

A New Ant Colony Optimization with Global Exploring Capability and Rapid Convergence

Xiang-yang Deng¹, Wen-long Yu¹, Li-min Zhang²

¹⁾ Department of Electric and Information Engineering, ²⁾ Department of Scientific Research,
Naval Aeronautical and Astronautical University, Yantai, Shangdong, China (xavior2012@yahoo.cn)

Abstract—ant colony optimization (ACO) is a meta-heuristic algorithm, and is widely applied in combinatorial optimization. To enhance the ACO's global exploiting capability and convergence, a new pheromone update strategy is presented, which results in a gradually transition of the ant colony's diversity, and an improved ACO algorithm called ACO+ is proposed. For a solution to the traveling salesman problem (TSP), a statistical model of traversed ants of sub-routes is introduced to rank the sub-routes, and an adaptive pheromone trails update mechanism is implemented, which integrates with the iteration-best pheromone update strategy. The algorithm can effectively combine the global exploring capability and convergence rate. Experiments show that the ACO+ has a good performance and robustness.

Keywords—ant colony optimization, discrete combinatorial optimization, travel salesman problem, meta-heuristic algorithm, pheromone trails

一种具有全局搜索能力和快速收敛的蚁群算法

邓向阳¹ 于文龙¹ 张立民²

¹⁾ 电子信息工程系, ²⁾ 科研部, 海军航空工程学院, 烟台, 山东, 中国

摘要 蚁群算法是一种现代启发式算法, 在求解组合优化问题中得到了广泛应用。本文针对蚁群算法收敛速度慢、容易陷入局部最优的缺点, 设计了一种新的信息素更新策略, 实现了种群多样性的平滑过渡, 提出了一种改进蚁群算法。论文以 TSP 问题为研究背景, 研究了一种子路径的蚂蚁经过数统计模型, 并以此为基础对子路径排序进行信息素更新, 结合本轮最优路径信息素更新策略, 实现了一种自适应信息素更新机制, 使算法在全局寻优和收敛速度二者上得到了有效的综合。实验表明该算法具有很好的求解能力和鲁棒性。

关键词 蚁群算法, 离散组合优化, 旅行商问题, 现代启发式算法, 信息素结构

1. 引言

在 1991 年的第一届欧洲人工生命会议上[1], 意大利学者 Colomi A, Dorigo M 和 Maniezzo V 首次提出了蚁群算法 (Ant Colony Optimization, ACO), 算法受自然界中的蚂蚁觅食行为的启发而来, 通过建立蚂蚁的个体选择建模和蚂蚁之间协商的信息素模型, 实现了一种群体智能涌现的机制, 用来解决各种优化问题, 尤其是在现阶段无法找到多项式时间算法的 NP-Complete 问题。旅行商问题 (Traveling Salesman Problem, TSP) 是一种经典的离散组合优化问题, 其模型简单, 易于理解, 且应用广泛, Dorigo M 等分别在文献[1, 2, 3]中提出了三种蚁群系统 (Ant-Density, Ant-Quality, Ant-Cycle) 模型用于求解 TSP 问题, 后来又在 1995 年提出了 Ant-Q 系统[4], 并在 1996

年的 ACS 系统[5]一文中对蚁群算法进行了全面细致的分析, 使蚁群算法成为解决 TSP 问题的经典算法。

蚁群算法具有系统性、自组织性、分布式计算、正反馈等特点, 因此自提出以来, 国内外大量学者对其进行了改进, 针对 TSP 问题的求解提出了多种模型, 如文献[6]提出了一种能够避免早熟收敛的现象发生的最大-最小蚂蚁系统, 文献[7]提出了一种基于个体排序的蚂蚁系统, 文献[8]采用去交叉策略, 实现了一种进行局部优化的蚁群算法, 文献[9]设计了一种随机选择策略和扰动策略, 提高了蚁群算法的全局搜索能力, 文献[10]讨论了一种带时间窗约束的 TSP 问题, 并提出了求解该问题的 ACS-TSPTW 算法, 文献[11]讨论了泛化 TSP 的解法, 提出了结合突变过程和本地搜索的 ACO 方法等。目前, 大部分算法主要围绕建立

不同的解结构、信息素更新策略和蚁群的初始状态等方面进行改进,以达到提高算法的全局搜索能力和收敛速度的目的,提高算法效率。但是,因为全局搜索和收敛速度在本质上是矛盾的双方,很难有很好的办法同时兼顾两种特性;还有一些算法是针对特定环境提出的,与问题本身特性的建模具有很大的关联,在应用到不同领域时,其初始环境和优化参数的设置不能通用,在普适性上不具有很好的性质。

本文将城市之间的连线定义为子路径,以分离的观点探讨了组成解序列的各个子路径在算法中的独立作用,将适应度最高的解序列和访问频度高的子路径结合考虑,建立了一种组合的多元信息素更新机制,并将子路径根据蚂蚁通过数进行排序,引入参与信息素更新的子路径数的自适应机制,使算法初期的全局探索能力和后期的稳定收敛能力得到了有效的结合,实现了一种改进蚁群算法。

2. 求解 TSP 问题的基本算法模型

蚁群系统(Ant Colony System, ACS)是由 Dorigo 和 Gambardella 提出的[5],它对早前的蚂蚁系统做了许多改进,成为后来广为研究的各类蚁群算法的基本模型。本文算法的求解思路为:初始将 m 只蚂蚁置于 n 个城市中的出发城市,每一只蚂蚁通过伪随机规则不断地选择下一个城市,每选中一个节点就在当前节点和选中节点之间建立了一条边,蚂蚁通过局部更新规则更新该边的信息素,直到创建一条完整的合法路径,当所有的蚂蚁都完成搜索之后,再对当前最佳路径上的边进行信息素更新。

模型的主要算法步骤为:

- 1) 初始化蚁群;
 - 2) 蚂蚁利用伪随机比例规则选择子路径,并利用局部信息素更新规则更新信息素;
 - 3) 重复构造子路径,直至形成完整解决方案;
 - 4) 根据目标函数计算蚂蚁的适应度得出最优路径;
 - 5) 根据全局信息素更新规则更新信息素;
 - 6) 根据算法终止条件完成或转 2);
 - 7) 结束。
- 蚂蚁 $k(1, \dots, m)$ 在构造子路径过程中,根据以下规则选择城市:

$$j = \begin{cases} \arg \max_{l \in J_k(i)} [(\tau_{il})^\alpha \times (\eta_{il})^\beta], & \text{if } q \leq q_0 \\ \text{according to formula (2),} & \text{otherwise} \end{cases} \quad (1)$$

其中 q 是在 $[0,1]$ 之间均匀分布的随机数, q_0 为算法给定概率。

算法根据各条路径上的信息量及路径的启发信息来计

算状态转移概率。 P_{ij}^k 表示蚂蚁 k 由城市 i 转移到城市 j 的状态转移概率为:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \times \eta_{ij}^\beta}{\sum_{j \in allowed} [\tau_{ij}^\alpha \times \eta_{ij}^\beta]}, & \text{if } j \in allowed \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

式中 α 为信息启发式因子,表示轨迹的相对重要性, β 为期望启发式因子,表示能见度的相对重要性, τ_{ij} 为弧 (i, j) 上的信息素密度, η_{ij} 为弧 (i, j) 上的启发函数的值,

一般情况下等于 $\frac{1}{d_{ij}}$, d_{ij} 表示弧 (i, j) 的长度。

在每只蚂蚁在完成对所有城市的遍历后,根据如下规则对信息素轨迹进行更新处理:

$$\begin{cases} \tau_{ij} = (1 - \rho) \times \tau_{ij} + \rho \times \Delta \tau_{ij} \\ \Delta \tau_{ij} = \frac{1}{L_{iterbest}} + \zeta \end{cases} \quad (3)$$

式中 ρ 表示信息素挥发系数, $\Delta \tau_{ij}$ 表示本次循环中路径 (i, j) 上的信息素增量, $L_{iterbest}$ 是本轮最佳路径的长度, ζ 为排序后的子路径信息素增量。

3. 基于子路径排序的信息素更新策略

蚁群算法在求解 TSP 问题过程中,通过不断的解序列的进化最后达到一定条件结束,这种进化主要通过同代蚁群之间的竞争和不同代蚁群之间的信息传承实现,同代蚁群的竞争主要采取了基于适应度的优胜劣汰机制,而不同代之间的信息传承则是因为采取了信息素的保持机制。信息素保持机制使得先代蚂蚁的经验可以指导后代蚂蚁的行为,使后代蚂蚁可以最大可能的沿着先代蚂蚁的路线寻求解决方案,避免盲目性,为了能够使算法尽快的得到一个稳定的解,往往需要加强这种相继多代蚁群之间的影响,这个过程需要通过子路径上的信息素量的差异才能够实现,而这种差异的加大又使蚁群远离当前最优解去搜索位置空间的可能性降低,因此,同时实现探索性和收敛性两个目标是矛盾的。

为了实现探索性和收敛性的有效折中,一般需要两种信息素的结合,以实现对信息素布局的有效控制,如 ACS 系统中的局部更新策略。本文通过考察子路径在形成最优路径中的独立作用,在每次迭代之后,采用迭代最优和排序较优子路径相结合的方式更新信息素。某代蚁群构建完所有路径后,在本次迭代的最优路径的子路径上更新信息素,同时,对经过蚂蚁数量比较多的子路径片段也释放信

息素, 从而影响下代蚁群的随机搜索行为, 建立了基于子路径排序的信息素更新方法, 据此建立的蚁群算法简称为ACO+。

3.1 子路径排序与信息素叠加

在蚁群寻优过程中, 不同的蚂蚁虽然形成解序列不同, 但是相互之间总会经过某些相同子路径, 从蚁群的整体认知来看, 经过子路径的蚂蚁数越多, 认为该子路径在构建全局最优路径中起着更加重要的作用。因此, 在某代蚁群搜索中, 当所有蚂蚁构造完解序列之后, 统计每条子路径经过的蚂蚁数, 并生成一种新的信息素增量, 可以用来调控信息素的布局。

设蚂蚁总数为 m , 子路径 r_{ij} 上 (i, j 为城市序号) 经过的蚂蚁数为 $m_{r_{ij}}$, 将各子路径基于 r_{ij} 进行排序, 取前 l 条子路径进行信息素更新, 更新的信息素量按下式计算:

$$\zeta = \begin{cases} Q, & \text{if } \text{rank} \leq l \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

该信息素增量的计算流程如下:

- 1) 初始化各子路径的经过数计数器矩阵元素为 0;
- 2) 在子路径矩阵中取出一条子路径;
- 3) 遍历所有蚁群个体, 如果该个体经过该子路径, 则子路径的计数器加 1, 如果子路径剩余数不为 0, 则转向第 2) 步;
- 4) 在得到的计数器矩阵中搜索最大数;
- 5) 更新该子路径信息素, 将该计数器清零, 如果未完成 l 条子路径, 转向第 4) 步;
- 6) 过程结束。

基于子路径排序的信息素更新流程图如图 1 所示。

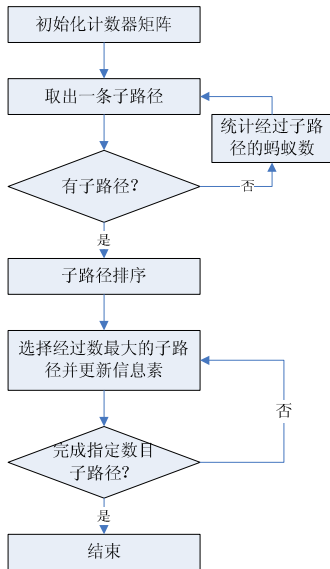


图 1 基于子路径排序的信息素更新流程图

蚁群搜索过程中, 在经过蚂蚁数量比较多的子路径上释放信息素, 会影响下代蚁群的随机搜索行为, 使远离当前最优路径的空间中的子路径也对蚁群的选择过程产生了影响, 达到了平衡最优解子路径与其他子路径之间概率差异的目的。

下图是算法叠加基于排序子路径的信息素更新策略前后的第 2 代的信息素分布示意图。

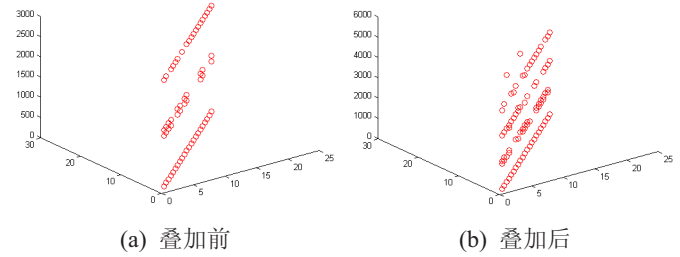


图 2 叠加子路径信息素前后比较图

从图中可以看出, 图 2(a)的信息素分布比较规则, 基本上在三层区域上, 比较均匀, 说明信息素集中在已找到的最优路径附近; 图 2(b)信息素分布相对较散乱, 说明在算法初期通过基于排序子路径的信息素, 使搜索的空间变大了, 提高了全局搜索能力。

但是随着解的不断进化, 蚁群访问的路径都会集中到最优路径附近, 使得基于蚂蚁经过数排序的子路径也集中到最优路径上, 此时, 信息素叠加机制将对随机搜索起到很大的限制作用, 不利于蚁群在局部范围内继续探索最优解, 因此, 需要对其进行进一步的改进。

3.2 自适应信息素更新机制

为了消除上述的信息素叠加机制在算法后期的不利影响, 可以通过不断减小叠加的信息素量得以实现, 但是, 应该注意到在算法后期叠加的信息素一方面会限制随机搜索的发生, 同时, 它也对算法收敛速度的提高和得到稳定解的过程具有正向作用, 因此, 不能完全将叠加的信息素去掉。

在蚁群算法中, 信息素分布是种群有效进化的外在因素, 基于子路径排序的信息素的主要功能是微调整体信息素布局, 因此本文建立自适应机制针对该信息素进行调控。在算法初期, 通过多数的子路径参与信息素叠加, 可以得到更强的全局搜索能力, 使得解空间的信息素均衡分布, 让搜索趋于随机性; 在算法后期, 通过少量的子路径参与信息素叠加, 可以提高算法收敛速度。因此, 设计 l 的变化关系如下所示:

$$l = a \times n - \text{iteration} \quad (5)$$

l 为该次迭代中参与信息素叠加的子路径条数, a 为参

与计算的子路径数的比例因子, n 为城市总数, $iteration$ 为迭代的次数。为了保持基于排序得到的子路径信息素在后期的作用, 本文设定了 l 的最小值。

通过自适应的信息素更新方式, 可以达到算法在前后两个阶段的信息素分布的平稳过渡, 使通过子路径叠加的信息素在算法初期利于全局搜索, 在算法后期影响越来越小, 达到帮助算法小范围收敛的作用。

4. 仿真实例

为了检验 ACO+蚁群算法的各种性能, 在如下的实验环境中对 ACO+和 ACS 系统进行了比较分析: MatLab 7.8.0 (R2009a), 运行在 Windows XP 环境下, HP540 笔记本, CPU T5470 1.6GHz, 内存 1G。

ACS 系统在本文的环境中重新实现, 主要参数包括:

1) 采用最近邻搜索算法构建初始路径长度 l_0 ;

2) 初始信息素设为 $\tau_0 = \frac{1}{n \times l_0}$;

3) 采用基于历史最优值的全局更新策略。

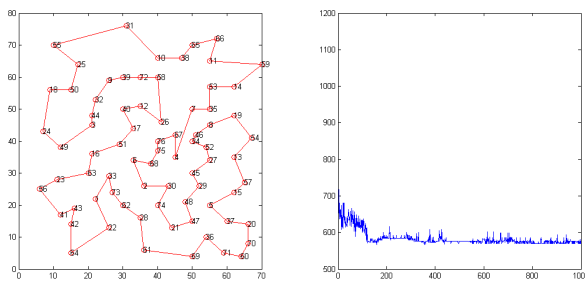
ACO+的参数设置为: $\alpha = 2$, $\beta = 1$, $\rho = 0.1$, $a = 2$,

蚂蚁数 $m = 20$, $q_0 = 0.4$, $Q = \frac{1}{L_{iterbest}}$, $iteration = 500$, τ_0

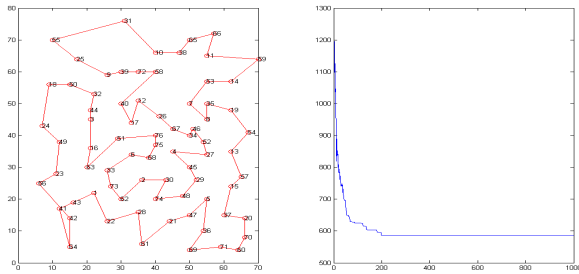
设置与 ACS 系统相同。

1) 与标准 ACS 算法的比较测试

实验中算法迭代 1000 次, 实验 20 次, 采用的城市坐标数据为 *eil76.tsp*, 本文算法 ACO+与 ACS 的实验最好结果如图 4 所示。



(a) ACO+算法



(b) ACS算法

图4 两种算法的测试结果

从图中可看出, ACO+算法能够收敛到一个相对较优的值, 其最优解为 557, 而 ACS 得到的最优值为 581。并且, ACO+算法在第 110 次循环左右达到收敛, 后期还能保持一定的寻优特性, 局部优化性能好。

2) 不同值 a 和不同 l_{\min} 下的测试

该实验中算法迭代 210 次, 采用 TSPLIB 中的 *ulysses22.tsp* 数据进行测试, a 取 5、10 和 15 共 3 种情况, num 最小值取 10、20 和 30 共 3 种情况, 得到的最优值变化结果如图 3 所示。

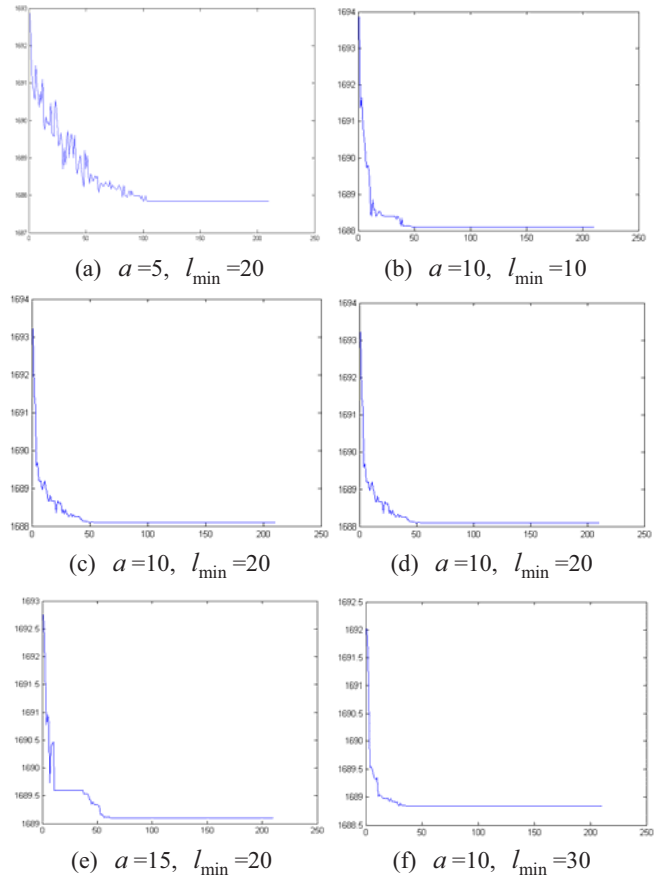


图3 不同自适应参数实验

从图中看出, 在子图(a)、(c)、(e)中, 当选定最低子路径数 l_{\min} 为 20, 随着 a 值增大, 在算法初期的收敛速度加快了, 但是会容易陷入局部最优, 这是因为随着 a 值增大, 参与信息素均衡的子路径数目增多, 使得算法初期的信息素在整个解空间泛滥, 容易使一些次优路径上的信息素增多, 进而快速收敛到局部解上。

在子图(b)、(d)、(f)中, 如果选定 a 值为 10, 随着最低子路径数 l_{\min} 的增加, 算法性能没有太大的影响。

5. 总结

本文总结了求解 TSP 问题的经典蚁群算法及各种改进

方案,并分析了蚁群算法容易出现早熟收敛的原因,深入探讨了信息素的不同结构对算法的影响,并对算法初期和后期的信息素布局特性与算法的全局寻优能力和收敛性能的关系进行了分析,通过引入单个子路径在最优解序列构造过程中的作用,建立了子路径蚂蚁经过数的排序模型,基于该模型实现了一种全局多元信息素更新机制,并通过建立自适应的信息素更新机制使算法获得了全局搜索能力和后期快速收敛的折中方案。通过标准 TSPLIB 测试平台的实验和与 ACS 系统的比较分析,验证了算法的可行性和鲁棒性。

但是,算法没有对自适应调节机制建立更加精细的模型,没有进行 TSPLIB 中的大型问题的测试,因此,其稳定性和适应性还需要做更多的实验进行验证。

参考文献

- [1] Colormi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life. Paris, 1991, 134-142.
- [2] Dorigo M, Maniezzo V, Colormi A. The Ant System: An autocatalytic optimizing process. Technical report 91-016 revised, Dipartimento di Elettronica, Politecnico di Milano, Milan.
- [3] Dorigo M. Optimizing, Learning and Natural Algorithms [in Italian]. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan.
- [4] Gambardella L M, Dorigo M. Ant-Q: a reinforcement learning approach to the traveling salesman problem. Proceedings of the 12th International Conference on Machine Learning. 1995:252-260.
- [5] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation. 1997 1(1):53-66.
- [6] Stützle T, Hoos H. H. The Max-Min Ant System and local search for the traveling salesman problem. In T. Back, Z. Michalewicz, & X. Yao (Eds.), Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97) (pp 309-314).
- [7] Bullnheimer B, Hartl R F, Strauss. A new rank-based version of the ant system: a computational study. Central European Journal of Operations Research and Economics, 1999, 7(1): 25-38.
- [8] 黄岚,王康平,周春光等. 基于蚂蚁算法的混合方法求解旅行商问题. 吉林大学学报, 2002, 40(4): 369-373.
- [9] 郝晋,石立宝. 求解复杂 TSP 问题的随机扰动蚁群算法. 系统工程理论与实践, 2002, 9: 88-91.
- [10] Chi-Bin Cheng, Chun-Pin Mao. A modified ant colony system for solving the travelling salesman problem with time windows. Mathematical and Computer Modelling, 2007, 46: 1225-1235.
- [11] Jinhui Yang, Xiaohu Shi, Maurizio Marchese, etc. An ant colony optimization method for generalized TSP problem. Progress in Natural Science, 2008, 18: 1417-1422.