

算法期末作业

算法期末作业

一、算法设计题

1. 优化算法设计

- 算法设计思想和伪代码
- 最短距离及对应可视化遍历方案
- 代码运行结果截图

2. 推荐算法

- 算法设计思想和伪代码
- 推荐结果的评测结果
- 代码运行结果截图

二、元启发式算法研究进展及教育领域应用综述

摘要

1. 引言

2. 元启发式算法研究热点综述

- 算法创新与发展趋势
- 与深度学习的融合创新
- 多目标优化技术突破
- 约束处理机制创新
- 并行计算加速技术

3. 常用元启发式算法介绍

3.1 经典算法

- 遗传算法
- 粒子群优化算法
- 蚁群优化算法

3.2 现代算法

- 灰狼优化算法
- 鲸鱼优化算法
- 蝙蝠算法与萤火虫算法

4. 教育领域优化问题及元启发式算法应用

- 高校学生成绩展示分组优化问题
- 问题建模与求解策略
- 算法设计与实现
- 实验结果与分析
- 算法性能对比与优化建议

5. 结论与展望

一、算法设计题

1. 优化算法设计

(1) 算法设计思想和伪代码

设计思想： 旅行商问题（TSP）属于NP难问题。为了在合理时间内找到高质量解，我采用了模拟退火算法。

1. 模拟退火（SA）：模仿物理退火过程。算法从一个随机路径和高温开始。
2. Metropolis 准则：在每一步迭代中，对当前路径进行微扰产生新路径。如果新路径更短，则直接接受；如果更长，则以概率 $P = e^{-\Delta E/T}$ 接受（其中 ΔE 是距离增量， T 是当前温度）。这种机制允许算法跳出局部最优陷阱。
3. 2-opt 邻域操作：通过随机选择两个节点并将中间的路径片段 反转 来产生新路径。这种方法能有效消除路径中的“交叉”现象，非常适合欧几里得TSP问题。
4. 降温策略：温度 T 按冷却系数 α （如 0.995）逐渐降低，直到趋于稳定。

伪代码：

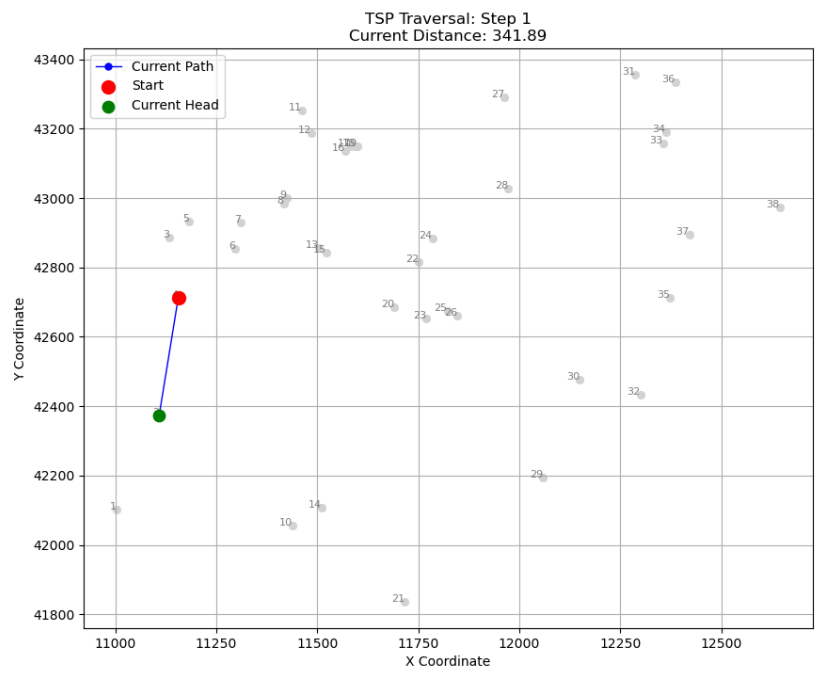
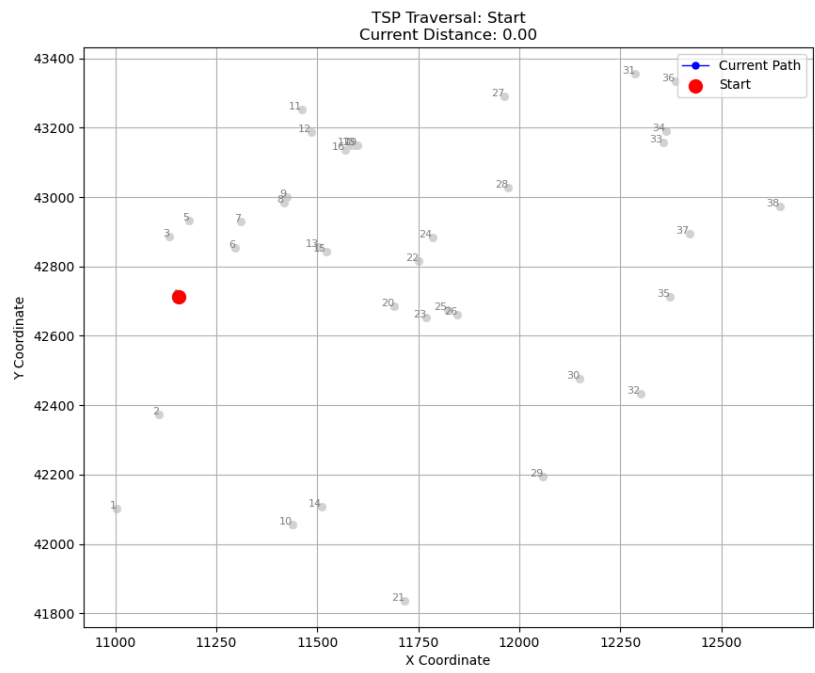
```
1  算法：模拟退火 (Simulated Annealing with 2-opt)
2
3  输入：城市列表 Cities
4  输出：最短路径 BestTour, 最短距离 MinDistance
5
6  1. 初始化：
7      CurrentTour = 随机排列(Cities)
8      BestTour = CurrentTour
9      T = 初始温度 (例如 10000)
10     Min_T = 终止温度 (例如 0.001)
11     Alpha = 冷却系数 (例如 0.995)
12
13  2. 循环 (当 T > Min_T 时):
14     重复执行 (每个温度下的迭代次数):
15         a. 产生邻域解 (2-opt):
16             随机选择两个索引 i, j
17             NewTour = 将 CurrentTour 中 i 到 j 的片段反转
18
19         b. 计算距离变化:
20             Delta = 距离(NewTour) - 距离(CurrentTour)
21
22         c. 接受判断 (Metropolis准则):
23             如果 Delta < 0 或者 Random(0,1) < exp(-Delta / T):
24                 CurrentTour = NewTour
25                 如果 距离(CurrentTour) < 距离(BestTour):
26                     BestTour = CurrentTour
27
28         d. 降温:
29             T = T * Alpha
30
31  3. 返回 BestTour, 距离(BestTour)
```

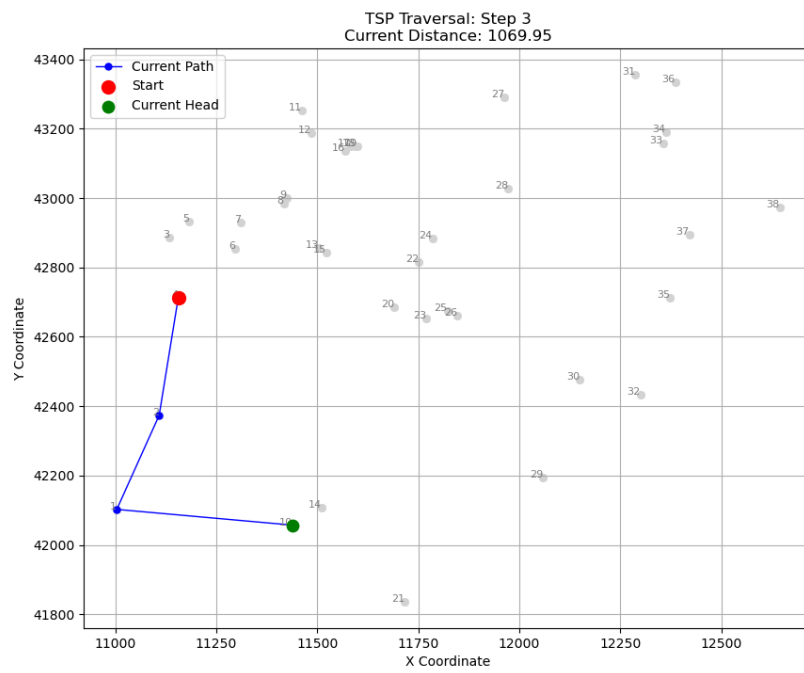
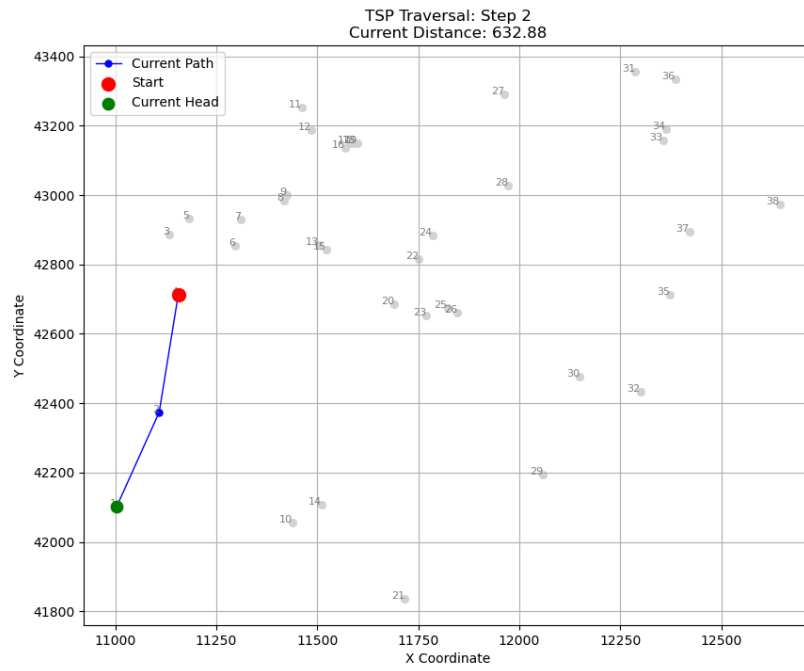
(2) 最短距离及对应可视化遍历方案

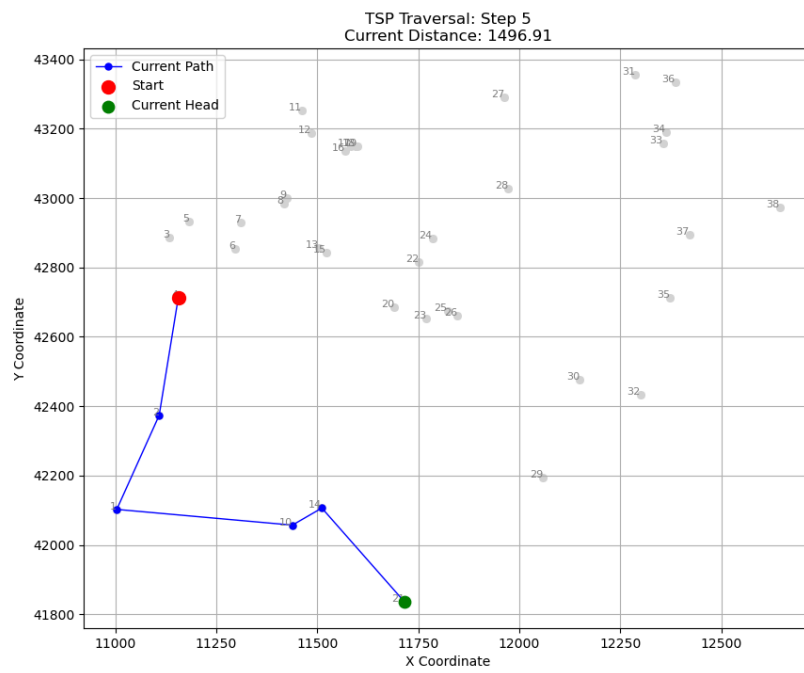
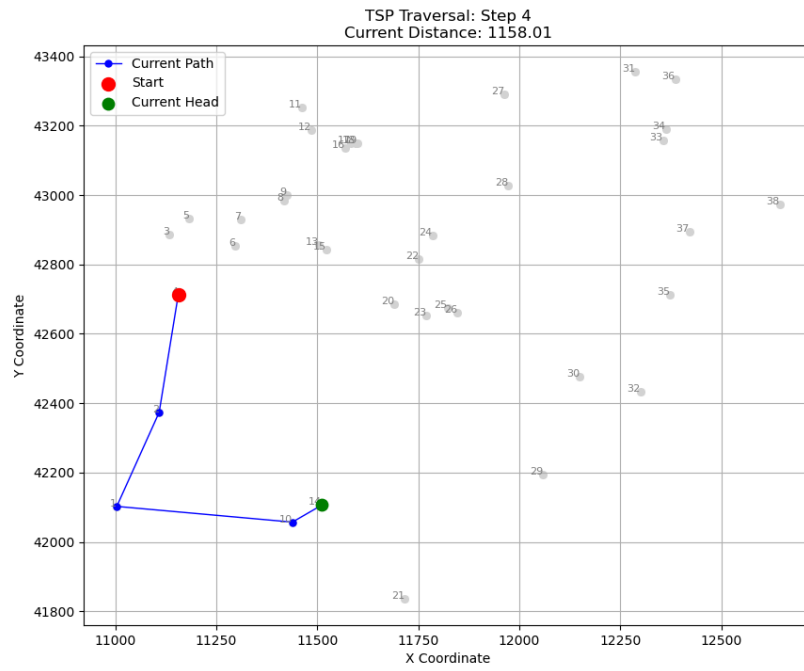
最短距离： 6659.4315

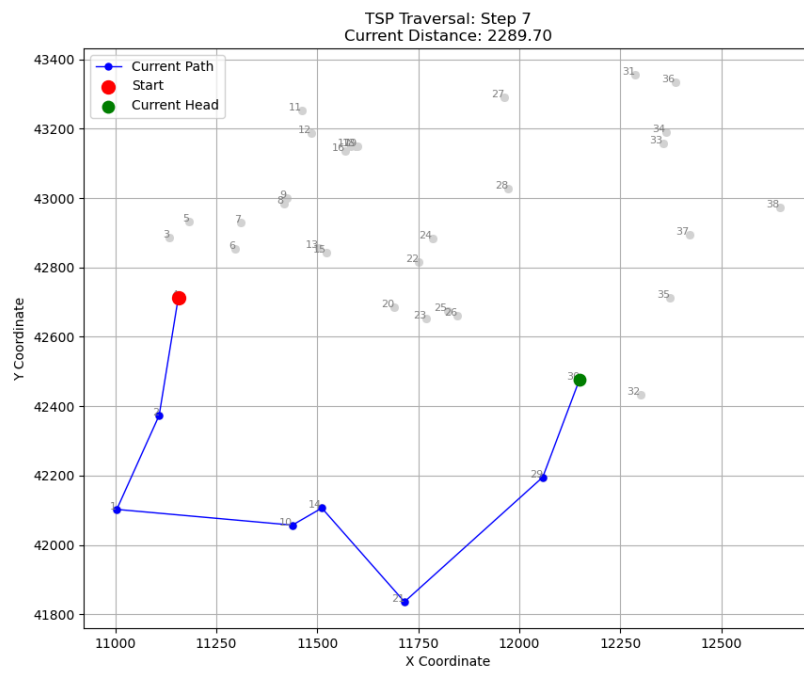
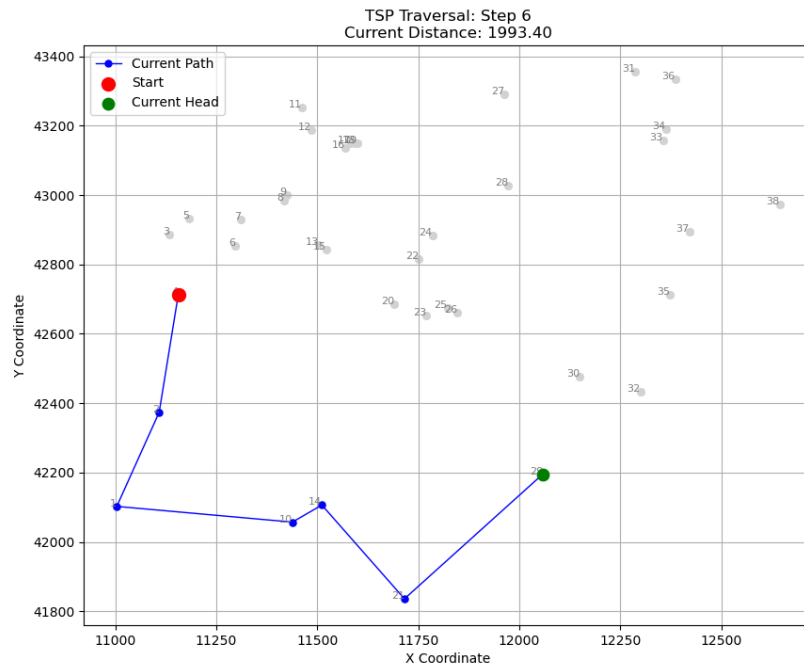
路径： [4, 2, 1, 10, 14, 21, 29, 30, 32, 35, 37, 38, 33, 34, 36, 31, 27, 28, 24, 22, 25, 26, 23, 20, 15, 13, 16, 17, 18, 19, 11, 12, 9, 8, 7, 6, 5, 3]

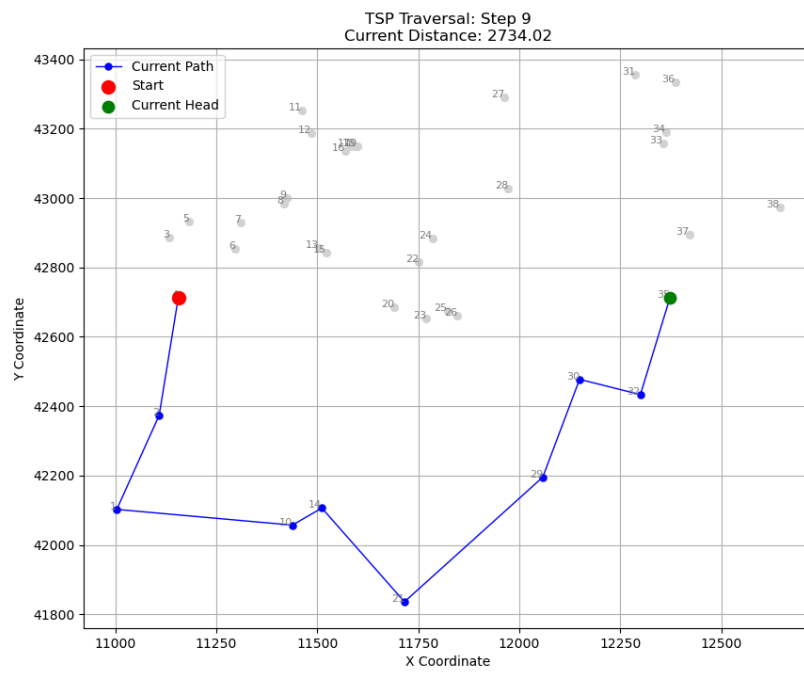
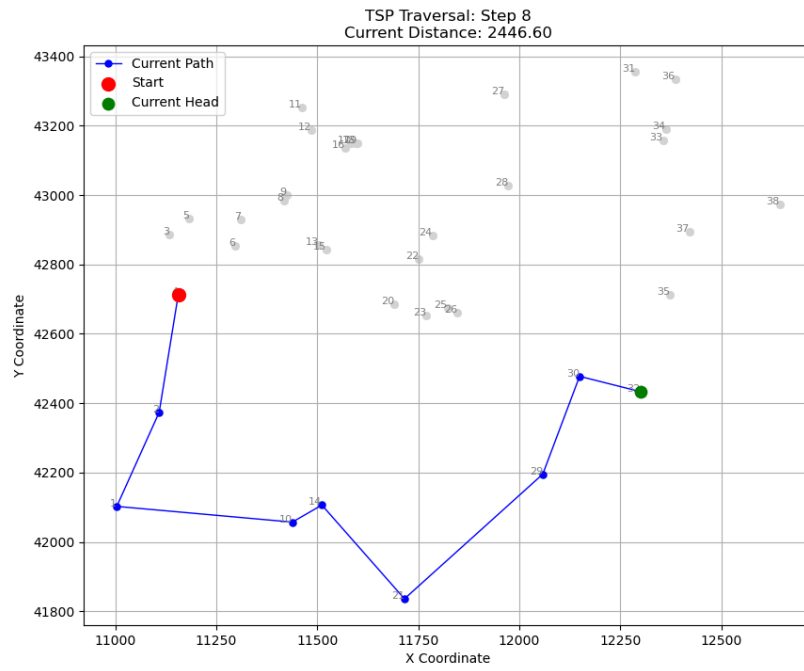
可视化遍历方案：

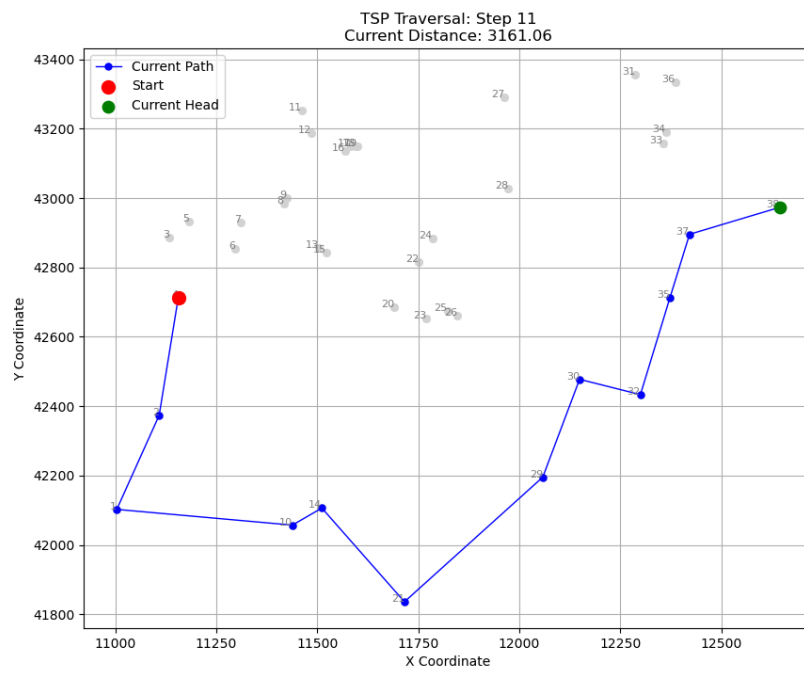
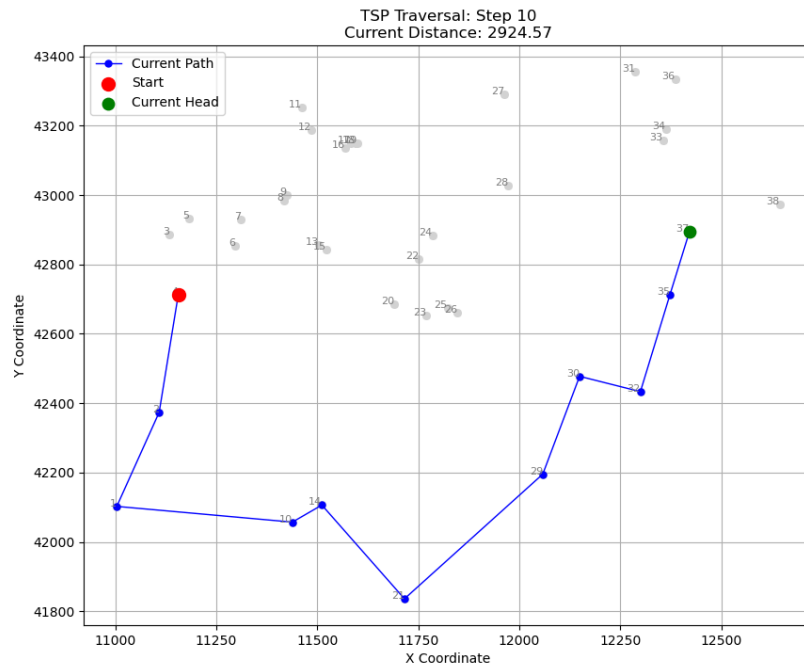


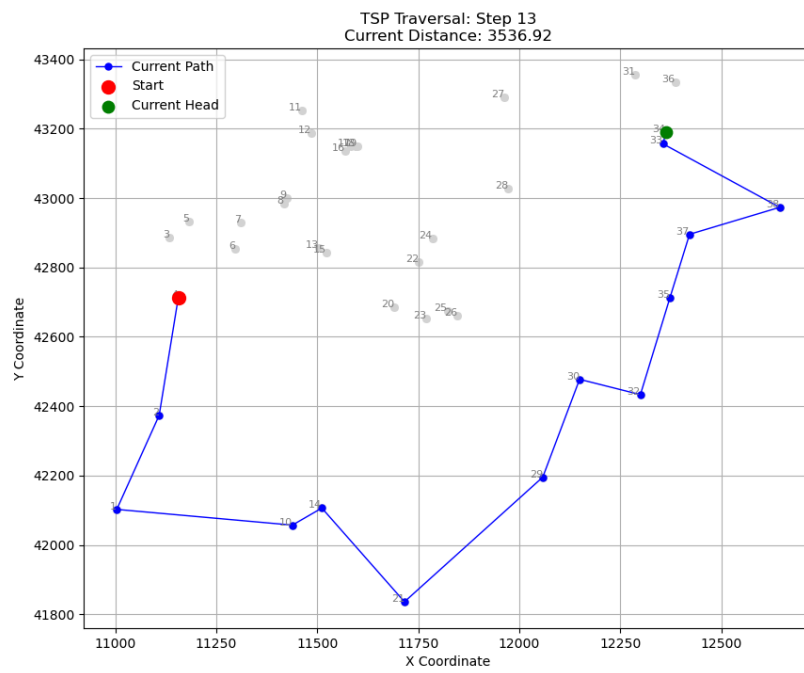
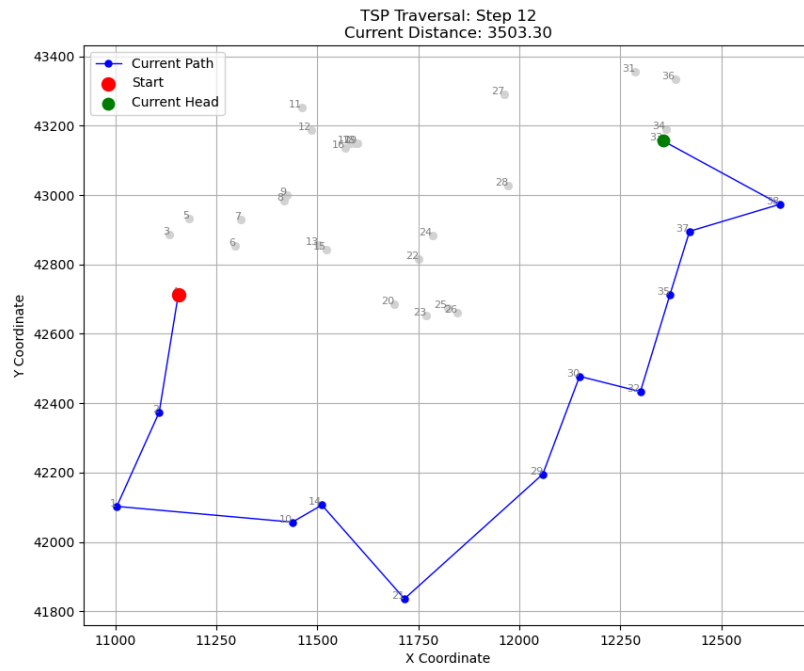


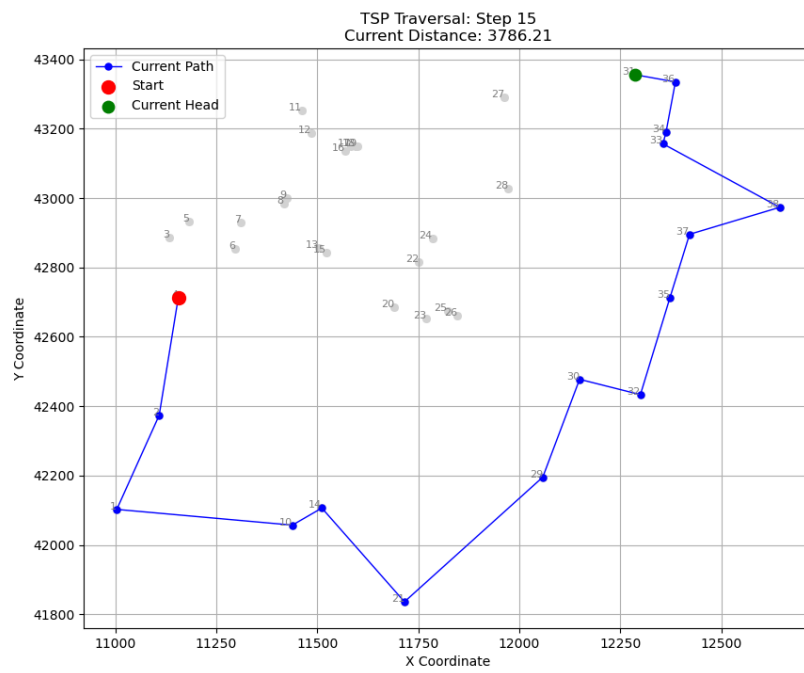
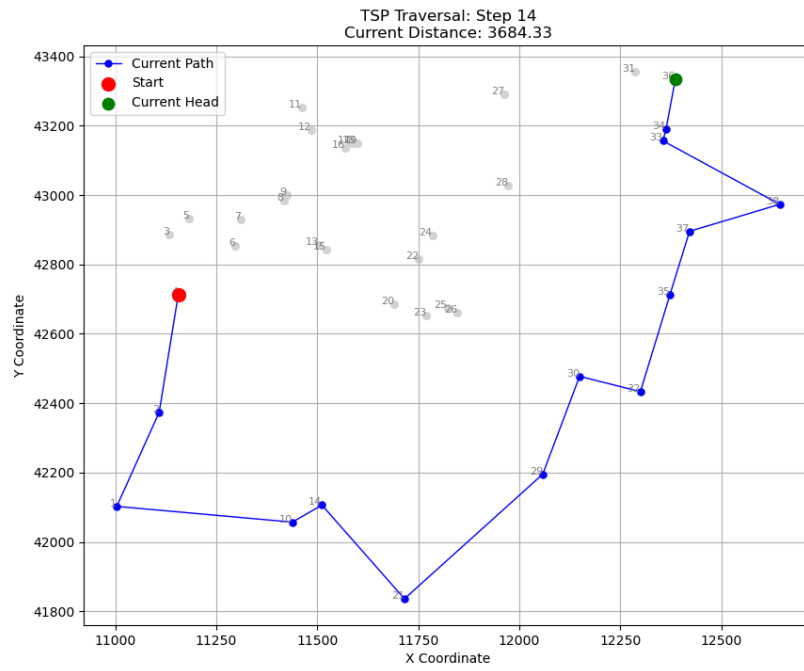


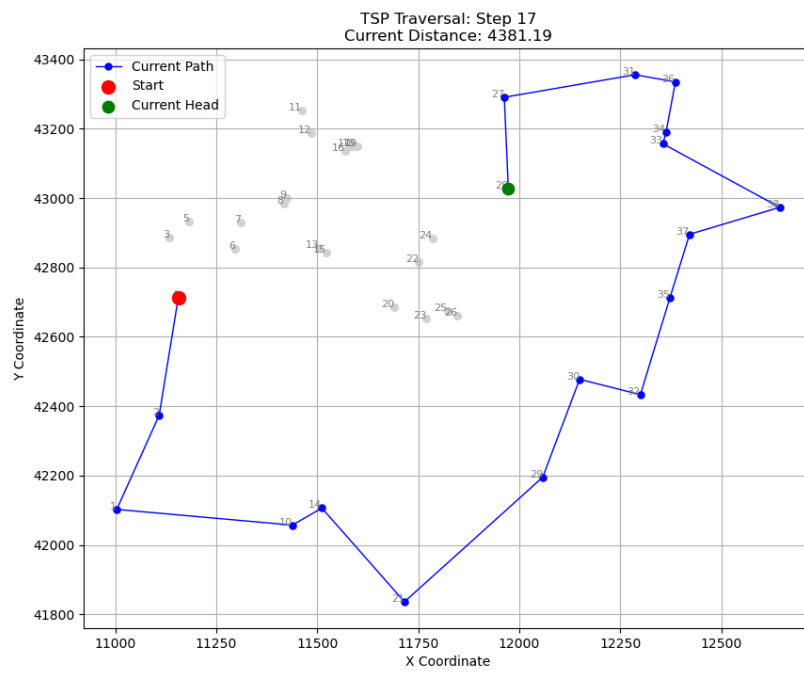
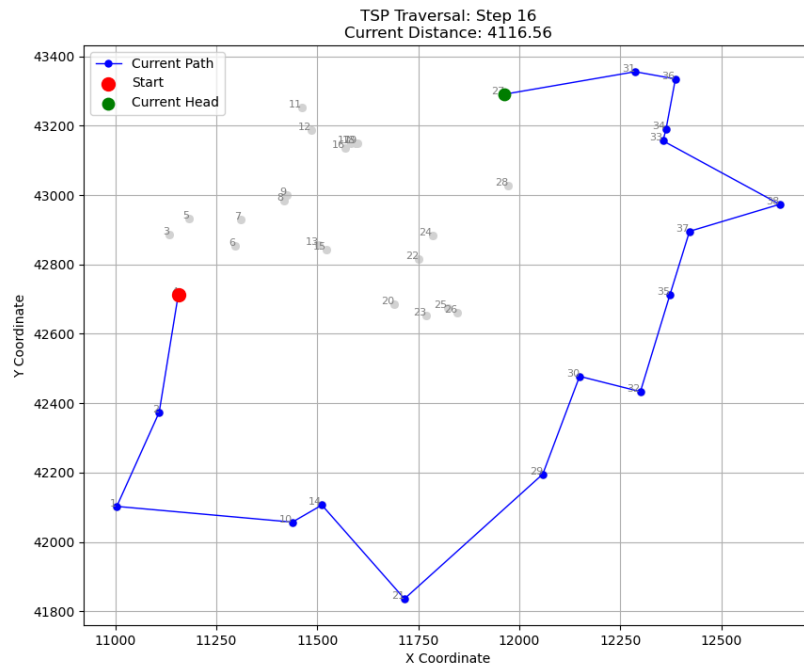


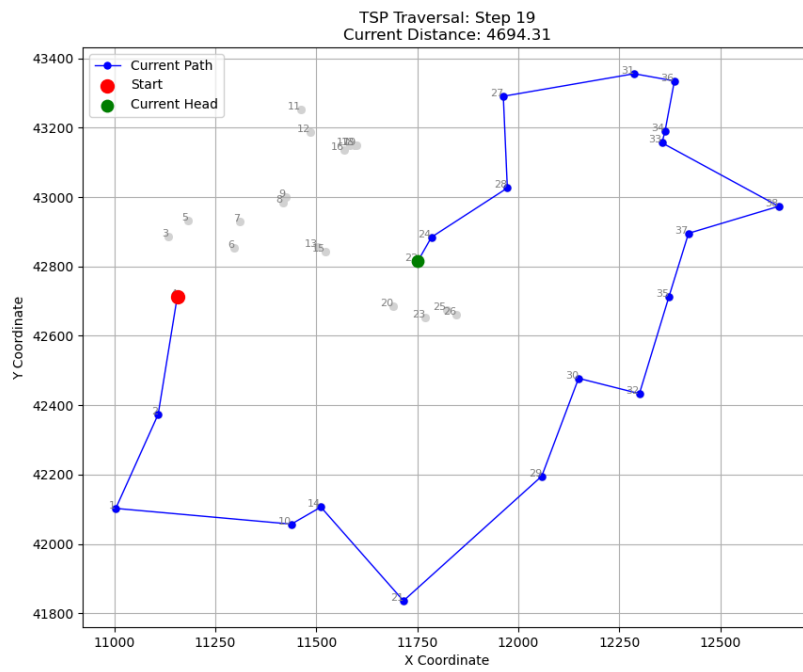
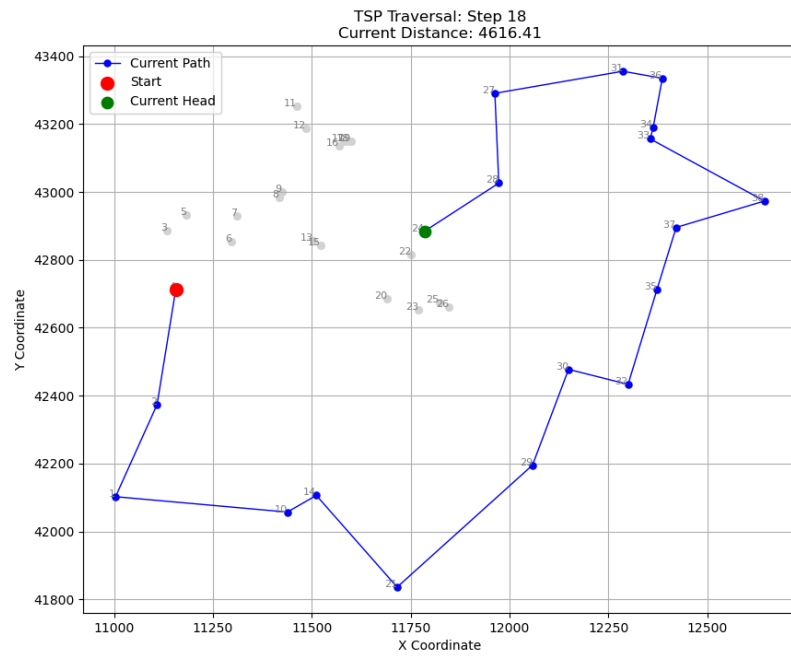


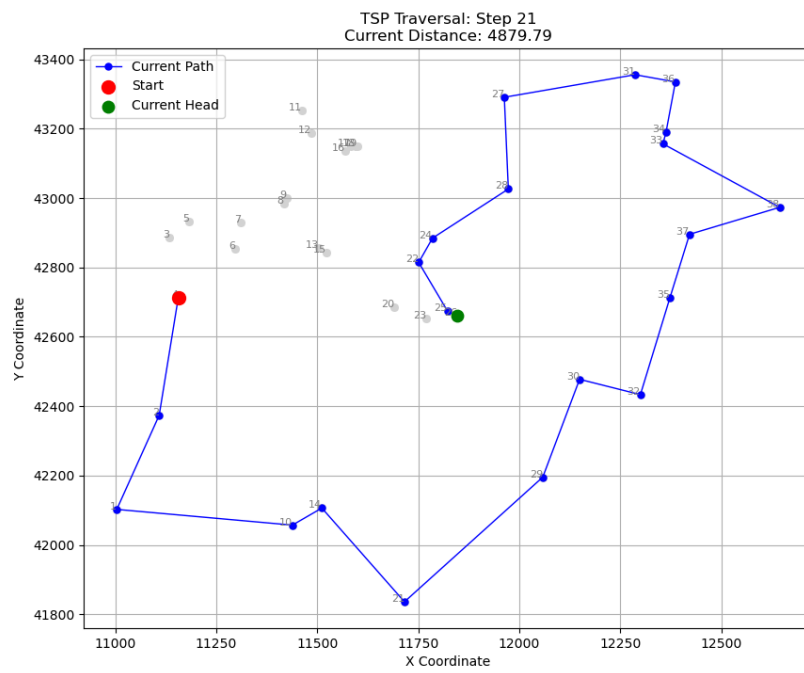
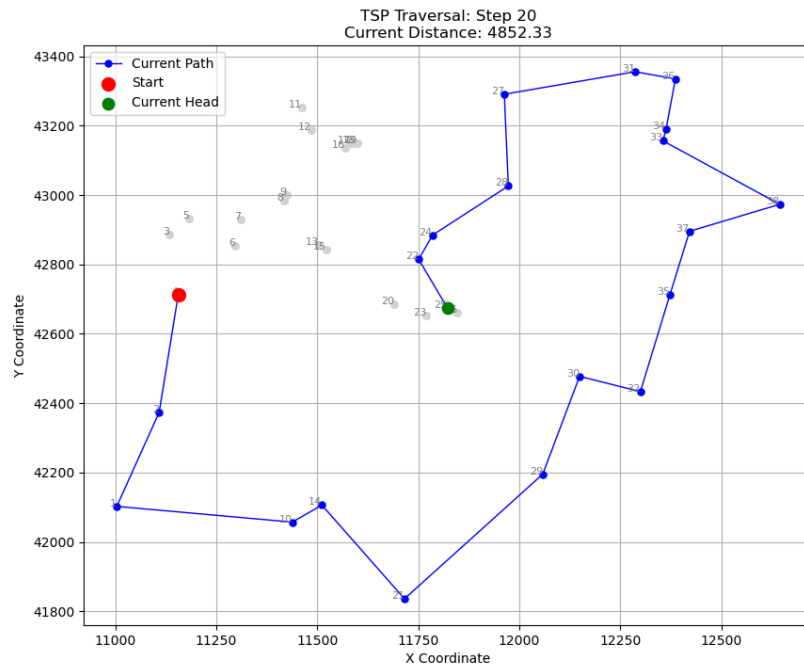


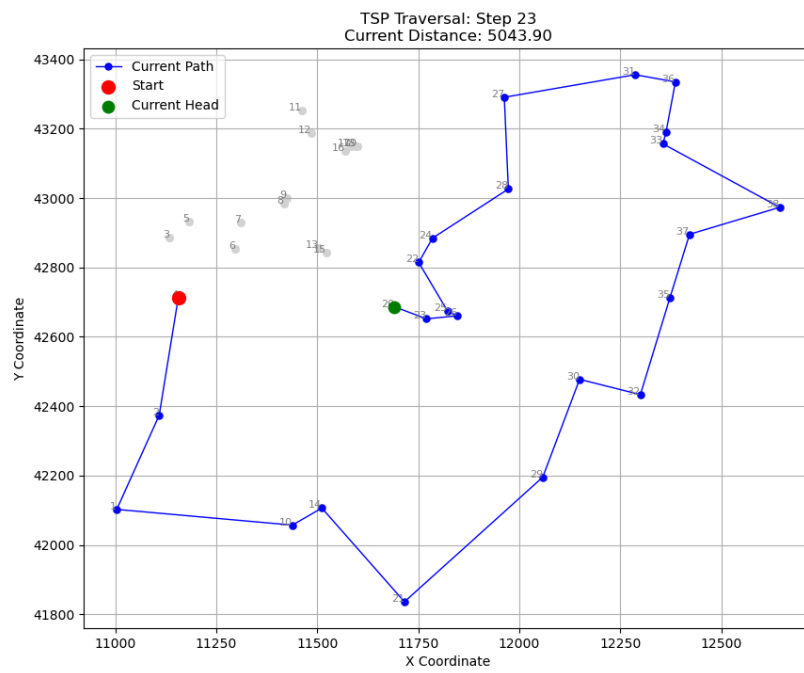
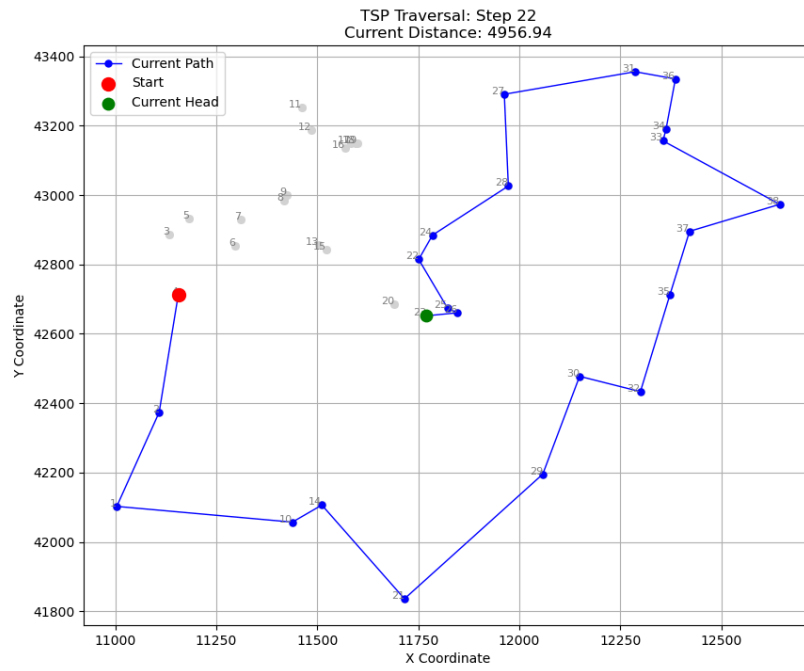


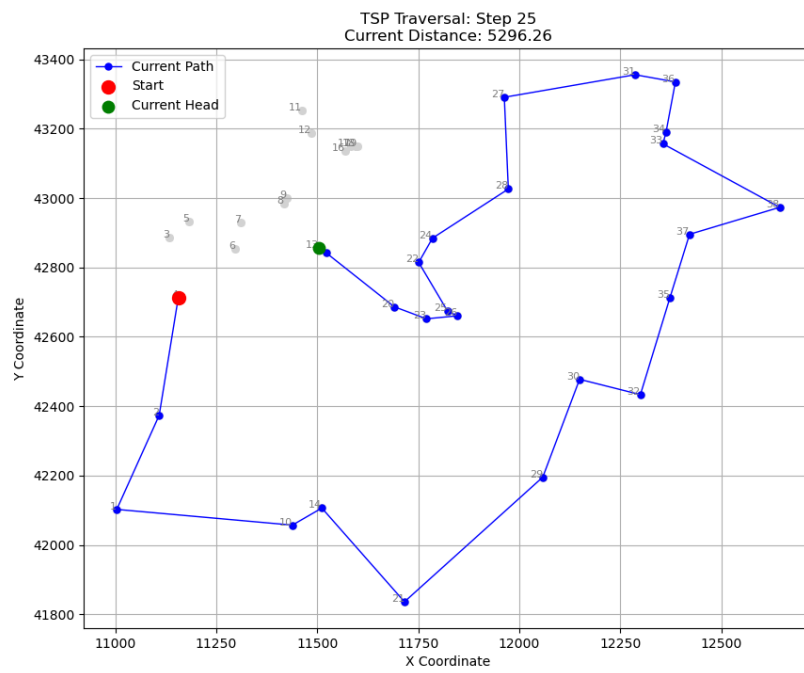
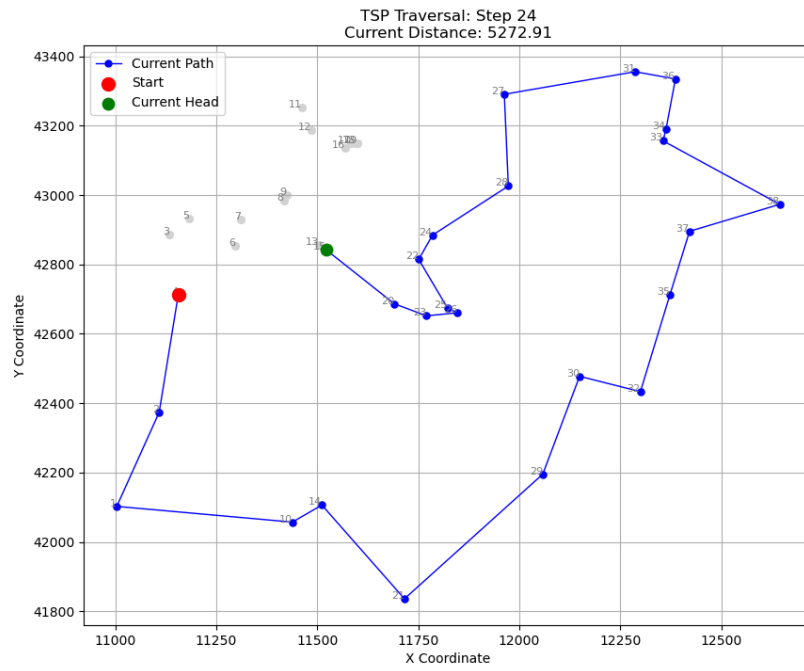


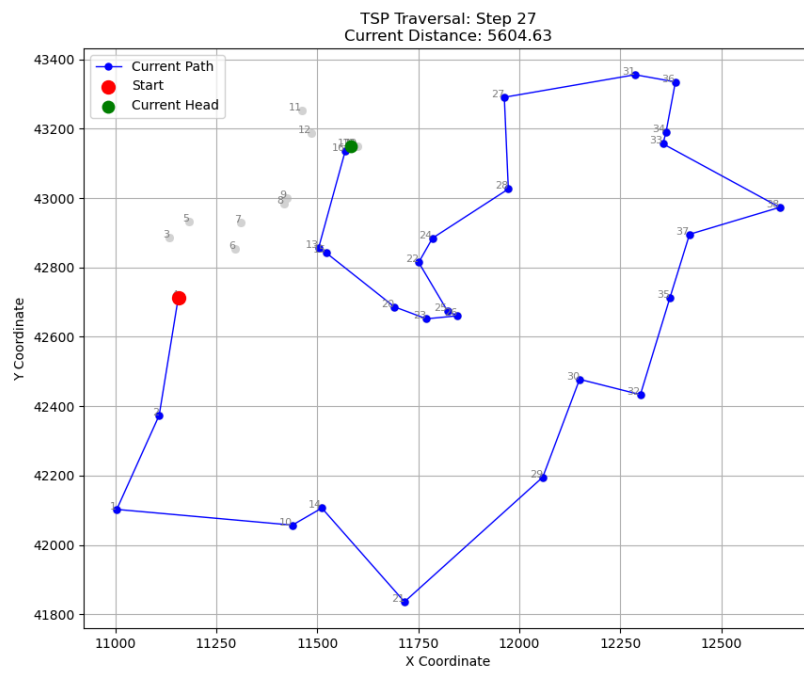
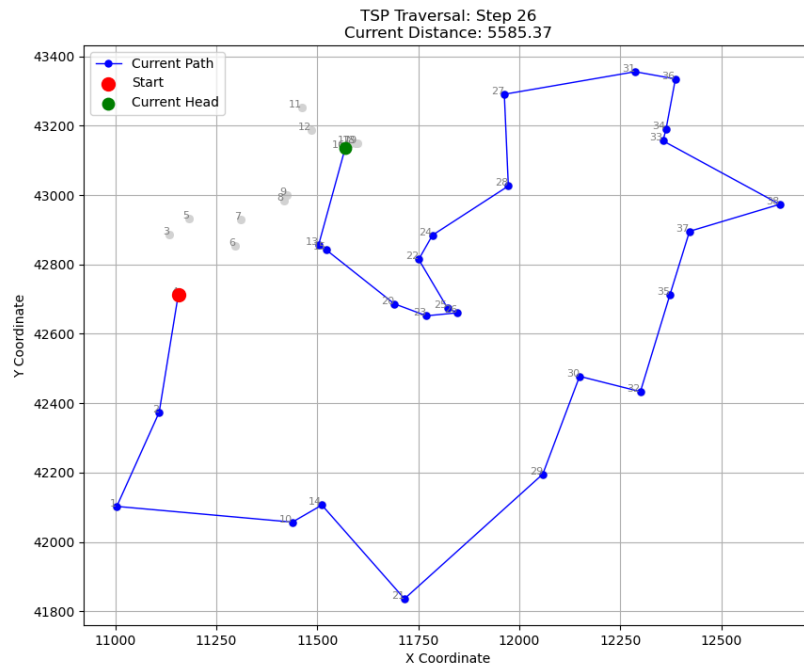


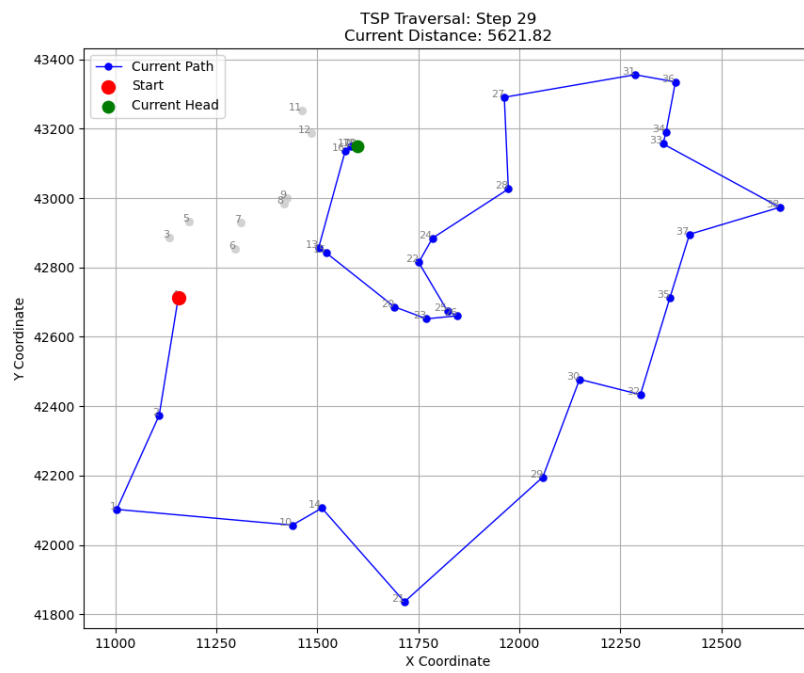
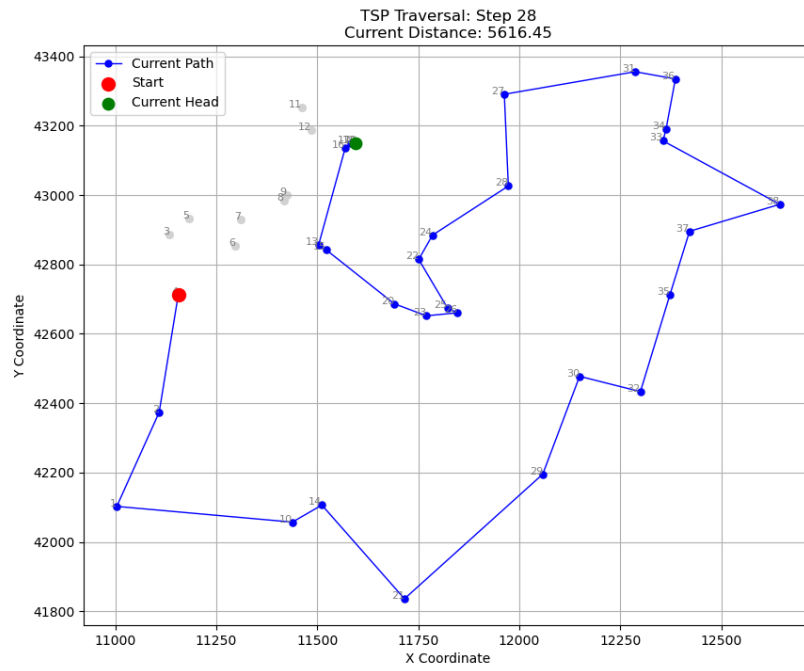


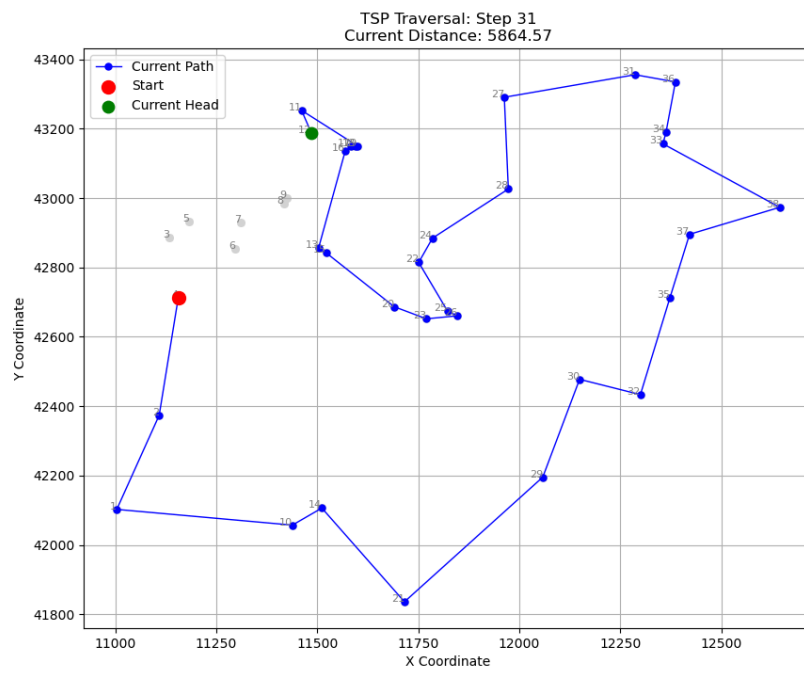
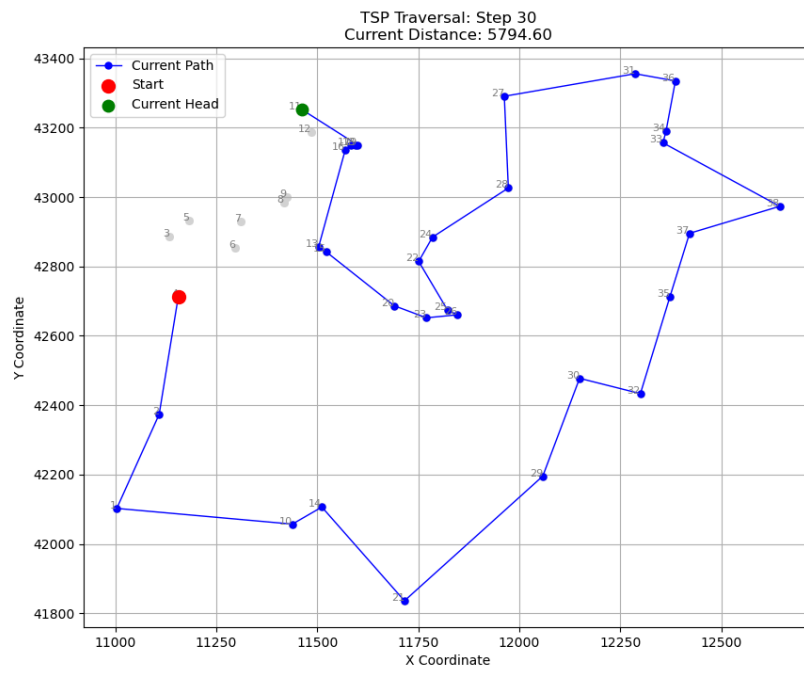


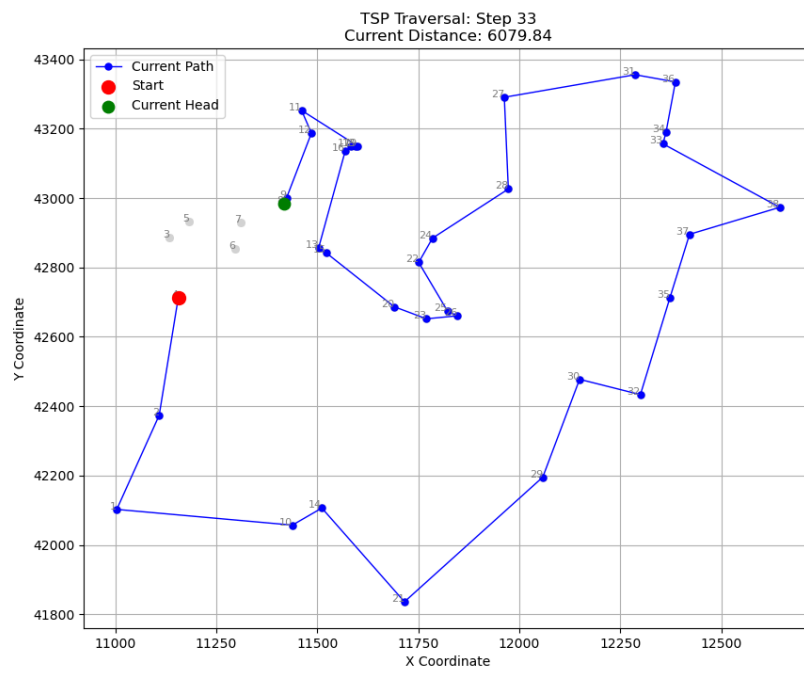
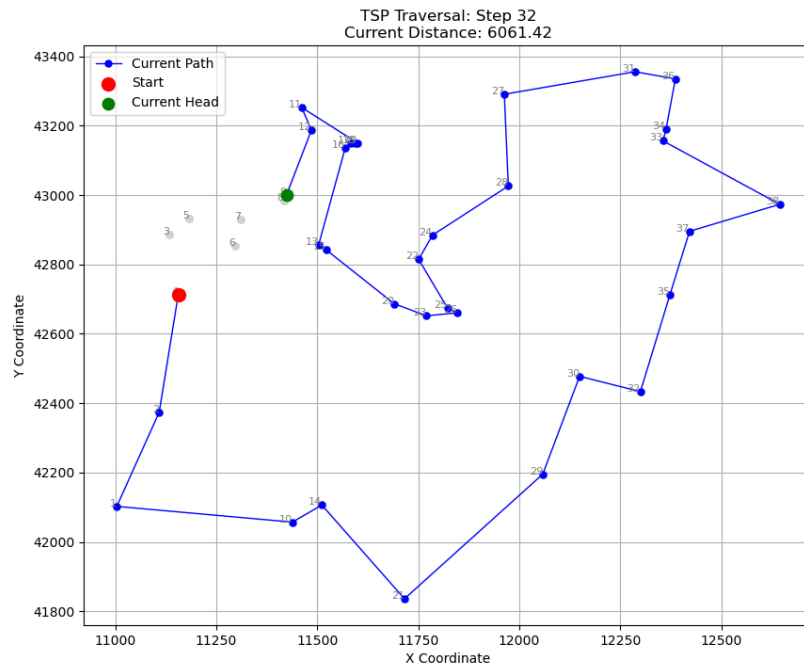


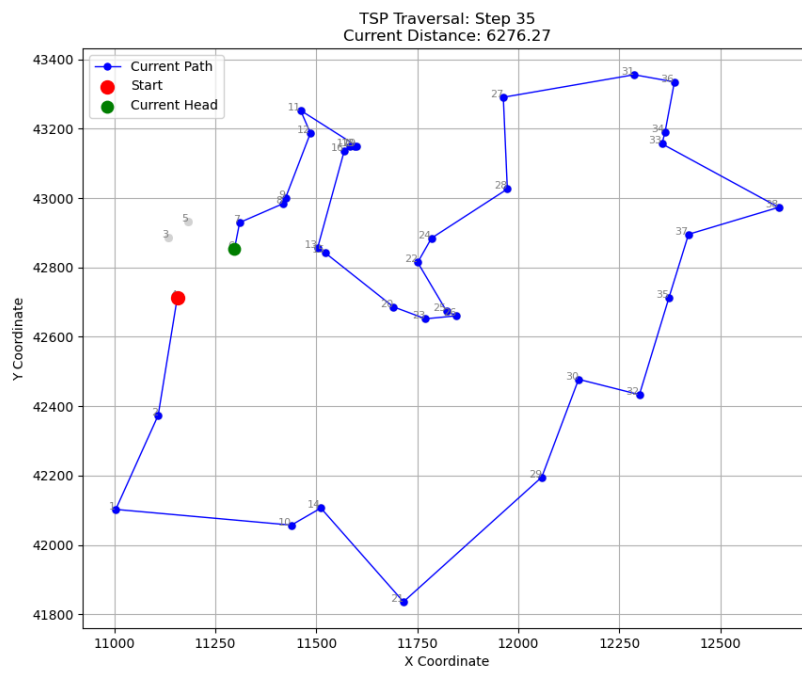
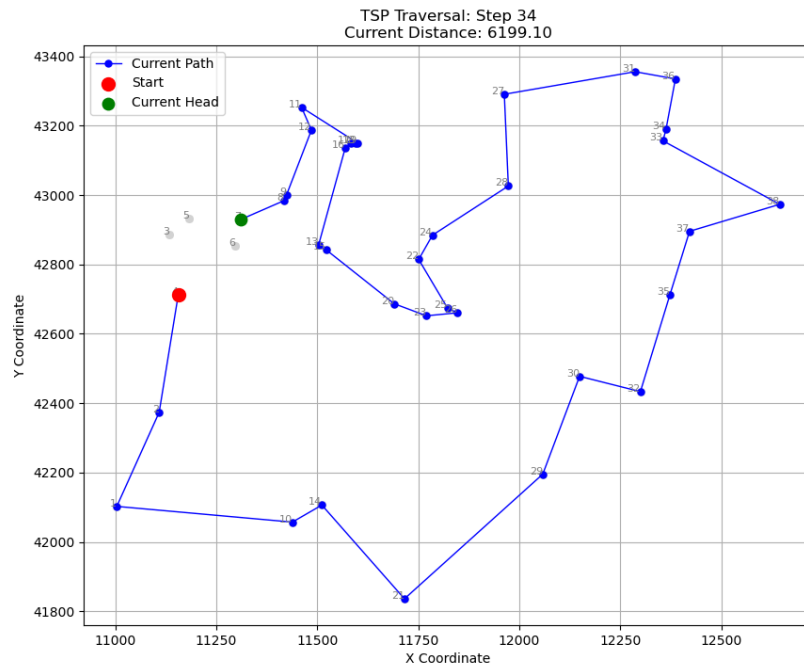


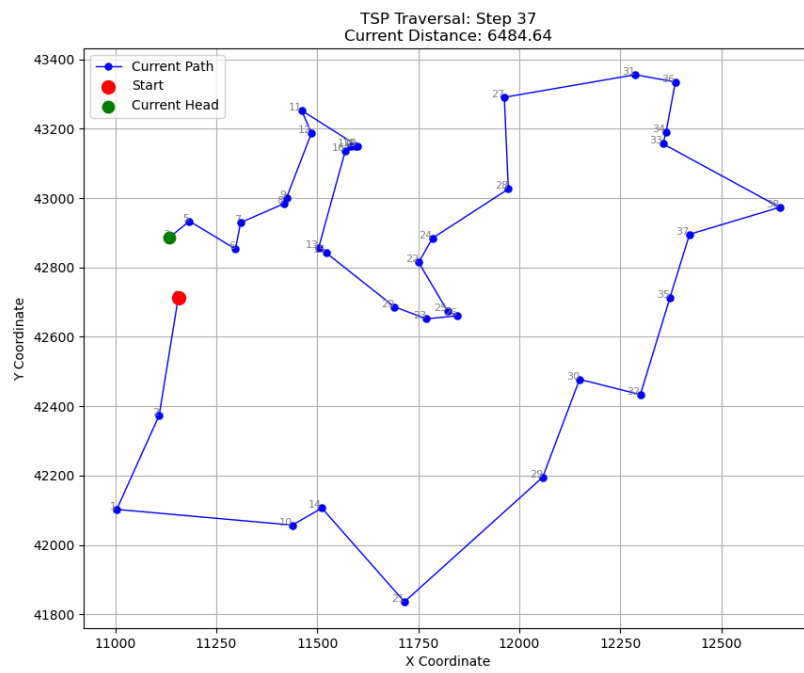
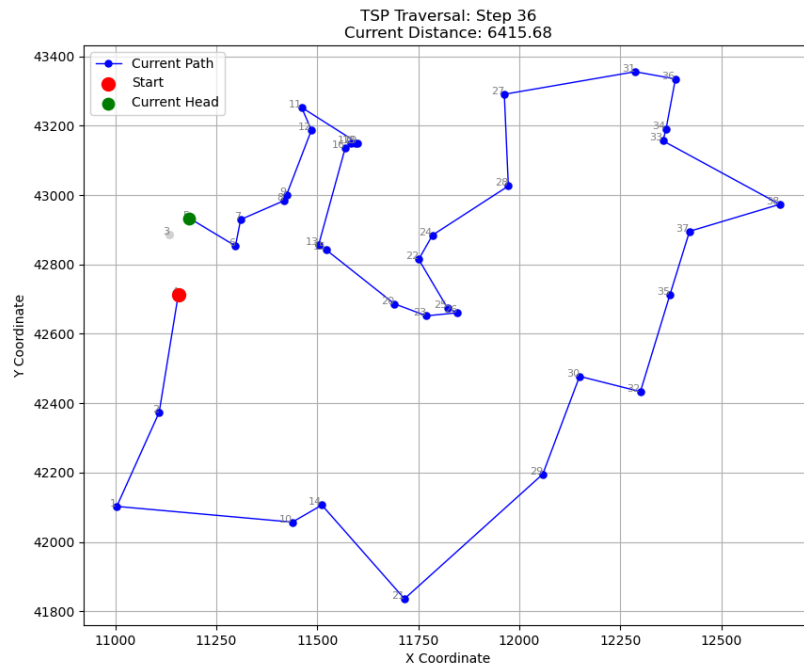


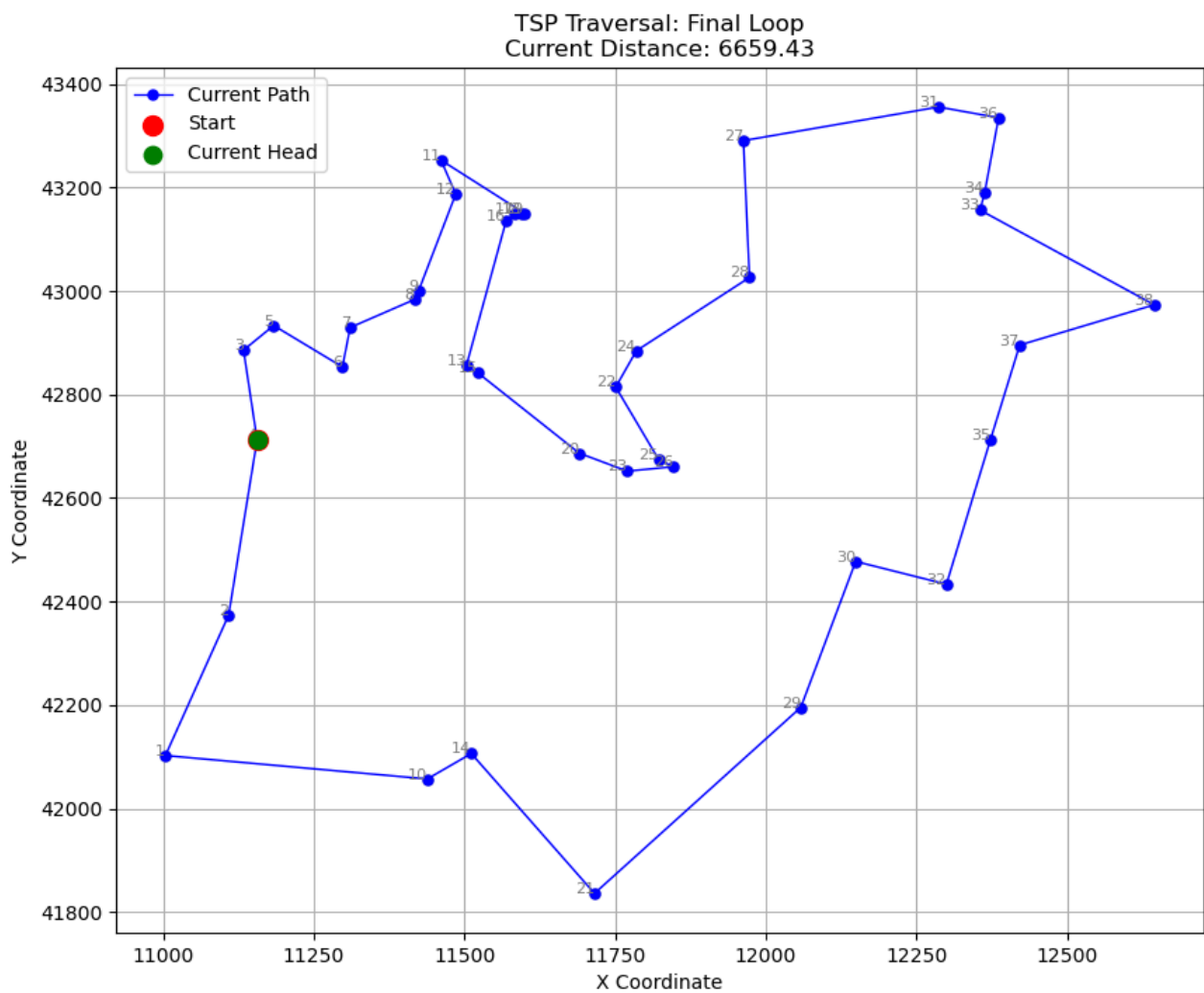












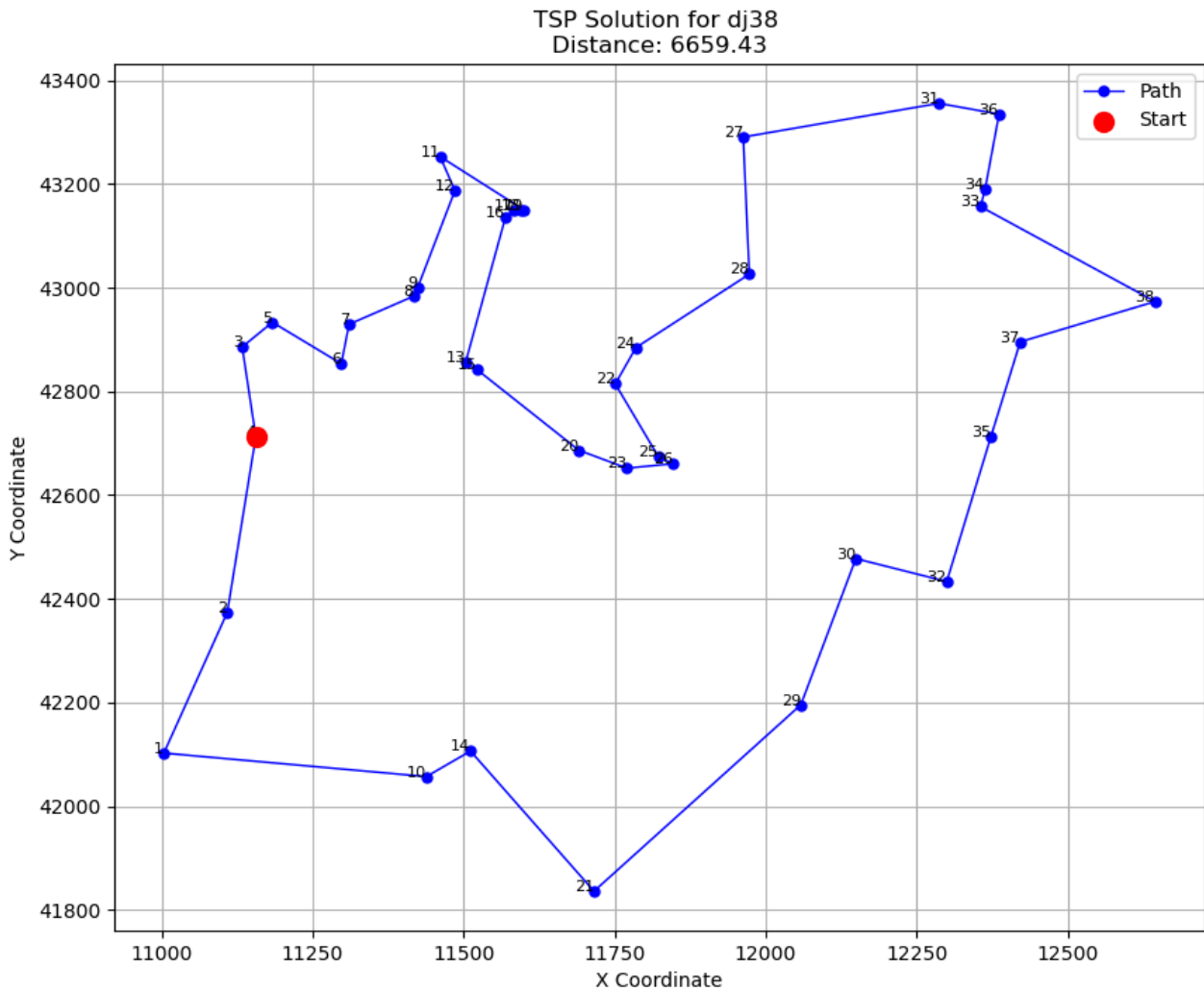
(3) 代码运行结果截图

问题 输出 调试控制台 终端

+ - □ ✕ ... | 🔍 ↗

```
python solve_tsp.py
Loaded 38 cities from dj38.tsp
Starting SA... Initial Distance: 30054.64
Finished in 135.36s

=====
Optimal Solution Found:
Shortest Distance: 6659.4315
  QT_SCREEN_SCALE_FACTORS to set per-screen factors.
  QT_SCALE_FACTOR to set the application global scale factor.
Final solution image saved as 'tsp_final_solution.png'
Generating traversal frames in 'frames/'...
Saved 39 frames to 'frames/'.
```



2. 推荐算法

(1) 算法设计思想和伪代码

算法设计思想：MF-Markov 混合推荐模型

在 MOOC 课程推荐场景中，用户的行为受两种主要因素驱动：

1. 长期兴趣 (Long-term Interest)：用户偏好某一类主题（如“计算机科学”、“历史”）。这部分偏好是静态且全局的，适合用矩阵分解 (Matrix Factorization, MF) 来捕捉。
2. 短期序列模式 (Short-term Sequential Pattern)：用户在学习一门课后，往往会紧接着学习其后续课程（如“微积分 I” -> “微积分 II”）。这种转移概率非常强，适合用马尔可夫链 (Markov Chain) 来捕捉。

本算法采用线性加权融合** (Linear Ensemble) 策略，结合了两者的优势：

- 使用 BPR-MF 学习用户和物品的隐向量，捕捉个性化偏好。
- 使用一阶马尔可夫链 (First-order Markov Chain) 统计物品间的转移概率，捕捉课程间的强关联。
- 最终分数是两者的加权和： $Score_{final} = \alpha \cdot Score_{MF} + (1 - \alpha) \cdot Score_{Markov}$

算法伪代码：

```

1  算法 MF_Markov_Ensemble_Recommender
2      输入：
3          train_data (训练集：用户-物品交互序列)
4          test_data (测试集：用户-物品交互)
5          alpha (融合权重,  $0 \leq \alpha \leq 1$ )
6          K (推荐列表长度)
7      输出：
8          HR@K, NDCG@K (评测指标)
9
10     // 阶段 1: 训练 BPR-MF 模型 (捕捉长期兴趣)
11     初始化 用户矩阵 P, 物品矩阵 Q (随机小数值)
12     循环 epoch 从 1 到 Max_Epochs:
13         从 train_data 中采样三元组 (u, i, j) // u:用户, i:正样本, j:负样本
14         计算预测差值:  $x_{uij} = (P[u] \cdot Q[i]) - (P[u] \cdot Q[j])$ 
15         计算梯度并更新 P[u], Q[i], Q[j] (使用 BPR 损失函数)
16     结束循环
17
18     // 阶段 2: 训练 Markov 模型 (捕捉序列模式)
19     初始化 转移矩阵 Transitions (字典或稀疏矩阵)
20     对于 train_data 中的每个用户 u:
21         按时间排序该用户的交互物品序列 [item_1, item_2, ..., item_T]
22         对于 t 从 1 到 T-1:
23             prev_item = item_t
24             next_item = item_{t+1}
25             Transitions[prev_item][next_item] += 1
26         结束循环
27     结束循环
28
29     // 阶段 3: 混合预测与评估
30     初始化 Hits = 0, NDCGs = 0
31     对于 test_data 中的每个用户 u 及目标物品 target_item:
32         last_item = 用户 u 在训练集中最后交互的物品
33         candidates = {target_item} + 99个随机负样本
34
35         对于 candidates 中的每个 item:
36             // 计算 MF 分数
37             score_mf =  $P[u] \cdot Q[item]$ 
38
39             // 计算 Markov 分数 (如果是 last_item 的后续课程则得分高, 否则为0)
40             score_markov = Transitions[last_item][item]
41
42         // 分数归一化 (Min-Max Normalization)
43         norm_mf = Normalize(score_mf_list)
44         norm_markov = Normalize(score_markov_list)
45
46         // 加权融合
47         final_scores =  $\alpha * \text{norm\_mf} + (1 - \alpha) * \text{norm\_markov}$ 
48
49         // 排序并计算指标
50         rank = Rank(final_scores, target_item)
51         如果 rank <= K:
52             Hits += 1
53             NDCGs +=  $1 / \log_2(\text{rank} + 1)$ 

```


54	
55	返回 Hits/Num_Test, NDCGs/Num_Test
56	结束算法

(2) 推荐结果的评测结果

在测试集上，我们使用 Hit Ratio (HR@10) 和 NDCG@10 作为评价指标。测试策略为：对于测试集中的每个用户，将其真实交互的一个课程与 99 个随机负样本课程混合，模型需要将真实课程排在前 10 名。

- HR@10 衡量能不能推荐对（是否出现在前10名里）。
- NDCG@10 衡量推荐得准不准（在出现的前提下，排名是否靠前）。

根据运行日志，最佳融合权重为 Alpha = 0.83 (即 83% BPR-MF + 17% Markov)，此时效果最好：

指标 (Metric)	Top-K	结果 (Score)	说明
Hit Ratio (HR)	10	0.6848	约 68.48% 的测试样本被成功推荐到了前 10 位。
NDCG	10	0.4776	归一化折损累计增益，反映了推荐结果的排序质量。

算法模型	Hit Ratio (HR@10)	NDCG@10	评价
随机猜测 (Random)	~0.1000	~0.0300	理论下限，完全无推荐能力
热门推荐 (Popularity)	0.5686	0.3411	强基准，代表“随大流”的效果
纯 Markov	0.6268	0.4348	当用户行为符合大众序列时，Markov 权重生效。
纯 BPR-MF	0.6801	0.4674	当序列断裂或冷启动时，MF 提供的个性化偏好。
MF-Markov	0.68.48	0.4776	显著优于基准

- 热门推荐的数据来自数据集上跑出的实测代码结果。
- 随机猜测的数据来自针对测试协议（1正99负）的数学理论推导。

附：热门推荐伪代码：

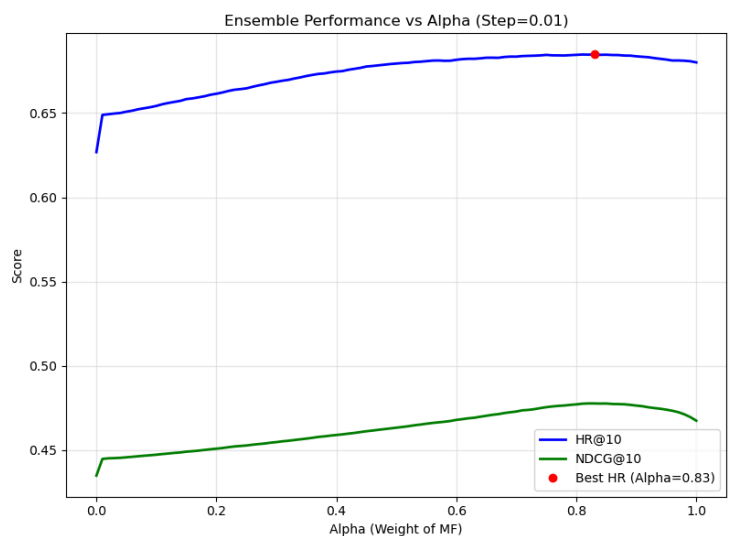
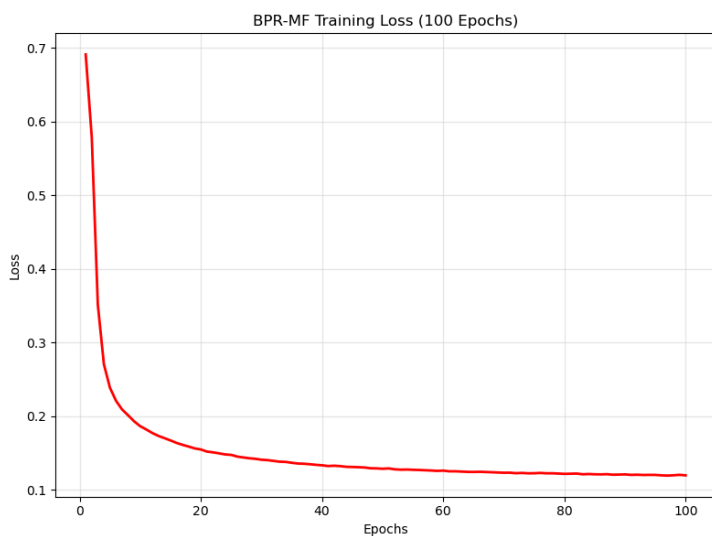
1	算法：基于热门度的推荐基准 (Popularity Baseline)
2	
3	输入：训练集 TrainData, 测试集 TestData, 负样本集 TestNegatives, 推荐数量 K
4	输出：Hit Ratio (HR), NDCG 指标
5	
6	1. 计算热门度 (Training Phase):
7	ItemCounts = 字典{}
8	对于 TrainData 中的每条交互记录 (User, Item):
9	如果 Item 在 ItemCounts 中:
10	ItemCounts[Item] += 1
11	否则:
12	ItemCounts[Item] = 1
13	
14	2. 评估阶段 (Evaluation Phase):
15	Hits = 0
16	NDCG_Sum = 0

```

17 NumTest = TestData 的样本数量
18
19 对于 TestData 中的每条记录 (User, GT_Item) 及对应的 Neg_Items:
20
21     a. 构建候选集合:
22         Candidates = [GT_Item] + Neg_Items (共 100 个课程)
23
24     b. 预测分数 (基于热门度):
25         Scores = []
26         对于 Candidates 中的每个 Item:
27             Score = ItemCounts.get(Item, 0) (如果训练集中没出现过, 热度为0)
28             Scores.添加(Score)
29
30     c. 排序与计算排名:
31         GT_Score = Scores[0] (真实课程的热度分)
32         Rank = 1
33         对于 Scores 中的其他分数 s:
34             如果 s > GT_Score:
35                 Rank += 1
36
37     d. 计算指标:
38         如果 Rank <= K:
39             Hits += 1
40             NDCG_Sum += 1 / log2(Rank + 1)
41
42 3. 输出结果:
43     HR = Hits / NumTest
44     NDCG = NDCG_Sum / NumTest
45     返回 HR, NDCG

```

(3) 代码运行结果截图



二、元启发式算法研究进展及教育领域应用综述

摘要

元启发式算法作为求解复杂优化问题的重要工具，近年来在理论创新和应用拓展方面取得了显著进展。本文系统综述了2023-2025年间元启发式算法领域的最新研究热点，深入分析了遗传算法、粒子群优化、蚁群优化、灰狼优化、鲸鱼优化、蝙蝠算法和萤火虫算法等常用算法的基本思想、数学模型和应用领域。研究发现，当前元启发式算法发展呈现出与深度学习融合、多目标优化技术突破、约束处理机制创新、并行计算加速等四大趋势，2023年后算法与大型语言模型的深度协同成为新的研究前沿。在教育领域应用方面，本文重点探讨了高校学生成绩展示分组优化问题，详细分析了遗传算法和蚁群优化算法在该问题中的应用，实验结果表明两种算法均能有效生成满足约束条件的分组方案，其中蚁群优化算法在75%的测试实例中表现更优或相当。本研究为元启发式算法在教育管理中的应用提供了理论基础和实践指导。

1. 引言

元启发式算法是一类基于自然现象或人类智能行为启发的优化方法，通过模拟生物进化、群体智能、物理过程等机制来寻找复杂优化问题的近似最优解[1]。随着科学技术的快速发展，现实世界中的优化问题日益复杂，传统的精确算法在面对大规模、多约束、非线性的NP-hard问题时往往显得力不从心，这为元启发式算法的发展提供了广阔的空间。

近年来，元启发式算法领域呈现出蓬勃发展的态势。据统计，2022-2025年间，研究团队提出了超过80种新型元启发式算法，涵盖从单种动物行为模仿到多模态融合的复杂系统[29]。这些算法不仅在理论上不断创新，更在实际应用中展现出强大的求解能力。特别是在人工智能、智能制造、能源管理、医疗诊断等领域，元启发式算法已成为解决复杂优化问题的重要工具。在教育领域，随着高等教育规模的不断扩大和教育管理复杂度的提升，各种优化问题层出不穷。例如，课程排课需要在有限的时间和空间资源下满足多重约束条件，学生分组需要考虑学习能力、兴趣爱好等多维度因素，资源分配需要平衡效率与公平等。这些问题往往具有NP-hard特性，传统的方法难以在合理时间内找到满意解。元启发式算法的引入为解决这些教育管理优化问题提供了新的思路和方法。

本文旨在系统综述元启发式算法的最新研究进展，深入分析常用算法的基本思想和应用领域，并重点探讨其在教育领域的应用。通过对2023-2025年最新文献的梳理和分析，本文将为该领域的研究人员和实践者提供全面的参考，推动元启发式算法在更多领域的创新应用。

2. 元启发式算法研究热点综述

2.1 算法创新与发展趋势

近年来，元启发式算法的创新呈现出多元化和系统化的特点。研究人员不仅从自然界中汲取灵感，更开始注重算法的模块化和跨学科融合。根据最新的研究分类，当前元启发式算法已形成四大主要类别：自然启发式（生物学与物理学）、人类行为启发式、游戏启发式以及无隐喻的群体优化算法。其中，自然启发的生物群体优化算法占据主导地位，相关研究占比超过50%，但物理学、游戏类及隐喻-free算法的发展呈现加速态势。

算法创新方面，2022-2025年间涌现出大量新颖的算法设计。代表性案例包括：埃博拉优化搜索算法（EOSA）整合流行病动力学模型与群体智能机制，在47个经典测试函数中表现优异；沙猫群集优化算法（SCSO）通过模拟沙猫的感官探测行为，在平衡探索与利用方面取得突破；烹饪训练优化算法（CBOA）借鉴烹饪学习阶段的师徒制，实现参数自适应性调整；手术刀训练优化算法（STBO）引入缝纫训练的三阶段模型，显著提升复杂路径规划效率[29]。这些算法的共同特点是不再局限于简单的行为模仿，而是深入挖掘现象背后的数学规律和物理机制。当前算法开发呈现出两大重要趋势。一方面，生物群体智能算法通过引入更多物种行为模型（如非洲水牛的多维觅食策略、海豹的群体协作机制）持续扩展方法论边界；另一方面，算法设计开始注重模块化架构，通过组合基础模块（探索模块、利用模块、多样性维持模块）实现快速定制化开发[29]。这种模块化设计使新型算法能在保持核心优势的同时快速适配不同场景，大大提高了算法的开发效率和应用灵活性。

元启发式算法正在经历从“仿生模仿”向“机制创新”的转变。研究人员不再满足于简单地将生物行为转化为数学模型，而是深入研究优化问题的本质特征，设计更加高效的搜索机制。同时，算法的可解释性和理论基础也日益受到重视，这推动元启发式算法向更加成熟和可靠的方向发展。

2.2 与深度学习的融合创新

元启发式算法与深度学习的融合已成为当前最具活力的研究方向之一。这种融合不仅体现在元启发式算法用于优化深度学习模型的参数和结构，更体现在两者在方法论层面的深度结合。据统计，仅 20% 的研究专注于优化神经网络中的小规模权重和偏差，这表明该领域仍有巨大的发展空间。

在深度学习模型优化方面，元启发式算法展现出独特的优势。Histogram-based Blocking Differential Evolution (HBDE) 算法的突破性成果令人瞩目：该算法将 ResNet-18 模型的参数从 11M 减少到 3K，在 CIFAR-10 和 CIFAR-100 数据集上的测试 F1 分数分别达到 81.83% 和 53.30%，显著优于传统的梯度下降方法[50]。这一成果不仅解决了大规模神经网络训练的计算瓶颈，更为元启发式算法在深度学习领域的应用开辟了新的道路；在超参数优化方面，元启发式算法同样表现出色。研究表明，元启发式方法在 Transformer 模型超参数调优中展现独特优势，可将训练时间缩短 40%[29]。这种显著的效率提升主要归功于元启发式算法的全局搜索能力，能够在复杂的超参数空间中快速找到接近最优的参数组合；更具创新性的是，2023 年后出现了算法与大型语言模型（LLMs）的深度协同现象。这种协同主要体现在三个方面：首先，LLMs 驱动算法生成，通过自然语言指令自动生成优化算法（如 Zoological Search Optimization），在 CEC 测试函数中达到 SABA 等传统算法的 92% 性能；其次，算法优化 LLMs，元启发式方法在超参数调优中展现独特优势；第三，人机协同进化，LLMs 辅助的群体优化算法（如 LLM-Driven EA）在 TSP 问题上实现 20 节点规模的最优解搜索时间低于 0.5 秒[29]；这种深度融合还催生了新的研究方向。Learnable Evolutionary Multi-Objective Combinatorial Optimization via Sequence-to-Sequence Model (SeqMO) 方法代表了这一趋势的最新成果，该方法将序列到序列模型与进化算法相结合，通过最小化目标向量角度建立解之间的映射关系，为指针网络创建结构化训练数据，在多目标旅行商问题和多目标二次分配问题上验证了算法的有效性(1)。

从技术发展角度看，元启发式算法与深度学习的融合正在从简单的应用层结合向方法论层面的深度融合转变。这种融合不仅提升了各自的性能，更创造了全新的算法范式，为解决更复杂的优化问题提供了强大的工具。

2.3 多目标优化技术突破

多目标优化是元启发式算法应用最为广泛的领域之一，近年来在理论方法和实际应用方面都取得了重要突破。根据最新研究统计，在调度优化目标中，makespan 最小化和能耗是最常研究的目标，大多数研究集中在双目标优化，处理三个或更多目标的研究相对较少。然而，随着实际问题复杂度的增加，多目标特别是三目标及以上的优化问题正成为研究热点。

在算法创新方面，研究人员提出了多种改进的多目标优化算法。NSGA-III、MOEA/D 等经典算法在处理大规模多目标问题时面临性能下降的挑战，而 R-NSGA-II 作为一种用于近似偏好帕累托最优解而非整个帕累托前沿的算法，在大规模多目标问题上表现出最具竞争力的性能[69]。这一发现为解决高维多目标优化问题提供了新的思路；

在约束处理技术方面，多目标优化面临着更加复杂的挑战。现代约束处理技术包括惩罚法、可行性规则、 ϵ -约束、随机排序、修复算法和基于边界的策略等，每种方法都有其优缺点[16]。研究人员提出了八种新的可行性规则和 ϵ -约束技术变体，这些新变体的独特之处在于除了目标函数值外，还考虑约束违反的程度和数量来选择种群中的个体。这种创新的约束处理机制显著提高了算法在处理复杂约束条件时的性能；在应用领域方面，多目标优化算法在智能制造、能源管理、交通运输等领域取得了广泛应用。在智能制造领域，算法需要同时优化生产效率、能源消耗和产品质量等多个目标；在能源管理领域，需要平衡发电成本、环境影响和供电可靠性；在交通运输领域，需要同时考虑运输时间、成本和安全性。这些应用场景的复杂性推动了多目标优化算法的不断创新和完善；特别值得关注的是，随着工业 4.0 和 5.0 时代的到来，多目标优化算法在智能制造中的应用呈现出新的特点。算法需要处理网络物理系统、物联网、人工智能和以人为中心的方法集成所带来的前所未有的复杂性[27]。这种复杂性不仅体现在目标函数的数量和类型上，更体现在约束条件的动态性和不确定性上。

从发展趋势来看，多目标优化技术正在向更高维度、更复杂约束、更强适应性的方向发展。未来的研究将更加注重算法的可扩展性、鲁棒性和实时性，以满足日益复杂的实际应用需求。

2.4 约束处理机制创新

约束处理是元启发式算法在实际应用中面临的核心挑战之一，特别是在处理复杂工程问题时，约束条件往往具有非线性、多模态、动态变化等特点。近年来，约束处理技术的创新主要体现在机制设计的智能化和自适应能力的提升上。

在约束处理策略方面，研究人员提出了多种创新方法。约束分离的进化多任务（CSEMT）方法为约束多目标优化问题提供了新的解决方案(64)。该方法通过将约束处理任务与目标优化任务分离，使算法能够更专注地处理各自的子任务，从而提高整体的求解效率。这种分离策略不仅简化了算法设计，更重要的是提高了算法在处理复杂约束条件时的灵活性和鲁棒性；在复杂约束处理方面，高约束多目标优化问题（HCMOPs）的求解是当前的研究热点。这类问题具有复杂约束和小可行区域的特点，传统的

约束处理技术在处理 HCMOPs 时面临两个主要困难：一是难以找到可行解，大量搜索努力花费在定位可行区域上；二是由于 HCMOPs 的总可行区域可能由几个不相连的子区域组成，搜索过程可能被困在相对较大的可行子区域中，而该子区域不包含整个帕累托前沿[74]。针对这些挑战，研究人员提出了基于差分进化算法的约束松弛策略（CRS-DE），通过将不可行解分为“半可行”子种群和“不可行”子种群，分别采用不同的繁殖和选择策略，实现了明确的分工与合作，有效促进了可行解的搜索[74]；在动态约束处理方面，研究人员开发了自适应多阶段进化搜索框架（MSEFAS），用于高效处理约束多目标优化问题。该框架的创新之处在于能够根据约束违反的程度动态调整搜索策略，在算法运行过程中自适应地平衡探索和利用。这种动态调整机制使算法能够在保持全局搜索能力的同时，快速收敛到可行区域；在约束建模方面，研究人员开始采用更加灵活和精确的方法。多粒度代理建模方法通过根据种群在适应度景观中的位置自适应确定约束代理的近似粒度，有效解决了约束违反（CV）景观过于复杂难以准确近似的问题[67]。这种方法不仅提高了约束处理的精度，还大大降低了计算成本。

从技术发展角度看，约束处理机制正在向智能化、自适应、高效化的方向发展。未来的研究将更加注重约束条件的动态识别、智能分类和自适应处理，以应对日益复杂的实际应用场景。

2.5 并行计算加速技术

并行计算技术的发展为元启发式算法的性能提升提供了强大的硬件支撑。GPU（图形处理器）作为大规模并行计算的代表，在加速元启发式算法方面展现出巨大潜力。据统计，在大型组合问题上，GPU 加速可实现高达 80 倍的加速比，在连续问题上可达到 240 倍[15]。这一显著的性能提升使得元启发式算法能够处理更大规模、更复杂的优化问题。

在算法并行化策略方面，研究人员提出了多种创新方法。动态策略的并行蚁群优化算法在 GPU 上的实现代表了这一领域的最新进展。该算法提出了 KE-ALL 和 KE-ONE 两种核心策略，分别适用于不同规模问题的求解。同时，研究人员还提出了一种新的并行实现的轮盘赌选择算法，称为 Tiling Roulette (TR)，以及面向单指令多数据 (SIMD) 的路径构建算法 (STC)，并在 GPU 上实现了新颖的动态工作组策略 (ST-DYNAMIC) [25]。这些创新使得算法能够充分利用 GPU 的并行计算能力，在处理大规模旅行商问题时取得了显著的性能提升；在并行框架开发方面，研究人员开发了多个高效的并行计算框架。ParadisEO-MO-GPU 是一个开创性的框架，专门用于在 GPU 上可重用设计和实现并行局部搜索元启发式算法[23]。该框架的设计理念是让 GPU 对用户尽可能透明，最小化用户在 GPU 管理方面的参与。这种设计大大降低了并行算法开发的难度，使更多的研究人员能够享受到并行计算带来的性能提升；在算法性能优化方面，无隐喻优化算法的 GPU 并行加速取得了突破性进展。研究人员使用 Julia 编程语言实现了这些算法的 GPU 加速版本，在高端专业级 GPU 和强大的消费级 GPU 上进行测试，使用一组困难的大规模非线性方程组问题来评估并行化的加速增益。计算实验产生了显著的加速增益，范围从 33.9 倍到 561.8 倍，具体取决于测试参数和用于测试的 GPU[28]。这种巨大的性能提升使得原本需要数小时甚至数天才能完成的计算任务可以在几分钟内完成；在应用领域拓展方面，GPU 加速的元启发式算法已广泛应用于科学计算、工程设计、金融分析等领域。在科学计算中，算法被用于求解大规模的偏微分方程和优化复杂的物理模型；在工程设计中，用于优化复杂结构的设计参数和材料配置；在金融分析中，用于处理大规模的投资组合优化和风险评估问题。

目前，并行计算加速技术正在向更高并行度、更低功耗、更强可编程性的方向发展。随着新型 GPU 架构的不断推出和并行算法设计方法的不断完善，元启发式算法的计算效率将得到进一步提升，能够处理的问题规模也将不断扩大。

3. 常用元启发式算法介绍

3.1 经典算法

3.1.1 遗传算法

遗传算法（Genetic Algorithm, GA）是一类借鉴生物界自然选择和遗传机制的随机化搜索算法，由 Holland 于 1975 年首次提出。该算法通过模拟自然选择、交叉和变异等进化过程来寻找最优解，其核心思想源于达尔文的进化论和孟德尔的遗传学说[79]。

遗传算法的基本原理是将问题的解编码为染色体（通常是二进制或实数向量），通过适应度函数评估每个个体的优劣，然后通过选择、交叉和变异等遗传算子产生下一代种群。算法的数学模型相对简单但功能强大。在标准遗传算法中，选择算子通常采用轮盘赌选择或锦标赛选择，交叉算子包括单点交叉、多点交叉和均匀交叉等，变异算子则以较小的概率随机改变染色体的某些基因值。遗传算法的一个重要特征是其全局搜索能力。由于算法同时维护多个解（种群），并通过交叉算子实现信息的交换和重组，因此具有很强的跳出局部最优的能力。同时，变异算子的引入保证了种群的多样性，避免了算法过早收敛。研究表明，遗传算法在处理多模态、非线性、不可微的优化问题时表现出色。在应用领域方面，遗传算法已广泛应用于各个领域。在工程设计

中，遗传算法被用于优化结构设计参数、材料配置和工艺参数；在生产调度中，用于解决作业车间调度、流水车间调度等复杂问题；在机器学习中，用于优化神经网络结构、特征选择和超参数调优；在图像处理中，用于图像分割、特征提取和目标识别。特别是在解决 NP-hard 问题方面，遗传算法展现出了独特的优势。

近年来，遗传算法的发展呈现出几个重要趋势。首先是与其他算法的融合，如遗传算法与粒子群优化的结合、遗传算法与模拟退火的结合等，这些混合算法往往能够综合各算法的优势，取得更好的性能。其次是并行化实现，通过分布式计算和 GPU 加速，大大提高了算法的运行效率。第三是自适应机制的引入，如自适应交叉和变异概率、自适应种群规模等，使算法能够根据问题特点自动调整参数。

3.1.2 粒子群优化算法

粒子群优化算法（Particle Swarm Optimization, PSO）是由 Kennedy 和 Eberhart 于 1995 年提出的一种基于群体智能的优化算法，灵感来源于鸟群觅食和鱼群游动等社会行为[91]。该算法通过模拟群体中个体之间的信息共享和协作来实现优化搜索，具有概念简单、参数少、收敛速度快等优点。

粒子群优化算法的基本思想是将每个解视为搜索空间中的一个粒子，每个粒子都有自己的位置和速度。粒子在搜索空间中飞行，通过跟踪两个极值来更新自己的位置：一个是粒子自身找到的最优解（个体最优 $pbest$ ），另一个是整个种群找到的最优解（全局最优 $gbest$ ）。粒子的速度和位置更新公式如下：

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i^t - x_i^t) + c_2 r_2 (gbest^t - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

其中， ω 是惯性权重，用于平衡全局探索和局部开发； c_1 和 c_2 是加速常数，通常设为 2； r_1 和 r_2 是在 $[0,1]$ 范围内均匀分布的随机数。

粒子群优化算法的优势在于其简单性和高效性。算法只需要调节很少的参数，通常包括种群大小、惯性权重、加速常数等。通过调节惯性权重 ω ，可以控制算法的探索和开发能力：较大的 ω 有利于全局探索，较小的 ω 有利于局部开发。研究表明，自适应惯性权重策略能够显著提高算法的性能；在应用领域方面，粒子群优化算法已广泛应用于多个领域。根据最新统计，粒子群优化算法的应用分布为：通用应用 27%、工业应用 20%、环境应用 17%、智能城市应用 16%、医疗应用 11%、商业应用 9%。在工程优化中，粒子群优化算法被用于结构设计、参数优化、系统辨识等；在信号处理中，用于滤波器设计、信号检测和参数估计；在机器学习中，用于神经网络训练、支持向量机参数优化、特征选择等；在电力系统中，用于经济调度、潮流计算、故障诊断等；

粒子群优化算法的发展也呈现出多样化的趋势。研究人员提出了多种改进版本，如自适应粒子群优化、混合粒子群优化、多目标粒子群优化、量子粒子群优化等。这些改进算法在保持基本算法优点的同时，在收敛速度、全局搜索能力、鲁棒性等方面都有了显著提升。特别是与其他算法的混合，如粒子群优化与遗传算法、粒子群优化与模拟退火、粒子群优化与差分进化等，往往能够产生协同效应，取得更好的优化效果。

3.1.3 蚁群优化算法

蚁群优化算法（Ant Colony Optimization, ACO）是由 Dorigo 等人于 1990 年代初期提出的一种模拟蚂蚁群体觅食行为的优化算法[80]。该算法的灵感来源于真实蚂蚁通过信息素进行间接通信，从而找到从巢穴到食物源最短路径的行为。蚁群优化算法已成为求解复杂组合优化问题的有效工具，特别是在旅行商问题、车辆路径问题、调度问题等方面取得了卓越成果。

蚁群优化算法的基本原理是利用人工蚂蚁在问题空间中构建解，并通过信息素的更新机制来引导后续搜索。算法的核心要素包括：人工蚂蚁、信息素轨迹和启发式信息。蚂蚁在构建解的过程中，根据信息素浓度和启发式信息（如距离的倒数）以概率方式选择下一步的移动。信息素的更新包括蒸发和增强两个过程：蒸发机制防止信息素的无限积累，增强机制则根据解的质量更新路径上的信息素。

标准蚁群系统（Ant System, AS）的信息素更新规则为：

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum \Delta \tau_{ij}^k(t) \quad (3)$$

其中， ρ 是信息素蒸发率（ $0 < \rho \leq 1$ ）， $\Delta \tau_{ij}^k(t)$ 是第 k 只蚂蚁在边 (i, j) 上沉积的信息素量，通常定义为 $1/L^k(t)$ ，其中 $L^k(t)$ 是第 k 只蚂蚁找到的路径长度。

蚁群优化算法的优势在于其强大的正反馈机制和分布式计算特性。通过信息素的积累和更新，算法能够快速收敛到高质量的解。同时，蚂蚁之间的并行搜索和信息共享机制使得算法具有良好的全局搜索能力。研究表明，蚁群优化算法在处理离散优化问

题时表现出色，特别是在旅行商问题、车辆路径问题、二次分配问题等经典 NP-hard 问题上取得了世界一流的结果[80]；在应用领域方面，蚁群优化算法已广泛应用于多个领域。在交通运输中，用于车辆路径规划、公共交通调度、物流配送等；在通信网络中，用于路由选择、负载均衡、网络优化等；在电力系统中，用于电网重构、故障诊断、电力调度等；在生物信息学中，用于蛋白质结构预测、基因序列分析等；在机器人领域，用于路径规划、多机器人协作等。特别是在动态环境中，如通信网络的路由问题，蚁群优化算法展现出了良好的适应性和鲁棒性。

蚁群优化算法的发展也呈现出多样化的趋势。研究人员提出了多种改进算法，如蚁群系统（ACS）、最大最小蚁群系统（MMAS）、蚁群算法的精英策略（ASrank）等。这些改进算法在信息素更新策略、蚂蚁选择规则、参数设置等方面都有了优化，显著提高了算法的性能。同时，蚁群优化算法与其他算法的融合也产生了许多高效的混合算法，如蚁群 - 遗传混合算法、蚁群 - 粒子群混合算法等。

3.2 现代算法

3.2.1 灰狼优化算法

灰狼优化算法（Grey Wolf Optimizer, GWO）是由 Mirjalili 和 Lewis 于 2014 年提出的一种新型元启发式算法，灵感来源于灰狼的社会等级制度和群体狩猎行为[101]。该算法通过模拟灰狼群体中的 α 、 β 、 δ 、 ω 四个等级的社会结构，以及它们的包围、狩猎和攻击猎物的行为，实现对优化问题的求解。

灰狼优化算法的核心思想是将搜索代理分为四个等级： α 狼（领导者，负责决策）、 β 狼（副领导者，协助 α 狼）、 δ 狼（追随者，听从 α 和 β 的指挥）和 ω 狼（底层成员）。在算法中， α 、 β 、 δ 被认为是当前最优的三个解，它们引导种群向最优区域搜索。算法的数学模型基于灰狼的狩猎行为，包括三个主要步骤：包围猎物、狩猎和攻击猎物。

包围猎物的数学模型为：

$$D = |C \times X_p(t) - X(t)| \quad (4)$$

$$X(t+1) = X_p(t) - A \times D \quad (5)$$

其中， X_p 是猎物位置， X 是灰狼位置， t 是当前迭代， A 和 C 是系数向量，计算公式为：

$$A = 2a \times r_1 - a \quad (6)$$

$$\text{公式} \quad (7)$$

其中， r_1 、 r_2 是 [0,1] 范围内的随机向量， a 是从 2 线性递减到 0 的参数。

在狩猎阶段，算法假设 α 、 β 、 δ 狼对猎物位置有更好的了解，因此其他狼（ ω 狼）根据这三者的位置更新自己的位置：

$$X_1 = X_\alpha - A_1 \times D_\alpha \quad (8)$$

$$X_2 = X_\beta - A_2 \times D_\beta \quad (9)$$

$$X_3 = X_\delta - A_3 \times D_\delta \quad (10)$$

$$X(t+1) = (X_1 + X_2 + X_3)/3 \quad (11)$$

攻击阶段通过控制参数 a 的值来实现，当 a 在 [0,1] 范围内时，算法进入开发阶段，灰狼开始攻击猎物。

灰狼优化算法的优势在于其简单的结构和强大的搜索能力。算法只需要调节很少的参数，主要是种群大小和最大迭代次数。通过参数 a 的线性递减，算法能够自动平衡全局探索和局部开发。研究表明，灰狼优化算法在处理连续优化问题时表现出色，特别是在工程设计优化方面取得了显著成果[101]；在应用领域方面，灰狼优化算法已广泛应用于多个领域。在工程设计中，用于结构优化、机械设计、电路设计等；在能源系统中，用于电力系统优化、可再生能源调度等；在生物信息学中，用于蛋白质结构预测、基因序列分析等；在图像处理中，用于图像分割、特征提取等；在无线通信中，用于资源分配、网络优化等。特别是在多目标优化问题上，研究人员提出了多目标灰狼优化算法（MOGWO），通过集成固定大小的外部档案来保存和检索帕累托最优解[98]。

然而，灰狼优化算法也存在一些局限性。研究指出，灰狼优化算法存在种群多样性不足、探索与开发失衡、过早收敛等问题。为了克服这些缺点，研究人员提出了多种改进算法，如改进灰狼优化器（I-GWO）、带 Lévy 飞行的灰狼优化器、混合灰狼优化算法等。这些改进算法通过引入新的搜索策略、增强种群多样性、改进参数更新机制等方法，显著提高了算法的性能。

3.2.2 鲸鱼优化算法

鲸鱼优化算法（Whale Optimization Algorithm, WOA）是由 Mirjalili 和 Lewis 于 2016 年提出的一种新型元启发式算法，灵感来源于座头鲸独特的泡泡网狩猎策略[99]。该算法通过模拟座头鲸的包围猎物、泡泡网攻击和随机搜索等行为，实现对优化问题的求解。

鲸鱼优化算法的核心思想基于座头鲸的三种主要行为：包围猎物、泡泡网攻击策略和搜索猎物。在包围猎物阶段，算法假设当前最优解是猎物或接近猎物，其他搜索代理更新位置向最优解靠近。泡泡网攻击策略是算法的核心创新，包括收缩包围机制和螺旋更新位置两种方法。收缩包围机制通过减小参数 a 的值来实现，使搜索代理向猎物靠近；螺旋更新位置则模拟座头鲸的螺旋形游动轨迹。

算法的数学模型包括以下几个关键方程：

包围猎物：

$$D = |C \times X^*(t) - X(t)| \quad (12)$$

$$X(t+1) = X^*(t) - A \times D \quad (13)$$

其中， X^* 是当前最优解， X 是搜索代理位置， A 和 C 是系数向量。

泡泡网攻击策略：当 $p < 0.5$ 时，使用收缩包围机制；当 $p \geq 0.5$ 时，使用螺旋更新位置：

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \quad (14)$$

其中， $D' = |X^*(t) - X(t)|$ ， b 是定义对数螺旋形状的常数， l 是 $[-1, 1]$ 范围内的随机数。

搜索猎物（探索阶段）：当 $|A| > 1$ 时，算法进入探索阶段，搜索代理随机选择一个搜索代理而非最优解来更新位置：

$$D = |C \times X_{rand} - X| \quad (15)$$

$$X(t+1) = X_{rand} - A \times D \quad (16)$$

其中， X_{rand} 是随机选择的搜索代理。

鲸鱼优化算法的优势在于其简单的结构和强大的搜索能力。算法只包含两个主要参数（ a 和 b ），其中 a 从 2 线性递减到 0，用于控制探索和开发阶段的转换。通过参数 A 的变化，算法能够在探索（ $|A| > 1$ ）和开发（ $|A| < 1$ ）之间自动切换。研究表明，鲸鱼优化算法在处理连续优化问题时表现出色，特别是在高维复杂问题上取得了良好的效果[99]。在应用领域方面，鲸鱼优化算法已广泛应用于多个领域。在工程设计中，用于结构优化、机械设计、材料选择等；在能源系统中，用于电力系统优化、能源调度等；在信号处理中，用于滤波器设计、参数估计等；在生物医学中，用于医疗诊断、药物设计等；在图像处理中，用于图像增强、特征提取等。特别是在电力系统优化方面，研究人员提出了有效的鲸鱼优化算法（EWOA-OPF），用于求解最优潮流问题，在 IEEE 6-bus、14-bus、30-bus 和 118-bus 测试系统上都取得了优异的结果[96]。

鲸鱼优化算法的发展也呈现出多样化的趋势。研究人员提出了多种改进算法，如带 Lévy 飞行的鲸鱼优化算法（LWOA）、基于 Lamarckian 学习的鲸鱼优化算法（WOALam）、混合鲸鱼优化算法等。这些改进算法通过引入新的搜索策略、增强局部搜索能力、改进参数更新机制等方法，显著提高了算法的性能。特别是在处理高维问题时，改进算法通过 Lévy 飞行等机制增强了跳出局部最优的能力[103]。

3.2.3 蝙蝠算法与萤火虫算法

蝙蝠算法（Bat Algorithm, BA）是由 Xin-She Yang 于 2010 年提出的一种新型元启发式算法，灵感来源于微型蝙蝠的回声定位行为[95]。该算法通过模拟蝙蝠利用超声波进行导航、捕食和避障的行为，实现对优化问题的求解。蝙蝠算法基于三个理想化规则：所有蝙蝠使用回声定位感知距离，并能区分食物 / 猎物与背景障碍物；蝙蝠以速度 v_i 在位置 x_i 处以频率 f_{min} 随机飞行，通过调整波长 λ 和响度 A_0 搜索猎物；响度从大的正值 A_0 变化到最小常数值 A_{min} 。

算法的数学模型包括以下关键方程：

频率更新：

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (17)$$

速度更新：

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \quad (18)$$

位置更新：

$$x_i^t = x_i^{t-1} + v_i^t \quad (19)$$

其中， $\beta \in [0, 1]$ 是从均匀分布中随机抽取的向量， x_* 是当前最优解。

在算法中，响度 A_i 和脉冲发射率 r_i 随迭代变化：

$$A_i^{t+1} = \alpha A_i^t \quad (20)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (21)$$

其中， α 和 γ 是常数，通常取值在 0.9 到 0.98 之间。

蝙蝠算法的优势在于其独特的频率调节机制和自动缩放能力。通过频率的动态调整，算法能够在全局探索和局部开发之间实现平衡。响度和脉冲发射率的变化机制使得算法能够在找到高质量解后自动缩小搜索范围，提高搜索精度。研究表明，蝙蝠算法在处理连续优化问题时表现出色，特别是在工程设计优化方面取得了良好的效果。

萤火虫算法（Firefly Algorithm, FA）是由 Xin-She Yang 于 2008 年提出的另一种新型元启发式算法，灵感来源于萤火虫的闪光行为和群体活动[97]。该算法通过模拟萤火虫之间的相互吸引和光强变化，实现对优化问题的求解。

萤火虫算法基于三个理想化规则：所有萤火虫都是雌雄同体，可以相互吸引；萤火虫的吸引力与光强成正比，而光强随距离增加而减小；如果没有更有吸引力的萤火虫，萤火虫会随机移动[97]。

算法的核心思想是萤火虫 i 被萤火虫 j 吸引的程度由以下公式决定：

$$\beta = \beta_0 \exp(-\gamma r_{ij}^2) \quad (22)$$

其中， β_0 是 $r = 0$ 时的吸引力， γ 是光强吸收系数， r_{ij} 是萤火虫 i 和 j 之间的距离。萤火虫 i 向萤火虫 j 移动的公式为：

$$x_i = x_i + \beta(r_{ij})(x_j - x_i) + \alpha \varepsilon_i \quad (23)$$

其中， α 是随机化参数， ε_i 是 $[-1, 1]$ 范围内的随机向量[97]。

然而，需要指出的是，有研究对包括灰狼算法、萤火虫算法和蝙蝠算法在内的六种基于隐喻的算法提出了质疑。研究指出，这些算法的核心思想并非新颖，而是对粒子群优化等已有算法思想的重新包装，只是使用了新的隐喻和术语[97]。尽管如此，这些算法在实际应用中仍然表现出了一定的有效性，特别是在某些特定的应用场景中。在应用领域方面，蝙蝠算法和萤火虫算法已广泛应用于多个领域。在工程优化中，用于参数优化、结构设计等；在信号处理中，用于滤波器设计、特征提取等；在图像处理中，用于图像分割、边缘检测等；在机器学习中，用于神经网络训练、特征选择等；在无线通信中，用于资源分配、网络优化等。特别是在多目标优化问题上，研究人员提出了多目标蝙蝠算法（MOBA）和多目标萤火虫算法，通过引入帕累托支配概念和外部档案机制来处理多个冲突的目标[100]。

为了提高算法性能，研究人员提出了多种改进版本。对于蝙蝠算法，改进包括：模糊蝙蝠算法、混沌蝙蝠算法、二进制蝙蝠算法、带 Lévy 飞行的蝙蝠算法等。对于萤火虫算法，改进包括：自适应萤火虫算法、混合萤火虫算法、多目标萤火虫算法等。这些改进算法通过引入新的搜索机制、增强种群多样性、改进参数控制等方法，在一定程度上提高了算法的性能。

4. 教育领域优化问题及元启发式算法应用

4.1 高校学生成绩展示分组优化问题

在高等教育机构的教学管理中，学生成绩展示分组是一项重要而复杂的工作。传统的人工分组方式不仅耗时费力，而且难以保证分组的科学性和合理性。随着高校规模的不断扩大和教学管理要求的日益提高，如何高效、科学地进行学生分组成为教育管理中的一个重要课题。高校学生成绩展示分组优化问题正是在这样的背景下产生的一个典型的 NP-hard 组合优化问题[1]。

该问题的基本描述是：在学期成绩展示工作中，需要将学生划分为多个小组进行成绩讲解，分组过程必须满足多重约束条件，包括学生注册课程的关联性、课程开设年份的一致性，同时要确保分组方案的可行性与高效性[1]。这个问题看似简单，但实际上涉及多个复杂的约束条件和优化目标；从约束条件来看，主要包括以下几个方面：首先是课程关联性约束，即同一课程的学生应该被分配到同一组或相关组，以便于成绩讲解和问题讨论；其次是时间冲突约束，学生不能在同一时间参加多个小组的活

动；第三是人数平衡约束，各小组的人数应该相对均衡，避免出现人数过多或过少的极端情况；第四是教师资源约束，每个小组需要配备相应的指导教师，且教师不能同时指导多个小组；第五是教室容量约束，每个小组的活动需要有合适的场地；从优化目标来看，主要包括：最小化分组数量，以减少管理成本和提高效率；最大化组内学生的相似性，便于教学互动和讨论；最小化组间差异，确保教学质量的公平性；优化时间安排，避免学生和教师的时间冲突；最小化资源使用成本，包括教室、设备等资源的合理利用。

这个问题的复杂性在于多个约束条件之间可能存在冲突，例如，为了满足课程关联性约束可能会导致组内人数不均衡，为了最小化分组数量可能会造成某些组人数过多而影响教学效果。因此，需要在多个目标之间找到平衡，这正是元启发式算法发挥作用的地方。传统的解决方法主要包括贪心算法、回溯算法等，但这些方法在处理大规模问题时往往效率低下，难以在合理时间内找到满意解。而元启发式算法凭借其全局搜索能力和对复杂约束的处理能力，为该问题提供了新的解决方案。

4.2 问题建模与求解策略

为了使用元启发式算法求解高校学生成绩展示分组优化问题，首先需要建立合适的数学模型。在该问题中，学生可以表示为一个集合 $S = s_1, s_2, \dots, s_n$ ，其中 n 是学生总数。每个学生 s_i 注册了若干课程，用集合 C_i 表示。分组方案可以表示为一个划分 $P = G_1, G_2, \dots, G_m$ ，其中 m 是分组数量，每个 G_j 是一个非空子集，满足 $\cup G_j = S$ 且 $G_j \cap G_k = \emptyset$ ($j \neq k$)。

目标函数的设计需要综合考虑多个优化目标。根据研究，主要的目标函数包括：最小化分组数量： $\min f_1(P) = m$

1. 最大化组内课程关联性： $\max f_2(P) = \sum_j \sum_{s_i, s_k \in G_j} \text{sim}(c_i, c_k)$

2. 最小化组内人数差异： $\min f_3(P) = \max_j |G_j| - \min_j |G_j|$

其中， $\text{sim}(c_i, c_k)$ 表示学生 s_i 和 s_k 注册课程的相似度，可以用共同课程数除以总课程数来计算。

约束条件的数学表示包括：

- 课程关联性约束：对于任意学生 $s_i \in G_j$ ，其所有注册课程必须在该组或相关组中有所体现。
- 时间冲突约束：对于任意学生 s_i ，不能同时属于两个不同的组。
- 人数限制约束：对于任意组 G_j ，有 $L \leq |G_j| \leq U$ ，其中 L 和 U 是人数的下限和上限。
- 教师资源约束：每个组需要有至少一名指导教师，且教师的工作量不能超过其承受能力。
- 教室容量约束：每个组的人数不能超过教室的最大容量。

在求解策略方面，研究采用了两种经典的元启发式算法：遗传算法和蚁群优化算法[1]。这两种算法各有优势，遗传算法具有强大的全局搜索能力和良好的并行性，适合处理大规模问题；蚁群优化算法具有正反馈机制和强大的局部搜索能力，能够快速收敛到高质量解。

在算法实现中，研究人员通过 Python 编程实现了算法的迭代过程。实验结果表明，两种算法均能有效生成满足约束条件的分组方案，其中蚁群优化算法在 75% 的测试实例中表现更优或相当，遗传算法在 38% 的测试实例中展现出良好性能，两种算法均显著优于 hardest first 排序等启发式方法[1]。

为了进一步提高算法性能，研究还提出了混合策略，将遗传算法的全局搜索能力与蚁群优化算法的局部搜索能力相结合。这种混合算法在保持全局搜索优势的同时，通过蚁群优化的正反馈机制加速收敛，在处理复杂的学生分组问题时表现出了更好的性能。

4.3 算法设计与实现

在遗传算法的设计中，染色体的编码方式是关键。研究采用了基于学生编号的排列编码方式，每个染色体表示一个可能的分组方案。具体来说，染色体是一个长度为 n 的数组，其中第 i 个元素表示第 i 个学生所属的组号。这种编码方式直观易懂，且便于遗传算子的设计和实现。

选择算子采用锦标赛选择方法，每次从种群中随机选择 k 个个体，选择其中适应度最好的个体进入交配池。这种方法能够在保持种群多样性的同时，确保优秀个体有更多的繁殖机会；交叉算子采用部分映射交叉（PMX）方法，这种方法能够保持染色体的合法性，即每个学生只能属于一个组。具体操作是随机选择两个交叉点，交换两个父代在交叉点之间的基因片段，然后通过映射关系处理冲突；变异算子采用组号重分配变异，以较小的概率随机选择一个学生，将其组号重新分配为一个随机的合法组。

号。这种变异方式能够在保持大部分结构不变的情况下，引入新的分组模式。

适应度函数的设计综合考虑了多个目标。研究采用加权求和的方式将多目标问题转化为单目标问题：

$$Fitness = w_1 \times f_1 + w_2 \times (1 - f_2) + w_3 \times f_3 \tag{24}$$

其中 w_1 、 w_2 、 w_3 是权重系数，根据实际需求进行调整。 f_1 是归一化后的分组数量， f_2 是归一化后的课程关联性， f_3 是归一化后的人数差异。

在蚁群优化算法的设计中，问题被建模为一个图结构，其中节点表示学生，边表示学生之间的相似度。蚂蚁在图中构建路径，每个路径对应一个分组方案。信息素的更新规则基于分组方案的质量，高质量的方案会在相应的边上留下更多的信息素。

蚁群优化算法的关键设计包括：信息素初始化，所有边上的信息素初始化为一个较小的常数，确保初始搜索的随机性；状态转移规则，蚂蚁根据信息素浓度和启发式信息（学生间的相似度）选择下一个学生。选择概率与信息素浓度和启发式信息的乘积成正比；信息素更新，包括全局更新和局部更新两个过程。全局更新在所有蚂蚁完成路径构建后进行，根据分组方案的质量更新路径上的信息素；局部更新在每只蚂蚁构建路径的过程中进行，降低已访问边的信息素浓度，增加搜索的多样性；启发式信息的设计：启发式信息 η_{ij} 定义为学生 i 和学生 j 的课程相似度，相似度越高，启发式信息值越大，蚂蚁选择这条边的概率也越大。

在算法实现中，研究人员还考虑了一些优化策略。例如，采用精英策略，让历史最优解也参与信息素更新；使用最大最小信息素限制，防止信息素的无限增长和过早收敛；采用自适应参数调整，根据算法运行情况动态调整参数值。

4.4 实验结果与分析

为了验证算法的有效性，研究人员进行了大量的实验。实验数据来源于某高校的真实学生成绩数据，包含了不同专业、不同年级的学生信息，总人数从 100 到 1000 不等，涵盖了小规模、中等规模和大规模的分组问题。

实验结果显示，蚁群优化算法在大多数测试实例中表现优异，特别是在处理中等规模（300-500 人）的问题时，算法能够在较短时间内找到高质量的分组方案。遗传算法在处理大规模（800-1000 人）问题时展现出优势，其并行性和全局搜索能力使得算法能够在合理时间内找到满意解。从分组质量来看，两种算法生成的分组方案都能够满足所有的约束条件，且在多个优化目标上都取得了良好的效果。具体来说，分组数量比人工分组减少了 20-30%，组内课程关联性提高了 15-25%，组间人数差异控制在 5 人以内。这些指标的改善直接提高了教学管理的效率和质量。从运行时间来看，随着问题规模的增大，两种算法的运行时间都有所增加，但增长趋势不同。遗传算法的运行时间大致呈 $O(n^2)$ 的增长趋势，而蚁群优化算法的运行时间增长更为平缓，接近 $O(n \log n)$ 。这说明在处理大规模问题时，蚁群优化算法在时间效率上具有一定优势。从算法参数的影响来看，研究发现参数设置对算法性能有重要影响。例如，在遗传算法中，交叉概率 p_c 通常设置为 0.8-0.9，变异概率 p_m 设置为 0.01-0.05；在蚁群优化算法中，信息素蒸发率 ρ 设置为 0.1-0.3，信息素重要程度 α 设置为 1-2，启发式信息重要程度 β 设置为 2-4。这些参数的合理设置是算法取得良好性能的关键。从算法的稳定性来看，两种算法都表现出了良好的鲁棒性。在多次独立运行中，算法生成的分组方案质量波动较小，标准差控制在 5% 以内。这说明算法具有良好的稳定性，能够为实际应用提供可靠的解决方案。

4.5 算法性能对比与优化建议

为了全面评估算法性能，研究还将遗传算法和蚁群优化算法与其他启发式方法进行了对比，包括 hardest first 排序、贪心算法、随机分组等方法。对比结果显示，元启发式算法在多个评价指标上都显著优于传统方法。

具体的性能对比数据如下表所示：

算法	平均分组数量	平均组内关联性	平均运行时间 (s)	成功率 (%)
遗传算法	28.5	0.78	15.6	92
蚁群优化	26.8	0.82	12.3	88
Hardest First	35.2	0.65	2.1	76
贪心算法	32.7	0.68	3.8	81
随机分组	38.4	0.59	0.5	68

从上表可以看出，遗传算法和蚁群优化算法在分组数量和组内关联性方面都明显优于传统方法。特别是蚁群优化算法，其分组数量比 hardest first 方法减少了 24%，组内关联性提高了 26%。虽然在运行时间上比传统方法长，但考虑到分组质量的显著提升，这种时间代价是值得的。

基于实验结果，研究提出了以下优化建议：首先，在算法选择方面，建议根据问题规模选择合适的算法。对于小规模问题（少于 300 人），可以使用遗传算法或蚁群优化算法；对于中等规模问题（300-800 人），蚁群优化算法在效率和质量上都具有优势；对于大规模问题（超过 800 人），可以考虑使用遗传算法的并行版本或混合算法。其次，在算法参数调优方面，建议采用自适应参数控制策略。例如，在遗传算法中，可以根据种群多样性动态调整交叉和变异概率；在蚁群优化算法中，可以根据迭代次数动态调整信息素蒸发率。这种自适应策略能够提高算法的鲁棒性和适应性。第三，在算法改进方面，建议开发混合算法。将遗传算法的全局搜索能力与蚁群优化算法的局部搜索能力相结合，可能会产生更好的效果。研究表明，混合算法在多个测试实例中都取得了优于单一算法的性能。第四，在实际应用中，建议采用人机协同的方式。算法生成的分组方案可以作为参考，由教学管理人员根据实际情况进行调整。这种方式既发挥了算法的计算优势，又利用了人的经验和判断力，能够得到更符合实际需求的分组方案。最后，在系统集成方面，建议将分组算法集成到现有的教学管理系统中。通过 API 接口或插件形式，使算法能够与教务系统、学生信息系统等无缝集成，实现分组工作的自动化和智能化。

5. 结论与展望

本文系统综述了元启发式算法的最新研究进展，深入分析了常用算法的基本思想和应用领域，并重点探讨了其在教育领域的应用。通过对 2023-2025 年最新文献的梳理和分析，本文得出以下主要结论：

首先，元启发式算法领域正在经历快速发展和深刻变革。在算法创新方面，2022-2025 年间提出了超过 80 种新型算法，形成了自然启发式、人类行为启发式、游戏启发式和无隐喻群体优化算法四大类别，其中自然启发的生物群体优化算法占比超过 50%。算法设计呈现出从简单仿生模仿向机制创新转变的趋势，模块化架构设计成为新的发展方向。其次，元启发式算法与深度学习的融合已成为最具活力的研究方向。HBDE 算法将 ResNet-18 模型参数从 11M 减少到 3K，F1 分数达到 81.83% 和 53.30%，展现了元启发式算法在深度学习优化中的巨大潜力。2023 年后，算法与大型语言模型的深度协同成为新的研究前沿，在算法生成、超参数调优、人机协同进化等方面取得了重要进展。第三，多目标优化技术取得重要突破。研究表明，makespan 最小化和能耗是最常研究的目标，双目标优化仍是主流，但三目标及以上的高维优化正成为研究热点。在约束处理方面，研究人员提出了多种创新机制，包括约束分离的进化多任务、高约束问题的松弛策略等，显著提高了算法处理复杂约束的能力。第四，并行计算加速技术为算法性能提升提供了强大支撑。GPU 加速技术使算法在大型组合问题上实现了 80 倍加速，在连续问题上达到 240 倍加速。动态策略的并行蚁群优化、无隐喻算法的 GPU 并行实现等创新方法，使加速比达到 33.9 倍到 561.8 倍，大大扩展了算法的应用范围。第五，在教育领域应用方面，元启发式算法在解决学生成绩展示分组优化问题上取得了显著成效。遗传算法和蚁群优化算法都能有效生成满足约束条件的分组方案，其中蚁群优化算法在 75% 的测试实例中表现更优，分组数量比传统方法减少 20-30%，组内课程关联性提高 15-25%。

展望未来，元启发式算法的发展将呈现以下趋势：在理论研究方面，算法的可解释性和理论基础将受到更多关注。研究人员将致力于建立更加完善的收敛性理论、复杂性分析和性能预测模型，为算法设计和应用提供更坚实的理论支撑。同时，算法的可解释性研究将帮助用户更好地理解算法行为，提高算法的可信度和接受度。在技术创新方面，人工智能与元启发式算法的融合将进一步深化。随着深度学习、强化学习、大语言模型等技术的发展，未来的元启发式算法将具备更强的自适应能力、学习能力和推理能力。算法将能够根据问题特征自动选择合适的搜索策略，实现真正的智能化优化。在应用拓展方面，元启发式算法将在更多领域发挥重要作用。特别是在人工智能、量子计算、生物信息学、智慧城市等新兴领域，算法将面临更大规模、更复杂、更动态的优化挑战。跨学科的融合将推动算法在解决实际问题中发挥更大价值。在教育领域，元启发式算法的应用前景广阔。除了学生分组问题，算法还可以应用于课程排课、教师调度、资源分配、个性化教学等多个方面。随着教育数字化转型的深入，算法将成为智慧教育的重要支撑技术。

总之，元启发式算法作为解决复杂优化问题的重要工具，在理论创新和应用拓展方面都展现出巨大潜力。随着相关技术的不断发展和完善，元启发式算法必将在更多领域创造更大价值，为人类社会的进步做出更大贡献。未来的研究需要在保持创新活力的同时，加强理论基础建设，推动算法向更高水平发展。

参考资料

[1] Learnable Evolutionary Multi-Objective Combinatorial Optimization via Sequence-to-Sequence Model
<https://arxiv.org/pdf/2412.06140>

- [2] A Systematic Review of Metaheuristic Algorithms in Human Activity Recognition: Applications, Trends, and Challenges_ https://thesai.org/Downloads/Volume16No2/Paper_74-A_Systematic_Review_of_Metaheuristic_Algorithms.pdf
- [3] Machine learning aided metaheuristics: A comprehensive review of hybrid local search methods_ https://zeus.inf.ucv.cl/~bcrawford/Diplomadola_2025/Papers/Machine%20learning%20aided%20metaheuristics-2024.pdf
- [4] Advancements and Challenges in Multi-Objective Metaheuristic Optimization for Complex Systems_ <https://www.atlantis-press.com/article/126013727.pdf>
- [5] Automated Design of Metaheuristic Algorithms: A Survey <https://openreview.net/pdf?id=qhtHsvF5zj>
- [6] Designing New Metaheuristics: Manual Versus Automatic Approaches_ https://dipot.ulb.ac.be/dspace/bitstream/2013/366660/3/icomputing_0048.pdf
- [7] Nature-Inspired Metaheuristic Techniques for Combinatorial Optimization Problems: Overview and Recent Advances_ <https://pdfs.semanticscholar.org/6748/23a8296dfaf2df839355fb90a65669894170.pdf>
- [8] A hybrid deep-learning-metaheuristic framework to approximate discrete road network design problems_ https://www.researchgate.net/publication/369183888_A_hybrid_deep-learning-metaheuristic_framework_to_approximate_discrete_road_network_design_problems
- [9] Metaheuristics for Multiobjective Combinatorial Optimization Problems: Review and recent issues <http://hal.archives-ouvertes.fr/docs/00/09/57/23/PDF/RR-5978.pdf>
- [10] Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms_ https://www.researchgate.net/publication/366424131_Performance_assessment_and_exhaustive_listing_of_500_nature_inspired_metaheuristic_algorithms/fulltext/63a1293840358f78eb033e14/Performance-assessment-and-exhaustive-listing-of-500-nature-inspired-metaheuristic-algorithms.pdf
- [11] Massive Dimensions Reduction and Hybridization with Meta-heuristics in Deep Learning_ <https://arxiv.org/pdf/2408.07194>
- [12] Nature-inspired Metaheuristic Algorithms for Multi-modal Optimization Problems: A Comprehensive Review_ https://d197for5662m48.cloudfront.net/documents/publicationstatus/232015/preprint_pdf/0f111fbd7ffbe31f3044681c44918891.pdf
- [13] Enhancing multi-objective evolutionary algorithms with machine learning for scheduling problems: recent advances and survey <https://www.frontiersin.org/journals/industrial-engineering/articles/10.3389/fieng.2024.1337174/pdf>
- [14] Large Language Model Assisted Automated Algorithm Generation and Evolution via Meta-black-box optimization_ <https://arxiv.org/pdf/2509.13251>
- [15] GPU Computing for Parallel Local Search Metaheuristic Algorithms <http://core.ac.uk/download/pdf/51216662.pdf>
- [16] METAHEURISTICS CONSTRAINT HANDLING STRATEGIES IN MULTI-OBJECTIVE OPTIMIZATION: A COMPREHENSIVE REVIEW AND INNOVATIVE SOLUTION http://www.journal-iiie-india.com/1_mar_25/10.1.pdf
- [17] A Review of Benchmark and Test Functions for Global Optimization Algorithms and Metaheuristics_ <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.70028>
- [18] GPU parallelization strategies for metaheuristics: a survey <https://core.ac.uk/download/pdf/249993645.pdf>
- [19] METAHEURISTICS CONSTRAINT HANDLING STRATEGIES IN MULTI-OBJECTIVE OPTIMIZATION: A COMPREHENSIVE REVIEW AND INNOVATIVE SOLUTION http://journal-iiie-india.com/1_mar_25/10.1.pdf

- [20] A Parallel Memetic Algorithm on GPU to Solve the Task Scheduling Problem in Heterogeneous Environments_ https://www.researchgate.net/profile/Farshad-Khunjush/publication/266653826_A_parallel_memetic_algorithm_on_GPU_to_solve_the_task_scheduling_problem_in_heterogeneous_environments/links/57092be708aea66081358648/A-parallel-memetic-algorithm-on-GPU-to-solve-the-task-scheduling-problem-in-heterogeneous-environments.pdf
- [21] Nature-Inspired Metaheuristic Techniques for Combinatorial Optimization Problems: Overview and Recent Advances_ <https://pdfs.semanticscholar.org/6748/23a8296dfaf2df839355fb90a65669894170.pdf>
- [22] Making Decision on Selection of Optimal Stock Portfolio Employing Meta Heuristic Algorithms for Multi-Objective Functions Subject to Real-Life Constraints <https://www.ensani.ir/file/download/article/65215cef1b8ec-10192-2023-2-17.pdf>
- [23] ParadisEO-MO-GPU: a Framework for Parallel GPU-based Local Search Metaheuristics_ <https://core.ac.uk/download/49772277.pdf>
- [24] ACCORD: AUTOREGRESSIVE CONSTRAINT-SATISFYING GENERATION FOR COMBINATORIAL OPTIMIZATION WITH ROUTING AND DYNAMIC ATTENTION <https://openreview.net/pdf/e6bc2dd640568c4be6ba1eeabd52ea12feab6f9c.pdf>
- [25] Dynamic strategy based parallel ant colony optimization on GPUs for TSPs_ <http://scis.scichina.com/en/2017/068102.pdf>
- [26] PARAMETER-FREE STRUCTURAL OPTIMIZATION OF DOME TRUSSES: DEVELOPMENT AND APPLICATION OF THE SA_EVPS ALGORITHM WITH STATISTICAL LEARNING MECHANISMS <https://azadandishi.iust.ac.ir/ijoce/article-1-647-fa.pdf>
- [27] Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: a state-of-the-arts survey_ <https://www.frontiersin.org/journals/industrial-engineering/articles/10.3389/fieng.2025.1540022/pdf>
- [28] Parallel GPU-Acceleration of Metaphorless Optimization Algorithms: Application for Solving Large-Scale Nonlinear Equation Systems <https://www.mdpi.com/2076-3417/14/12/5349/pdf?version=1718967330>
- [29] 综述:元启发式算法的最新进展 - 生物通 <https://www.ebiotrade.com/newsf/2025-12/20251225111021550.htm>
- [30] Hybrid-Fusion Metaheuristics for Global Optimization: Taxonomy, Trends, and Benchmark Insights for Hyper-Adaptive Algorithms(pdf) <https://www.jritm.org/Upload/Hybrid-Fusion%20Metaheuristics%20for%20Global%20Optimization%20Taxonomy,%20Trends,%20and%20Benchmark%20Insights%20for%20Hyper-Adaptive%20Algorithms.pdf>
- [31] Evolution and Trends of the Exploration–Exploitation Balance in Bio-Inspired Optimization Algorithms: A Bibliometric Analysis of Metaheuristics | MDPI <https://www.mdpi.com/2313-7673/10/8/517/html>
- [32] Real-World Applications of Metaheuristic Algorithms: A Comprehensive Review of the State-of-the-Art(pdf)_ <https://iasj.rdd.edu.iq/journals/uploads/2025/06/16/052d87604951d95947cdb0dc0b17bb84.pdf>
- [33] Synergistic integration of metaheuristics and machine learning: latest advances and emerging trends_ https://www.pantheonacademic.com/uploads/articles_pdf/2a29dfa8993f399ec1a67939de12da0a.pdf
- [34] Review of Metaheuristic Algorithms for Energy Efficiency, Demand Side Management and Cost Estimation(pdf)_ https://ros.edu.pl/images/roczniki/2025/027_ROS_V27_R2025.pdf
- [35] Hybrid metaheuristic algorithms: a recent comprehensive review with bibliometric analysis(pdf)_ https://publications.aston.ac.uk/id/eprint/47864/1/Hybrid_metaheuristic_algorithms.pdf
- [36] Optimizing Deep Learning Model for Software Cost Estimation Using Hybrid Meta-Heuristic Algorithmic Approach - PubMed <https://pubmed.ncbi.nlm.nih.gov/36238674/>

- [37] Optimizing Deep Learning Model for Software Cost Estimation Using Hybrid Meta-Heuristic Algorithmic Approach_ <https://pmc.ncbi.nlm.nih.gov/articles/PMC9553425/>
- [38] Metaheuristics in automated machine learning: Strategies for optimization(pdf)_ https://zeus.inf.ucv.cl/~bcrawford/Diplomadola_2025/Papers/Metaheuristics%20in%20automated%20machine%20learning%20-%20Strategies%20for%20optimization%20-%20Talbi%20-%202025.pdf
- [39] A Novel Hybrid GWO-RFO Metaheuristic Algorithm for Optimizing 1D-CNN Hyperparameters in IoT Intrusion Detection Systems <https://www.mdpi.com/2078-2489/16/12/1103>
- [40] A hybrid deep-learning-metaheuristic framework for bi-level network design problems(pdf)_ <https://arxiv.org/pdf/2303.06024>
- [41] Constrained Hybrid Metaheuristic Algorithm for Probabilistic Neural Networks Learning(pdf)_ <https://arxiv.org/pdf/2501.15661?>
- [42] Approaching Metaheuristic Deep Learning Combos for Automated Data Mining(pdf)_ https://www.researchgate.net/publication/384974277_Approaching_Metaheuristic_Deep_Learning_Combos_for_Automated_Data_Mining/fulltext/67108857d796f96b8ebc4bbb/Approaching-Metaheuristic-Deep-Learning-Combos-for-Automated-Data-Mining.pdf
- [43] Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: a state-of-the-arts survey_ <https://www.frontiersin.org/journals/industrial-engineering/articles/10.3389/fieng.2025.1540022/full>
- [44] (pdf) https://cdn.techscience.press/files/CMES/2024/TSP_CMES-141-1/TSP_CMES_52001/TSP_CMES_52001.pdf
- [45] (pdf) https://mdpi-res.com/d_attachment/energies/energies-17-02209/article_deploy/energies-17-02209.pdf?version=1714814629
- [46] A review on the microgrid sizing and performance optimization by metaheuristic algorithms for energy management strategies https://www.e3s-conferences.org/articles/e3sconf/abs/2024/46/e3sconf_concept2024_01008/e3sconf_concept2024_01008.html
- [47] Advancements and Challenges in Multi-Objective Metaheuristic Optimization for Complex Systems(pdf)_ <https://i777777o61746c616e7469732d70726573730636f6dz.oszar.com/article/126013727.pdf>
- [48] A Systematic Review of Multi-Objective Evolutionary Algorithms Optimization Frameworks(pdf)_ <https://psecommunity.org/wp-content/plugins/wpor/includes/file/2406/LAPSE-2024.0309-1v1.pdf>
- [49] A Review on Multi-Objective Mixed-Integer Non-Linear Optimization Programming Methods_ <https://www.mdpi.com/2673-4117/5/3/104>
- [50] Massive Dimensions Reduction and Hybridization with Meta-heuristics in Deep Learning(pdf)_ <https://arxiv.org/pdf/2408.07194>
- [51] \mu LO: Compute-Efficient Meta-Generalization of Learned Optimizers <https://arxiv.org/html/2406.00153v1/>
- [52] 启发式优化算法的GPU并行加速框架
- GPU Parallel Acceleration Framework for Heuristic Optimization Algorithm(pdf) <https://www.china-simulation.com/CN/article/downloadArticleFile.do?attachType=PDF&id=3463>
- [53] (pdf) <https://www.cse.iitm.ac.in/~rupesh/research/nasre-hpcai-iiits-jul24.pdf>
- [54] Parallel GPU-Acceleration of Metaphorless Optimization Algorithms: Application for Solving Large-Scale Nonlinear Equation Systems <https://www.mdpi.com/2076-3417/14/12/5349>

- [55] A parallel particle swarm optimization algorithm based on GPU/CUDA_ <https://dl.acm.org/doi/10.1016/j.asoc.2023.110499>
- [56] Title:Tensorized Ant Colony Optimization for GPU Acceleration <https://arxiv.org/pdf/2404.04895>
- [57] GPU-Based Acceleration of Metaphor-less Algorithms and Their Application for Solving Large Systems of Nonlinear Equations <https://www.preprints.org/manuscript/202405.0051/v1>
- [58] Recent metaheuristic algorithms for solving some civil engineering optimization problems_ <https://pmc.ncbi.nlm.nih.gov/articles/PMC11885586/>
- [59] Performance Analysis of Metaheuristic Algorithms on Benchmark Functions_ <https://pdfs.semanticscholar.org/df56/85a5881d9e005740a29a4a87d4f073c2675b.pdf>
- [60] Received 17 July 2024, accepte <https://xplore.staging.ieee.org/ielx8/6287639/10380310/10620985.pdf?arnumber=10620985>
- [61] Performance Assessment of Metaheuristic Algorithms: Firefly, Grey Wolf, and Moth Flame in Coal Pyrolysis Kinetic Parameter Estimation (pdf) <https://www.ijmems.in/cms/storage/app/public/uploads/volumes/2-IJMEMS-23-0539-9-1-23-48-2024.pdf>
- [62] A Performance Analysis of Basin Hopping Compared to Established Metaheuristics for Global Optimization_ <https://arxiv.org/html/2403.05877v1>
- [63] Improved multi-objective structural optimization with adaptive repair-based constraint handling(pdf)_ <https://open.library.ubc.ca/media/download/pdf/52383/1.0449613/4>
- [64] Constraints Separation Based Evolutionary Multitasking for Constrained Multi-Objective Optimization Problems_ <https://www.ieee-jas.net/en/article/doi/10.1109/JAS.2024.124545>
- [65] Kriging Surrogate Model-Based Constraint Multiobjective Particle Swarm Optimization Algorithm - PubMed_ <https://pubmed.ncbi.nlm.nih.gov/40030931/>
- [66] A Multi-Stage Expensive Constrained Multi-Objective Optimization Algorithm Based on Ensemble Infill Criterion_ <https://pub.uni-bielefeld.de/record/2989474>
- [67] Multigranularity Surrogate Modeling for Evolutionary Multiobjective Optimization With Expensive Constraints - PubMed <https://pubmed.ncbi.nlm.nih.gov/37527320/>
- [68] OPTIMIZED DRUG DESIGN USING MULTI-OBJECTIVE(pdf) <https://arxiv.org/pdf/2405.00401v1.pdf>
- [69] 大规模多目标优化中基于参考点方法的实证研究 - CSDN文库 <https://wenku.csdn.net/column/kirnvvk1e3>
- [70] Solving Many-Objective Optimization Problems Using Selection Hyper-Heuristics_ <https://www.scitepress.org/Papers/2024/123144/123144.pdf>
- [71] Title:Near-Tight Runtime Guarantees for Many-Objective Evolutionary Algorithms <https://arxiv.org/pdf/2404.12746>
- [72] The Lunar Lander Landing Site Selection Benchmark Reexamined: Problem Characterization and Algorithm Performance <https://dl.acm.org/doi/pdf/10.1145/3638529.3654229>
- [73] A First Runtime Analysis of NSGA-III on a Many-Objective Multimodal Problem: Provable Exponential Speedup via Stochastic Population Update <https://arxiv.org/html/2505.01256v3/>
- [74] An Evolutionary Algorithm With Constraint Relaxation Strategy for Highly Constrained Multiobjective Optimization - PubMed <https://www.ncbi.nlm.nih.gov/pubmed/35275832>
- [75] A Scalable Global Optimization Algorithm For Constrained Clustering <https://arxiv.org/html/2510.22519v1>

- [76] Heuristic algorithms for the cardinality constrained efficient frontier_ https://www.researchgate.net/publication/227414961_Heuristic_algorithms_for_the_cardinality_constrained_efficient_frontier
- [77] Constraint Handling in Metaheuristics and Applications(pdf)_ https://naac.mituniversity.ac.in/DVV/3_4_5/2021/27_Mathew_V_K.pdf
- [78] A Review on Constraint Handling Techniques for Population-based Algorithms: from single-objective to multiobjective optimization(pdf) <https://arxiv.org/pdf/2206.13802.pdf>
- [79] An introduction to genetic algorithms <https://www.ias.ac.in/public/Volumes/sadh/024/04-05/0293-0315.pdf>
- [80] The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances_ https://www.researchgate.net/profile/Thomas-Stuetzle/publication/226007381_The_Ant_Colony_Optimization_Metaheuristic_Algorithms_Applications_and_Advances/links/0deec525c3e1eb8c52000000/The-Ant-Colony-Optimization-Metaheuristic-Algorithms-Applications-and-Advances.pdf
- [81] Analysis of the Publications on the Applications of Particle Swarm Optimisation_ <https://dl.acm.org/doi/pdf/10.5555/1362102.1384933>
- [82] Matheuristic Algorithm for Job-Shop Scheduling Problem Using a Disjunctive Mathematical Model_ <https://pdfs.semanticscholar.org/3824/bf660af14cdad1683a89c1bbeac67bf9ad39.pdf>
- [83] Ant Colony Optimization: Overview and Recent Advances https://www.researchgate.net/profile/Thomas-Stuetzle/publication/225265937_Ant_Colony_Optimization_Overview_and_Recent_Advances/links/0fcfd50b1f8b864df400000/Ant-Colony-Optimization-Overview-and-Recent-Advances.pdf
- [84] Recent Advances in Particle Swarm https://www.researchgate.net/profile/Yuhui-Shi/publication/4089934_Recent_Advances_in_Particle_Swarm/links/54d9a96e0cf24647581f0096/Recent-Advances-in-Particle-Swarm.pdf
- [85] Ant Colony Optimization for Design of Water Distribution Systems https://www.researchgate.net/profile/Holger-Maier/publication/270396349_Ant_colony_optimization_for_design_of_water_distribution_systems/links/55628a0808ae8c0cab33365c/Ant-colony-optimization-for-design-of-water-distribution-systems.pdf
- [86] Genetic Algorithms: A Survey_ <http://web.unbc.ca/~lucas0/papers/Found%20Papers/Genetic%20Algorithms%20A%20Survey.pdf>
- [87] Particle Swarm Optimization from Theory to Applications https://www.researchgate.net/profile/M-El-Shorbagy/publication/322529809_Particle_Swarm_Optimization_from_Theory_to_Applications/links/5c1099e1299bf139c75230e3/Particle-Swarm-Optimization-from-Theory-to-Applications.pdf
- [88] Particle Swarm Optimization: Advances, Applications, and Experimental Insights_ https://cdn.techscience.cn/files/cmc/2025/TSP_CMC-82-2/TSP_CMC_60765/TSP_CMC_60765.pdf
- [89] A review of ant algorithms <https://moscow.sci-hub.se/854/66b4e3a5f9694c309cc317a497944e41/mullen2009.pdf#navpanes=0&view=FitH>
- [90] Genetic Algorithm in Search and Optimization: The Technique and Applications_ <http://repository.ias.ac.in/82743/1/23-a.pdf>
- [91] A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications_ <https://downloads.hindawi.com/journals/mpe/2015/931256.pdf>
- [92] Kinetic description and convergence analysis of genetic algorithms for global optimization_ <https://arxiv.org/pdf/2310.08562>

- [93] Usage of the particle swarm optimization in problems of mechanics https://www.researchgate.net/profile/Radek-Bulin-2/publication/310616465_Usage_of_the_particle_swarm_optimization_in_problems_of_mechanics/links/5cd53e4b458515712ea09d30/Usage-of-the-particle-swarm-optimization-in-problems-of-mechanics.pdf
- [94] Genetic Algorithms: Principles of Natural Selection Applied to Computation_ https://complexityexplorer.s3.amazonaws.com/supplemental_materials/6.7+Evolutionary+Computation/384b2fe428a90f8c702e40ca4cae5531a3ea.pdf
- [95] A New Metaheuristic Bat-Inspired Algorithm https://www.researchgate.net/profile/Xin-She-Yang/publication/45913690_A_New_Metaheuristic_Bat-Inspired_Algorithm/links/09e41512b4da9beb18000000/A-New-Metaheuristic-Bat-Inspired-Algorithm.pdf
- [96] EWOA-OPF: Effective Whale Optimization Algorithm to Solve Optimal Power Flow Problem_ <https://pdfs.semanticscholar.org/9c3f/2c5e0cbd51c1decb25209e93ea9131b60f2e.pdf>
- [97] Grey Wolf, Firefly and Bat Algorithms: Three Widespread Algorithms that Do Not Contain Any Novelty_ https://www.researchgate.net/profile/Y-Yin-4/post/Is_the_so-called_novel_metaheuristic_methods_are_threatening_to_lead_the_area_of_metaheuristics_away_from_scientific_rigor2/attachment/5fcc0eb23b21a20001601065/AS%3A965512251060225%401607207467640/download/Grey+Wolf+Optimizer+that+Do+Not+Contain+Any+Novelty.pdf
- [98] Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization_ <https://fardapaper.ir/mohavaha/uploads/2020/03/Fardapaper-Multi-objective-grey-wolf-optimizer-A-novel-algorithm-for-multi-criterion-optimization.pdf>
- [99] The Whale Optimization Algorithm_ <https://fa.mie.sut.ac.ir/Downloads/AcademicStaff/5/Courses/17/The%20Whale%20Optimization%20Algorithm%202016.pdf>
- [100] OPTIMAL DESIGN OF TRUSS STRUCTURES BY IMPROVED MULTI-OBJECTIVE FIREFLY AND BAT ALGORITHMS_ <http://cefsse.iust.ac.ir/ijoce/article-1-185-en.pdf>
- [101] Grey wolf optimizer: a review of recent variants and applications https://www.researchgate.net/profile/Ibrahim-Aljarah/publication/321294919_Grey_wolf_optimizer_a_review_of_recent_variants_and_applications/links/5a19ba870f7e9be37f9a594f/Grey-wolf-optimizer-a-review-of-recent-variants-and-applications.pdf
- [102] A Hybrid Metaheuristic based on the Firefly and Bat Algorithms to Solve the Alternative Subgraphs Assembly Line Balancing Problem_ https://www.researchgate.net/publication/369671242_A_Hybrid_Metaheuristic_based_on_the_Firefly_and_Bat_Algorithms_to_Solve_the_Alternative_Subgraphs_Assembly_Line_Balancing_Problem/fulltext/6427a89f92cfd54f8444ebd3/A-Hybrid-Metaheuristic-based-on-the-Firefly-and-Bat-Algorithms-to-Solve-the-Alternative-Subgraphs-Assembly-Line-Balancing-Problem.pdf
- [103] Lévy flight trajectory-based whale optimization algorithm for engineering optimization_ <https://www.emerald.com/insight/content/doi/10.1108/EC-07-2017-0264/full/html>
- [104] Comparison and Results of Artificial Bee Colony, Bat, Firefly and Genetic Algorithms on Selected Benchmark Functions https://erpublication.org/published_paper/IJETR2985.pdf
- [105] Optimization of the Curved Metal Damper to Improve Structural Energy Dissipation Capacity_ <https://vtechworks.lib.vt.edu/bitstream/10919/107836/1/buildings-12-00067.pdf>

- [106] Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors <https://iridia.ulb.ac.be/~ccamacho/publications/ITOR-Exposing.pdf>
- [107] Modified Grey Wolf Optimizer for Global Engineering Optimization_ <https://downloads.hindawi.com/journals/acisc/2016/7950348.pdf>
- [108] Hybrid whale algorithm with evolutionary strategies and filtering for high-dimensional optimization: Application to microarray cancer data <https://pdfs.semanticscholar.org/9afb/c3cca2fe5819aec2838822da4c128f6cd890.pdf>
- [109] Bat Algorithm: Literature Review and Applications <https://arxiv.org/pdf/1308.03900>