几个相同取值的参数为 $m = n$, 信息素浓度初值 = 3, $\theta = \pi/6$. 实验发现, 当 $\alpha < \beta$ 时, 本文算法能够获得更为理想的求解效率. 这主要是因为本文算法采用的信息素生成和扩散机制已经强化了各条路径上残留

的信息素的影响作用, 所以在蚁群寻优的过程中要增强路径启发信息的作用, 这样才能均衡两者在蚂蚁选路过程中的影响, 提高算法的收敛能力.

Table3 Parameter Setting for Our Algorithm on Various TSP Problems

表 3 本文算法在一些 TSP 问题上的参数设置

Instance of TSPs	α	β	ρ	γ	q_0	Q	w
Oliver30	2	4	0. 3	0. 4	0. 7	50	1
dantzig42	1	5	0. 3	0. 4	0. 7	150	1
eil51	1	5	0. 3	0. 4	0. 7	200	1
berlin52	2. 5	4	0. 2	0. 4	0. 6	100	1
st70	2. 5	5	0. 3	0. 4	0. 7	100	1
pr107	2	4. 5	0. 3	0. 4	0. 7	50	1
tsp225	2	5	0. 3	0. 4	0. 95	100	1

4 结 论

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.本文将信息素更新机制和蚁群随机搜索机制的改进相融合,提出一种新的快速求解旅行商问题的蚁群算法 PIPDMACO.首先,在分析已有信息素模型的基础上,给出了一种新的信息素的增量模型——蚁恒模型,以体现蚂蚁在不同路径上行走时所产生的信息量差异;同时,以蚂蚁经过的路径(直线段)作为信息素扩散浓度场的信源,改进了已有的信息素扩散模型,强化了蚂蚁间的协作和交流,提高了求解精度;最后采用较低复杂度的变异策略对迭代的结果进行优化,以加快可行解的搜索.在大量通用 TSP 数据集上的实验表明:该算法不仅能获得最优解,而且收敛速度有显著的提高.

本文工作是对蚁群算法快速收敛问题研究的一个新成果,为中小规模 TSP 优化问题的求解提供了一个有效的方法.另外,在大规模 TSP 问题(数千城市)上的实验表明,本文算法收敛速度较其他算法有很大提高,且求解精度也有所改进,但离最优解还有一定的距离,如何在有限的时间内大幅度提高大规模 TSP 问题求解精度,将是下一步研究的主要任务.

参 考 文 献

- [1] Dorigo M, Colnani A, Maniezzo V. Distributed optimization by ant colonies [C] //Proc of the 1st European Conf of Artificial Life. Paris: Elsevier, 1991: 134-142
- [2] Dorigo M, Maniezzo V, Colnani A. The ant system: Optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics, Part B, 1996, 26 (1): 29-41
- [3] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66
- [4] Duan Haibin. Ant Colony Algorithms: Theory and Applications [M]. Beijing: Science Press, 2005 (in Chinese) (段海滨. 蚁群算法原理及其应用 [M]. 北京: 科学出版社, 2005)
- [5] Gambardella L M, Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies [C] //Proc of the Int Conf on Evolutionary Computation. Piscataway, NJ: IEEE, 1996: 622-627
- [6] Stutzle T, Hoos HH. MAX-MIN ant system and local search for the traveling salesman problem [C] //Proc of the IEEE Int Conf on Evolutionary Computation. Piscataway, NJ: IEEE, 1997: 309-314

- [7] Huang Guorui, Cao Xianbin, Wang Xifa. An ant colony optimization algorithm based on pheromone diffusion [J]. Acta Electronica Sinica, 2004, 32(5): 865-868 (in Chinese) (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)
- [8] Wu Qinghong, Zhang Jihui, Xu Xinhe. An ant colony algorithm with mutation features [J]. Journal of Computer Research and Development, 1999, 36(10): 1240-1245 (in Chinese) (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(10): 1240-1245)
- [9] Wu Bin, Shi Zhongzhi. An ant colony optimization algorithm based partition algorithm for TSP [J]. Chinese Journal of Computers, 2001, 24(13): 1328-1333 (in Chinese) (吴斌, 史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法 [J]. 计算机学报, 2001, 24(13): 1328-1333)
- [10] Ding Jianli, Chen Zhengqing, Yuan Zhuzhi. On the combination of genetic algorithm and ant algorithm [J]. Journal of Computer Research and Development, 2003, 40 (9): 1351-1356 (in Chinese) (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)
- [11] 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 (in Chinese) (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)
- [12] Blum C, Dorigo M. The hyper-cube framework for ant colony optimization [J]. IEEE Trans on Systems, Man, and Cybernetics, 2004, 34(2): 1161-1172
- [13] Dorigo M, Birattari M, Stützle T. Ant colony optimization: Artificial ants as a computational intelligence technique [J]. IEEE Computational Intelligence magazine, 2006, 11: 28-39



Ji Junzhong, born in 1969. Ph. D., professor. Senior member of China Computer Federation. His main research interests include machine learning, Web intelligence and computation intelligence.

冀俊忠, 1969 年生, 博士, 教授, 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。



Huang Zhen, born in 1981. Master. His main research interests include data mining and Web intelligence.

黄 振, 1981 年生, 硕士, 主要研究方向为数据挖掘、Web 智能。



Liu Chunnian, born in 1944. Professor and Ph. D. supervisor. Received his B. S. degree in mathematics from Beijing University, his M. S. degree in computer science from Beijing University of

Technology, and Ph. D. degree in software engineering from Norwegian University of Science and Technology. His main research interests include data mining, inductive logic programming, constraint programming, machine learning and soft computing.

刘椿年, 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.

Research Background

Ant colony optimization (ACO) algorithm is a metaheuristic and stochastic search technology, which has been one of the effective tools for solving discrete optimization problems. However, there is an application bottleneck that the ACO algorithm costs too much time in order to get an optimal solution. Therefore, the authors propose a fast ACO algorithm for the traveling salesman problem (TSP). There are three major contributions. Firstly, by analyzing the pheromone increment model proposed by researchers, a new pheromone increment model called ant constant is presented. Secondly, based on info fountain of a path, a pheromone diffusion model is established to reflect the strength field of the pheromone diffusion faithfully, and is strengthens the collaboration among ants. Finally, a simple mutation strategy with lower computational complexity is adopted to optimize the evolution result. Compared with some recent algorithms, the proposed algorithm has the best convergence performance while getting better solutions. Our work is supported by the Major Research Program of National Natural Science Foundation of China (No. 60496322), the Beijing Natural Science Foundation (No. 4083034), and the Scientific Research Common Program of Beijing Municipal Commission of Education (No. KM200610005020).