

A Memory-Guided Ant Colony Algorithm for TSP Problem

Hong Gao Yingchun Li*

School of Computer Science and Technology, Liaoning University of Science and Technology, Anshan, Liaoning, 114051, China

Abstract

To address the problem of ant colony optimization (ACO) easily falling into local optima, this paper proposes a memory-guided adaptive restart ant colony algorithm to solve the traveling salesman problem (TSP). The algorithm stores excellent solutions and employs an intelligent restart strategy to escape local optima. Experimental results show that the proposed algorithm significantly improves solution quality and the hit rate of optimal solutions, accelerates the early convergence speed of the ACO algorithm, and has broad application potential.

Keywords

Memory-Guided Adaptive Restart; Ant Colony Optimization; Traveling Salesman Problem

基于记忆引导重启的蚁群算法求解 TSP 问题

高宏 李迎春*

辽宁科技大学计算机科学与技术学院, 中国·辽宁 鞍山 114051

摘要

针对蚁群算法 (ACO) 容易陷入局部最优的问题, 本文提出一种记忆引导的自适应重启蚁群算法求解旅行商问题 (TSP)。该算法通过存储优秀解并采用智能重启策略来跳出局部最优。实验结果表明, 所提算法显著提高了求解质量和最优解命中率, 加快了ACO算法的前期收敛速度, 具有广阔的应用潜力。

关键词

记忆引导自适应重启; 蚁群算法; 旅行商问题

1 引言

作为组合优化领域最经典的 NP-hard 问题之一, 旅行商问题 (TSP) 在印刷电路板布线设计和 DNA 测序等领域具有广泛的应用价值^[1]。随着现代工业体系规模的快速扩张和智能化管理需求的日益增长, 高效求解 TSP 已成为提升行业效率、降低运营成本的关键技术挑战。

蚁群算法 (ACO) 是一种模拟蚁群觅食行为的元启发式算法, 是求解 TSP 的常用方法^[2,3]。然而, 传统蚁群算法存在前期收敛速度慢、容易陷入局部最优等缺陷, 难以高效找到最优或接近最优的解^[4]。究其根源, 在于信息素分布与解的多样性之间的动态失衡: 算法初期信息素在各路径上初

始分布均匀, 蚂蚁的路径构建具有较强的随机性, 导致有效启发式信息积累缓慢, 这是造成前期收敛速度滞后的根本原因; 而随着迭代进行, 优质路径上的信息素浓度呈指数级增长, 正反馈机制虽能加速收敛, 却也极易导致种群迅速丧失多样性, 使算法陷入局部最优解无法自拔——这一现象在复杂对称 TSP 实例及大规模非对称 TSP 问题中尤为突出。尽管研究者提出了最大最小蚁群系统 (MMAS) 等多种策略来缓解早熟收敛问题^[5], 通过限定信息素轨迹范围和初始化全局最优解的信息素来试图延缓早熟, 但这些方法本质上仍属于被动防御机制, 无法在搜索彻底停滞时主动引导种群跳出局部陷阱, 因此在复杂搜索空间中的停滞问题仍然存在。

记忆引导的自适应重启算法是一种融合历史信息利用和动态重启机制的启发式优化方法。通过维护精英解的记忆库, 并根据搜索停滞指标智能触发重启, 旨在跳出局部最优, 引导搜索向更有前景的区域进行^[6]。与传统的随机重启或简单的变异操作不同, “记忆引导”强调对搜索历史的尊重与利用。具体而言, 记忆库不仅保存当前迭代中的最优解, 还系统性地记录历史迭代中出现的具有代表性或结构多样性的高质量解, 这一记忆库构成了算法对解空间认知“集体

【基金项目】2025年辽宁科技大学校级大学生创新创业训练计划项目。

【作者简介】高宏 (2004—), 男, 中国四川人, 在读本科, 从事聚焦于启发式智能算法的改进与混合优化策略研究。

【通讯作者】李迎春 (1973—), 女, 中国辽宁喀左人, 副教授, 从事自然语言处理研究。

记忆”。当检测到连续多代最优解无改进等搜索停滞指标时，自适应重启机制不再盲目地重置所有参数，而是依据记忆库中的精英解结构进行有导向的重启。这种重启可以表现为两种形式：其一，利用记忆库中解的共同边特征对信息素矩阵进行重新初始化，强化有前景的路径组合；其二，在种群重置时，部分蚂蚁直接继承记忆库中的优秀解片段作为模板进行局部扰动，从而在保留历史成果的基础上开辟新的搜索方向。这一概念与其他元启发式算法中用于平衡探索与开发的自适应机制有相似之处^[7]。

鉴于此，本文将记忆引导的自适应重启机制引入 ACO 框架，提出名为记忆引导自适应重启蚁群算法 (MGA-ACO) 的新算法。这种“记忆 - 检测 - 重启”的闭环设计，本质上是对算法探索与开发能力的一种动态调谐。在搜索前期，算法倾向于开发记忆库中的优质结构以加速收敛，有效提升前期搜索效率；在检测到停滞时，则利用记忆引导的多样性注入强制转移搜索重心，规避无效的随机徘徊，从而在保留历史成果的基础上持续开辟新的搜索方向。该算法通过系统保存高质量解、监控搜索进度并执行智能重启，旨在提升 ACO 求解 TSP 的性能，同时提高收敛速度和最终解质量，为印刷电路板高密度布线及 DNA 测序等领域的实际应用提供更为有效的解决方案。

2 算法设计

2.1 蚁群算法的基本原理 **

蚁群算法 (ACO) 受自然界蚂蚁觅食行为的启发，是由 Dorigo 等人首次提出的基于群体的启发式搜索方法^[2, 8]。在觅食过程中，蚂蚁会在所经过的路径上释放信息素。后续的蚂蚁倾向于以更高的概率选择信息素浓度较高的路径。这种正反馈机制加上信息素的挥发，使得蚁群能够共同发现巢穴与食物源之间的短路径。该算法已被形式化并扩展到求解像 TSP 这样的组合优化问题^[3]。

2.2 记忆引导自适应重启算法的基本原理 **

记忆引导的自适应重启算法是一种旨在克服迭代优化过程中搜索停滞的策略。其核心思想是持续记录关键历史信息，如精英解 (构成“记忆”) 和搜索状态指标^[6]。当预定义的停滞条件满足时 (例如，最佳解在若干次迭代中无改进)，算法不会简单地随机重置。相反，它利用存储的记忆来引导重启，这可能涉及在保留或强化优质解知识的同时，部分重置搜索状态。这种方法有助于跳出局部最优，并促进对搜索空间进行更有效的探索^[9]。

3 算法

3.1 参数设置

Name	Parameter settings
Number of cities	30 → 80 → 150
ALPHA	1.0 → 1.2 → 1.5
BETA	2.5 → 2.0 → 1.8
RHOc	0.6 → 0.55 → 0.5
Q	120 → 150 → 200
ant_num	60 → 120 → 200
MEMORY_SIZE	8 → 12 → 15
SIMILARITY_THRESHOLD	0.65 → 0.7 → 0.75
Restart threshold	20 → 30 → 40
Restart ratio	0.3 → 0.2 → 0.15
Exploitation ratio	0.9 → 0.85 → 0.8
Elite bias	2.0 → 1.8 → 1.6

3.2 算法步骤

步骤 1: 初始化阶段

距离矩阵初始化: 根据城市规模加载相应的坐标数据, 计算欧氏距离矩阵。

信息素矩阵初始化: 将所有边的初始信息素值设为 1.0。

组件初始化:

精英记忆库: 设置容量和相似度阈值。

智能重启管理器: 初始化基本重启阈值和最小多样性阈值 (0.15)。

自适应信息素管理器: 负责信息素更新和重置。

蚁群初始化: 创建增强型蚂蚁, 每只蚂蚁都与精英记忆库关联。

步骤 2: 蚂蚁路径构建

起点选择: 每只蚂蚁随机选择一个起始城市。

路径构建: 采用轮盘赌选择下一个城市, 选择概率综合考虑:

- 基础概率: $(\text{信息素}^\alpha) \times (1/\text{距离}^\beta)$
- 精英引导: 如果该边在精英边集中, 则概率增强。
- 最近邻启发式: 在早期阶段, 对短距离边额外增加 1.3 倍的增强。

局部优化: 以 30% 的概率执行 2-opt 局部搜索。

步骤 3: 记忆库更新与精英引导

1. 解筛选标准:

- 质量优先: 距离 $< 1.05 \times$ 记忆库中当前最优解
- 多样性补充: 与记忆库中所有解的最大相似度 $<$ 相似度阈值

2. 记忆库维护: 当记忆库满时, 移除最差的解。

3. 边频统计: 统计所有精英解中各条边的出现次数。

4. 精英边生成: 选择出现频率最高的边作为精英引导。

步骤 4: 智能重启监控与执行

收敛监控指标:

1. 连续无改进迭代次数 (监控阈值为重启阈值)
 2. 最近 10 次改进的平均改进幅度
 3. 记忆库多样性评分 (1 - 平均相似度)
- 自适应重启阈值调整:

1. 多样性 > 0.6: 阈值 × 1.5
2. 改进幅度大: 适当放宽阈值

重启触发条件:

1. 连续无改进迭代次数 ≥ 自适应阈值 → 停滞重启
2. 多样性评分 < 0.15 → 多样性重启
3. 平均改进幅度 < 1.0 → 收敛过慢重启

重启策略选择:

1. 软重启: 重置重启比例的信息素边, 保留精英引导。
2. 多样性重启: 增强长距离边的信息素, 促进探索。
3. 硬重启: 将信息素矩阵完全重置为 1.0 × 随机值。
4. 蚂蚁重启: 同时重启重启比例的蚂蚁。

步骤 5: 动态信息素更新

1. 信息素挥发: 将所有信息素乘以挥发系数 ρ 。
2. 信息素沉积:
 - a. 只奖励高质量解 (距离 < 1.2 × 当前最优解)
 - b. 沉积量 = $Q / \text{路径距离}$
3. 精英边增强: 将精英边集中的边信息素值乘以 1.5 倍。
4. 信息素裁剪: 将信息素值强制限制在 [0.01, 20.0] 范围内, 防止极端值。

步骤 6: 参数自适应调整机制

1. α 自适应: 当搜索停滞时适当增加 α 值 (最大 2.0)。
2. 发 - 探索平衡: 根据搜索阶段动态调整开发比例。
3. 重启重置比例: 根据停滞程度动态调整。

3.3 实验数据对比

Number of Cities	Running Time (seconds)	SOA		MGA-ACO	
		Best	AVG	Best	AVG
30	0.5	3099	3187	3024	3024
	1	3045	3059	3024	3024
	2	3024	3024	3024	3024
80	5	5039	5454	4357	4572
	15	4428	4768	4336	4426
	50	4451	4542	4336	4343
150	15	8302	9391	7956	9295
	50	5534	5940	5019	5286
	100	4985	5097	4845	4985

4 实验设计与结果分析

本文提出的 MGA-ACO 算法在 PyCharm 集成开发环境中使用 Python 3.11.8 实现。为测试算法求解 TSP 的性能并验证其相对于传统蚁群算法的优化效果, 实验采用标准城市

坐标数据集, 选取包含 3 组城市的测试样本, 通过计算欧氏距离构建初始距离矩阵。

表 2 展示了蚁群算法和 MGA-ACO 算法在 30、80 和 150 个城市数据集上生成的路径最优结果。实验表明, 在相同参数设置下, MGA-ACO 算法具有更好的全局搜索能力和收敛速度, 能有效提高求解质量和求解效率。

5 结语

针对蚁群算法收敛速度慢、易陷入局部最优的缺陷, 本文引入记忆引导的自适应重启算法进行优化, 有效提升了蚁群算法的收敛速度和全局搜索能力。实验数据表明, 对于 TSP 问题, 本文提出的 MGA-ACO 算法的求解结果优于传统蚁群算法, 且最优解命中概率更高。

这证明 MGA-ACO 算法融合了记忆引导自适应重启算法的监控、存储和重启机制, 在提升全局搜索能力的同时提高了搜索效率, 验证了算法的有效性。

然而, MGA-ACO 算法的时间复杂度较高。随着城市规模的增大, 时间复杂度问题更为突出。未来的工作需要继续优化其时间复杂度。

参考文献

- [1] Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2006). The traveling salesman problem: a computational study. Princeton university press.
- [2] Dorigo, M., Maniezzo, V., & Colomi, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29-41.
- [3] Dorigo, M., & Stützle, T. (2004). Ant colony optimization. MIT press.
- [4] Stützle, T., & Hoos, H. H. (2000). MAX-MIN ant system. Future generation computer systems, 16(8), 889-914.
- [5] Ouyang, X., & D. L. (2013). A novel hybrid algorithm based on ant colony optimization and Nelder-Mead simplex search for traveling salesman problem. Journal of Computational Information Systems, 9(5), 1867-1874.
- [6] de O. Campos, P. R. A., & Nascimento, M. Z. (2017). A restart strategy for enhancing the performance of population-based metaheuristics. Applied Soft Computing, 61, 1142-1154.
- [7] Lü, Z., & Hao, J. K. (2010). Adaptive tabu search for the traveling salesman problem. Computers & Operations Research, 37(7), 1225-1232.
- [8] Dorigo, M. (1992). Optimization, Learning and Natural Algorithms (Ph.D. thesis). Politecnico di Milano, Italy.
- [9] Rios, L. H., & Sahinidis, N. V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. Journal of Global Optimization, 56(3), 1247-1293.