

求解旅行商问题的混合蚁群优化算法*

郭娜苹^{1,2}, 马小华², 高岳林^{1,2}

(1. 北方民族大学数学与信息科学学院, 银川 750021;

2. 宁夏科学计算与智能信息处理协同创新中心, 银川 750021)

摘要:针对蚁群算法在求解旅行商问题时存在收敛速度慢、易陷入局部最优等不足,提出一种混合蚁群优化算法(PSO-ACO-3opt)。首先,引入自适应权重以增强粒子群算法的全局搜索能力和局部搜索能力;其次,采用基于复合函数的状态转移公式,平衡信息素与启发因子之间的关系,提升算法的鲁棒性;同时,加入信息素重置比率因子,增强蚂蚁的搜索能力,避免算法过早陷入局部最优;最后,通过3-opt局部搜索策略进一步优化路径。实验结果表明,改进的算法在TSPLIB数据集上的性能优于基本蚁群算法。

关键词:旅行商问题; 蚁群优化算法; 粒子群优化算法; 自适应; 3-opt

中图分类号: TP18 **文献标志码:** A

组合优化是在离散状态下求极值的问题,其目的是在离散变量可行解集中寻找最优解。作为运筹学和科学研究的重要分支,组合优化在实际问题上具有重要意义^[1]。旅行商问题(Traveling salesman problem, TSP)是组合优化领域的经典问题之一,在物流配送、车辆路径规划、电路板制造和机器人路径规划等众多领域中得到了广泛应用。

目前有关TSP的求解方法主要分为:(1)精确算法,如整数线性规划法、动态线性规划法、分支定界法和图算法等。这些方法在小规模问题上能够提供高效的解决方案,但在较大规模或数据精度不足的情况下,通常会出现计算复杂度呈指数级增长的问题,进而导致运算效率大幅度降低。(2)智能优化算法,旨在可接受的计算时间内找到令人满意的或接近最优解的解,如蚁群优化(Ant colony optimization, ACO)算法、粒子群优化(Particle swarm optimization, PSO)算法、黏菌算法(Slime mould algorithm, SMA)、萤火虫算法(Firefly algorithm, FA)等,这些算法通过借鉴自然界中特定现象或行为模式,运用其内在优化原理,可以有效地解决复杂问题。为了解决多旅行商问题,Meng等^[2]引入城市颜色矩阵,以提供各个推销员城市可达性的差异信息,从而提高可行解的质量;Pandiri等^[3]提出了一种具有扰动策略的人工蜂群算法,以减少迭代过程中相邻解的影响程度,增强了算法的稳定性;张硕航等^[4]提出将K-means聚类和改进ACO算法相结合,不仅有效缩短了求解TSP的运行时间,还显著提高了求解精度;Gong等^[5]将SMA与分数阶蚂蚁系统相结合,设计了一种新的信息素更新规则,实现了搜索与开发之间的动态平衡。针对参数优化问题,Wang等^[6]提出一种混合蚁群生物共生搜索算法,利用生物共生搜索算法自适应优化ACO算法的重要参数,这种设计不但提高了求解精

* 收稿日期:2024-07-01; 修回日期:2024-10-16

基金项目:宁夏自然科学基金项目(2024AAC03150);北方民族大学重大专项(ZDZX201901);宁夏一流学科建设(NXYLXK2017B09);南京证券支持基础学科项目(N-JZQJCK202201);北方民族大学创新创业项目(YCX24084)

作者简介:郭娜苹(1998—),女,宁夏吴忠人,2022级硕士研究生。E-mail: chuichui0810@163.com

通信作者:马小华(1964—),女,陕西榆林人,教授,硕士。主要从事系统工程的研究。E-mail: mxh6464@163.com

com

度,而且降低了参数分配的复杂度;Zhang等^[7]通过使用反转算子、交叉算子以及三选算子,有效缓解了参数敏感性问题,提升了算法的局部搜索能力。众所周知,局部优化算法能够显著提升启发式算法对TSP的求解性能。Dong等^[8]借助3-opt优化算法的求解速度和求解精度,避免狼个体陷入局部最优困境;Dahan等^[9]在应用TSP的动态飞行ACO算法中引入3-opt算法,从而减轻停滞问题对求解效果的影响;Hougardy等^[10]在求解TSP时提出上下限近似比的2-opt改进算法,以一种更加严谨的方式证明了改进算法的有效性。

尽管上述文献的各类算法在求解TSP时已取得一定效果,但ACO算法因其仿生机理仍展现出独特的优势。然而,ACO算法中的正反馈机制可能导致其搜索易陷入局部最优,这一局限性主要源于算法对参数设置的敏感性,尤其是在应对不同问题时,参数选择对算法性能有着显著影响。因此,本文从多启发式算法的角度出发,在ACO算法的框架下,利用改进粒子群优化ACO算法的两个重要参数,对状态转移公式进行改进,以避免单一因素过度主导路径的选择;当算法陷入局部最优时,通过信息素重置因子重新调整部分信息素浓度,从而增强蚂蚁的搜索能力;最后利用局部搜索3-opt优化策略解的质量。基于标准TSPLIB数据集的仿真实验结果表明,PSO-ACO-3opt算法在不同规模以及不同分布的实例中均可得到不错的结果,验证了所提算法的可行性和有效性。

1 旅行商问题

TSP可以表述为:一个推销员从某个城市出发,需遍历完其他城市,使得每个城市恰好被访问一次,最终返回到起始城市。其目标是找到一条最短的路径,使得行走的总路线长度最短。设 $G=(V, E)$ 为赋权完全图, V 为 n 个城市构成的顶点的集合, E 是所有城市间形成的边的集合, d_{ij} 为城市 i 和城市 j 之间的距离, u_i 用于确保城市的访问顺序。TSP具体模型如下:

$$\min \sum_{i=1}^n \sum_{j=1, j \neq i}^n d_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1, j \neq i}^n x_{ij} = 1, \forall i \in \{1, 2, \dots, n\} \quad (2)$$

$$\sum_{j=1, i \neq j}^n x_{ji} = 1, \forall j \in \{1, 2, \dots, n\} \quad (3)$$

$$u_i - u_j + nx_{ij} \leq n - 1, \forall i \neq j, i, j \in \{2, 3, \dots, n\} \quad (4)$$

$$u_1 = 1 \quad (5)$$

$$x_{ij} = \begin{cases} 1, & \text{经过路径}(i, j) \\ 0, & \text{其他} \end{cases} \quad (6)$$

其中,式(1)为目标函数,旨在最小化距离成本;式(2)和(3)共同确保了每个城市恰好被访问和离开一次,避免了城市被重复访问或遗漏的问题;式(4)用于防止路径中出现子环;式(5)规定起始城市为1;式(6)定义了二元决策变量,用于表示是否选择某一条边作为路径的一部分。

2 基本算法介绍

2.1 蚁群算法

ACO算法是一种模拟蚂蚁觅食行为的群智能优化算法。针对TSP问题,ACO算法需要构建特定的数学模型,该模型构建主要包括路径构建和信息素更新两个阶段。在路径构建阶段,每只蚂蚁从起始点出发,利用状态转移公式,在信息素和启发式函数的共同指导下,逐步构建出一条从起点城市到终点城市的完整路径。为提高算法的搜索能力并防止过早收敛于局部最优解,ACO算法采用轮盘赌选择策略来增加随机性,使得蚂蚁能够更全面地搜索不同的路径组合,从而提高发现全局最优解的可能。状态转移公式如式(7)所示:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in J_k(i)} [\tau_{is}(t)]^\alpha [\eta_{is}]^\beta}, & j \in J_k(i) \\ 0, & j \notin J_k(i) \end{cases} \quad (7)$$

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (8)$$

其中, $J_k(i)$ 表示蚂蚁 k 被允许访问的城市集合; $P_{ij}^k(t)$ 表示在第 t 次迭代中, 蚂蚁 k 从城市 i 转移到下一城市 j 的概率; τ_{ij} 为路径 (i, j) 上的信息素的浓度; η_{ij} 表示从城市 i 到城市 j 的启发式信息, 通常取值为两个城市间距离 d_{ij} 的倒数, 距离越小, 期望程度越大, 相应被选择的概率也越大; α 为信息素启发因子, 表示路径上累积信息素对蚂蚁选择决策的影响强度; β 为期望启发因子, 反映启发式信息在路径选择中的相对权重。

在信息素更新阶段, 当所有蚂蚁完成路径构建后, 算法将对路径上的信息素进行迭代更新, 该过程的目的是增强那些表现优异的路径上的信息素, 以便在未来的搜索中更可能被其他蚂蚁选择。更新公式如下:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij} + \Delta\tau_{ij}(t) \quad (9)$$

$$\Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \quad (10)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{蚂蚁 } k \text{ 在本次循环经过 } (i, j) \\ 0, & \text{其他} \end{cases} \quad (11)$$

其中, ρ 为信息素蒸发系数, $1 - \rho$ 表示路径上剩余信息素的含量。 $\Delta\tau_{ij}(t)$ 表示第 t 次迭代中所有蚂蚁在路径 (i, j) 上释放的信息素总量, L_k 为在本次迭代中蚂蚁 k 走过的路径长度, Q 表示信息素强度。

2.2 标准粒子群优化算法

PSO 算法是一种模拟鸟群觅食行为的群智能优化算法, 因其强大且独特的路径优化能力而备受研究者青睐。在 PSO 算法的寻优过程中, 每个粒子代表在 D 维解空间中的一个可能解, $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ 和 $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ 分别表示粒子当前的位置和速度。粒子的速度同时受个体认知(个体最优位置)和社会认知(全局最优位置)影响, 二者共同指导粒子的搜索方向, 促使种群向更有希望的区域移动。迭代更新公式如式(12)、式(13)所示:

$$v_{ij}^{k+1} = \omega v_{ij}^k + c_1 r_1 (Pbest_i^k - x_{ij}^k) + c_2 r_2 (Gbest^k - x_{ij}^k) \quad (12)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (13)$$

其中, x_{ij}^k 及 v_{ij}^k 表示粒子 i 在第 k 次迭代时所处的位置及速度, $Pbest_i^k$ 为粒子 i 在前 k 次迭代中的个体最优位置, $Gbest^k$ 为截止到第 k 次迭代时找到的全局最优位置, c_1, c_2 为学习因子, r_1, r_2 为 $[0, 1]$ 之间的随机数。

2.3 3-opt 算法

局部搜索策略通常被用于提高 TSP 的求解性能。3-opt 算法是一种常用的局部搜索方法, 其基本思想是选取生成路径中不相邻的 3 个节点并删除节点之间的连线, 在不断重组连接中, 3-opt 算法逐渐接近最优解。以一个闭合六边形路径为例, 如图 1(a) 所示。假设初始路径为 $[1, 2, 3, 4, 5, 6, 1]$, 随机选择 3 个不相邻的节点 2、4、6, 并删除在原路径上连接这 3 个不相邻节点的边 $2-3$ 、 $4-5$ 及 $6-1$, 将原路径分为 3 个小路径段 $[1, 2]$ 、 $[3, 4]$ 、 $[5, 6]$ 。通过再连接节点 2、4、6 形成新的子路径得到图 1 中 8 种不同的连接情况。启发式算法在求解 TSP 时, 通常会得到最优解或者接近最优解的解。当得到的解不是最优解时, 其缘由是因为生成的最短路径仍然存在可优化的现象。3-opt 算法通过尝试不同的边组合, 从而达到提高路径质量的目的。

为更直观地理解 3-opt 算法的工作原理, 通过一个实例进行说明。如图 2 左图所示, 蚂蚁从节点 A 出

发,生成的回路为[A, B, C, D, E, F, A],显然该路径不是最优路径。经过 3-opt 优化后生成如图 2 右图所示的闭合路径[A, C, B, D, E, F, A],有效提高了解的质量。对于局部搜索策略 3-opt 而言,一个好的初始解能够有效地帮助该算法快速找到更满意的解,因此 3-opt 算法常用来优化启发式算法得到的解^[6]。

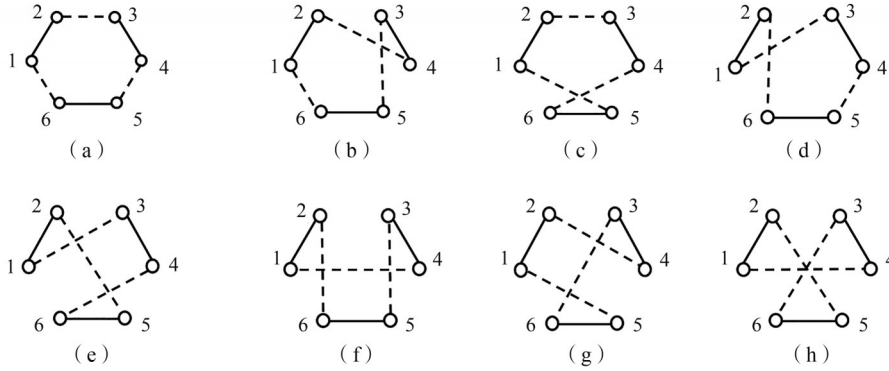


图 1 3-opt 所有连接可能

Fig. 1 All possible connections in 3-opt

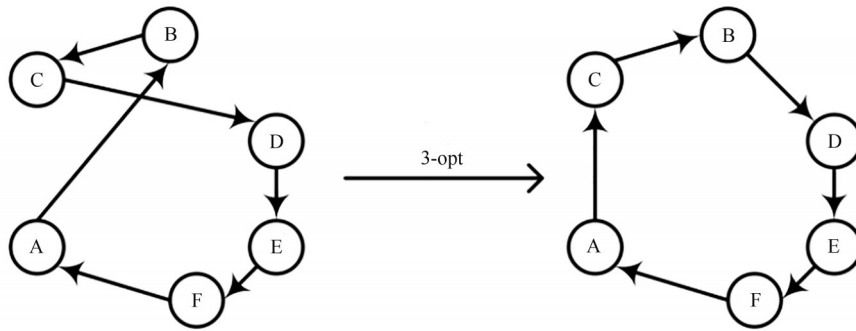


图 2 3-opt 在旅行商前后的路径

Fig. 2 Path transformation before and after 3-opt in the traveling salesman problem

3 PSO-ACO-3opt 算法

3.1 改进的粒子群算法

在粒子群算法中,惯性权重是一个关键参数,它决定了粒子在前进方向上的惯性,并影响粒子的搜索能力。然而,过大或过小的惯性权重都会导致算法性能下降。文献[11]提出了一种自适应权重的改进粒子群算法,能够应对快速趋同问题。借鉴该思路,本文提出了一种改进的自适应惯性权重策略,进一步优化算法性能,公式如(14)和(15)所示:

$$\omega_a = \omega e^{-\frac{0.9iter+1}{iter_{max}}} \tag{14}$$

$$v_{ij}^{k+1} = \omega_a v_{ij}^k + c_1 r_1 (Pbest_i^k - x_{ij}^k) + c_2 r_2 (Gbest^k - x_{ij}^k) \tag{15}$$

其中, $iter_{max}$ 为最大迭代次数。在改进的算法中,惯性权重会随着迭代次数的增加而逐渐减小,使得算法在早期阶段能够保持较大的权重,有助于广泛搜索整个解空间。

3.2 改进蚁群算法

原始状态转移概率公式在解决某些复杂问题时,会出现信息素过高或启发式信息过高的极端现象。为了解决这一问题,采用对数函数对信息素和启发式信息进行处理,从而减弱这些极端现象带来的影响。对数函数能够有效减小大数和小数之间的差距,平滑极端值的影响。在此基础上,为了在状态转移公式中使用这些处理后的数值,引入指数函数进行恢复。这样,状态转移结果能够合理

地反映信息素和启发式信息的相对影响,避免单一因素主导路径选择,提高算法的全局搜索能力和适应性。改进后的公式为

$$P_{ij}^k(t) = \frac{e^{\alpha \ln(\tau_{ij}(t)) + b_1} e^{\beta \ln(\eta_{ij}(t)) + b_2}}{\sum_{s \in J_k(i)} e^{\alpha \ln(\tau_{is}(t)) + b_1} e^{\beta \ln(\eta_{is}(t)) + b_2}} \quad (16)$$

其中, $\ln x$ 为对数变化,用于对信息素 $\tau_{ij}(t)$ 和启发式函数 $\eta_{ij}(t)$ 的转化。分子部分将转化后的函数进行线性组合, b_1 、 b_2 为线性组合的偏置项,用于调节线性组合的结果以增加公式的灵活性。

3.3 改进粒子群算法优化蚁群参数

在求解 TSP 时,ACO 算法是一种有效的启发式算法,但其性能受参数设置的影响较大。由于 TSP 的实例特性不同,同一组参数设置可能对不同问题的适应性存在差异,因此确定最佳参数组合需要进行大量实验。手动调参的过程费时费力,且结果可能不稳定。为克服这一不足,采用改进 PSO 算法对 ACO 算法的两个重要参数进行优化。通过这种方式,可以更有效地确定最佳参数组合 (α, β),提高算法的适用性,从而更好地求解 TSP。

在使用 PSO 算法优化 ACO 算法参数之前,需要生成一组随机解。每个粒子代表一个可能的参数组合,而粒子的位置则对应了一组具体的参数值。使用每个粒子的参数值用于 ACO 算法求解路径长度,并记录路径长度 $Pbest$ 和 $Gbest$,同时根据边界检查公式对粒子的速度和位置进行更新。

3.4 PSO-ACO-3opt 算法流程

PSO-ACO-3opt 算法流程如图 3 所示,算法步骤如下:

步骤 1:初始化参数。初始化蚁群信息素、信息素启发式因子 $[\alpha_{min}, \alpha_{max}]$ 和期望启发因子 $[\beta_{min}, \beta_{max}]$,设置禁忌表及信息素重置比例因子。

步骤 2:初始粒子种群在 α, β 的取值范围内随机生成。

步骤 3:将 m 只蚂蚁随机放到 n 个城市。通过式(16)选择需要访问的下一个节点,并将已访问过的城市放入禁忌表中,直至访问完所有的城市。

步骤 4:遍历完所有粒子后,根据边界对粒子的位置和速度进行更新。

步骤 5:迭代过程中保留最佳路径长度及其所对应的路径。

步骤 6:根据式(9)(10)以及(11)更新信息素。

步骤 7:当搜索过程陷入停滞状态时,通过信息素重置比例,帮助算法跳出局部最优。

步骤 8:若未达到迭代终止条件,则返回步骤 2 继续迭代并清空禁忌表,反之输出最优路径。

步骤 9:3-opt 对生成的全局最优路径进行优化。

步骤 10:输出优化后的路径。

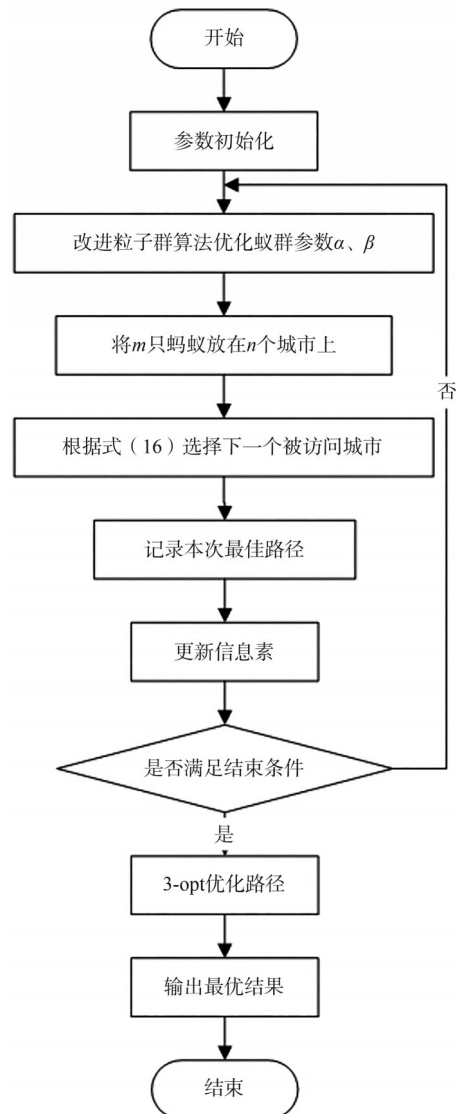


图 3 PSO-ACO-3opt 算法流程

Fig. 3 Flowcharts of the PSO-ACO-3opt algorithm

4 数值实验

4.1 TSP测试集和实验环境

本实验在 Matlab 的 R2020a 平台上进行。为验证 PSO-ACO-3opt 算法的性能,本文在 Windows 10 的计算机(Intel(R) Xeon(R) E-2224 CPU @ 3.40GHz, 3.41 GHz)上进行测试,并从 TSPLIB 中选取不同规模的 TSP 实例进行实验。

4.2 实验方法

BKS 表示截至目前已知的最优解;独立运行 20 次得到的最优值用 $Best$ 表示;平均值记为 Avg ; $Error$ 表示算法平均解与已知最优解之间的相对误差,公式为 $Error=(Avg - BKS)/BKS$; DE 表示算法最优解与已知最优解之间的相对误差,公式为 $DE=(Best - BKS)/Best$ 。实验结果中较优值以粗体显示。为更好地比较数据,所有实验结果均忽略 BKS 、 $Best$ 的小数部分。

4.2.1 PSO-ACO-3opt 在不同数据集上的实验

从表 1 可以看出,在 Eil51、Berlin52、St70、Eil76、KroA100、Lin105、Ch130 测试实例中,PSO-ACO-3opt 算法找到的 $Best$ 最接近 BKS 。对于 Att48 和 KroD100,改进后的算法求得的 $Best$ 与 BKS 相同。在 14 个实例中,所有的测试结果与 BKS 的误差均在 0.01 以内,这表明该算法具有较强的稳定性。当城市数量在 100 个及以上时,除 Gil262 和 Eil76 之外,其他实例得到的 $Error$ 都在 2% 以内,再次表明 PSO-ACO-3opt 算法的稳定性较好。

表 1 PSO-ACO-3opt 算法数值实验结果

Table 1 Numerical experimental results of the PSO-ACO-3opt algorithm

实例	BKS	$Best$	Avg	$Error/\%$	DE
Att48	33523	33523	33618.2	0.28	0
Eil51	426	428	433.05	1.65	0
Berlin52	7542	7544	7549.4	0.10	0
St70	675	677	681.35	0.94	0
Eil76	538	545	550.5	2.32	0.01
KroA100	21282	21285	21395.75	0.53	0
KroE100	22068	22124	22319.4	1.14	0
KroD100	21294	21294	21497.55	0.96	0
Lin105	14379	14382	14461.33	0.57	0
Ch130	6110	6159	6182.8	1.19	0.01
KroA150	26524	26615	26911.05	1.46	0
KroA200	29368	29387	29763.35	1.35	0
Gil262	2378	2409	2436.35	2.45	0.01
Pr299	48191	48675	48960	1.60	0.01

4.2.2 PSO-ACO-3opt 算法有效性对比实验

为验证所提出策略的有效性,在 Att48、Eil51、St70、Eil76、KroA100 和 Ch130 6 个测试实例上,将 PSO-ACO-3opt 算法与 PSO-ACO 算法和 ACO 算法进行对比实验。从表 2 可以清晰地看出,无论是在 $Best$ 、 Avg 还是 $Error$,PSO-ACO-3opt 算法所获得的结果都明显优于未加入局部优化策略的 PSO-ACO 算法。这表明本文提出的改进策略是有效的。同时,通过比较 PSO-ACO 及 ACO 算法,可以进

一步说明加入改进策略后算法的有效性。最后,表2中数据也证实了3-opt在提升优化解决方案质量方面的贡献。

表2 PSO-ACO-3opt与PSO-ACO和ACO算法数值实验结果对比

Table 2 Comparison of numerical experimental results between PSO-ACO-3opt with PSO-ACO and ACO algorithms

实例	BKS	PSO-ACO-3opt			PSO-ACO			ACO		
		Best	Avg	Error/%	Best	Avg	Error/%	Best	Avg	Error/%
Att48	33523	33523	33618.20	0.28	33523	33752.25	0.68	34714	35359.99	5.48
Eil51	426	428	432.90	1.65	429	433.05	1.62	442	448.39	5.26
St70	675	677	681.35	0.94	682	693.25	2.70	704	728.64	7.95
Eil76	538	545	550.50	2.32	549	554.10	2.99	561	570.08	5.96
KroA100	21282	21285	21395.70	0.53	21334	21908.05	2.94	22387	22532.95	5.88
Ch130	6110	6159	6182.80	1.19	6209	6277.90	2.75	6412	6577.54	7.65

4.2.3 PSO-ACO-3opt与ACO算法的对比

从表3中可以看出,PSO-ACO-3opt和ACO算法在求解Att48、Eil51、Berlin52等9个测试实例上,前者取得的Best均优于后者。对于DE来说,PSO-ACO-3opt同样取得了优于ACO的结果,并且所有测试实例的Best与BKS的误差均在0.01以内,这表明PSO-ACO-3opt算法对TSP有着较强的求解能力。此外,PSO-ACO-3opt算法相较于ACO算法的Avg也更小,进一步说明改进后的PSO-ACO-3opt算法拥有较强的求解能力,同时也具有较好的求解稳定性。

表3 ACO和PSO-ACO-3opt算法数值实验结果对比

Table 3 Comparison of numerical experimental results between ACO and PSO-ACO-3opt algorithms

实例	BKS	算法	Best	Avg	Error/%	DE
Att48	33523	ACO	34714	35359.99	5.48	0.03
		PSO-ACO-3opt	33523	33618.20	0.28	0
Eil51	426	ACO	442	448.39	5.26	0.04
		PSO-ACO-3opt	428	433.05	1.65	0
Berlin52	7542	ACO	7663	7701.93	2.12	0.02
		PSO-ACO-3opt	7544	7549.40	0.10	0
St70	675	ACO	704	728.64	7.95	0.04
		PSO-ACO-3opt	677	681.35	0.94	0
Eil76	538	ACO	561	570.08	5.96	0.04
		PSO-ACO-3opt	545	550.50	2.32	0.01
Lin105	14379	ACO	14774	15069.72	4.80	0.03
		PSO-ACO-3opt	14382	14461.33	0.57	0
Ch130	6110	ACO	6412	6577.54	7.65	0.05
		PSO-ACO-3opt	6159	6182.80	1.19	0.01
KroB150	26130	ACO	27916	28154.92	7.75	0.07
		PSO-ACO-3opt	26327	26506.75	1.44	0.01
Pr299	48191	ACO	53188	53548.68	11.12	0.10
		PSO-ACO-3opt	48675	48960.00	1.60	0.01

4.2.4 PSO-ACO-3opt与传统算法的对比实验

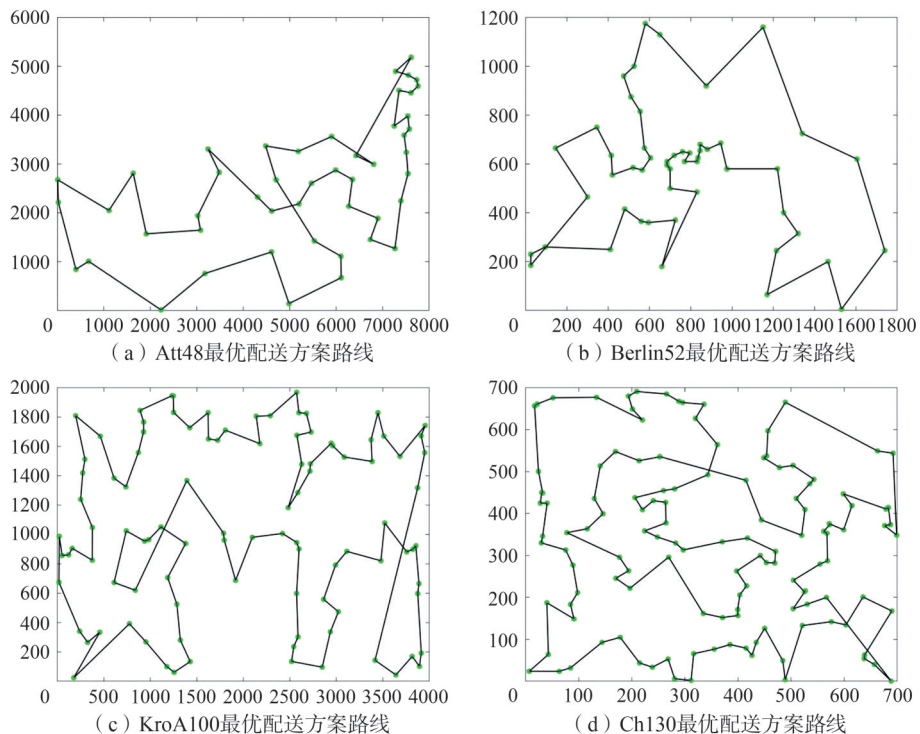
为了进一步说明 PSO-ACO-3opt 算法的有效性,将该算法与目前在 TSP 中应用较广泛的遗传算法(GA)、人工蜂群算法(ABC)以及模拟退火算法(SA)进行对比,对比结果如表 4 所示。结果表明,在求解 Eil51 测试实例时,ABC 算法较本文算法更有效,能够得到更接近 *BKS* 的值。但随着问题规模的增加以及节点数目的增加,ABC 算法在求解 TSP 时所得到的解的质量会逐渐变差。对于所求得的 *Best* 来说,GA 和 SA 均差于 PSO-ACO-3opt 算法,且随着城市规模的增加,GA 和 SA 的 *Best* 逐渐偏离 *BKS*。同时还可以看到,PSO-ACO-3opt 算法在表 4 中所有的测试实例上均取得了非常接近 *BKS* 的值。因此,在一定程度上,PSO-ACO-3opt 算法的求解性能优于 ABC、GA 和 SA。

表 4 PSO-ACO-3opt 与传统算法数值实验结果对比

Table 4 Comparison of numerical experimental results between PSO-ACO-3opt and traditional algorithms

算法	指标	Eil51	Berlin52	Eil76	KroA100	Pr226
PSO-ACO-3opt	<i>BKS</i>	426	7542	538	21282	80369
	<i>Best</i>	428	7544	545	21285	80411
	<i>Avg</i>	437.34	7583.84	555.44	21494.77	80895.55
ABC	<i>Best</i>	428	7544	552	22010	92941
	<i>Avg</i>	432.86	7612.80	564.69	22493.33	99054.16
GA	<i>Best</i>	482	8948	601	32235	359248
	<i>Avg</i>	590.87	10172.08	648.46	38980.39	404118.21
SA	<i>Best</i>	492	8742	691	34169	376082
	<i>Avg</i>	534.76	9695.24	751.05	38619.06	415247.70

为了更直观地看到算法改进前后的效果,选取了 6 个不同节点分布的实例,分别用 ACO 和 PSO-ACO-3opt 算法进行寻优。图 4 为 ACO 算法寻优效果,图 5 为 PSO-ACO-3opt 算法寻优效果,显然后者能够取得更好的效果。



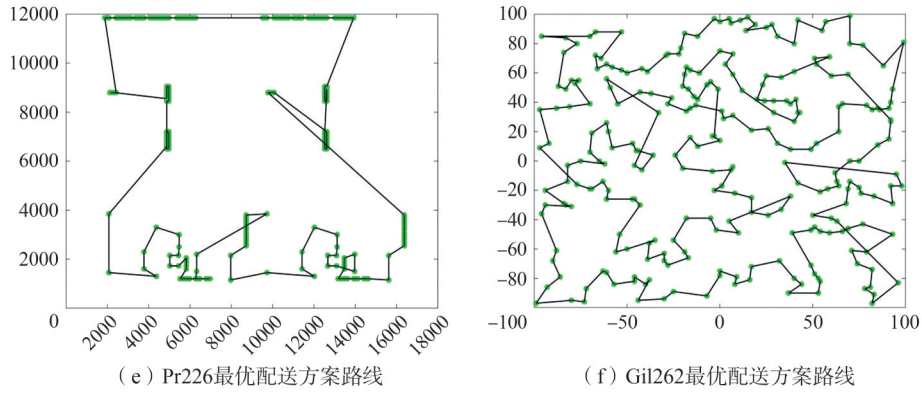


图 4 ACO算法寻优效果

Fig. 4 Optimization effect of the ACO algorithm

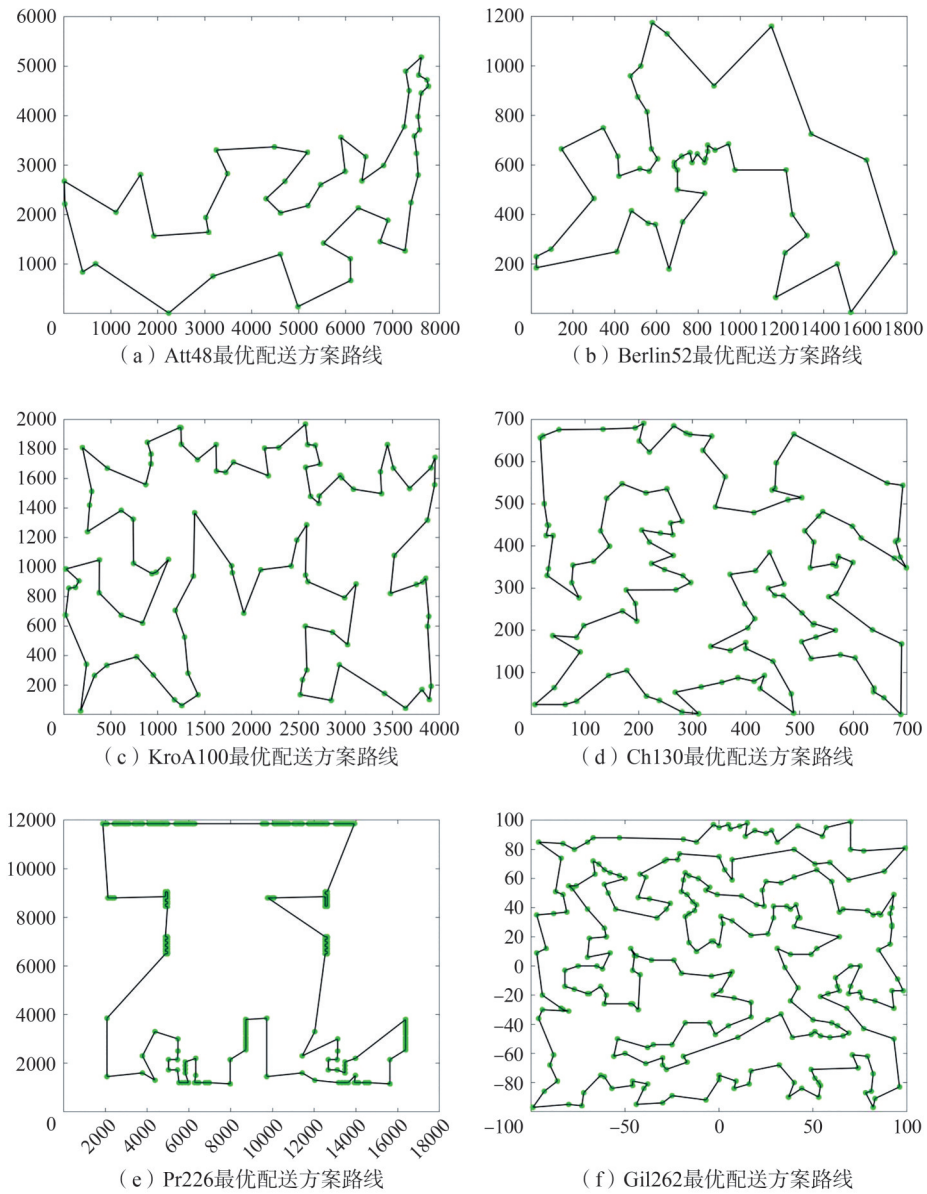


图 5 PSO-ACO-3opt算法寻优效果

Fig. 5 Optimization effect of the PSO-ACO-3opt algorithm

4.2.5 PSO-ACO-3opt算法与其他算法的对比实验

为再次验证所改进算法的有效性,将PSO-ACO-3opt算法与ACO-ABC^[12]算法、RAB-TSP^[13]算法得到的数据进行对比,具体实验结果如表5所示。通过对5个不同规模和分布情况的实例进行对比,可以看出,在求解Berlin52测试实例时,ACO-ABC算法取得了较其他两种算法更好的结果;在剩余的4个实例中,PSO-ACO-3opt算法所得的*Avg*、*SD*(标准差)、*Error*均优于其他两种算法。此外,本文所提算法在求解表5中所有测试实例的结果均优于RAB-TSP算法。因此,可以得出,当测试实例规模较大时,PSO-ACO-3opt算法具有一定的优势。

表5 PSO-ACO-3opt与其他算法数值实验结果对比

算法	指标	Eil51	Berlin52	Eil76	KroA100	KroB150
PSO-ACO-3opt	<i>Avg</i>	433.05	7549.40	550.50	21395.75	26506.75
	<i>SD</i>	3.07	16.20	3.60	86.14	121.45
	<i>Error</i> /%	1.65	0.10	2.32	0.53	1.19
ACO-ABC	<i>Avg</i>	443.40	7544.40	558.00	22435.30	-
	<i>SD</i>	5.25	0	4.10	231.30	-
	<i>Error</i> /%	4.08	0.03	3.17	5.42	-
RAB-TSP	<i>Avg</i>	437.47	7932.50	-	21522.73	26631.87
	<i>SD</i>	4.20	277.25	-	93.34	232.86
	<i>Error</i> /%	2.69	5.18	-	1.13	1.92

5 总结

本文针对旅行商问题,提出一种PSO-ACO-3opt算法。该算法以蚁群算法为框架,利用改进的粒子群算法优化蚁群算法的参数。为更好地平衡信息素和启发因子之间的关系,对状态转移概率公式进行了改进。当算法陷入局部最优时,通过信息素重置比例因子帮助算法跳出局部最优。然后,借助3-opt算法对蚂蚁生成的路径应用3-opt局部搜索,通过片段重组进一步优化路径长度,提高了算法的求解性能。最后,将PSO-ACO算法与PSO-ACO-3opt算法进行比较,证明了本文算法的有效性。但在个别实例中本文提出的算法表现得还不够理想,因此,在未来的研究中将继续对混合算法进行改进,以提升算法在不同场景实例下的求解性能。

参考文献:

- [1] 张虎,吕丽平. 基于改进长果蝇算法的冗余机器人逆运动学求解[J]. 机械设计与研究,2021,37(3):74-77.
- [2] MENG X H, LI J, DAI X Z, et al. Variable neighborhood search for a colored traveling salesman problem[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(4): 1018-1026.
- [3] PANDIRI V, SINGH A. An artificial bee colony algorithm with variable degree of perturbation for the generalized covering traveling salesman problem[J]. Applied Soft Computing, 2019, 78: 481-495.
- [4] 张硕航,郭改枝,张朋. K-Means聚类下的改进蚁群算法优化TSP问题[J]. 内蒙古大学学报(自然科学版), 2021, 52(6): 609-616.
- [5] GONG X L, RONG Z H, WANG J, et al. A hybrid algorithm based on state-adaptive slime mold model and fractional-order ant system for the travelling salesman problem[J]. Complex & Intelligent Systems, 2023, 9(4): 3951-3970.
- [6] WANG Y, HAN Z P. Ant colony optimization for traveling salesman problem based on parameters optimization[J].

- Applied Soft Computing, 2021, 107:107439.
- [7] ZHANG T, ZHOU Y Q, ZHOU G, et al. Discrete mayfly algorithm for spherical asymmetric traveling salesman problem[J]. Expert Systems with Applications, 2023, 221:119765.
- [8] DONG R Y, WANG S S, WANG G Y, et al. Hybrid optimization algorithm based on wolf pack search and local search for solving traveling salesman problem[J]. Journal of Shanghai Jiaotong University (Science), 2019, 24(1): 41-47.
- [9] DAHAN F, EL HINDI K, MATHKOUR H, et al. Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem[J]. Sensors, 2019, 19(8):1837.
- [10] HOUGARDY S, ZAISER F, ZHONG X H. The approximation ratio of the 2-opt heuristic for the metric traveling salesman problem[J]. Operations Research Letters, 2020, 48(4):401-404.
- [11] 杨晓光, 张奇松, 张益民, 等. 基于改进混合粒子群算法的云计算任务调度问题研究[J]. 内蒙古大学学报(自然科学版), 2019, 50(6):674-678.
- [12] GUNDÜZ M, KIRAN M S, ÖZCEYLAN E. A hierarchic approach based on swarm intelligence to solve the traveling salesman problem[J]. Turkish Journal of Electrical Engineering and Computer Sciences, 2015, 23(1):103-117.
- [13] PASTI R, NUNES DE CASTRO L. A Neuro-immune network for solving the traveling salesman problem[C]// The 2006 IEEE International Joint Conference on Neural Network Proceedings. Vancouver, BC, Canada: IEEE, 2006:3760-3766.

(责任编辑 黄宝琦)

Hybrid Ant Colony Optimization Algorithm for Solving the Traveling Salesman Problem

GUO Naping^{1,2}, MA Xiaohua², GAO Yuelin^{1,2}

(1. *College of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China;*
2. *Ningxia Collaborative Innovation Center for Scientific Computing and Intelligent Information Processing, Yinchuan 750021, China*)

Abstract: To address the slow convergence and susceptibility to local optima in ant colony optimization (ACO) when solving the traveling salesman problem (TSP), an improved ACO algorithm optimized with particle swarm parameters is proposed. Firstly, adaptive weights are introduced to enhance the global and local search capabilities of the particle swarm optimization (PSO). Secondly, a composite function state transition formula is used to balance the relationship between pheromone and heuristic factors, improving the algorithm's robustness and global search capability. Additionally, a pheromone reset ratio factor is included to enhance the ants' exploratory ability, preventing premature convergence to local optima. Finally, the 3-opt local search strategy is employed to further optimize the generated paths. Experimental results show that the improved algorithm outperforms the basic ACO in terms of performance on the TSPLIB dataset.

Key words: traveling salesman problem; ant colony optimization algorithm; particle swarm optimization algorithm; adaptive; 3-opt