

# 基于变邻域改进的蚁群算法的TSP问题仿真

王霖, 付杰, 金中

上海海事大学理学院, 上海

收稿日期: 2025年9月10日; 录用日期: 2025年9月25日; 发布日期: 2025年10月30日

## 摘要

旅行商问题(TSP)作为组合优化领域的经典难题, 对物流路径规划等领域具有重要意义。传统蚁群算法(ACO)虽具备全局搜索能力, 但存在易陷入局部最优和收敛速度不足的问题。为此, 本文提出一种融合变邻域搜索(VNS)的改进蚁群算法(ACO-VNS), 旨在通过动态调节信息素挥发系数、并在算法停滞时引入邻域扰动策略, 以增强全局寻优能力与收敛效率。实验基于TSPLIB实例库中的典型问题, 对比传统ACO、ACS、Rank-AS及ACO-VNS的性能。结果表明: ACO-VNS在解质量与收敛速度上均显著优于其他方法, 且解稳定性更高。结论表明, ACO-VNS通过动态参数调整与变邻域搜索的有效结合, 成功克服了传统算法易陷入局部最优和收敛速度不足的缺陷, 为复杂TSP问题提供了高效解决方案。

## 关键词

蚁群算法, 变邻域操作, TSP问题, 自适应性

# Simulation of TSP Problem Based on Ant Colony Algorithm Improved by Variable Neighborhood

Lin Wang, Jie Fu, Zhong Jin

School of Science, Shanghai Maritime University, Shanghai

Received: September 10, 2025; accepted: September 25, 2025; published: October 30, 2025

## Abstract

The Traveling Salesman Problem (TSP), a classic challenge in combinatorial optimization, plays a vital role in logistics path planning and related fields. While traditional Ant Colony Optimization (ACO) algorithms demonstrate global search capabilities, they often suffer from being prone to local optima and insufficient convergence speed. To address these limitations, this study proposes an enhanced ACO variant (ACO-VNS) that integrates variable neighborhood search (VNS). The proposed algorithm

dynamically adjusts pheromone decay rates and incorporates neighborhood disturbance strategies during stagnation phases, thereby improving global optimization capabilities and convergence efficiency. Experimental results based on typical problems from the TSPLIB benchmark demonstrate that ACO-VNS significantly outperforms conventional methods like ACO, Ant Colony Search (ACS), Rank-AS, and ACO-VNS in both solution quality and convergence speed, with superior solution stability. The conclusions indicate that ACO-VNS effectively overcomes traditional algorithms' limitations of local optima trapping and insufficient convergence through dynamic parameter adjustments combined with variable neighborhood search, providing an efficient solution for complex TSP problems.

## Keywords

Ant Colony Algorithm, Variable Neighborhood Operation, TSP Problem, Adaptability

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

作为组合优化领域的经典难题，旅行商问题(Traveling Salesman Problem, TSP)一直是一个经典且复杂的组合优化问题，在物流配送、路径规划等众多领域都有着广泛的实际应用场景。随着社会发展，人们对高效解决 TSP 问题的需求日益迫切，因为它直接关系到资源的合理分配与利用效率的提升。因此人们通常利用智能算法求解 TSP 问题[1]，其中蚁群算法就是一种常用的智能算法。

与粒子群优化模仿鸟群迁徙类似，蚁群算法的核心机制源自自然界中蚂蚁群体的协作觅食行为。具体而言，单只蚂蚁在路径探索过程中会释放信息素作为路径优劣的标记，而群体中多只蚂蚁的独立探索行为共同构成了对解空间的分布式搜索。例如，在 TSP 问题中，每只蚂蚁的移动路径可视为一个潜在解，而信息素残留量的高低则通过正反馈机制引导后续蚂蚁的决策，最终促使群体收敛至近似最优路径。在演化过程中，路径长度越短的蚂蚁会在沿途释放更高浓度的信息素标记，这种物质会随迭代次数增加而持续累积。由于后续蚂蚁倾向于选择信息素残留量高的路径(即更优解)，逐渐形成“强者愈强”的正反馈效应[2]，最终蚁群将高度聚集于信息素残留量最高的路径上，该路径即为对应优化问题的全局最优解。

蚁群算法在解决 TSP 问题时具有一定优势，有较强的全局搜索能力和对复杂环境的适应性。然而，它也存在诸多局限性，如容易陷入次优解，导致无法找到全局最优路径；收敛速度较慢，迭代时存在大量冗余(即多次迭代后最优解没有发生改变)。针对此问题，本文引入变邻域搜索[3]，在蚁群算法一定次数的迭代且最优解没有发生变化时通过变邻域搜索得到新解从而加速蚁群算法的迭代、减少迭代次数以及得到更好的最优解。

## 2. 蚁群算法及变邻域搜索算法

### 2.1. 蚁群算法的基本思想

蚁群算法是一种模拟蚂蚁觅食行为的群智能优化算法，该算法的仿生学基础源于蚂蚁群体在觅食过程中通过信息素标记路径实现分布式协作的群体智能行为。在真实蚁群觅食时，蚂蚁群体觅食时通过分泌信息素标记路径，其浓度随时间推移呈指数衰减，路径越短，信息素积累效应越显著。后续蚂蚁在选择路径时，会以较高概率选择信息素残留量高的路径，这样就形成了一种正反馈机制[4]，促使群体行为逐渐收敛至全局最优路径，对应 TSP 问题的最短闭合回路和优化问题的最优解。

在蚁群算法应用于求解 TSP 的情境中, 路径信息素的动态衰减与更新机制是算法实现全局寻优能力的关键[5]。信息素的动态衰减不仅直观地反映了历史上蚂蚁选择特定路径的频率, 还通过群体经验的协同积累[6]引导后续蚂蚁的决策。例如, 在 TSP 中, 若某条路径被频繁选择且长度较短, 其信息素残留量将指数级增长, 形成“优质路径自增强”效应, 最终促使算法收敛至近似最优解。

在蚁群算法中, 为了不失一般性, 假设蚂蚁群体中有  $m$  只蚂蚁, 城市数量为  $n$ , 城市  $i$  与城市  $j$  之间的距离为  $d_{ij}$ 。蚂蚁  $k$  在时刻  $t$  从城市  $i$  转移到城市  $j$  的概率  $P_{ij}^k(t)$  可以通过公式(1) [5]计算:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & \text{if } j \in allow_k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

其中,  $\tau_{ij}(t)$  表示时刻  $t$  城市  $i$  与城市  $j$  之间路径上的信息素残留量;  $\eta_{ij}(t) = \frac{1}{d_{ij}}$  是启发函数, 表示蚂蚁从城市  $i$  转移到城市  $j$  的路劲偏好;  $\alpha$  和  $\beta$  分别控制信息素残留量与启发式因子对路径选择概率的影响程度;  $allow_k$  是蚂蚁  $k$  尚未访问的节点集合。

在蚂蚁完成一次循环后, 路径上的信息素残留量会根据公式(2)进行更新:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

其中,  $\rho$  是信息素衰减系数,  $\Delta\tau_{ij}$  是所有蚂蚁在路径  $(i, j)$  上释放的信息素总量。信息素的更新通常采用蚁周模型(Ant-Cycle Model) [7]即公式(3):

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{if 蚂蚁 } k \text{ 经过路径 } (i, j) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

其中,  $Q$  是信息素总量常数,  $L_k$  是蚂蚁  $k$  经过的路径总长度[8]。

## 2.2. 变邻域搜索算法

变邻域搜索算法(Variable Neighborhood Search, 简称 VNS)是一种用于解决优化问题的元启发式算法由 Mladenovic 和 Hansen 在 1997 年提出[9]。它的核心思想是通过系统地改变搜索领域来避免算法陷入次优解。

变邻域搜索作为一种元启发式算法, 近年来在优化领域备受关注。其核心特点在于通过动态切换邻域结构(如交换、插入、逆序等操作)逐步逼近全局最优解。变邻域搜索中包含了动态变化的邻域结构, 算法通用性强, 自由度大, 可针对特殊问题设计多种变型。因原理简洁(仅需邻域定义与切换策略)、实现门槛低(无需复杂参数调优)及扩展性强等特性, 持续成为组合优化领域的研究热点[10]。主要操作有: 1. 邻域定义: 首先需要定义问题的邻域结构。对于不同的优化问题, 邻域结构的定义方式不同。例如在旅行商问题(TSP)中, 常见的邻域结构可以是交换两个城市的顺序、反转一段路径等。通常会定义多个不同的邻域结构, 这些邻域结构的复杂程度和搜索范围逐渐增大。2. 局部搜索: 从一个初始解开始, 在当前的邻域结构内进行局部搜索, 找到该邻域内的次优解。局部搜索可以采用多种方法, 如贪心算法、爬山法等。3. 邻域切换: 如果在当前邻域内找到的次优解没有比当前全局最优解更好, 那么就切换到下一个邻域结构继续搜索。如果已经到达了最大的邻域结构, 则可以重新从最小的邻域结构开始, 或者采用一些策略来调整邻域结构的顺序。4. 扰动操作: 为了进一步扩大搜索范围, 有时会在切换邻域结构时对当前解进行一些扰动操作, 使得算法能够跳出次优解的吸引区域。例如在 TSP 问题中, 可以随机交换几个城

市的位置作为扰动。

该算法认为在一个次优解的邻域内可能不存在更好的解，但在更大或不同结构的邻域中可能存在。所以，它会在不同的邻域结构之间进行切换搜索，以增加找到全局最优解的可能性。同时该算法简单，可扩展性强(可结合遗传算法、模拟退火等混合策略)，易于与其他优化技术结合。

### 3. 蚁群算法和变邻域搜索算法融合解决 TSP 问题

蚁群算法具有以下几个显著特点：

- 1) 正反馈机制：蚂蚁通过释放信息素来增强较优路径的选择概率，形成正反馈，使得搜索过程不断收敛，最终逼近最优解。
- 2) 分布式计算：通过蚂蚁群体的分布式路径探索，提高了算法的并行处理力能与整体计算速率。
- 3) 启发式搜索：蚂蚁在选择路径时不仅依赖信息素残留量，还考虑路径的启发式信息(如距离)，使得算法不容易陷入局部最优。

- 4) 自适应性：信息素的挥发和更新机制使得算法能够动态调整搜索策略，适应不同的优化问题。

尽管蚁群算法在 TSP 问题中展现出较强的全局搜索能力，但其仍存在局限性：

- 1) 容易陷入局部最优：由于信息素的累积和挥发机制，蚂蚁可能会过早地集中在某一条路径上，导致算法陷入次优解。

- 2) 收敛速率较慢：蚁群算法需要多次迭代才能找到较优解，尤其是在大规模问题中，收敛速度较慢。

- 3) 参数敏感性：蚁群算法的性能依赖于多个参数，参数的选择对算法的效果有较大影响。

为有效解决问题，我们对原本的蚁群算法进行了一些优化。

为了解决蚁群算法在全局搜索能力、收敛速度和解质量等方面的不足，我们采用已有的策略[11]从两个方面对蚁群算法进行改进：(1) 在每次循环结束后只保留最优解。(2) 对信息量挥发系数自适应调整。

研究数据充分证明：信息素挥发系数  $\rho$  在路径寻优过程中表现出显著波动性。当  $\rho$  取值偏高时，信息素挥发速率加快，这将影响算法对解空间的全面探索，使已搜索过的解被重复搜索；而降低  $\rho$  值虽有助于扩大搜索范围，但同时会造成信息素残留积累，进而延缓算法收敛进程。因此，必须对  $\rho$  值进行全局搜索效率与收敛速率的协同优化。为此我们采用参数  $\rho$  的自适应调节机制——当最优解在  $N$  次迭代中未出现显著改进时，即按公式(4)进行参数校准：

$$\rho(t) = \begin{cases} 0.35\rho(t+1), & \text{当 } 0.35\rho(t+1) \geq p_{\min} \text{ 时} \\ p_{\min} = 0.5, & \text{其他} \end{cases} \quad (4)$$

为防止  $\rho$  值过低导致收敛效率下降，特别设置了  $\rho$  的最小阈值即  $p_{\min} = 0.5$ 。

同时研究引入了变邻域搜索算法[12]。

通过上述改进措施，算法在获取最优解的过程中，迭代次数得到了有效减少，提升了算法求解效率，增强了算法在实际应用中的性能表现。

图 1 为融合算法的流程图。

### 4. 基于融合算法求解 TSP 问题

#### 4.1. 融合算法中的变邻域搜索算法规则

变邻域搜索算法主要有下面三种类型[13]：

交换：得到对路径中的节点进行任意交换两个位置后的路径和距离，如图 2：有 8 个城市，当前解为 12345678，随机选择两个位置，然后将这两个位置上的元素进行交换。交换 2 和 5 两个位置上的元素，

则交换后的解为 15342678。

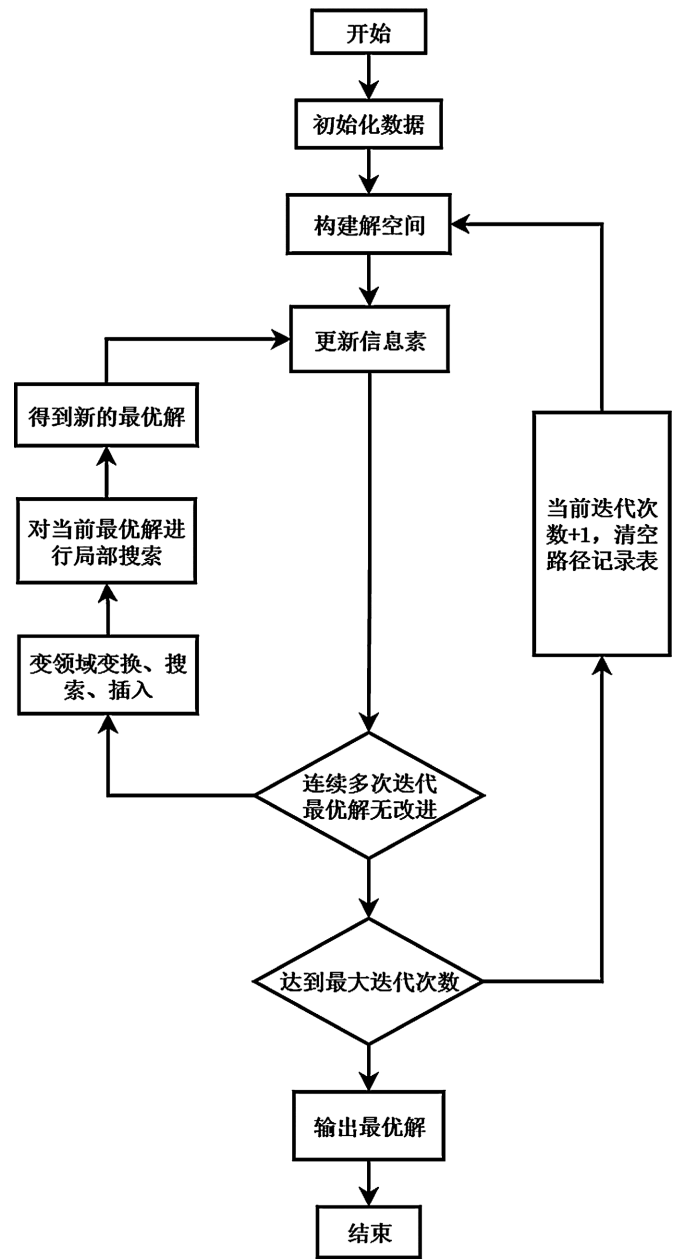


Figure 1. Algorithm flow chart  
图 1. 算法流程图

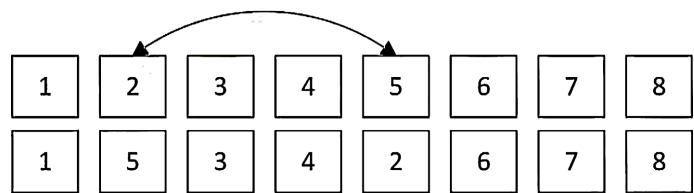


Figure 2. Schematic diagram of exchange operation  
图 2. 交换操作示意图

插入: 得到对路径中的节点进行插入操作后的路径和距离, 如图 3: 有 8 个城市, 当前解为 12345678, 我们随机选择两个位置, 然后将第一个位置上的元素插入到第二个元素之后。第一个选择 5 这个位置, 第二个选择 2 这个位置, 则插入后的解为 12534678。

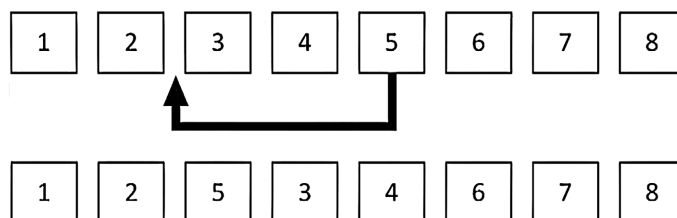


Figure 3. Schematic diagram of insertion operation

图 3. 插入操作示意图

逆转: 得到对路径中的节点进行逆转操作后的路径和距离, 如图 4: 有 8 个城市, 当前解为 12345678, 我们随机选择两个位置, 然后将这两个位置之间的元素进行逆排列。逆转 2 和 5 之间的所有元素, 则逆转后的解为 15432678。

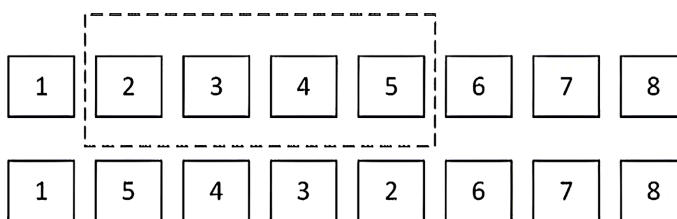


Figure 4. Schematic diagram of reversal operation

图 4. 逆转操作示意图

## 4.2. 融合算法求解 TSP 问题的算法过程

首先执行蚁群算法, 在蚁群算法一定迭代次数没有更优解的时候, 进行变邻域搜索算法。具体实施策略为: 当蚁群算法在设定的一定迭代次数内未能实现路径缩短时, 将此时的最优路径作为初始路径, 开启变邻域搜索, 变邻域搜索的过程按顺序进行交换、插入和逆转操作。在完成变邻域搜索获取新的最优路径后, 按照原顺序将原路径的信息素导入新得到的最优路径之中, 以此构建全新的路径与信息素分布。

### 4.2.1. 蚁群算法主循环

- 1) 初始化蚁群算法各参数, 初始化起点。
- 2) 路劲选择规则: 使用轮盘赌法选择下一个城市, 概率公式为公式(1)。
- 3) 路径长度计算: 计算每只蚂蚁的路径长度, 确保路径闭合。
- 4) 更新最优路径: 记录每代最短路径和平均路径长度。
- 5) 判断是否陷入局部最优: 如果连续 5 次迭代最优路径长度没有更新, 则认为蚁群算法陷入局部最优, 启动变邻域搜索算法(VNS)模块, 同时调整挥发系数  $\rho$ 。
- 6) 更新信息素: 根据信息素更新公式更新路径上的信息素残留量。

### 4.2.2. 变邻域搜索(VNS)模块

- 1) 初始化: 构造一个初始解, 将蚁群算法得到的当前最优路径作为变邻域搜索算法的初始解。
- 2) 外层循环: 变邻域优化启动后, 进行 MAXGEN 次外层迭代。

3) 内层循环与邻域切换机制: 内层迭代按交换操作(Swap); 反转操作(Reversion); 插入操作(Insertion) 三种领域操作进行搜索, 核心策略为:

(1) 设置初始领域索引  $k = 1$  (对应一种领域操作, 如交换操作)。

(2) 当  $k \leq 3$  时, 执行循环: 在当前邻域结构内, 对当前解进行  $M$  次邻域操作, 寻找该邻域内的局部最优解, 如果局部最优解优于初始解, 则接受该解(替代初始解), 并立即将领域索引重置为  $k = 1$ , 重新从第一个邻域开始搜索。否则(即在当前领域中未找到改进解), 则切换到下一个邻域结构(令  $k = k + 1$ ), 以在更大或不同的搜索范围内继续搜索寻找改进机会[14]。

4) 更新全局最优: 每次外层迭代后, 比较并择优更新最优解。

5) 输出: VNS 模块结束后, 将得到的最优解返回给 ACO 主流程。

#### 4.2.3. 信息素更新与动态参数调整策略

通过变领域搜索提高解的质量以后, 算法执行信息素更新和动态改变参数:

1) 信息素全局更新: 所有蚂蚁完成路径构建后, 按照公式(2)和(3)对全局信息素进行挥发和更新。

2) 针对 VNS 新解的信息素的导入机制: 新路径上的边若在原路径中也存在, 则保留原信息素的一部分, 并加上由新路径长度决定的信息素增量, 新路径上新增的边(原路径中没有的边)则赋予基础信息素值加上增量, 不在新路径上的边仅进行信息素挥发, 不增强。设新路径为  $T_{new}$ , 其长度为  $L_{new}$ , 则信息素更新公式为公式(5):

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \frac{Q}{L_{new}} \quad (5)$$

其中,  $Q$  是信息素总量常数,  $\rho$  是挥发系数,  $\tau_{ij}$  是路径  $(i, j)$  上的信息素。

3) 动态挥发系数( $\rho$ )随迭代次数调整, 平衡探索与开发, 加速收敛。

基于上述流程, 在 MATLAB 平台进行仿真实验, 结果如图 5、图 6。其中图 5 为城市节点分布,

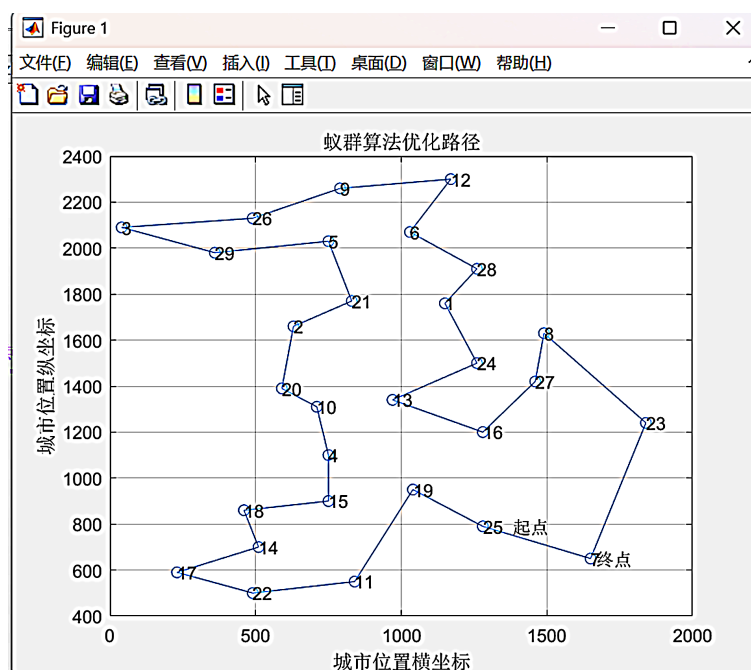


Figure 5. Coordinate diagram

图 5. 坐标图

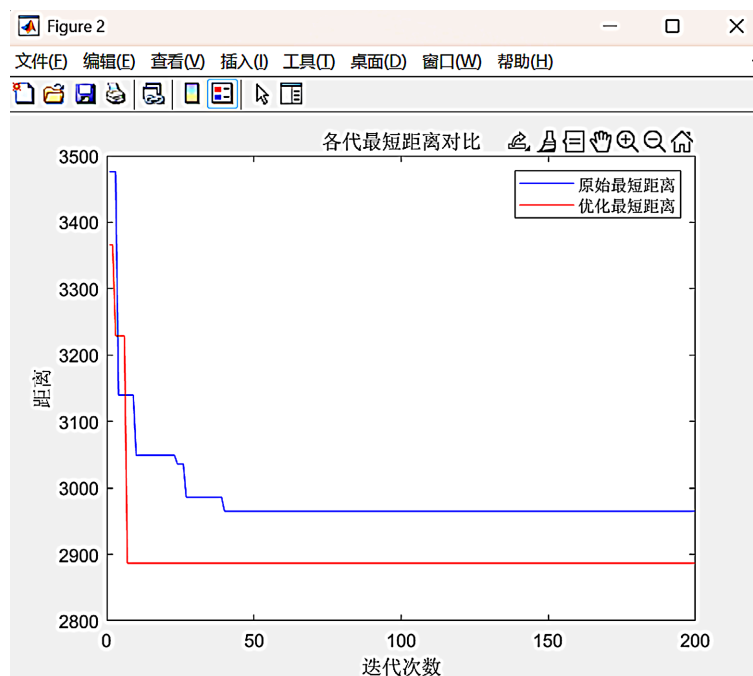


Figure 6. Iteration diagram

图 6. 迭代图

蓝色线段为起始坐标点(城市)的连线,表示算法求得的最优闭合路径[15]。图 6 收敛特性曲线,可获得各代最短距离及最短距离所对应的迭代次数。得到最优解 2887。

## 5. 实验结果及分析

为了验证融合算法的有效性,我们使用 MATLAB 程序分别实现了传统蚁群算法(ACO)、蚁周系统(ACS)、基于排序的蚂蚁系统(Rank-AS)和本文提出的融合算法(ACO-VNS)两种改进算法,并从 TSPLIB 实例库中选取了两个典型的 TSP 问题进行实验。表 1 是四种算法的对比测试结果。表中第 2 列是指到目前为止对某个 TSP 问题已知的最优解,第 4 列记录算法首次达到最优解所需的迭代次数,第 5、6、7 和最后一列表示最优解、标准差、迭代次数范围和最优解范围。

实验结果表明,融合算法在求解 TSP 问题时表现出色,具有较高的求解效率和较好的解质量。

参数设置:  $m = 40$ ,  $\alpha = 1$ ,  $\beta = 5$ ,  $Q = 1$ ,  $\text{iter} = 1$ ,  $\text{iter\_max} = 200$ ,  $\rho$ (动态调整),  $\text{MAXGEN} = 50$ ,  $M = 50$ ,  $n_1 = 3$ 。

根据图 7 的实验数据对比分析,本研究提出的改进的蚁群算法在全局搜索能力方面表现突出。在 att48、bier127 两个典型的 TSP 问题中,改进算法得到的最优解最为接近目前此 TSP 问题的理论最优值。对于 att48 问题,ACO-VNS 是唯一能够稳定找到已知最优解 10628 的算法,其平均最优解与已知最优解完全一致,且标准差为 0,这证明了其在解决该问题时具有完美的收敛性和稳定性,ACO-VNS 的平均迭代次数(31.4 次)远低于传统 ACO (121.9 次)和蚁周系统(115.5 次),虽然略高于基于排序的蚂蚁系统(23.0 次),但后者解的质量远逊于 ACO-VNS。这体现了 ACO-VNS 高效收敛至全局最优的能力;对于规模更大的 bier127 问题,ACO-VNS 取得的平均最优解(37551.7)显著优于其他对比算法,比表现次优的基于排序的蚂蚁系统提升了近 4.4%。同时,其标准差(42.65)远低于其他算法(均高于 300),这表明 ACO-VNS 每次运行都能稳定地输出高质量的解,足以表明改进的新蚁群算法的全局搜索能力更强。由此可见新蚁群算法在收敛速度、全局优化能力和稳定性上均显著优于传统蚁群算法。

| 问题实例    | 已知最短距离 | 算法        | 平均迭代次数 | 平均最短距离  | 距离标准差  | 迭代次数范围 | 最短距离范围      |
|---------|--------|-----------|--------|---------|--------|--------|-------------|
| att48   | 10628  | 原始蚁群算法    | 121.9  | 11305.5 | 85.92  | 10-198 | 11191-11493 |
|         |        | 蚁周系统      | 115.5  | 11281.4 | 120.45 | 15-199 | 11135-11478 |
|         |        | 基于排序的蚂蚁系统 | 23     | 11096.6 | 142.67 | 18-35  | 10929-11389 |
|         |        | 变邻域蚁群系统   | 31.4   | 10628   | 0      | 10-49  | 10628-10628 |
| bier127 | 118282 | 原始蚁群算法    | 110.9  | 44312   | 480.67 | 48-191 | 43753-45365 |
|         |        | 蚁周系统      | 109.3  | 39878.6 | 302.92 | 25-196 | 39327-40238 |
|         |        | 基于排序的蚂蚁系统 | 44.8   | 39270.4 | 450.12 | 33-85  | 38545-40067 |
|         |        | 变邻域蚁群系统   | 118.2  | 37551.7 | 42.65  | 62-187 | 37478-37623 |

Figure 7. Algorithm comparison results

图 7. 算法比对结果

6. 结语

本文针对传统蚁群算法在求解 TSP 问题时因路径多样性不足导致容易陷入局部最优、收敛速度较不足等问题，提出了一种融合蚁群算法与变邻域搜索算法的混合算法(ACO-VNS)。通过引入变邻域搜索的局部搜索策略，融合算法能够在蚁群算法陷入局部最优时跳出局部解，进一步探索解空间，从而找到更优的解。实验结果表明，融合算法在多个 TSP 实例上均表现出色，能够有效减少迭代次数并提高解的质量。未来的研究可以进一步优化算法参数，探索其在多目标优化问题中的应用，并利用并行计算技术提高算法的求解效率。

基金项目

本文工作得到 2024 年度上海高校市级重点课程项目“数学建模实训”资助。

参考文献

[1] 冯月华. 一种求解 TSP 问题的改进蚁群算法[J]. 电子测试, 2014(8): 38-40.

[2] 谢立, 刘政, 张继君, 等. 蚁群算法在宽弦风扇叶片优化排序中的应用[J]. 航空动力学报, 2021, 36(8): 1680-1689.

[3] 边锦华, 张晓霞. 求解 TSP 问题的一种变邻域遗传算法[J]. 福建电脑, 2023, 39(12): 24-27.

[4] Schermer, D., Moeini, M. and Wendt, O. (2020) A Branch-and-Cut Approach and Alternative Formulations for the Traveling Salesman Problem with Drone. *Networks*, **76**, 164-186. <https://doi.org/10.1002/net.21958>

[5] 郁磊, 史峰, 王辉, 胡斐. MATLAB 智能算法 30 个案例分析[M]. 第 2 版. 北京: 北京航空航天大学出版社, 2015.

[6] 郝扬瑞. 基于信息素更新的双种群蚁群算法求解 TSP 的分析[J]. 信息记录材料, 2025, 26(2): 114-116.

[7] Dorigo, M., Maniezzo, V. and Colorni, A. (1996) Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **26**, 29-41. <https://doi.org/10.1109/3477.484436>

[8] 王胜, 谭家政, 刘勇, 等. 求解 TSP 问题的改进蚁群算法[J]. 武汉理工大学学报(信息与管理工程版), 2013, 35(3): 340-344.

[9] Mladenović, N. and Hansen, P. (1997) Variable Neighborhood Search. *Computers & Operations Research*, **24**, 1097-1100. [https://doi.org/10.1016/s0305-0548\(97\)00031-2](https://doi.org/10.1016/s0305-0548(97)00031-2)

[10] 董伟. 变邻域搜索算法研究及在组合优化中的应用[D]: [硕士学位论文]. 阜新: 辽宁工程技术大学, 2011.

[11] 王婉婷. 若干组合优化问题的混合蚁群算法研究[D]: [硕士学位论文]. 银川: 北方民族大学, 2023.

[12] 刘卫林, 董增川, 王德智. 混合智能算法及其在供水水库群优化调度中的应用[J]. 水利学报, 2007, 38(12): 1437-1443.

- 
- [13] 于继江. 一种新型的变领域搜索算法[J]. 通信技术, 2011, 44(9): 129-131.
  - [14] 圣文顺, 徐爱萍, 徐刘晶. 基于蚁群算法与遗传算法的 TSP 路径规划仿真[J]. 计算机仿真, 2022, 39(12): 398-402, 412.
  - [15] 郑娟毅, 程秀琦, 付姣姣. 改进蚁群算法在 TSP 中的应用研究[J]. 计算机仿真, 2021, 38(5): 126-130, 167.