

6-28-2024

Cooperative Ant Colony Algorithm Combining Evaluation Reward and Punishment Mechanism and Neighborhood Dynamic Degradation

Yujie Wang

Shanghai University of Engineering Science, Shanghai 201620, China, Wj13942013593@163.com

Xiaoming You

Shanghai University of Engineering Science, Shanghai 201620, China, Yxm7520253@163.com

Sheng Liu

Shanghai University of Engineering Science, Shanghai 201620, China

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

Cooperative Ant Colony Algorithm Combining Evaluation Reward and Punishment Mechanism and Neighborhood Dynamic Degradation

Abstract

Abstract: To address the slow convergence and the tendency to fall into local optimality in solving TSP, a cooperative ant colony algorithm combining evaluation reward and punishment mechanism and neighborhood dynamic degradation (ENCACO) is proposed. The paths are classified into active and abandon paths according to the path evaluation value, and with the path evaluation value as the weight, the different pheromone reward and punishment strategies are adopted for the two types of paths to accelerate the convergence speed of the algorithm. Through the neighborhood dynamic degradation strategy, and the neighborhood radius is used to divide the set of cities into exploration and degradation zones. The search range of ants is adaptively reduced, some cities in the degradation zone are dynamically refined by retention probability, and the state transfer probability is calculated together with the population to balance the convergence speed and the population diversity of the algorithm. The interspecies co-evolution mechanism is adopted to determine the interaction period between populations according to Tanimoto correlation coefficient, and the appropriate interaction is selected at different stages of the algorithm to help the algorithm jump out of the local optimum and improve the solution accuracy to achieve the purpose of effective communication between populations.

Keywords

evaluation of rewards and punishment, neighborhood degradation, Tanimoto correlation coefficient, co-evolution, ant colony algorithm, TSP

Recommended Citation

Wang Yujie, You Xiaoming, Liu Sheng. Cooperative Ant Colony Algorithm Combining Evaluation Reward and Punishment Mechanism and Neighborhood Dynamic Degradation[J]. Journal of System Simulation, 2024, 36(6): 1475-1492.

结合评估奖惩机制和邻域动态退化的协同蚁群算法

王育洁, 游晓明*, 刘升

(上海工程技术大学, 上海 201620)

摘要: 针对蚁群算法在求解旅行商问题中出现收敛速度慢以及易陷入局部最优等问题, 提出一种结合评估奖惩机制和邻域动态退化的协同蚁群算法。根据路径评估值将路径划分为活跃路径和舍弃路径, 以路径评估值作为权重对两类路径采取不同的信息素奖惩策略, 加快算法的收敛速度。采用邻域动态退化策略, 利用邻域半径将城市集分为探索区和退化区, 自适应缩小蚂蚁的搜索范围, 通过保留概率动态保留部分退化区中的城市, 结合探索区中的城市一并计算状态转移概率, 平衡算法的收敛速度和种群的多样性。采取种间协同进化机制, 根据 Tanimoto 相关系数确定种群间的交互周期, 并在算法的不同阶段选择合适的交互方式帮助算法跳出局部最优, 提高算法的求解精度, 达到种群间有效交流的目的。

关键词: 评估奖惩; 邻域退化; Tanimoto 相关系数; 协同进化; 蚁群算法; 旅行商问题

中图分类号: TP18 文献标志码: A 文章编号: 1004-731X(2024)06-1475-18

DOI: 10.16182/j.issn1004731x.joss.23-0524

引用格式: 王育洁, 游晓明, 刘升. 结合评估奖惩机制和邻域动态退化的协同蚁群算法[J]. 系统仿真学报, 2024, 36(6): 1475-1492.

Reference format: Wang Yujie, You Xiaoming, Liu Sheng. Cooperative Ant Colony Algorithm Combining Evaluation Reward and Punishment Mechanism and Neighborhood Dynamic Degradation[J]. Journal of System Simulation, 2024, 36(6): 1475-1492.

Cooperative Ant Colony Algorithm Combining Evaluation Reward and Punishment Mechanism and Neighborhood Dynamic Degradation

Wang Yujie, You Xiaoming*, Liu Sheng

(Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: To address the slow convergence and the tendency to fall into local optimality in solving TSP, a cooperative ant colony algorithm combining evaluation reward and punishment mechanism and neighborhood dynamic degradation (ENCACO) is proposed. The paths are classified into active and abandon paths according to the path evaluation value, and with the path evaluation value as the weight, the different pheromone reward and punishment strategies are adopted for the two types of paths to accelerate the convergence speed of the algorithm. Through the neighborhood dynamic degradation strategy, and the neighborhood radius is used to divide the set of cities into exploration and degradation zones. The search range of ants is adaptively reduced, some cities in the degradation zone are dynamically refined by retention probability, and the state transfer probability is calculated together with the population to balance the convergence speed and the population diversity of the algorithm. The interspecies co-evolution mechanism is adopted to determine the interaction period between populations

收稿日期: 2023-05-06 修回日期: 2023-08-06

基金项目: 国家自然科学基金(61673258, 61075115); 上海市自然科学基金(19ZR1421600)

第一作者: 王育洁(2000-), 女, 硕士生, 研究方向为智能算法、移动机器人路径规划。E-mail: WYj13942013593@163.com

通讯作者: 游晓明(1963-), 女, 教授, 硕士生导师, 博士, 研究方向为群智能系统、进化算法。E-mail: Yxm7520253@163.com

according to Tanimoto correlation coefficient, and the appropriate interaction is selected at different stages of the algorithm to help the algorithm jump out of the local optimum and improve the solution accuracy to achieve the purpose of effective communication between populations.

Keywords: evaluation of rewards and punishment; neighborhood degradation; Tanimoto correlation coefficient; co-evolution; ant colony algorithm; TSP

0 引言

旅行商问题(TSP)^[1]是组合优化中经典的NP难问题,旅行商人从规定的城市列表中选择起点出发,不重复地遍历所有城市,最终返回起点城市,并找到其中最短的闭合回路。目前,有大量的智能优化方法应用于TSP问题的解决,例如,模拟退火算法、遗传算法、蚁群算法和粒子群算法等。其中,由于蚁群算法具有较强的鲁棒性、良好的并行性、算法正反馈性和易于与其他算法结合等优点,成为求解TSP问题最有效的方法之一,进而被广泛应用于车间调度问题^[2]、分配问题^[3]、路径规划问题^[4]、多目标优化^[5]等组合优化问题。

文献[6]提出蚂蚁系统(ant system, AS),他们在研究蚂蚁觅食行为的过程中发现,虽然一只蚂蚁的行为较为简单,但蚁群整体却可以体现出群体智能。然而,蚂蚁系统存在易陷入局部最优、收敛速度慢等明显缺陷。文献[7]提出改进的蚁群系统(ant colony system, ACS),采用伪随机比例进行状态转移并使用全局和局部两种信息素更新方式,一定程度上加快了算法的收敛速度。文献[8]提出了最大最小蚂蚁系统(max-min ant system, MMAS),它作为AS的改进算法,对信息素取值设定最大值和最小值,将其控制在一个区间内,避免信息素过度累积,从而提高算法的多样性。虽然以上经典算法已具备较强的搜索能力,但在解决大规模TSP问题时仍存在收敛速度慢、易陷入局部最优等问题。

针对上述不足,大量学者采用不同策略对算法进行改进并取得一定成果。文献[9]在每次迭代中引入异构种群自动化,以生成更好的候选解,同时引入差分信息更新机制,利用从候选解中获

得的差分边缘信息合理指导信息素轨迹挥发,并在状态调整阶段应用进化状态估计和调整策略监测停滞状态的出现,增加及时探索全局最优的可能性。文献[10]提出相遇蚂蚁的概念,当两只蚂蚁禁忌表中城市之和为完整城市集时跳出当前迭代,形成一个新的旅程,两只蚂蚁共同搜索新路径将有效缩短搜索时间,加快算法收敛速度,同时蚁群的搜索多样化也得到一定的扩展。文献[11]提出一种基于秩的蚂蚁系统,将蚂蚁按照适应度排序,排名与蚂蚁释放的信息素数量呈正相关,则蚂蚁排名越高其路径被选择的概率就越大,一定程度上提高了解的精度,并将模糊逻辑和改进的蚁群算法相结合,使虚拟路径中的动态因素精确数字化。文献[12]自适应优化禁忌搜索算子的调用,设计了5个邻域算子,在并行的蚁群算法中实现串行搜索,提高了算法的收敛速度,并引入模拟退火机制,调用知识模型重新优化找到的路线集,根据优化的解集更新全局信息素,增加种群多样性。以上改进的蚁群算法虽然在收敛速度和解的精度上均有一定的提升,但对单一种群进行优化仍存在一定的局限性。

为了平衡收敛速度和精度以达到更好的效果,有学者提出结合多策略的多种群蚁群优化算法。文献[13]提出一种自适应制导机制,使用动态评估网络对解进行评估和划分,按照评估值采取正负激励策略,增强更高评估值解的领导作用,并提出一种基于博弈模型的种间协同进化机制跳出局部最优,提高解的精度。文献[14]根据种群间的汉明距离自适应调节异构多种群的交互周期,采用竞争交互策略并根据竞争系数匹配交互对象,平衡算法速度和精度,同时对多种群采取淘汰重组

机制, 提高算法的求解精度。文献[15]使用有限历史存档策略增强路径引导, 实现算法快速收敛, 并通过智猪博弈实现种群间的协同进化, 提高算法性能, 通过均衡机制帮助算法在后期跳出局部最优, 提高解的精度。文献[16]对种群采取分工合作机制, 将种群按比例划分为采用不同策略搜索的子种群, 引入分级更新策略提高算法求解速度, 设计自适应变异算子, 采用随机扰动策略增加算法多样性。以上改进的蚁群算法虽取得了一些成果, 但随着 TSP 问题规模的扩大, 算法的求解精度和收敛速度之间无法很好平衡, 解的质量有待提高。因此, 本文提出一种结合评估奖惩机制和邻域动态退化的协同蚁群算法 (cooperative ant colony optimization combining evaluation reward and punishment mechanism and neighborhood dynamic degradation, ENCACO)。

1 相关工作

1.1 蚁群系统(ACS)

1.1.1 路径构建规则

蚂蚁在 ACS 中采用伪随机状态转移策略选择下一个待访问的城市 j , 该状态转移策略为

$$S = \begin{cases} \arg \max_{j \in C_k} \{\tau_{ij} \cdot \eta_{ij}^\beta\}, & q \leq q_0 \\ s, & q > q_0 \end{cases} \quad (1)$$

式中: τ_{ij} 为城市 i 和城市 j 间的信息素浓度值; β 为期望启发式因子, 它的大小决定启发式信息在蚂蚁选择城市时的影响程度; C_k 为蚂蚁 k 可以选择的城市的集合; q 为一个随机数, 分布在 $[0, 1]$ 之间, 当 $q \leq q_0$ 时, 蚂蚁会倾向于选择信息素浓度较高且距离较短的城市作为下一个要访问的城市, 当 $q > q_0$ 时, 蚂蚁则会按照轮盘赌的方式选择城市; q_0 ($0 \leq q_0 \leq 1$) 为一个可调参数, 其值影响蚂蚁选择路径的方式, 从而影响算法的收敛速度和多样性; s 为根据式(3)按轮盘赌规则选择的一个随机变量; η_{ij} 为算法的启发式信息。

$$\eta_{ij} = 1/d_{ij} \quad (2)$$

式中: d_{ij} 为城市 i 和城市 j 间的欧式距离。

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{l \in C_k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}, & j \in C_k \\ 0, & j \notin C_k \end{cases} \quad (3)$$

式中: P_{ij}^k 为蚂蚁 k 在城市 i 通过轮盘赌方式选择城市 j 的概率; α 为信息素启发式因子, 它的大小决定信息素在蚂蚁选择城市时的影响程度。

1.1.2 信息素更新规则

(1) 局部信息素更新

蚂蚁在完成一次遍历后, 对它走过的路径进行局部信息素更新, 减少当前路径上的信息素, 增加算法的多样性。局部信息素更新式为

$$\tau_{ij}(t+1) = (1 - \zeta)\tau_{ij}(t) + \zeta\tau_0 \quad (4)$$

式中: ζ 为局部信息素挥发系数, 取值范围 $[0, 1]$; τ_0 为初始信息素浓度。

(2) 全局信息素更新

当所有蚂蚁均完成城市遍历后, ACS 会对最优路径进行全局信息素更新, 全局信息素更新式为

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} \quad (5)$$

$$\Delta\tau_{ij} = \begin{cases} 1/L_{gb}, & i, j \in L_{gb} \\ 0, & \text{else} \end{cases} \quad (6)$$

式中: ρ 为全局信息素挥发因子; $\Delta\tau_{ij}$ 为最优路径上的信息素增量; L_{gb} 为全局最优路径的长度。

1.2 最大最小蚂蚁系统(MMAS)

1.2.1 信息素范围限制

MMAS 将信息素限制在 $[\tau_{\min}, \tau_{\max}]$ 范围内, 其中, τ_{\min} 保证每条路径都有一定的信息素值, 可增加解的多样性, 避免算法停滞; τ_{\max} 可避免某些路径上的信息素过度累积使算法多样性下降。MMAS 将信息素取值规范化, 若 $\tau_{ij} < \tau_{\min}$, 则将其值更新为 τ_{\min} ; 若 $\tau_{ij} > \tau_{\max}$, 则将其值更新为 τ_{\max} 。

$$\tau_{\max} = (1/\rho) \times (1/L_{gb}) \quad (7)$$

$$\tau_{\min} = \tau_{\max}/2n \quad (8)$$

式中: n 为城市数。

1.2.2 信息素更新规则

MMAS 只更新全局最优路径或迭代最优路径上的信息素，信息素更新公式为

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

$$\Delta\tau_{ij}^{\text{best}} = 1/L_{\text{gb}} \quad (10)$$

式中： L_{gb} 为全局最优或迭代最优路径长度。

1.3 Tanimoto 相关系数

Tanimoto 相关系数是一种用来评估两个集合相似度的方法，通常用于度量两组数据之间的相关程度，表示的是 X 、 Y 的公共特征占 X 、 Y 所有特征的比例，Tanimoto 相关系数的取值范围为 $[0, 1]$ ，其计算方式为

$$\text{Tan}(X, Y) = \frac{X \cdot Y}{\|X\|^2 + \|Y\|^2 - X \cdot Y} \quad (11)$$

式中： X 、 Y 分别为 2 个 n 维向量； $\|X\|$ 和 $\|Y\|$ 分别为 X 和 Y 的向量范数。

$$\|X\| = \sqrt{\sum_{i=1}^n X_i^2} \quad (12)$$

目前有很多相似度的计算方法，例如，Pearson 相关系数(式(13))、余弦相似度(式(14))等。相比来说，由于上述方法的分子均为乘积形式，即若分母在计算过程中为 0 时，会导致结果出错，不利于算法求解，而 Tanimoto 相关系数采用加减法构造分母，因此避免了上述问题的出现。

$$r_{xy} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\left(\sqrt{\sum (X - \bar{X})^2}\right)\left(\sqrt{\sum (Y - \bar{Y})^2}\right)} \quad (13)$$

$$\cos(x, y) = \frac{\sum_{i=1}^n (x_i \cdot y_i)}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (14)$$

Tanimoto 相关系数还具有计算简单，在处理稀疏数据时较为有效等优势，因此本文使用 Tanimoto 相关系数计算两种群间的相似度，用于种群进化程度的度量。相似系数越低，说明种群间差异越大，解的多样性越好；反之，相似系数越高，说明种群进化程度较高，两种群差异较小，

算法易于陷入局部最优。

2 改进的蚁群算法

本文在传统蚁群算法的基础上提出一种多策略融合的协同蚁群算法，采用 ACS 和 MMAS 组成双种群协同演化。由于 ACS 算法收敛速度较快，而 MMAS 算法拥有较好的多样性，因此，将两类种群的优势结合互补，并采取有效策略促进两种群交互，从而提高算法在大规模 TSP 问题上寻优的能力。

2.1 自适应评估奖惩机制

2.1.1 自适应路径评估

每次迭代结束后，通过式(15)计算蚂蚁得到的解(路径)的评估值，并按评估值将路径分类。路径评估函数为

$$E_i = (L_i - L_{\text{best}})/L_{\text{best}} \quad (15)$$

式中： E_i 为每条路径的评估值； L_i 为第 i 次迭代时蚂蚁得到的路径长度； L_{best} 为当前最优解的长度。 E_i 取值在 $[0, 1]$ 之间，且路径评估值越小，则 L_i 越小，即得到解的质量越好。

设置阈值 θ ，将每条路径的评估值与之比较。若 $E_i \leq \theta$ ，则该路径为活跃路径；反之，路径为舍弃路径。该机制示意图如图 1 所示。

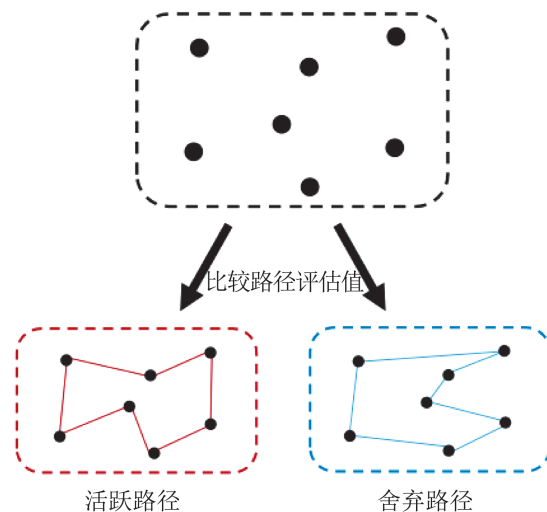


图1 自适应路径评估

Fig. 1 Adaptive path evaluation

2.1.2 基于路径评估值的信息素奖惩机制

信息素是蚁群算法中蚂蚁寻找路径时必不可少的信息指引, 传统蚁群算法的信息素更新规则相对简单, 无法体现出较优路径的优势, 导致算法搜索时间长, 收敛速度慢。本节提出一种基于路径评估值的信息素奖惩策略, 即经过路径的自适应评估后, 对不同类型的路径采用不同的信息素更新方式: 若该路径被评为活跃路径, 则对它进行额外的信息素奖励; 若该路径被评为舍弃路径, 则会受到相应的信息素惩罚。

$$\Delta\tau_{ij}^{\text{active}} = E_i \cdot \Delta\tau_{ij} \quad (16)$$

$$\Delta\tau_{ij}^{\text{abandon}} = -E_i^2 \cdot \Delta\tau_{ij} \quad (17)$$

式中: $\Delta\tau_{ij}^{\text{active}}$ 为活跃路径上的信息素增量; $\Delta\tau_{ij}^{\text{abandon}}$ 为舍弃路径上的信息素增量; $\Delta\tau_{ij}$ 为最优路径上的信息素增量, $\Delta\tau_{ij} = 1/L_{\text{gb}}$ 。

在改进的信息素更新策略中, 使用路径的评估值作为权重对不同路径采取不同的更新规则。由于活跃路径是包含最优路径在内的全局较优路径的集合, 故在算法运行过程中, 对活跃路径采取一定的正向信息素增量, 增加活跃路径对蚂蚁的吸引力, 提高算法收敛速度。由于路径越优, 其评估值越小, 例如最优路径的评估值为0。由式(16)可知, 活跃路径集合中较差路径信息素奖励较多, 而较优路径信息素奖励较少, 增加了了解的多样性, 提高了算法精度, 避免算法陷入局部最优。对舍弃路径施加反向信息素增量, 减少其对蚂蚁的吸引力, 由式(17)可知, 舍弃路径集合中较差路径将受到较多的信息素惩罚, 其被探索的可能性大幅降低, 可有效提高算法的收敛速度。随着迭代的不断进行, 活跃路径上的信息素不断累积, 引导蚂蚁快速寻得最优解, 而舍弃路径随着迭代的进行逐渐被淘汰, 从而减少蚂蚁的搜索范围, 有效地提高了算法的收敛速度, 缩短了求解时间。

2.2 基于保留概率的邻域动态退化策略

由于蚂蚁在选择路径时会对待访问城市列表中的所有城市计算状态转移概率, 对于小规模城

市集来说, 城市之间相对聚集, 不会产生太多冗余的时间开销, 而对于城市规模较大的问题, 依次计算所有城市的转移概率会增加不必要的运行时间, 导致算法收敛速度变慢。针对此问题, 本节提出一种邻域动态退化策略, 以当前城市为中心, 根据邻域半径动态划分区域, 对半径内的城市计算转移概率, 同时按照保留概率随机失活半径外邻域中的部分城市, 即将其移出待访问城市列表, 不再计算其转移概率。该策略有效缩小了蚂蚁的搜索范围, 同时也保留了一定的探索可能性, 在提高收敛速度的同时平衡了解的多样性。

2.2.1 保留概率

在深度神经网络中, 由于节点数较多易出现过拟合现象, 一般采用随机失活策略^[17]将隐含层的部分权重或输出随机归零, 且在遍历每层节点时设置一个节点保留概率, 保证神经网络不会偏向于某一个节点, 实现神经网络的正则化。在进行城市邻域动态退化策略时引入保留概率, 按保留概率在退化区内随机保留部分城市继续探索, 并将退化区中剩余城市移出待访问城市列表, 不再计算其状态转移概率, 达到随机失活的目的。保留概率的计算公式为

$$P_{\text{remain}} = n_{\text{out}}/n \quad (18)$$

式中: n_{out} 为在退化区中的城市个数。

保留概率随着退化区中的城市数量占有所有城市的比例动态改变, 退化区中城市占有所有城市的比例越大, 则保留概率越大; 反之, 保留概率越小。

2.2.2 邻域动态退化

邻域动态退化策略的工作方式如图2所示。该策略仅针对在待访问城市列表中的城市。以当前蚂蚁所在城市为中心, 邻域半径 R 为半径的圆内范围称为探索区, 在圆外范围称为退化区。图2中用黑色细实线与蚂蚁当前所在城市相连的黑色城市即为蚂蚁下一步可选的城市集合, 而在退化区中的白色城市为邻域动态退化的结果。

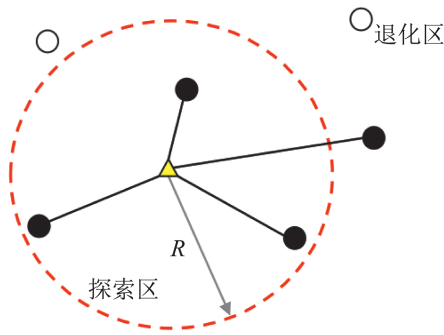


图2 邻域动态退化策略

Fig. 2 Neighborhood dynamic degradation strategy

邻域半径 R 随迭代的进行自适应变化:

$$R = D[\sin(i/i_{\max}) + 0.5] \quad (19)$$

$$D = \frac{1}{2} \cdot [\max(d_{i,n}) + \min(d_{i,n})] \quad (20)$$

式中: i 为当前迭代次数; i_{\max} 为最大迭代次数; $\max(d_{i,n})$ 为当前城市 i 与其距离最远城市间的距离; $\min(d_{i,n})$ 为当前城市 i 与其距离最近城市间的距离; D 为调节因子。

由于邻域半径 R 和保留概率 P_{remain} 均随参数动态变化, 因此在算法的整个运行过程中, 邻域动态退化策略在不同时期发挥不同的作用。在算法初期, 各路径信息素浓度较为平均, 蚂蚁选择路径概率相对均等, 此时邻域半径设置较小, 使蚂蚁在探索区的较小范围内寻找下一个目标城市, 算法快速收敛至最优解。同时, 为了避免退化区城市的过度影响, 保留概率保持在一个相对较低的水平, 从而加快算法的收敛速度。在算法后期, 由于较优路径上信息素累积较多, 蚂蚁逐渐集中在较优解附近, 算法易陷入局部最优, 此时邻域半径自适应扩大, 蚂蚁搜索范围增大, 探索区城市选择可能性增多。同时, 保留概率也相应增大, 按较大比例保留退化区城市, 增加解的多样性, 提高算法的求解精度。邻域半径随着迭代进行自适应增大, 其能达到的最大值为当前城市与其最远城市距离及与其最近城市距离之和的一半。因此, 在迭代后期为了跳出局部最优, 即使蚂蚁选择范围扩大, 最终也不会选择较远城市作为其目

标城市, 因此邻域动态退化策略不仅不会降低求解精度, 反而提高了算法求解速度。

通过引入邻域动态退化策略, 算法动态调节邻域半径和保留概率, 根据进化的不同阶段控制蚂蚁搜索范围的大小和内容, 有效平衡了算法收敛速度和求解精度。

2.3 种间协同进化机制

在进行 ACS 和 MMAS 双种群蚁群算法过程中, 为保证算法的搜索质量和效率, 需要定期进行种群间的信息交互。本节提出一种种间协同进化机制, 使用 Tanimoto 相关系数判断两个种群的进化程度, 确定交互周期, 选择合适的交互方式和内容, 促进种群间的信息交流, 实现协同进化。

2.3.1 Tanimoto 相关系数

使用 Tanimoto 相关系数判断两种群得到的最优解的相似度, 判断种群的进化程度。Tanimoto 相关系数越小, 说明两种群的最优解差异越大, 最优解存在较少的重叠路径, 种群进化程度低, 两种群多样性较好; Tanimoto 相关系数越大, 说明两种群得到的最优解存在较多的重叠路径, 种群间相似度较高, 具有较高水平的进化程度, 算法易陷入局部最优。两种群间的 Tanimoto 相关系数计算式为

$$\begin{cases} \text{Tan}(\mathbf{X}, \mathbf{Y}) = \frac{\mathbf{X} \cdot \mathbf{Y}}{\|\mathbf{X}\|^2 + \|\mathbf{Y}\|^2 - \mathbf{X} \cdot \mathbf{Y}} \\ \mathbf{X} = \text{Trace}_{\text{ACS}}^{\text{best}}[C_1, C_2, \dots, C_n] \\ \mathbf{Y} = \text{Trace}_{\text{MMAS}}^{\text{best}}[C_1, C_2, \dots, C_n] \\ \mathbf{X} \cdot \mathbf{Y} = \text{Trace}_{\text{ACS}}^{\text{best}} \cap \text{Trace}_{\text{MMAS}}^{\text{best}} \end{cases} \quad (21)$$

式中: $\text{Trace}[C^1, C^2, \dots, C^n]$ 为组成某个解的城市队列; \mathbf{X} 为 ACS 种群组成最优解的城市队列; \mathbf{Y} 为 MMAS 种群组成最优解的城市队列; $\mathbf{X} \cdot \mathbf{Y}$ 为两种群最优解城市队列的重叠路径。

2.3.2 自适应交互周期

在双种群蚁群算法中, 为了提高算法的求解精度, 确定交互周期尤为重要。若交互周期较大,

两种群之间交流较弱, 会削弱种群间的学习促进作用; 若交互周期较小, 两种群交流频繁, 会造成算法多样性较低, 导致求得的解出现同质化现象。因此, 综合考虑种群间的进化程度以及迭代次数等因素, 确定种群间的交互周期:

$$T = 150 \cdot \left[\frac{i_{\max} - i}{i_{\max}} \times \frac{1}{\Theta_{\max} - \Theta_{\text{ave}}} \right] \quad (22)$$

$$\Theta_{\text{ave}} = \frac{1}{i_{\max}} \cdot \sum_{i=1}^{i_{\max}} \Theta_i \quad (23)$$

式中: i_{\max} 为最大迭代次数; i 为当前迭代次数; Θ_{\max} 为当前最大 Tanimoto 相关系数值; Θ_{ave} 为当前平均 Tanimoto 相关系数值; Θ_i 为当前迭代时的 Tanimoto 相关系数值。

算法运行初期, 种群间多样性较好, 蚂蚁有较大的潜力各自寻得最优解, 因此, 交互周期应间隔较大, 保证解的多样性; 算法运行后期, 路径信息素累积较为集中, 算法易陷入局部最优, 此时交互周期应间隔较小, 促进种群间的交流, 保证求解质量。通过 Tanimoto 相关系数判断种群间多样性, 值越小, 则种群相似度越低, 此时给予种群适当空间探索各自的最优解; 值越大, 则进化程度越高, 此时促进种群间交流, 提高算法求解精度。

2.3.3 交互方式的确定

在算法运行前期, 即 Tanimoto 相关系数较小时, 此时希望算法快速收敛至最优解, 因此采用交换两种群间最优解的方式进行, 即 $\text{Solution}_{\text{ACS}}^{\text{best}} \leftrightarrow \text{Solution}_{\text{MMAS}}^{\text{best}}$ 。交互后, 较优种群的较优解会引导较差种群搜索, 同时较差种群的解也会为较优种群提供多样性, 共同起到协同进化的作用。

在算法后期, 即 Tanimoto 相关系数较大时, 经过自适应评估奖惩机制进行路径评估和信息素奖惩后, ACS 种群中的最优路径和最差路径间的信息素浓度差距较大, 算法易陷入局部最优。而由于 MMAS 可以控制路径上信息素的最大值和最小值, 因此路径上信息素不会过度累积也不会过度挥发,

每段路径间的信息素差异不会过于明显, 即信息素矩阵较为均匀。此时通过交流两种群间的信息素矩阵进行信息交互, 保持 MMAS 种群的信息素矩阵不变, 将 MMAS 种群的信息素矩阵通过系数加到 ACS 种群的信息素矩阵中形成新的信息素矩阵:

$$\Gamma_{\text{MMAS}}^{\text{new}} = \Gamma_{\text{MMAS}} \quad (24)$$

$$\Gamma_{\text{ACS}}^{\text{new}} = \Gamma_{\text{ACS}} + \frac{\Theta_i}{\Theta_{\max} + \Theta_{\min}} \Gamma_{\text{MMAS}} \quad (25)$$

式中: $\Gamma_{\text{MMAS}}^{\text{new}}$ 为交互完成后 MMAS 的信息素矩阵; Γ_{MMAS} 为交互之前 MMAS 的信息素矩阵; $\Gamma_{\text{ACS}}^{\text{new}}$ 为交互完成后 ACS 的信息素矩阵; Γ_{ACS} 为交互之前 ACS 的信息素矩阵。

由于蚁群算法的正反馈特性, 随着迭代的进行, 信息素会大量累积在几条较优路径上, 导致蚂蚁不断被吸引从而频繁选择该路径, 使算法陷入局部最优。根据迭代次数自适应选择交互方式, 在前期加快算法收敛速度, 在后期提高算法的多样性, 实现了两种群间的协同进化。

2.4 算法流程

本文提出的改进算法(ENCACO)的流程图如图3所示。

算法流程如下。

step 1: 初始化参数, 计算城市间距离;

step 2: 进行第一次迭代, 蚂蚁生成解并进行局部信息素更新;

step 3: 根据式(15)计算每条路径的评估值, 并根据评估值将路径划分为活跃路径和舍弃路径;

step 4: 对2种不同类型的路径根据不同的信息素更新规则进行全局更新;

step 5: 每次迭代后, 根据式(18)和式(19)计算保留概率和邻域半径, 并根据式(21)计算种间 Tanimoto 相关系数;

step 6: 根据邻域半径确定探索区和退化区, 并根据保留概率动态保留退化区中的城市;

step 7: 根据 Tanimoto 相关系数计算交互周期;

step 8: 根据交互周期和算法运行阶段选择合适的交互方式;

step 9: 判断是否达到最大迭代次数, 若是, 进入step 10, 否则转step 3;

step 10: 输出全局最优解。

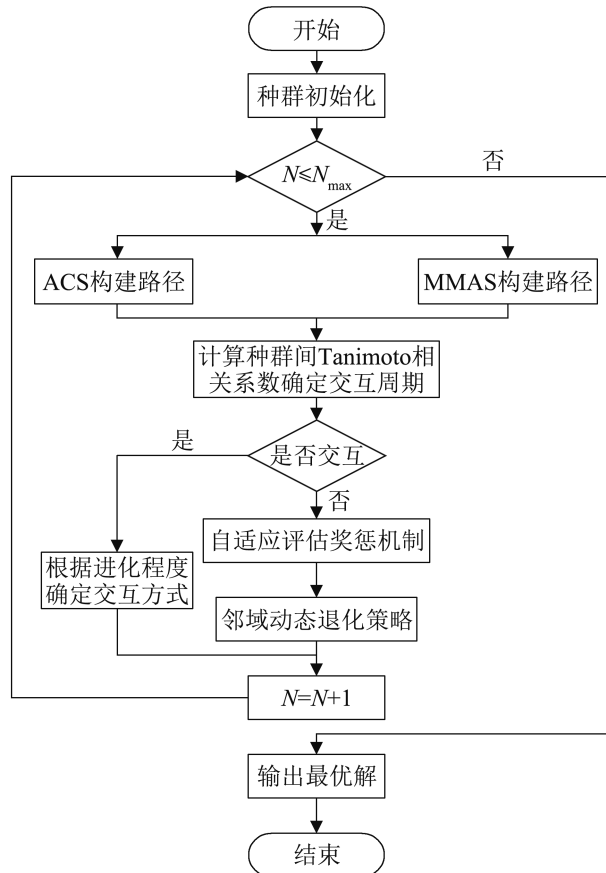


图3 算法流程图

Fig. 3 Algorithm flow chart

2.5 算法复杂度分析

通过对算法步骤以及算法流程图分析, 得到算法各步骤的时间复杂度如表1所示, 且ENCACO算法中ACS和MMAS两种群并行求解, 因此, 得出ENCACO的最大时间复杂度为 $O(m \times n^2 \times N_{\max})$, 其中, m 为蚂蚁个数, n 为TSP城市个数, N_{\max} 为最大迭代次数。由于传统算法ACS和MMAS的最大时间复杂度为 $O(m \times n^2 \times N_{\max})$, 因此, ENCACO算法没有增加算法的最大时间复杂度。

表1 ENCACO的时间复杂度
Table 1 Time complexity of ENCACO

内容	时间复杂度
参数初始化	$O(n^2 + m)$
设置禁忌表	$O(n)$
蚂蚁构建路径	$O(n^2 m)$
信息素更新	$O(n^2)$
计算路径评估值	$O(n)$
计算Tanimoto相关系数	$O(n)$
执行自适应评估奖惩机制	$O(n^2)$
执行邻域动态退化策略	$O(n^2)$
输出结果	$O(1)$

3 实验分析

本文使用MATLAB2018b进行代码编写和仿真调试。为了展示ENCACO的有效性以及在解决大规模TSP问题上的优势, 选取TSPLIB数据库中不同规模的测试集对算法的性能进行验证。

3.1 实验参数设置

3.1.1 公共参数设置

在ACS和MMAS的基础上对算法进行改进, 并通过采用不同参数对大量TSP实例进行对比仿真分析, 最终得出整体效果较好的算法解, 如表2所示, 后续的仿真实验中, 保持公共参数不变。

表2 ENCACO中的公共参数设置
Table 2 Public parameter settings in ENCACO

参数	α	β	ρ	ζ	q_0	N_{\max}	m
ACS	1	4	0.3	0.1	0.8	2 000	20
MMAS	1	5	0.2	null	null	2 000	20

3.1.2 评估阈值 θ 的确定

每次迭代结束后, 通过评估函数计算每条路径的评估值, 并将其与 θ 进行比较, 其中小于等于 θ 的路径为活跃路径, 在后续对其进行信息素奖励; 大于 θ 的路径为舍弃路径, 后续对其进行信息素惩罚。两类路径的数量比例对算法的收敛速度和求解精度都会产生一定的影响。若阈值设置较小, 算法中舍弃路径较多, 则受信息素惩罚

的路径较多, 算法收敛速度慢, 寻找最优解的时间较长; 算法中活跃路径较多, 信息素过多奖励在某些路径上, 会导致算法陷入局部最优的陷阱中, 不利于求解精度的提高。因此, 为了更好地平衡算法的收敛速度和求解精度, 本节通过实验仿真确定 θ 的最优取值, 以获得质量更优的解。

算法误差率的计算式为

$$\psi = (l_{\text{best}} - l_{\text{min}}) / l_{\text{min}} \times 100\% \quad (26)$$

式中: l_{best} 为算法得到的最优解; l_{min} 为 TSPLIB 标准最优解。

本节选取规模不同的 3 组 TSP 测试集 eil51、kroB150 和 lin318 进行仿真实验, 通过表 3 设定 θ 的不同组合, 分别测试解的精度, 实验结果如表 4 所示。

图 4 展示了不同阈值组合下 3 个测试集的收敛情况, 对于中规模测试集 kroB150, 由图 4(b) 可知,

图 5 展示了 3 个测试集下不同阈值组合的最优解和平均解对比。结合表 4 和图 4~5 可知, 对于小规模测试集 eil51, 虽然算法在 θ 的不同组合下均能找到最优解, 但从图 4(a) 能明显看出组合 C 曲线最先收敛至最优解, 且由图 5(a) 可知其找到解的平均值更优。

表 3 阈值 θ 的组合

Table 3 Combination of threshold θ		
阈值 θ 组合	阈值 θ 取值	
A	θ_{ACS}	0.18
	θ_{MMAS}	0.18
B	θ_{ACS}	0.20
	θ_{MMAS}	0.20
C	θ_{ACS}	0.20
	θ_{MMAS}	0.25
D	θ_{ACS}	0.30
	θ_{MMAS}	0.30

表 4 3 种城市集中不同阈值 θ 组合的结果

Table 4 Results for different combinations of thresholds θ for 3 urban concentrations

TSP	标准最优解	阈值 θ 组合	最优解	平均解	误差率%	迭代次数
eil51	426	A	426	426.9	0	422
		B	426	427.5	0	416
		C	426	426.8	0	60
		D	426	427.3	0	63
kroB150	26 130	A	26 249	26 324.3	0.46	1 634
		B	26 193	26 342.8	0.24	1 151
		C	26 130	26 296.9	0	867
		D	26 178	26 336.2	0.18	1 074
lin318	42 029	A	42 425	42 962.0	0.94	1 631
		B	42 634	43 013.1	1.44	1 635
		C	42 236	42 793.5	0.49	1 864
		D	42 342	42 949.9	0.74	1 628

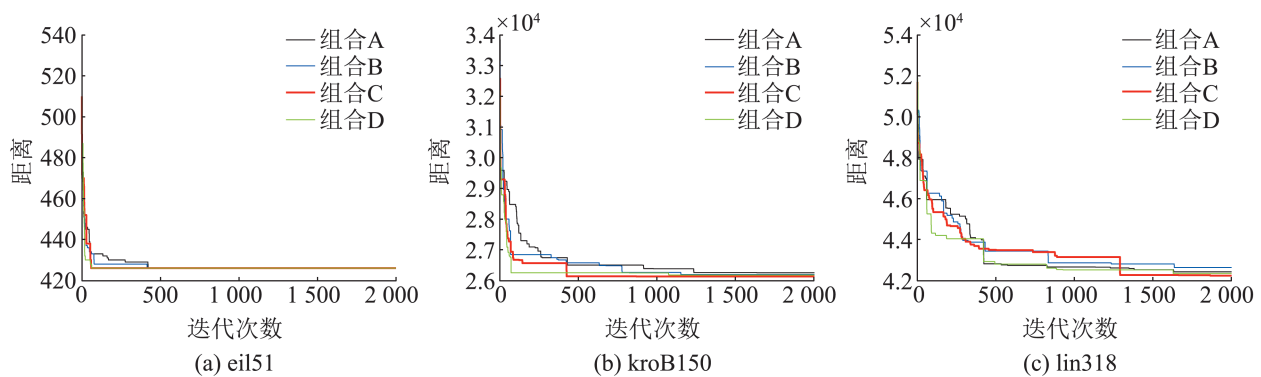


图 4 阈值 θ 不同组合的收敛图

Fig. 4 Convergence plots for different combinations of threshold θ

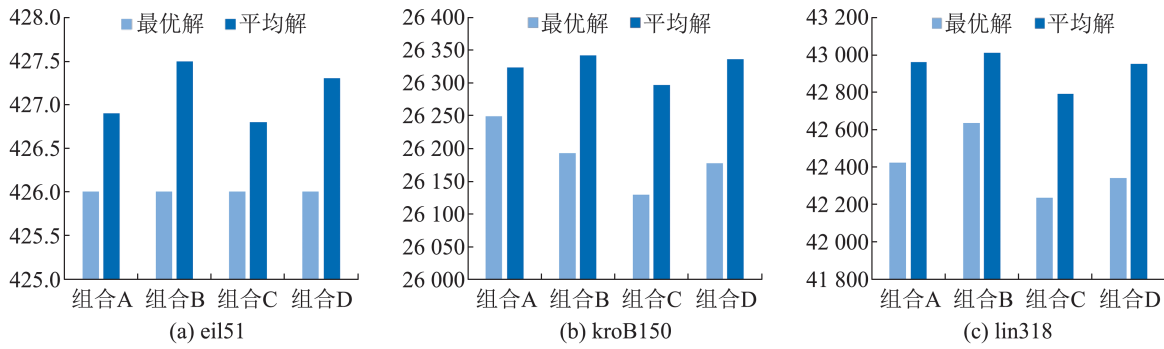


图5 3个测试集最优解及平均解对比图

Fig. 5 Comparison of optimal and average solutions of 3 test sets

在其他组合均未找到最优解的情况下，组合C仍可在最小迭代次数内寻得最优解。当 θ 设置偏小时，如组合A和组合B，算法收敛速度较慢，且后期跳出局部最优的能力差。当 θ 设置偏大时，如组合D，算法虽在初期收敛速度较快，但在1100代左右陷入局部最优，进而停滞，求解精度较差。且从图5(b)可以看出，组合C得到的最优解和平均解均更优。

在较大规模测试集lin318中，由图4(c)可以看出，对于组合B，算法收敛速度较慢，且解的精度不高。对于组合D，虽然前期收敛速度较快，但算法在100代左右就陷入局部最优，这是由于活跃路径和舍弃路径之间的比值没有达到平衡。而组合C虽然前期的收敛速度并不是很快，但在后期能一直保持探索能力，最终收敛至较优解。图5(c)的对比分析也同样体现了组合C策略的优越性。

综上，结合表4和图4-5综合分析，当 θ 取组合C的值时，算法能很好地平衡收敛速度和多样性，确保了解的精度，因此，评估阈值 θ 按组合C的取值选取。

3.2 ENCACO改进策略的有效性分析

为了验证ENCACO算法中不同改进策略的有效性，将算法提出的3个策略(自适应评估奖惩机制、邻域动态退化策略和种间协同进化机制)分别组合成4种不同的优化方案进行实验仿真，具体优化方案如表5所示。选取eil51、ch150和tsp225

这3个不同规模的测试集分别按照表5的4种方案各进行15次实验，实验结果如表6所示。图6为不同方案下各测试集的收敛图。

表5 优化方案表

Table 5 Optimization scheme table

方案	优化组合
A	ENCACO(评估奖惩机制+种间协同进化)
B	ENCACO(邻域动态退化+种间协同进化)
C	ENCACO(评估奖惩机制+邻域动态退化)
D	ENCACO

表6 3种优化组合的性能对比

Table 6 Performance comparison of 3 optimized combinations

TSP	标准最优解	方案	最优解	平均解	误差率/%	迭代次数
eil76	538	A	538	539.8	0	413
		B	538	540.5	0	415
		C	538	543.2	0	738
		D	538	538.7	0	267
ch150	6 528	A	6 532	6 580.7	0.06	569
		B	6 544	6 578.9	0.25	433
		C	6 553	6 583.2	0.38	994
		D	6 528	6 554.5	0	549
tsp225	3 916	A	3 919	3 973.9	0.08	1 624
		B	3 921	3 980.2	0.13	1 629
		C	3 935	3 996.8	0.49	1 231
		D	3 916	3 965.3	0	1 235

结合图6和表6分析可知，方案D的最优解和平均解均优于另外3个方案。由于方案A包含评估奖惩机制，因此算法在前期收敛速度较快，因

为通过评估值将路径划分并进行不同的信息素奖惩, 导致蚂蚁被信息素累积较多的路径吸引, 从而提高了算法的收敛速度。方案B包含邻域动态退化策略, 在具有较快的收敛速度的同时也提高了算法的求解精度。方案C仅含有2个附加策略而没有种群间的协同进化机制, 因此, 相当于2种群并行求解, 求解效率较低。

以测试集 ch150 为例, 如图 6 (b)所示, 可以看出方案A在算法初期收敛速度较快, 但在后期多样性较差, 陷入局部最优; 方案B中的邻域动态退化策略虽平衡了收敛速度和解的多样性, 但求解精度仍存在不足; 方案C虽前期收敛速度较快, 后期因为两种群无法进行协同进化而在200代左右便快速进入停滞状态, 后期难以跳出局部最优; 方案D综合以上3种策略, 由于评估奖惩

机制的作用, 前期算法快速收敛至较优解附近, 中期通过自适应改变的邻域半径扩大搜索范围, 收敛曲线呈持平后下降状态, 由于种间协同机制促进两种群间进行交流, 算法仍能在中后期跳出局部最优, 增加次优解被选择的可能, 在快速收敛到最优解的同时保证了解的精度。方案D证明了3种改进策略均有一定的作用, 同时使用获得解的质量更优。综上, 方案D不论是在求解速度还是在平均解的质量上均有一定提高。

3.3 与经典蚁群算法的对比

3.3.1 实验结果分析

选取不同的 TSP 城市集, 将改进算法 ENCACO 与经典蚁群算法 ACS 和 MMAS 的实验结果进行比较分析, 对比结果如表 7 所示。

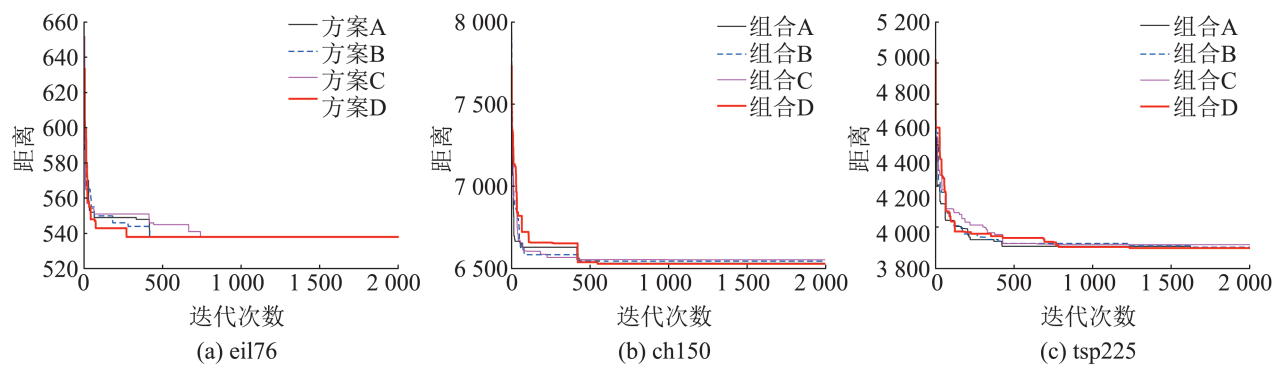


图6 不同方案下各测试集收敛图

Fig. 6 Convergence diagram of each test set under different schemes

表7 ENCACO 与经典算法 ACS 和 MMAS 在不同测试集下的性能对比

Table 7 Performance comparison of ENCACO with classical algorithms ACS and MMAS under different test sets

TSP	标准最优解	算法	最优解	平均解	误差率/%	迭代次数
eil51	426	ACS	426	428.1	0	964
		MMAS	426	429.8	0	870
		ENCACO	426	426.8	0	60
eil76	538	ACS	538	543.3	0	285
		MMAS	538	544.3	0	1 813
		ENCACO	538	538.7	0	267
kroA100	21 282	ACS	21 282	21 476.1	0	1 436
		MMAS	21 356	21 845.4	0.35	1 235
		ENCACO	21 282	21 397.8	0	589

续表

TSP	标准最优解	算法	最优解	平均解	误差率/%	迭代次数
kroB100	22 141	ACS	22 237	22 351.5	0.43	1 565
		MMAS	22 304	22 381.2	0.74	1 393
		ENCACO	22 141	22 289.9	0	820
ch130	6 110	ACS	6 156	6 218.5	0.75	1 089
		MMAS	6 197	6 234.0	1.42	1 250
		ENCACO	6 110	6 198.5	0	535
kroA150	26 524	ACS	26 761	27 199.9	0.89	1 460
		MMAS	26 877	27 223.1	1.33	1 665
		ENCACO	26 525	26 781.7	0	820
kroB150	26 130	ACS	26 248	26 715.8	0.45	1 892
		MMAS	26 289	26 826.4	0.61	1 750
		ENCACO	26 130	26 296.9	0	867
ch150	6 528	ACS	6 547	6 626.9	0.29	1 465
		MMAS	6 533	6 585.7	0.08	894
		ENCACO	6 528	6 554.5	0	549
kroA200	29 368	ACS	29 528	29 901.7	0.54	1 516
		MMAS	29 450	29 862.1	0.28	1 219
		ENCACO	29 368	29 673.4	0	1 498
kroB200	29 437	ACS	29 751	30 333.3	1.07	1 960
		MMAS	29 838	30 207.4	1.36	1 648
		ENCACO	29 437	29 815.9	0	1 115
tsp225	3 916	ACS	3 934	3 998.7	0.46	1 470
		MMAS	3 978	4 093.0	1.58	1 195
		ENCACO	3 916	3 965.3	0	1 235
a280	2 579	ACS	2 583	2 713.8	0.16	1 862
		MMAS	2 590	2 632.2	0.43	570
		ENCACO	2 579	2 613.1	0	1 039
lin318	42 029	ACS	42 708	43 322.2	1.62	1 818
		MMAS	43 111	44 640.5	2.57	1 643
		ENCACO	42 236	42 793.5	0.49	1 864
fl417	11 861	ACS	12 005	12 105.4	1.21	1 919
		MMAS	12 189	12 359.8	2.77	1 970
		ENCACO	11 963	12 059.2	0.86	1 478
pr439	107 217	ACS	108 711	110 501.8	1.39	1 816
		MMAS	108 857	110 614.5	1.53	1 850
		ENCACO	107 961	109 329.6	0.69	1 233
rat575	6 773	ACS	7 093	7 350.4	4.72	1 907
		MMAS	7 101	7 366.3	4.84	1 878
		ENCACO	6 922	7 009.8	2.20	1 676
p654	34 643	ACS	35 445	36 564.4	2.32	1 924
		MMAS	35 912	36 944.5	3.66	1 895
		ENCACO	34 793	35 147.4	0.43	1 692
u724	41 910	ACS	43 598	44 200.7	4.03	1 945
		MMAS	43 654	44 967.2	4.16	1 957
		ENCACO	42 895	43 377.7	2.35	1 900

<http://www.china-simulation.com>

• 1486 •

可以看出, 对于小规模测试集, 相较于 ACS 和 MMAS 传统算法, 改进算法 ENCACO 均能在较小的迭代次数内找到最优解, 且平均解均优于 2 种传统算法, 这是由于评估奖惩机制的存在, 在算法运行过程中拉开较优路径和较差路径间的差距, 加快了收敛速度。

在中规模测试集中, 传统算法已难以保证算法求解精度, 而 ENCACO 算法均能找到标准最优解, 对于 kroA150 测试集, ENCACO 虽未能找到标准最优解, 但其得到的解跟标准最优解差距很小, 误差率接近 0。对于中规模测试集, ENCACO 得到解的平均值均优于传统蚁群算法, 且在收敛速度上存在明显优势。

对于较大规模测试集, ENCACO 算法均能将解的精度控制在 1% 之内, 并且 tsp225 和 a280 更是可以找到最优解。相比 2 种传统算法, ENCACO 除了在 tsp225 和 a280 测试集上收敛速度没有达到最优, 其余测试集均在收敛速度和解的质量上表现更优。

对于大规模测试集, ENCACO 的总体表现均优于其他 2 种传统算法。其中, ENCACO 在 p654 测试集中表现优秀, 将最优解误差率控制到 0.5% 之内, 其平均解以及收敛速度均比传统算法更优。对于 rat575 和 u724 测试集来说, 传统算法已经无法很好地找到较优解, 而 ENCACO 仍可以在相对较短的时间内找到较优解, 并将最优解的误差率控制在 2% 左右。

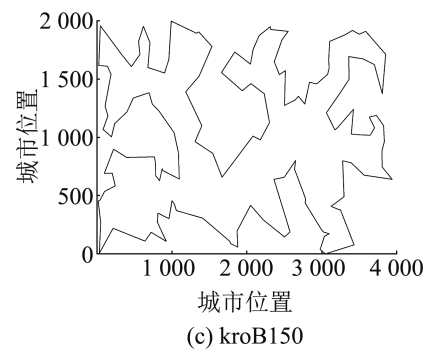
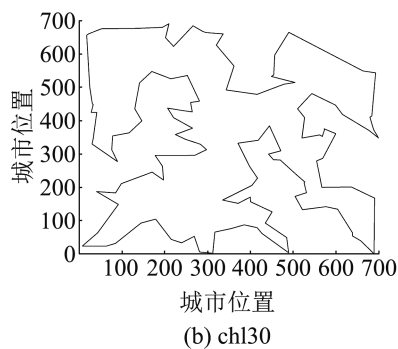
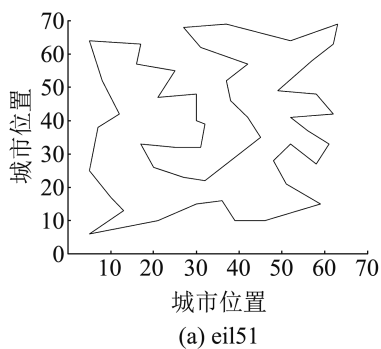
通过以上分析, ENCACO 在求解精度和收敛速度上均有一定提高。部分测试集的路径如图 7 所示, 2 种传统算法与 ENCACO 算法最优解误差率的比较如图 8 所示。

3.3.2 ENCACO 算法多样性和收敛性分析

为了更全面地验证 ENCACO 算法的性能, 从算法的收敛性和多样性进行分析, 分别选取 kroA150、tsp225 和 pr439 不同规模的测试集, 分别给出他们的多样性折线图和收敛曲线图, 多样性为测试集的标准差值。

由图 9(a)(b), 可以看出, ACS 算法多样性在后期有所下降, 而 ENCACO 的标准差变化在后期仍然较大, 这是因为邻域动态退化策略增加了算法的多样性。结合图 9(c) 可知, ENCACO 在算法初始时就以较快的速度收敛到较优解附近, 这是因为评估奖惩机制的引入, 对不同路径采取不同的信息素奖惩, 进而提高了算法的收敛性。后期算法多样性较好, 能够得到质量较好的解, 这是因为对于奖励的路径来说, 越优的路径受到的奖励反而越少, 在一定程度上保护了次优路径的探索机会, 对于惩罚的路径也采取同样的方式, 有效提高了算法的多样性。

由图 10(a)(b) 可以明显看出, 随着迭代的进行, ENCACO 算法多样性呈上升趋势, 而 ACS 的多样性随着迭代进行逐步降低, 且结合图 10(c) 迭代情况可知 ENCACO 可以快速收敛并保持稳定, 相比传统算法优势明显。



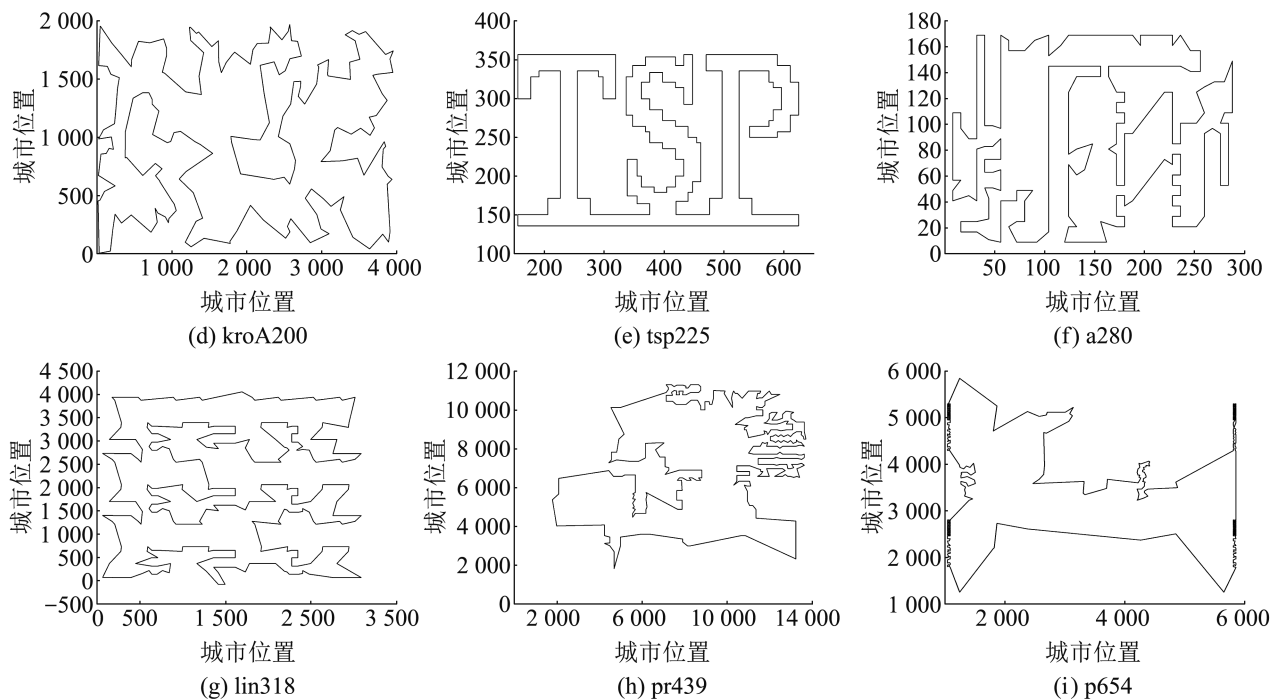


图7 ENCACO部分最优路径
Fig. 7 Partial optimal path of ENCACO

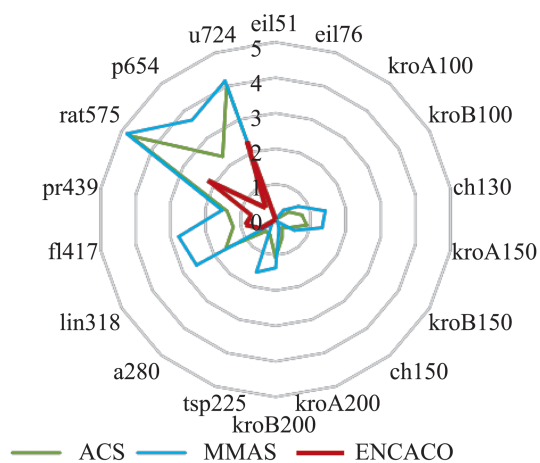


图8 最优解误差率对比图
Fig. 8 Comparison of error rates of optimal solutions

由图11(a)(b)可知, ENCACO的多样性相较于ACS的多样性大, 且有上升趋势, 说明ENCACO在迭代中不断优化路径, 在算法后期多样性也较为丰富, 结合图11(c)可以看出ENCACO在后期仍能跳出局部最优, 表明改进算法拥有较好的多样性。

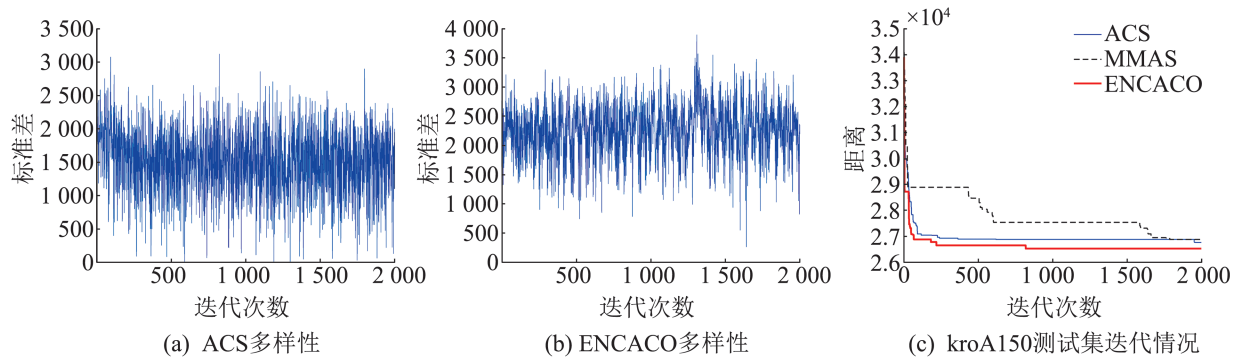
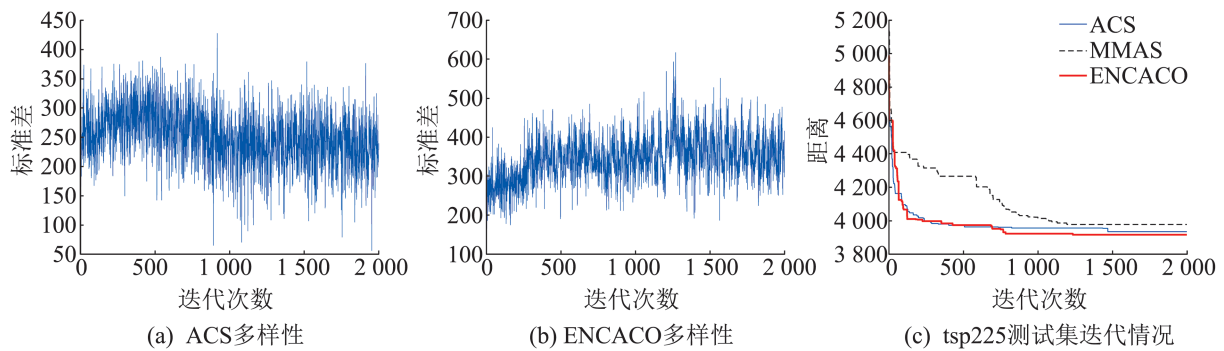
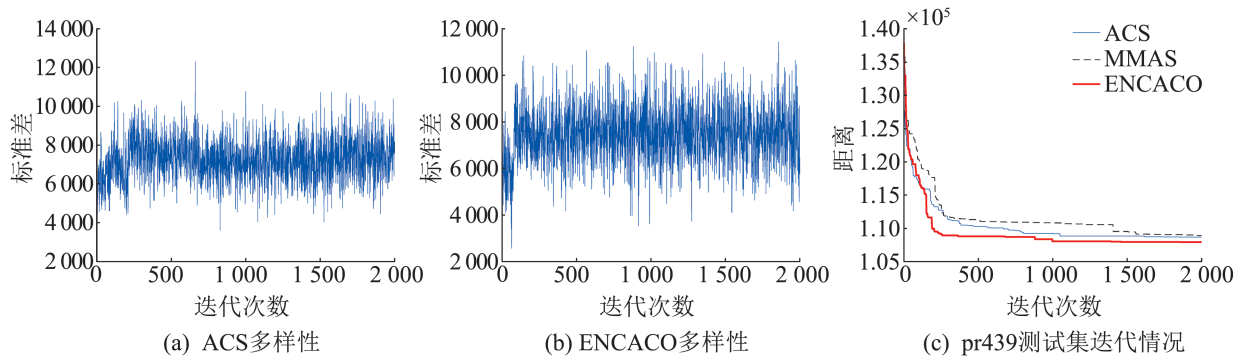
文中策略自适应评估奖惩机制在算法进行中对活跃路径进行信息素奖励而对舍弃路径进行信息素惩罚, 若活跃路径和舍弃路径之间出现重叠

路径段, 根据式(17)和(18)可知, 重叠路径段上的信息素值会发生相应变化, 但信息素发生变化的个别小段重叠路径不会对整体的算法多样性造成较大影响, 图12展示了kro150、tsp225和lin318测试集在没有评估奖惩机制下的算法多样性。综上所述, ENCACO拥有良好的收敛性和多样性。

3.3.3 ENCACO算法稳定性分析

分析ENCACO算法和2种传统算法的平均误差率来判断算法稳定性。平均误差率指平均解与标准最优解间的误差率, 平均误差率越小, 表明算法越稳定。

图13为3种算法在上述TSP测试集中的平均误差率对比, 传统算法ACS和MMAS的平均误差率各有高低, 但ENCACO算法得到的平均误差率一直处于一个较低的水平, 中小规模及较大规模测试集的平均误差率均控制在2%以内, 优于传统蚁群算法。对于3个大规模测试集来说, 虽然ENCACO的平均误差率有一定提高, 但相比传统算法的误差率提高相对较小, 表明ENCACO得到解的质量较优且拥有良好的稳定性。

图9 kroA150 测试集
Fig. 9 kroA150 test set图10 tsp225 测试集
Fig. 10 tsp225 test set图11 pr439 测试集
Fig. 11 pr439 test set

3.4 与其他改进算法的对比

为了进一步验证算法性能, 将ENCACO与其他最新改进算法进行对比, 对比数据取自基于近邻牵引算子的离散黑猩猩优化算法(discrete chip optimization algorithm, DChOA)^[18]、结合价格波动策略与动态回溯机制的蚁群算法(ant colony algorithm based on price fluctuation strategy and

dynamic backtracking mechanism, PBACO)^[19]、结合邻域耦合机制与双边滤波的双蚁群算法(dual ant colony optimization with neighborhood coupling mechanism and bilateral filtering, NBACO)^[20]和基于参数优化的改进蚁群算法(the hybrid symbiotic organisms search and ACO, SOS-ACO)^[21], 以上算法均为近年来发表的群智能算法, 在求解TSP问题中表现出良好的性能, 对比结果如表8~10所示。

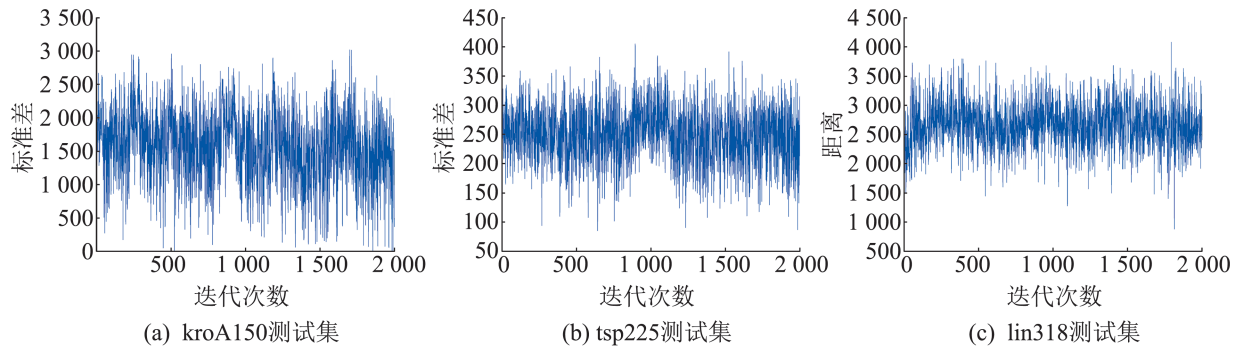


图 12 不包含自适应评估奖惩机制的算法多样性

Fig. 12 Algorithms diversity without adaptive evaluation reward and punishment mechanism

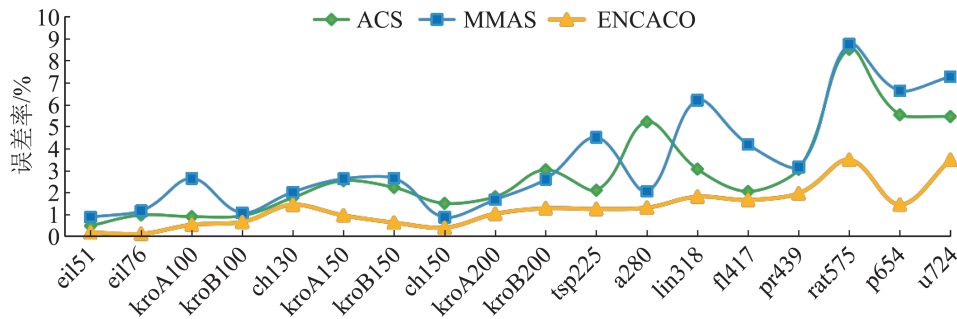


图 13 平均误差率对比图

Fig. 13 Average error rate comparison chart

表 8 ENCACO 与其他改进算法比较

Table 8 Comparison of ENCACO with other improved algorithms

TSP	标准解	ENCACO	误差率/%	DChOA	误差率/%	PBACO	误差率/%
eil76	538	538	0	538	0	538	0
kroA150	265 24	26 525	0	27 231	2.67	26 605	0.30
ch150	6 528	6 528	0	6 715	2.86	6 533	0.07
kroA200	29 368	29 368	0	29 565	0.67	29 383	0.05
tsp225	3 916	3 916	0	3 997	2.07	3 923	0.17
lin318	42 029	42 236	0.49	44 136	5.01	42 384	0.84
pr439	107 217	107 954	0.69	111 262	3.77	—	—

注：“—”表示对比文献未对该城市集进行测试。

表 9 ENCACO 与 NBACO 比较

Table 9 Comparison of ENCACO with NBACO

TSP	标准解	ENCACO			NBACO		
		最优解	误差率/%	平均解	最优解	误差率/%	平均解
eil51	426	426	0	426.8	426	0	428
kroA150	26 524	26 525	0	26 781.7	26 528	0.02	26 989
tsp225	3 916	3 916	0	3 965.3	3 924	0.20	3 992
a280	2 579	2 579	0	2 613.1	2 581	0.07	2 630
lin318	42 029	42 236	0.49	42 793.5	42 370	0.81	43 222
fl417	11 861	11 963	0.86	12 059.2	11 967	0.89	12 097
pr439	107 217	107 961	0.69	109 329.6	107 842	0.58	110 684

表10 ENCACO与SOS-ACO比较
Table 10 Comparison of ENCACO with SOS-ACO

TSP	标准解	ENCACO			SOS-ACO		
		最优解	误差率/%	平均解	最优解	误差率/%	平均解
eil51	426	426	0	426.8	426	0	428.1
eil76	538	538	0	538.7	538	0	541.7
kroA100	21 282	21 282	0	21 397.8	21 282	0	21 290.1
ch150	6 528	6 528	0	6 554.5	6 558	0.46	6 571.2
kroA200	29 368	29 368	0	29 673.4	29 413	0.15	29 520.2
lin318	42 029	42 236	0.49	42 793.5	42 473	1.05	42 762.7
pr439	107 217	107 961	0.69	109 329.6	107 978	0.71	108 873.8

4 结论

针对传统蚁群算法在求解TSP问题收敛速度慢以及易陷入局部最优等问题, 提出结合评估奖惩机制和邻域动态退化的协同蚁群算法(ENCACO)。通过计算路径评估值将路径划分为活跃路径和舍弃路径, 对活跃路径进行信息素奖励, 对舍弃路径进行信息素惩罚, 通过信息素奖惩机制加快算法的收敛速度。引入邻域动态退化策略, 根据城市规模、迭代进行程度等因素设置动态邻域半径, 并对半径外的退化区按保留概率自适应退化城市, 退化区保留下来的城市结合半径内的探索区一起计算状态转移概率, 平衡了算法的收敛速度和解的多样性。同时, 为了提高两种种群间交流的效率, 提出种间协同进化机制, 根据种群间进化程度设置交互周期并选择合适的交互方式, 从而提高算法解的精度。通过仿真实验的结果可以看出ENCACO在收敛速度以及解的精度上均有一定提升。在中小规模TSP问题的求解上, ENCACO可以表现出色, 但在求解大规模或超大规模TSP问题时, 算法精度还有待提高。下一步尝试引入博弈策略改进多种群间的协同进化机制, 以提高算法在大规模TSP问题上的求解能力。

参考文献:

- [1] Yun Xiaoyan. Research on Traveling Salesman Problem Algorithm[J]. *Advanced Materials Research*, 2013, 694-697: 2901-2904.
- [2] 张丽萍, 朱振威, 周雄辉. 基于蚁群算法的冲压车间模具协同调度优化研究[J]. *模具技术*, 2021(3): 1-8.

- Zhang Liping, Zhu Zhenwei, Zhou Xionghui. Collaborative Die Scheduling Optimization of Twin Cranes in Stamping Workshop Based on Ant Colony Optimization[J]. *Die and Mould Technology*, 2021(3): 1-8.
- [3] 梅前, 董宝力. 基于混合蚁群遗传算法的多AGV任务分配[J]. *物流工程与管理*, 2022, 44(8): 1-5, 9.
Mei Qian, Dong Baoli. Multi-AGV Task Assignment Based on Hybrid Ant Colony Genetic Algorithm[J]. *Logistics Engineering and Management*, 2022, 44(8): 1-5, 9.
- [4] 雷金羨, 孙宇, 朱洪杰. 改进蚁群算法在带时间窗车辆路径规划问题中的应用[J]. *计算机集成制造系统*, 2022, 28(11): 3535-3544.
Lei Jinxian, Sun Yu, Zhu Hongjie. Improved Ant Colony Optimization Algorithm for Vehicle Routing Problems with Time Window[J]. *Computer Integrated Manufacturing Systems*, 2022, 28(11): 3535-3544.
- [5] 赵小惠, 卫艳芳, 赵雯, 等. 基于混合遗传蚁群算法的多目标FJSP问题研究[J]. *组合机床与自动化加工技术*, 2023(1): 188-192.
Zhao Xiaohui, Wei Yanfang, Zhao Wen, et al. Research on Multi-objective Flexible Job Shop Scheduling Problem Based on Hybrid Genetic Ant Colony Algorithm[J]. *Combined Machine Tools & Automated Machining Technology*, 2023(1): 188-192.
- [6] Dorigo M, Maniezzo V, Colomi A. Ant System: Optimization by a Colony of Cooperating Agents[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 1996, 26(1): 29-41.
- [7] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 53-66.
- [8] Thomas Stützle, Hoos H H. MAX - MIN Ant System[J]. *Future Generation Computer Systems*, 2000, 16(8): 889-914.

- [9] Li Wei, Wang Cancan, Huang Ying, et al. Heuristic Smoothing Ant Colony Optimization with Differential Information for the Traveling Salesman Problem[J]. *Applied Soft Computing*, 2023, 133: 109943.
- [10] Gao Wei. New Ant Colony Optimization Algorithm for the Traveling Salesman Problem[J]. *International Journal of Computational Intelligence Systems*, 2020, 13(1): 44-55.
- [11] Song Qi, Zhao Qinglei, Wang Shuxin, et al. Dynamic Path Planning for Unmanned Vehicles Based on Fuzzy Logic and Improved Ant Colony Optimization[J]. *IEEE Access*, 2020, 8: 62107-62115.
- [12] Ren Teng, Luo Tianyu, Jia Binbin, et al. Improved Ant Colony Optimization for the Vehicle Routing Problem with Split Pickup and Split Delivery[J]. *Swarm and Evolutionary Computation*, 2023, 77: 101228.
- [13] Li Shundong, You Xiaoming, Liu Sheng. Co-evolutionary Multi-colony Ant Colony Optimization Based on Adaptive Guidance Mechanism and Its Application[J]. *Arabian Journal for Science and Engineering*, 2021, 46(9): 9045-9063.
- [14] 冯晨, 游晓明, 刘升. 结合竞争交互策略和淘汰重组机制的异构多蚁群算法[J]. *系统仿真学报*, 2024, 36(1): 232-248.
Feng Chen, You Xiaoming, Liu Sheng. Heterogeneous Multi-ant Colony Algorithm Combining Competitive Interaction Strategy and Eliminating-reconstructing Mechanism[J]. *Journal of System Simulation*, 2024, 36(1): 232-248.
- [15] Li Hanke, You Xiaoming, Liu Sheng. Multi-ant Colony Optimization Algorithm Based on Finite History Archiving and Boxed Pigs Game[J]. *Applied Soft Computing*, 2023, 138: 110193.
- [16] 石美凤, 肖诗川, 冯欣. 基于多种群的随机扰动蚁群算法求解分布式约束优化问题[J]. *计算机应用研究*, 2022, 39(9): 2683-2688.
- Shi Meifeng, Xiao Shichuan, Feng Xin. Random Disturbance Based Multi-population Ant Colony Algorithm to Solve Distributed Constraint Optimization Problems[J]. *Application Research of Computers*, 2022, 39(9): 2683-2688.
- [17] 王凯, 何宏, 殷静. 基于改进LeNet-5神经网络的微表情识别研究[J]. *中国设备工程*, 2022(4): 258-259.
- [18] 沈孝凯, 张纪会, 郭乙运, 等. 基于近邻牵引算子的离散黑猩猩优化算法[J/OL]. *控制与决策*. (2023-01-07) [2023-04-24]. <https://doi.org/10.13195/j.kzyjc.2022.1417>.
Shen Xiaokai, Zhang Jihui, Guo Yiyun, et al. A Discrete Chimp Optimization Algorithm Based on Neighbor Traction Operator[J/OL]. *Control and Decision*. (2023-01-07) [2023-04-24]. <https://doi.org/10.13195/j.kzyjc.2022.1417>.
- [19] 赵家波, 游晓明, 刘升. 结合价格波动策略与动态回溯机制的蚁群算法[J]. *计算机科学与探索*, 2022, 16(6): 1390-1404.
Zhao Jiabo, You Xiaoming, Liu Sheng. Ant Colony Algorithm Based on Price Fluctuation Strategy and Dynamic Back-tracking Mechanism[J]. *Journal of Frontiers of Computer Science & Technology*, 2022, 16(6): 1390-1404.
- [20] 吴立胜, 游晓明, 刘升. 结合邻域耦合机制与双边滤波的双蚁群算法[J]. *计算机科学与探索*, 2023, 17(9): 2092-2106.
Wu Lisheng, You Xiaoming, Liu Sheng. Dual Ant Colony Optimization with Neighborhood Coupling Mechanism and Bilateral Filtering[J]. *Journal of Frontiers of Computer Science & Technology*, 2023, 17(9): 2092-2106.
- [21] Wang Yong, Han Zunpu. Ant Colony Optimization for Traveling Salesman Problem Based on Parameters Optimization[J]. *Applied Soft Computing*, 2021, 107: 107439.