 DOI: https://doi.org/10.12349/iser.v7i3.9479

A hybrid ant colony algorithm based on black widow optimization and memory guidance for solving traveling salesman 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 shortcomings of traditional ant colony optimization (ACO) in solving the Traveling Salesman Problem (TSP), including slow convergence and susceptibility to local optima, this paper proposes a hybrid ant colony algorithm incorporating Black Widow Optimization (BWO) and Memory-Guided Mechanism (MGA), termed MGA-BWO-ACO. The algorithm establishes a "warm-up-memory-search" tripartite collaborative mechanism: BWO warm-up generates high-quality initial solutions to accelerate convergence, the memory bank dynamically accumulates search experience to guide ant path selection, and the ant colony algorithm performs global search, integrated with local search and adaptive restart strategies. Experimental results demonstrate that the proposed algorithm significantly enhances global search capability and convergence speed, improving solution quality and efficiency, exhibiting broad application potential.

Keywords

Traveling Salesman Problem; Black Widow Optimization; Ant Colony Algorithm; Memory-Guided Mechanism; Hybrid Algorithm

融合黑寡妇优化与记忆引导的混合蚁群算法求解旅行商问题

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

针对旅行商问题求解中传统的蚁群算法收敛速度慢、容易陷入局部最优的缺陷，本文提出一种融合黑寡妇优化算法（BWO）和记忆引导机制（MGA）的混合蚁群算法（MGA-BWO-ACO）。该算法构建“预热-记忆-搜索”三位一体协同机制：BWO预热生成高质量初始解提升收敛速度，记忆库动态积累搜索经验引导蚂蚁选择路径，蚁群算法进行全局搜索，并集成局部搜索与自适应重启策略。数据实验结果表明，该算法显著提升了全局搜索能力与收敛速度，优化了求解的质量与效率，具有广泛应用潜力。

关键词

旅行商问题；寡妇优化算法；蚁群算法；记忆引导机制；混合算法

1 引言

旅行商问题（Traveling Salesman Problem, TSP）作为组合优化领域的经典 NP-hard 问题，其核心目标是寻找一条经过所有给定城市且仅一次的最短闭合路径。该问题不仅在理论上具有重要研究价值，更在物流配送、路径规划、电路板布线、DNA 测序等众多实际领域中有着广泛应用 [1]。随着各领域规模的持续扩大和智能化需求的日益增长，TSP 求解规模不断攀升，传统的精确算法难以在可接受时间内求得最优解，因此启发式智能算法成为研究热点 [2]。

蚁群算法（Ant Colony Optimization, ACO）作为群智能优化算法的典型代表，由 Dorigo 等人于 20 世纪 90 年代初提出，其灵感来源于蚂蚁觅食过程中的信息素通信机制 [3,4]。该算法凭借正反馈、并行性和鲁棒性等优势，在 TSP 求解中取得广泛应用。然而，传统蚁群算法存在固有缺陷：算法初期信息素分布均匀，缺乏有效引导，导致前期收敛速度缓慢；正反馈机制虽有助于收敛，却也容易使算法陷入局部最优，出现早熟停滞现象；求解稳定性受参数设置影响较大，

【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，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。

限制了其普适性 [5,6]。

黑寡妇优化算法（Black Widow Optimization, BWO）是 Hayyolalam 等人于 2020 年提出的新型元启发式算法，模拟黑寡妇蜘蛛独特的交配行为与生存策略，包括交配繁殖、同类相食、变异更新等机制 [7]。该算法通过独特的种群更新策略，在全局勘探与局部开发之间取得良好平衡，具有较强的全局搜索能力和较快的收敛速度，已在函数优化、工程设计等领域得到成功应用 [8]。

记忆引导机制（Memory-Guided Mechanism, MGM）是一种通过动态存储和利用历史优质信息来优化搜索过程的有效策略。其核心思想是构建精英解记忆库，持续记录搜索过程中发现的优质解及其特征信息，当算法陷入搜索瓶颈时，记忆库中的历史经验能够提供有价值的参考，引导种群向更有前景的区域推进 [9,10]。研究表明，融合记忆机制的优化算法能够有效提升搜索效率和求解质量 [11]。

基于上述分析，本文创新性地提出一种融合黑寡妇优化算法和记忆引导机制的混合蚁群算法(MGA-BWO-ACO)，构建“预热 - 记忆 - 搜索”三位一体协同机制：BWO 预热生成高质量初始解，提升算法前期收敛速度；记忆库动态积累搜索经验，通过记忆奖励因子引导蚂蚁路径选择；蚁群算法进行精细化全局搜索，并集成局部搜索与自适应重启策略。三种机制优势互补，力求改善传统蚁群算法的性能，提升 TSP 求解的效率与质量，为相关领域实际应用提供一种更具性能的新方案。

2 算法设计

2.1 蚁群算法的基本原理

蚁群算法（Ant Colony Optimization, ACO）是一种启发式优化算法，由 Marco Dorigo 等人在 1992 年提出 [3,4]。蚁群算法模拟蚂蚁觅食过程，蚂蚁在路径上释放信息素，根据赌轮盘的方式选择路径，信息素浓度高的路径被选择概率大。随着蚂蚁不断选择路径，较优路径上信息素浓度逐渐提高，从而引导蚁群找到从蚁巢到食物源的最短路径。该算法已被广泛应用于求解 TSP 等组合优化问题 [5,6]。

2.2 黑寡妇优化算法的基本原理

黑寡妇优化算法（Black Widow Optimization, BWO）是一种新型元启发式算法，由 Hayyolalam 等人于 2020 年提出，模拟黑寡妇蜘蛛独特的交配与捕食行为 [7,8]。算法核心包括三个阶段：种群初始化，生成候选解个体；交配行为，通过独特的交叉操作产生子代，模拟蜘蛛繁殖过程；同类相食机制，淘汰适应度较差的个体，保留优质解。这种机制使得 BWO 算法在全局勘探与局部开发之间取得平衡，具有较强的收敛能力和求解精度，适用于函数优化和组合优化问题 [8]。

2.3 记忆引导机制的基本原理

记忆引导机制（Memory-Guided Mechanism, MGM）是一种通过动态存储和利用历史搜索经验来优化算法性能的策略 [9,10]。其核心思想是构建一个记忆库，用于存储算法在搜索过程中发现的优质解及其特征信息。在后续迭代中，算法通过记忆奖励因子对记忆库中的优质边或路径片段给予额外选择概率，引导蚂蚁或个体向历史经验中的优质区域搜索。记忆库采用优胜劣汰的更新策略，不断保留更优解、淘汰劣质解，实现搜索经验的动态积累与有效利用，从而提升算法的收敛速度和解的质量 [11]。

3 MGA&BWO&ACO 算法

3.1 算法简介

MGA-BWO-ACO 算法是融合记忆引导机制（MGA）、黑寡妇优化算法（BWO）、蚁群算法（ACO）的混合算法，创建预热 - 记忆 - 搜索三位一体协同机制。BWO 预热快速生成高质量初始解，提升蚁群算法前期收敛速度 [7,8]；MGA 记忆动态积累搜索经验，利用经验引导智能搜索 [9,10]；ACO 进一步搜索全局，求解最终的最优解 [3,4]。三者通过信息素增强和记忆奖励因子实现双向信息流动，实现优势互补，能有效提升 TSP 问题的求解质量与效率。

3.2 参数设置

参数名称	参数设置
信息素权重 α	1
启发信息权重 β	2
挥发系数 ρ	0.5
沉积常数 Q	100
蚂蚁数量	min(150, 城市数 ×2)
种群规模	60
迭代代数	6
变异率	0.3
精英比例	0.2
记忆库容量	50
精英偏置	5
局部搜索概率	0.3
2-opt 迭代	100
开发蚂蚁比例	0.4
BWO 边奖励	3
BWO 初始增强	8
最优路径增强	3
停滞检测阈值	3
重启比例	0.5

3.3 算法设计

步骤一：BWO 全局预热初始化

构建 BWO 种群，采用混合初始化策略，50% 个体使用贪心算法构建（每次选择最近未访问的城市），余下的 50% 个体随机生成，以此来保障种群多样性 [7,8]；执行 BWO 优化迭代；提取 BWO 最优路径及其包含的所有边；将 BWO 最优解存入记忆库，作为初始经验。边；将 BWO 最优解存入记忆库，作为初始经验。

步骤二：信息素与记忆库初始化

初始化信息素矩阵（所有边设置为 1.0）[3,4]；将 BWO 优质边进行强化；对记忆库进行初步填充 [9,10]；构建混合蚂蚁群（MGA 蚂蚁、BWO 蚂蚁、普通蚂蚁）。步骤三：循环迭代搜索

每只蚂蚁独立构建完整路径：

a.MGA 蚂蚁：状态转移概率 = (信息素 ^α) × (启发信息 ^β) × 记忆奖励因子

b.BWO 蚂蚁：给 BWO 优质边 3 倍转移概率奖励

c. 普通蚂蚁：利用赌轮盘的方式随机生成概率，60% 概率直接使用记忆库最优解，20% 概率交叉 top5 路径，20% 概率标准行走 [3,4]

计算每只蚂蚁的路径长度，若新的解优于记忆库中最差的解，则加入记忆库，更新边频统计 [10]；对全局最优解进行深度局部搜索（2-opt）[12]，对其他优质解以一定概率进行轻度局部搜索；执行信息素更新，遵循 ACO 框架的挥发与沉积规则 [5,6]；若连续 3 次迭代无改进，判定为陷入局部最优，随机选择 50% 的蚂蚁进行重置，保留记忆库历史经验 [11]。步骤四：结果输出

循环迭代，直至达到预设运行时间，终止循环，统计数据并返回实验结果

城市数量	运行时间（秒）	ACO		BWO&ACO		MGA&BWO&ACO	
		Best	AVG	Best	AVG	Best	AVG
30	5	3024	3024	3024	3024	3024	3024
50	10	3654	3667	3633	3652	3633	3640
100	30	4782	4848	4578	4630	4550	4605
150	5	6970	7917	4784	4858	4820	4868
150	10	5666	5811	4821	4864	4827	4869
150	30	5100	5272	4790	4865	4762	4821
150	60	5026	5136	4778	4838	4755	4813

4 实验设计与结果分析

本为提出的 MGA&BWO&ACO 算法采用 Python 语言 3.11.8 版本在 Pycharm 集成开发环境中实现，为测试该算法求解旅行商问题的性能，并验证其与传统蚁群算法相比的优化效果，实验采用标准的城市坐标数据集，选取包含 4 组城市数据的测试样本。

5 结语

针对传统蚁群算法在求解旅行商问题时前期收敛速度较慢且容易陷入局部最优的缺陷，本文提出了一种融合黑寡妇优化算法和记忆引导机制的混合蚁群算法（MGA-BWO-ACO）。本文引入黑寡妇优化算法进行全局预热，生成高质量初始路径并强化初始信息素分布，能有效提升算法前期的收敛速度 [7,8]；引入记忆引导机制动态积累搜索经验，通过记忆奖励因子引导搜索方向，局部最优情况显著下降 [9,10]；集成自适应重启机制进一步提高了解的质量 [11]。实验数据表明，对于不同规模的旅行商问题，本文提出的 MGA-BWO-ACO 算法求解结果优于传统的蚁群算法和 BWO-ACO 算法，且命中最优解的概率更高，平均解的质量也更高。这证明 MGA-BWO-ACO 算法融合了黑寡妇优化算法的快速勘探能力、记忆引导机制的经验积累能力与蚁群算法的并行精细搜索能力，实现了优势互补，验证了算法的有效性和鲁棒性。

然而，MGA-BWO-ACO 算法的时间复杂度仍然较高，随着城市规模的扩大，记忆库维护和局部搜索的计算开销问题愈发突出。未来工作将聚焦于算法的时间复杂度优化 [2]，在保持解的质量的同时提升运行效率，并探索自适应参数调节机制以进一步提升算法的性能 [6]。

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] Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2), 231-247.

[3] Dorigo, M., Maniezzo, V., & Colorni, 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.

[4] Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT press.

[5] Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. *Future generation computer systems*, 16(8), 889-914.

[6] Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4), 353-373.

[7] Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 87, 103249.

[8] Abdollahzadeh, B., Soleimanian Gharehchopogh, F., & Mirjalili, S. (2021). Black widow optimization: a comprehensive review, applications, and open issues. *Artificial Intelligence Review*, 54(8), 6169-6214.

[9] 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.

[10] Lü, Z., & Hao, J. K. (2010). Adaptive tabu search for the traveling salesman problem. *Computers & Operations Research*, 37(7), 1225-1232.

[11] 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.

[12] Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6(6), 791-812.