# 用于路径规划的信息素聚焦蚁群优化算法

Yi Liu[1] Hongda Zhang[1] Zhongxue Gan[1,*] Yuning Chen[1] Ziqing Zhou[1] Chunlei Meng[1] Chun Ouyang[1,*]

arXiv:2601.07597v1 [cs.NE] 12 Jan 2026

摘要—**蚁群优化（ACO）是一种重要的群体智能算法，广泛应用于路径规划。然而，传统蚁群优化方法往往存在盲目搜索、在复杂环境中收敛缓慢等缺点。为应对这些挑战，本文提出信息素聚焦蚁群优化（PFACO）算法，通过引入三项关键策略来增强蚁群的问题求解能力。首先，根据节点到起点和终点的欧几里得距离，将初始信息素分布集中在更有前景的区域，从而平衡探索与利用之间的权衡。其次，在蚁群迭代过程中强化有前景的解，增加高质量路径上的信息素沉积，在保持解的多样性的同时加快收敛。第三，采用前瞻机制对冗余的路径转弯施加惩罚，以促进生成更平滑、高效的解。这些策略共同形成聚焦的信息素，引导蚁群搜索，从而增强PFACO算法的全局优化能力，并显著提高其在各类优化问题中的收敛速度和解的质量。实验结果表明，在收敛速度和解的质量方面，PFACO始终优于对比蚁群优化算法。**

## I. 引言

蚁群优化（ACO）算法 [1],受蚂蚁协同觅食行为的启发，因其分布式智能、并行性、正反馈和鲁棒性而备受赞誉。随着蚁群优化算法 [2],的广泛应用，研究人员逐渐发现，蚁群优化算法存在搜索效率低、收敛速度慢、停滞以及易陷入局部最优等缺点。因此，针对蚁群优化提出了多种改进方法 [3]。例如，精英蚁群优化（EliteACO） [4],通过根据已发现的最佳解调整信息素更新来加快收敛；最大最小蚁群优化（MMACO） [5], 则通过限制信息素浓度来缓解过早停滞。这些变体已得到广泛应用。

近年来，蚁群优化算法已广泛用于路径规划 [6], [7], ，其目标是找到一条从起点到终点的最优无碰撞路径。例如，GSACO [8],基于自适应贪心策略，在搜索过程中动态调整蚂蚁的偏好程度。通过利用这一贪心策略，GSACO 鼓励蚂蚁探索信息素浓度较高的区域。然而，实验结果表明，这种方法容易使蚁群陷入局部最优。其他研究人员 [9] 尝试控制新构建的解与先前选定的解之间的变化程度，以实施更有针对性的搜索，从而在保留现有解质量的同时发现潜在改进。然而，以这种方式直接限制搜索范围会增加过早收敛至局部最优的风险。此外，IHMACO [10] 采用了四种主要策略来提升算法性能，并包含 11 个可调参数。然而，将其应用于不同的问题实例时，需要进行相应的参数配置。此外，IHMACO 执行时间较长，限制了其在现实应用中的实用性。此参数调优过程需要经验丰富的研究人员分析每个实例的具体特征并作出适当调整。由于不同问题实例各具特点，此类蚁群优化算法难以在复杂环境中有效泛化。同时，该领域研究人员所提出的改进日益复杂 [11], 给其他科学家复现和实施这些策略带来了挑战。这种复杂性阻碍了这些算法的进一步推广。例如，PF3SACO [12] 采用一种动态参数调整机制，将粒子群优化（PSO） [6]与模糊系统 [13],相结合，从而加快收敛并提高准确性和稳定性。然而，其较高的计算复杂度限制了实际应用的可行性。

为提升路径规划中蚁群的解质量和收敛速度，本文提出信息素聚焦蚁群优化（PFACO）算法。为提高搜索效率，基于环境信息构建初始信息素矩阵，使信息素分布聚焦于有前景的解，从而有效平衡探索与利用。从种群层面优化的角度出发，该算法采用受生物进化启发的有前景解强化策略。具体而言，生成高质量解的个体会繁衍至后续世代，增强其在蚁群中的影响力；而生成较差解的个体则更有可能被淘汰。这种强化机制通过加大对有前景路径的信息素强化来加速收敛，同时保留多样性。此外，在个体层面，为每只蚂蚁赋予短期记忆能力，以实现局部解优化。这一策略

[1]复旦大学智能机器人与先进制造学院，中国上海

*通讯作者

通过对过多转弯点施加惩罚，减少不必要的路径偏移，从而确保轨迹更加平滑、高效。总体而言，所提出的策略使蚁群的信息素集中在接近最优解或近似最优解的区域，显著提升了算法的整体优化性能。

本文其余部分组织如下：第二节概述传统蚁群算法。第三节介绍 PFACO 的详细内容。第四节开展实验并分析结果。最后，第五节给出结论。

## II. 初步知识

蚁群优化算法受到蚁群协同行为的启发，采用动态信息素更新机制，并利用蚂蚁数量 $M$ 进行 $K$ 次迭代搜索，以获得最优解或近似最优解。

本节介绍蚁群优化 [14]的基本过程。在初始迭代中，每只蚂蚁从可用集合中随机选择一个动作，所有选项的概率均相等。当所有蚂蚁完成一次迭代内的搜索后，将相应地更新信息素浓度。信息素更新由以下公式（1）控制：

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \sum_{m=1}^{M}\Delta\tau_{ij}^{m},\ 0<\rho<1 \tag{1}$$

其中，$M$ 表示蚂蚁最大数量，$\rho$ 表示信息素挥发因子，$\tau_{ij}$ 表示节点 $i$ 与节点 $j$之间的信息素水平，$\Delta\tau_{ij}^{m}$ 对应于蚂蚁 $m$ 在从 $i$ 到 $j$的路径上释放的信息素。

$$\Delta\tau_{ij}^{m} = \begin{cases} Q/L_m, & \text{condition a} \\ 0, & otherwise \end{cases} \tag{2}$$

[a]如果蚂蚁 $m$ 在其巡游路径中经过边 (i,j)。

其中，$Q$是一个超参数，$L_m$ 是蚂蚁 $m$所走路径的长度。

在后续迭代中，蚂蚁的行动由状态转移规则引导，该规则受前几次迭代中沉积的信息素影响。此规则定义如下：

$$p_{ij}^{m} = \begin{cases} \frac{\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}{\sum_{l\in allowed_m}\tau_{il}^{\alpha}\cdot\eta_{il}^{\beta}}, & if\ j\in allowed_m \\ 0, & otherwise \end{cases} \tag{3}$$

其中，$\alpha$ 和 $\beta$ 是用于调节信息素浓度与启发函数之间平衡的超参数。启发函数 $\eta$ 定义为节点 $i$和 $j$ 之间距离的倒数（$d_{ij}$），表示为：

$$\eta_{ij} = 1/d_{ij} \tag{4}$$

$allowed_m$ 定义为蚂蚁$m$ 从当前节点 $i$可访问的邻接节点集合。

## III. 信息素聚焦蚁群优化算法（PFACO）

本节介绍PFACO算法的详细内容。PFACO通过三种策略增强最优解或近似最优解周围的信息素浓度：自适应距离信息素初始化（ADPI）、承诺解信息素强化策略（PSPRS）和前瞻转弯优化策略（LTOS）。PFACO的伪代码如算法1所示。

### *A.* 自适应距离信息素初始化（*ADPI*）

在传统蚁群优化算法中，环境中所有位置的初始信息素浓度要么相同（通常设为 0 或 1 等整数值），要么根据节点之间的距离确定（如公式 (4)）。这种统一初始化会导致蚂蚁在早期探索阶段选择行动时具有较强的随机性，从而减慢收敛速度。为解决这一问题，本文提出一种新颖的信息素分配方法，以平衡蚁群中随机性与收敛速度之间的权衡。由于两点间的直线距离代表最短路径，并且在理想条件下是最优解，PFACO 将起点与终点之间的欧几里得距离作为基准。通过计算该基准与其他位置到起点和终点的欧几里得距离之和的比值，评估这些位置与最优解的接近程度。这种方法称为自适应距离信息素初始化（ADPI），它根据公式 (5) 确定环境中的初始信息素浓度值。

$$\tau_{ij}^{0} = \frac{a\times Euc(ST)}{Euc(Sj)+Euc(jT)}, \tag{5}$$

如需更详细地解释式 (6) 中的 $a$:

$$\begin{cases} a=2 & d_{iT} > d_{jT} \\ a=1 & \text{else} \end{cases} \tag{6}$$

其中，$S$ 表示起始节点，$T$ 表示终止节点，$i$ 和 $j$ 为可用节点，$Euc(xy)$表示 $x$ 与 $y$之间的欧几里得距离。当 $d_{iT}$ >$d_{jT}$时，表明节点 $j$更靠近终止节点 $T$，因此会为节点 $j$ 分配更高的权重（其中 $a=2$）。该加权机制促使蚁群优先朝节点 $j$的方向进行探索。

在蚁群运行初期，该策略使信息素浓度分布集中于理想的最优解或近似最优解。因此，它增强了信息素对蚂蚁探索方向的引导作用，同时显著提高了算法的收敛速度。

### *B.* 有前景解信息素强化策略（*PSPRS*）

为促进蚁群中信息素浓度的显著分布，引入有前景的解信息素强化策略（PSPRS）。

**算法1：PFACO 的伪代码。输入：实例（地图 $W \times H$、起点 $S$ 和终止点 $T$）以及蚁群参数，包括迭代次数 $K$ 和蚂蚁数量 $M$。输出：最优解或近似最优解。: 1 初始化：自适应距离信息素初始化；2 对 $w \leftarrow 1$ 到 $W$ 执行 3 对 $h \leftarrow 1$ 到 $H$ 执行 4 如果节点 $i$ 可用，则 5 计算欧几里得距离 $Euc$（$ST$）、$Euc$（$Si$）、$Euc$（$iT$）、$Euc$（$jT$）；6 根据式（5）初始化非均匀信息素矩阵；7 否则 8 信息素浓度为 0；9 结束 10 结束 11 结束 12 初始化 $Global\ Elite\ Solution$ 和 $Top\ Quality\ Solution$ 列表；13 对 $k \leftarrow 1$ 到 $K$ 执行 14 对 $m \leftarrow 1$ 到 $M$ 执行 15 如果蚂蚁 $m$ 尚未到达终止点 $T$，则 16 蚂蚁 $m$ 根据式（1）、（7）和（3）选择下一个节点；17 结束 18 通过 LTOS（第三节-C）更新蚂蚁 $m$ 的路径；19 结束 20 如果所有蚂蚁均到达终止点 $T$，则 21 更新 $Global\ Elite\ Solution$ 和 $Top\ Quality\ Solution$ 集合；**

22 $New\ Set = Top\ Quality\ Solution + 5 \times Global\ Elite\ Solution$；**23 根据 $New\ Set$，通过式（1）和（7）更新信息素矩阵；24 初始化 $New\ set$；25 结束 26 结束**

在 PFACO算法中，蚁群实时记录其发现的排名前 ($0.1 \times M$) 个全局解，并将其列为全局精英解。每次迭代后，蚂蚁发现的解会按质量排序，并保留当前迭代中排名前一半的解，作为高质量解。随后，将全局精英解集合中的每个解复制五次，形成一个规模相当于蚁群中蚂蚁数量一半的集合。这些复制得到的有前景的解与全局精英解共同参与更新环境中的信息素矩阵。如算法1第20至25行所示，该策略增强了高质量解周围的信息素浓度，从而提升有前景的解的长期影响，并加快最优解上的信息素积累。此外，复制机制不仅扩大了高质量解的影响，还保留了多样性，并防止过早收敛到局部最优。

通过整合 PSPRS，PFACO 提高了全局搜索效率并加快了收敛速度，从而增强了其在复杂环境中优化路径的能力。总体而言，该策略优化了信息素分布，强化了蚁群对高质量解的偏好，并显著提升了 PFACO 在复杂优化任务中的性能。

### C。前瞻转弯优化策略（ LTOS)

在传统蚁群算法中，状态转移规则（例如式 4）除了考虑信息素浓度外，还会使选择过程倾向于距离较短的相邻节点（如上、下、左、右）。这种偏好可能导致全局解中转弯次数过多。为解决这种短视问题并进一步优化信息素浓度，本文提出了一种前瞻性节点重配置机制，称为前瞻转弯优化策略（LTOS）。该机制在搜索过程中动态调整局部搜索空间中的父节点，从而减少冗余转弯，并防止局部最优导致过度绕行。

此外，信息素更新由

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/(L_k + Turn_k), & \text{condition b} \\ 0, & otherwise \end{cases} \tag{7}$$

[b]如果蚂蚁 $k$ 在其巡游路径中访问了边 (i,j)。

该公式不仅考虑了路径长度，还引入转弯点数量（$Turn_k$）作为惩罚项，从而有效平衡路径长度与转弯冗余。通过采用这种方法，PFACO 大幅增强了蚁群搜索高质量路径的能力。因此，整体路径更加平滑，质量也更高。

## IV. 实验

### A. 数据集

在路径规划任务中，地图尺度通常与任务复杂度呈正相关。地图越大，通常涉及的范围越广，整体挑战也随之增加。为全面评估所提出的 PFACO算法在不同路径规划尺度下的性能，本文基于 [15]和 [16]中提及的网格世界环境，构建了三个不同尺寸的地图数据集。这些数据集由尺寸分别为 $10 \times 10$、 $15 \times 15$和 $20 \times 20$的地图组成。例如，$10 \times 10$ 地图数据集包含一张无障碍地图、五张障碍物配置各不相同的地图，以及四张含有混合障碍物的地图。图 1展示了一张示例地图。空间被划分为 $N \times N$ 个方块，其中黑色网格表示障碍物，白色网格表示

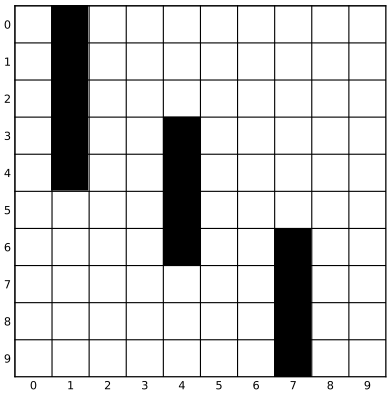


图 1. 地图数据集示例。

可通行区域。在每个时间步，智能体可以沿八个方向移动：(0,1)、(0,-1)、(1,1)、(1,-1)、(-1,1)、(-1,-1)、(1,0)、(-1,0)。

在对比实验（第四节-C）中，为确保测试实例具有多样性，从每个数据集中随机选取起点和终点，抽取 100 个实例。这些实例作为评估的基础，确保实验结果兼具代表性和全面性。

### *B.* 实验设置

本研究的所有实验均在一台配备 Intel Core i7-8700 @ 3.20GHz 中央处理器和 RTX 2080 SUPER 图形处理器的个人计算机上进行，所有算法均使用 PyTorch 在 Python 3.7 中实现。

本文讨论的对比算法和 PFACO算法的参数选择主要依据已有研究 [17]。PFACO算法的参数设置如下：$Q = 2$, $\rho = 0.2$, $\alpha = 1$,$\beta = 3$, 以及 $T = 2$ *or* 1。对于蚁群系统（AS），如第四节- C 所述，其配置包括 30 只蚂蚁和 20 次迭代，记为“AS-30-20”。本研究采用“算法名称 - 种群规模 - 搜索迭代次数”的格式标记蚁群优化算法变体。蚁群优化算法的参数定义如下：$\alpha = 1$, $\beta = 3$, 信息素挥发率设为 $[0.1, 0.4)$, 以及超参数 $Q = 2$。

为评估 PFACO 在路径规划中的性能，选取了五种蚁群优化算法作为基准方法：AS、精英蚁群系统、最大最小蚁群系统、NCAACO [18], 和 IHMACO [10]。此外，在对比实验中还纳入了 A*算法 [19] ，以更全面地评估 PFACO 在机器人路径规划中的整体性能。

通过比较 100 个随机生成实例中的关键指标来评估 PFACO 的性能，包括平均路径长度（平均路径）、每个实例的平均耗时（时间*(%)*）、转弯次数（转弯次数）、成功率（成功率*(%)*）和路径改进率（路径改进率（%））。平均路径长度越短，解的质量越高；而平均耗时反映了该算法的计算效率。转弯次数 并非完全独立的评价指标，而是结合平均路径长度来分析算法性能。此外，路径改进率（路径改进率（%））用于比较同一蚁群优化算法在不同迭代次数和蚁群规模下的性能，以评估其收敛速度和全局优化能力。其定义如下：

$$\frac{AveragePath(a') - AveragePath(a'')}{AveragePath(a'')} \tag{8}$$

其中， $a'$ 和 $a''$ 分别表示参数集不同的算法 a。通过分析解的质量随迭代次数和蚁群规模增加而发生的变化，本研究进一步评估算法的稳定性。此外，还考察路径长度的标准差（*SD-P*）和耗时的标准差（*SD-T*），以评估结果的一致性和稳定性。为确保研究结果具有统计显著性，采用曼-惠特尼U检验比较各算法结果的均值差异，显著性水平设为 0.05。

为避免将蚁群优化算法应用于复杂实例时因潜在的死锁问题导致计算时间过长，本节为所有算法设置了截止阈值。具体而言，如果搜索过程超过 120 秒仍未达到目标，则视为未成功。不过，需要注意的是，这一机制可能会降低蚁群优化算法解决某些实例的成功率。

### *C.* 结果与讨论

本节比较 PFACO 与 AS、精英蚁群系统、最大最小蚁群系统、NCAACO 和 IHMACO 在地图尺寸为 $10 \times 10$、$15 \times 15$ 和 $20 \times 20$ 的数据集上的性能，结果见表 I。此外，还纳入 A*算法的结果，以全面评估 PFACO 在路径规划任务中的表现。比较采用多项性能指标，包括平均路径、时间（秒）、转弯次数、*SD-P*、*SD-T*、成功率（%）、路径改进率（%）和*p* 值。

如表 I 所示，在所有对比的蚁群优化算法中，PFACO-30-20 在三个地图尺度下的平均路径均最短。具体而言，在 $10 \times 10$ 地图集上，PFACO 在平均路径和转弯次数两项指标上均优于 A*算法。在 $15 \times 15$ 地图集上，PFACO-30-20 的平均路径 最接近 A*算法，差值仅为 0.215。在 $20 \times 20$ 地图集上，PFACO-30-20 的平均路径仍与 A*算法具有竞争力，差值为 2.134；同时，其平均转弯次数比 A*算法少 0.44。此外，与所有对比算法相比，PFACO-30-20 的*SD-P*最小，PFACO-15-10 次之，这表明 PFACO 生成的解比其他算法更加稳定。这些结果进一步证实，PFACO 在不同地图尺度下均能稳定地提供高质量的路径规划解。

就平均计算时间（时间（秒））而言，PFACO-15-10 在所有实验环境中的执行时间最短，其次是 PFACO-30-20，后者

表 I PFACO 与其他对比算法在三个不同地图规模的数据集上生成的 100 个随机实例中的性能。这些算法的评估指标包括平均路径长度（平均路径）、平均时间成本（时间（秒））、平均转弯次数（转弯次数）、路径长度标准差（*SD-P*）、时间成本标准差（*SD-T*）、实例完成成功率（成功率（%））、增加蚂蚁数量和迭代次数所带来的路径规划结果改进程度（路径改进率（%）），以及显著性结果（*p* 值）。蚁群优化算法中的最佳值以粗体标出。

| 10 × 10 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $AveragePath$ | 时间（秒） | 转弯次数 | *SD-P* | *SD-T* | 成功率 *(%)* | $PathImprove$ *(%)* | p 值 |
| A* | 5.217 | $6.65e^{-05}$ | 2.16 | 3.298 | $6.23e^{-05}$ | 100 | - | **0.314** |
| AS-15-10 | 6.157 | 0.260 | **1.18** | 4.267 | 0.209 | 100 | - | 0.014 |
| EliteACO-15-10 | 5.328 | 0.266 | 1.60 | 3.524 | 0.225 | 100 | - | **0.241** |
| MMACO-15-10 | 5.337 | 0.277 | 1.49 | 3.528 | 0.223 | 100 | - | **0.235** |
| NCAACO-15-10 | 5.255 | 0.986 | 1.58 | 3.404 | 0.843 | 100 | - | **0.309** |
| IHMACO-15-10 | 8.809 | 2.060 | 2.03 | 6.225 | 8.863 | 94 | - | $2.733e^{-07}$ |
| PFACO-15-10 | 5.068 | **0.103** | 1.84 | 3.231 | **0.136** | 100 | - | **0.437** |
| AS-30-20 | 5.766 | 1.040 | 1.37 | 3.921 | 0.829 | 100 | 6.35 | **0.063** |
| EliteACO-30-20 | 5.159 | 1.040 | 1.64 | 3.277 | 0.845 | 100 | 3.17 | **0.361** |
| MMACO-30-20 | 5.138 | 1.080 | 1.81 | 3.284 | 0.865 | 100 | 3.73 | **0.378** |
| NCAACO-30-20 | 5.135 | 3.890 | 1.72 | 3.272 | 3.398 | 100 | 2.28 | **0.403** |
| IHMACO-30-20 | 8.212 | 3.460 | 2.00 | 6.013 | 10.321 | 93 | 6.77 | $4.498e^{-06}$ |
| PFACO-30-20 | **5.013** | 0.320 | 1.94 | **3.095** | 0.294 | 100 | **1.09** | - |
| 15 × 15 | | | | | | | | |
| | $AveragePath$ | 时间（秒） | 转弯次数 | *SD-P* | *SD-T* | 成功率 *(%)* | $PathImprove$ *(%)* | p 值 |
| A* | 8.697 | $6.65e^{-05}$ | 2.16 | 3.298 | $2.12e^{-04}$ | 100 | - | **0.38** |
| AS-15-10 | 12.346 | 0.937 | **2.10** | 7.852 | 0.758 | 100 | - | $2.431e^{-04}$ |
| EliteACO-15-10 | 10.410 | 0.969 | 2.48 | 6.778 | 0.789 | 100 | - | 0.046 |
| MMACO-15-10 | 10.403 | 1.000 | 2.47 | 6.919 | 0.843 | 100 | - | 0.050 |
| NCAACO-15-10 | 10.362 | 3.470 | 2.47 | 6.723 | 2.750 | 100 | - | 0.048 |
| IHMACO-15-10 | 14.736 | 6.670 | 4.43 | 7.009 | 14.104 | 84 | - | $3.966e^{-09}$ |
| PFACO-15-10 | 9.328 | **0.215** | 3.27 | 5.910 | **0.173** | 100 | - | **0.303** |
| AS-30-20 | 11.435 | 3.840 | 2.22 | 7.246 | 3.145 | 100 | 7.37 | $3.439e^{-03}$ |
| EliteACO-30-20 | 9.559 | 3.850 | 2.67 | 5.916 | 3.189 | 100 | 8.17 | **0.213** |
| MMACO-30-20 | 9.653 | 3.940 | 2.77 | 6.137 | 3.332 | 100 | 7.21 | **0.184** |
| NCAACO-30-20 | 9.681 | 3.890 | 2.70 | 6.206 | 13.032 | 100 | 6.57 | **0.176** |
| IHMACO-30-20 | 13.301 | 7.220 | 4.11 | 7.494 | 14.269 | 82 | 9.74 | $1.475e^{-05}$ |
| PFACO-30-20 | **8.912** | 0.695 | 3.13 | **5.412** | 0.516 | 100 | **4.46** | - |
| 20 × 20 | | | | | | | | |
| | $AveragePath$ | 时间（秒） | 转弯次数 | *SD-P* | *SD-T* | 成功率 *(%)* | $PathImprove$ *(%)* | p 值 |
| A* | 10.982 | $2.30e^{-04}$ | 5.15 | 5.174 | $2.00e^{-04}$ | 100 | - | 0.003 |
| AS-15-10 | 19.167 | 2.21 | 3.47 | 11.523 | 1.660 | 100 | - | $9.751e^{-05}$ |
| EliteACO-15-10 | 16.234 | 2.18 | **3.31** | 9.902 | 1.638 | 100 | - | 0.025 |
| MMACO-15-10 | 16.231 | 2.20 | 3.52 | 9.706 | 1.607 | 100 | - | 0.025 |
| NCAACO-15-10 | 16.081 | 8.71 | 3.48 | 9.598 | 6.667 | 100 | - | 0.035 |
| IHMACO-15-10 | 20.449 | 15.20 | 5.74 | 9.605 | 26.907 | 58 | - | $2.341e^{-06}$ |
| PFACO-15-10 | 14.116 | **1.14** | 4.58 | 8.560 | **0.941** | 100 | - | **0.349** |
| AS-30-20 | 18.155 | 8.59 | 3.44 | 11.038 | 6.365 | 100 | 5.27 | $7.382e^{-04}$ |
| EliteACO-30-20 | 15.029 | 8.66 | 3.74 | 8.962 | 6.333 | 100 | 7.42 | **0.134** |
| MMACO-30-20 | 15.161 | 8.95 | 3.61 | 9.042 | 6.677 | 100 | 6.59 | **0.114** |
| NCAACO-30-20 | 15.317 | 31.30 | 3.34 | 9.229 | 18.373 | 100 | 4.75 | **0.095** |
| IHMACO-30-20 | 20.203 | 10.50 | 5.75 | 9.645 | 17.188 | 53 | **1.20** | $8.273e^{-06}$ |
| PFACO-30-20 | **13.664** | 1.39 | 4.71 | **8.017** | 1.557 | 100 | 3.20 | - |

在参与比较的蚁群算法中，效率排名前三。具体而言，PFACO-15-10 的 *SD-T* 最小，而 PFACO-30-20 的 *SD-T* 排名前三，这表明 PFACO 的计算时间波动较小。

通过对平均路径的改进以及路径提升幅度进行对比分析如表 I 所示，在增加蚁群规模和迭代次数前后比较路径长度时，PFACO 的改进幅度最小，同时保持了最短的平均路径。这一观察表明，与其他算法相比，PFACO 只需较少的蚂蚁和迭代次数即可实现更优的路径规划结果，无需

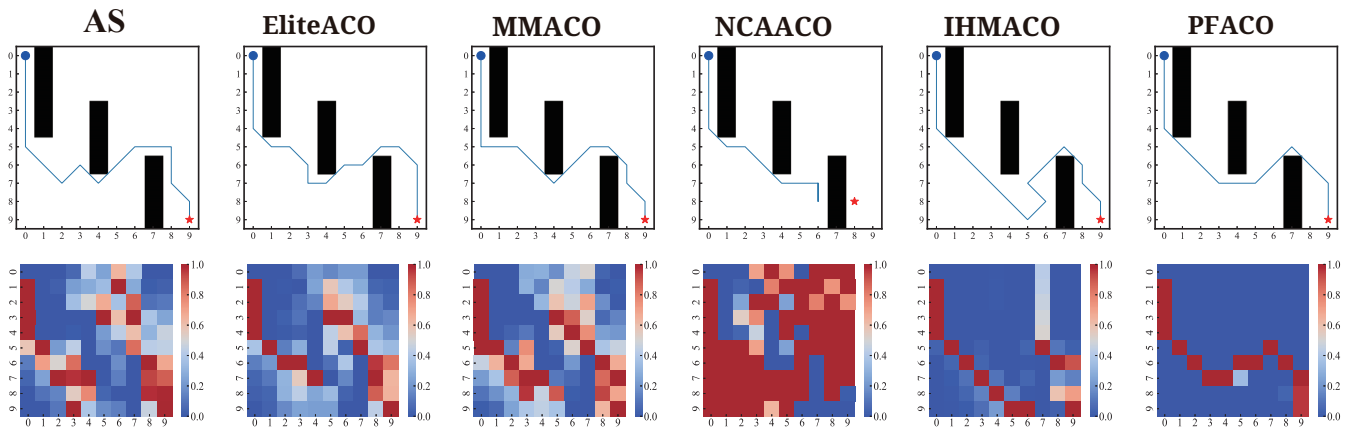



Fig. 2. 实例 1 中蚁群优化算法的路径规划结果及最终信息素浓度分布

大规模蚁群或大量迭代来显著提升路径质量。这些发现凸显了 PFACO 在计算资源利用方面的效率：它能够以更高的效率获得最优解，展现出更强的搜索性能和稳定性，适用于路径规划任务。

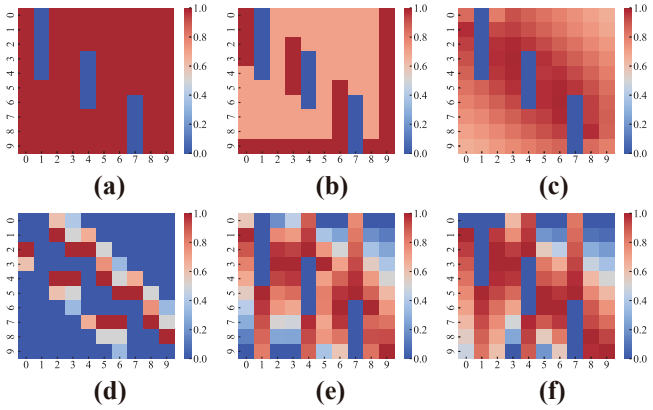



图3. 展示了实例 1 中不同策略下的初始信息素浓度分布。纵轴和横轴分别表示网格地图的 $x$ 和 $y$ 坐标。颜色渐变表示信息素浓度值，其中暖色代表较高的浓度，冷色代表较低的浓度。浓度值范围为 [0, 1]。

此外，表 I 展示了假设检验的结果。我们采用曼-惠特尼 U 检验，将取得最佳平均路径 （以粗斜体突出显示）的算法与其他算法进行比较。在显著性水平为 0.05 时，若未检测到显著差异，则以粗体显示结果。显著性检验结果表明，在 $10 \times 10$ 地图数据集上，A*算法、AS-30-20、EliteACO、MMACO、NCAACO、IHMACO、PFACO-15-10 和 PFACO-30-20 之间不存在统计学显著差异。同样，在 $15 \times 15$ 地图数据集上，A*算法、EliteACO-30-20、MMACO- 30-20、NCAACO-30-20、PFACO-15-10 和 PFACO-30-20 之间未观察到显著差异。在 $20 \times 20$ 地图数据集上，EliteACO-30-20、NCAACO-30- 20、PFACO-15-10 和 PFACO-30-20 之间不存在统计学显著差异。表 I 中的结果表明，PFACO-15-10 和 PFACO-30-20 在所有地图规模下均无显著劣势，这证明了 PFACO 在不同问题规模下具有很强的稳定性。

总体而言，表 I 中的实验结果凸显了 PFACO 在多个方面的优势。它能够获得更短的路径长度、更高的解稳定性、更优的计算效率以及更小的波动。此外，PFACO 在不同问题规模下均能保持稳定的性能，展现出应对搜索空间扩大的鲁棒性。这些特性共同确立了 PFACO 作为一种高效、可靠的路径规划算法的地位，尤其适用于计算资源有限或要求快速响应的应用场景。

本节考察一张 10尺度地图（图 1），起点位于 (0,0)，目标点位于 (9,9)，该地图称为**实例 1**。与边缘障碍物不同，边缘障碍物虽会增加障碍物密度，却不会显著影响路径规划难度；实例 1 的障碍物位于地图中央，形成了“C形陷阱”和迷宫式结构。这些障碍物可能误导算法的探索，使路径规划变得更具挑战性。为评估算法性能，本节将最终路径和信息素分布可视化，以便直观比较。图 2 展示了 AS、EliteACO、MMACO、NCAACO、IHMACO 和 PFACO 在实例 1 上的结果。上排展示最终路径规划结果，下排展示相应的信息素浓度分布。与其他算法相比，PFACO 在最优解或近似最优解周围呈现出更集中的信息素分布，表明其收敛性更强。相比之下，即使在搜索后期，AS、EliteACO 和 MMACO 的信息素分布仍较为分散。IHMACO 算法最终的信息素浓度分布在所比较的算法中表现出最强的收敛性。然而，在路径后半段，L形障碍物使其无法在给定的迭代次数和蚂蚁数量约束下收敛至最优解或近似最优解附近，因此可能需要额外的计算资源。因此，

其搜索效率略低于 PFACO。NCAACO 中出现了一个值得注意的情况：NCAACO 的超参数与实例 1 不匹配，可能导致转移概率计算失效，使蚂蚁在环境中无休止地探索，却无法到达目标点。PFACO 收敛更快、信息素分布更集中，这得益于其整合的策略：ADPI 根据具体实例自适应地优化信息素初始化；PSPRS 加速全局收敛；LTOS 则提升局部路径质量。这些因素共同使 PFACO 能够比其他算法更快收敛，并找到效率更高的解。

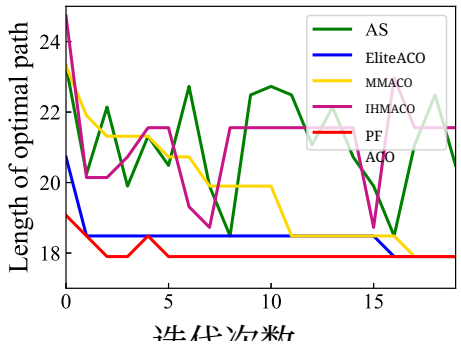



图4. PFACO 与其他基准算法在实例 1 上迭代过程中的路径长度变化曲线。

为展示 ADPI 的有效性，本节将 PFACO 的信息素矩阵与另外五种不同的信息素初始化策略进行比较。如图3所示，（f）表示 ADPI 下的信息素初始化矩阵，并与以下初始化方法进行对比：（a）常数均匀分布，（b）基于逆距离的信息素初始化（$\eta_{ij}$），（c）NCAACO 中使用的信息素初始化策略 [18]，（d）IACO-SFLA [20] 以及（e）MsAACO [21]。对比结果表明，在搜索开始之前，ADPI 为蚁群提供了具有针对性的信息素分布，有效过滤了潜力较小的搜索区域。该策略通过引导探索指向最有价值的区域，提高了搜索精度。相比之下，常数均匀分布（图3（a））在信息素浓度方面缺乏差异，无法为后续搜索提供额外引导。基于逆距离的初始化（图3（b））引入了一定的差异性，但仅提供了指向目标点的模糊方向偏差。NCAACO [22]（图3（c））和 IHMACO（图3（d））中的信息素分布相似，最高的信息素浓度位于起点与目标点之间的直线路径上。然而，图3（d）中的 IHMACO 比图3（c）中的 NCAACO 覆盖了更广泛的有价值探索区域。这种分布在应对L形或C形障碍物等复杂陷阱时效果较差。可视化结果表明，在 PFACO 的初始阶段，该策略有助于形成以期望的最优解或近似最优解为中心的信息素浓度分布。它增强了

表II PFACO 与其他对比算法在实例1上的10次可重复性实验结果。

| 算法 | 路径长度 | | | 转弯次数 | 时间（秒） |
|---|---|---|---|---|---|
| | Mean | Best | Std. | | |
| A* | 19.657 | 19.657 | 0.000 | 9.0 | $3.533e^{-04}$ |
| AS | 21.499 | 19.899 | 1.029 | 6.1 | 4.611 |
| 精英蚁群优化算法 | 18.802 | 18.485 | 0.439 | **6.0** | 4.852 |
| MMACO | 18.724 | **17.899** | 0.444 | 6.5 | 4.909 |
| IHMACO | 21.356 | 20.142 | 0.690 | 8.1 | 1.462 |
| PFACO | **18.134** | **17.899** | **0.302** | 6.8 | **1.322** |

蚂蚁探索的引导，使其能够高效地穿越具有潜力的搜索区域。

为说明PSPRS策略的有效性，本节通过绘制实例 1 中蚁群在迭代过程中所获得的解质量变化，对PFACO与其他竞争算法进行了可视化比较。如图4所示，在实例 1的求解过程中，与其他基准算法相比，PFACO获得了最小的初始解、最快的收敛速度以及最小的波动。这些结果进一步验证了PSPRS的有效性，因为该策略能够在搜索过程中引导蚂蚁朝向有希望的路径，从而更快地收敛到最优解或近似最优解。解质量波动的减小表明，PSPRS能够稳定搜索过程，防止解之间出现无规律的跳跃，并促进对解空间更加一致的探索。这说明PSPRS有效增强了蚁群对高质量解的偏好，从而提升了其解决复杂优化问题的性能。

为评估LTOS的有效性，本节展示了在实例 1上使用PFACO及其他算法开展的10次可重复性实验结果。如表II所示，尽管PFACO在所有算法中并未取得最低的转弯次数，但在重复实验中始终获得最短的平均路径长度和最佳值路径长度，且标准差最小。这些结果表明，尽管PFACO未将转弯次数降至最低，但其在路径质量和稳定性方面表现出色。具体而言，LTOS能够有效减少不必要的转弯，从而得到更加平滑、可靠的解，这一点体现在路径长度的最小变化上（如 $Std.$ 的 $PathLength$所示）。这表明，与其他算法相比，PFACO成功平衡了转弯次数和路径长度，实现了更加稳定、高效的路径规划。

## V. 结论

本文提出了信息素聚焦蚁群优化（PFACO）算法，该算法通过优化信息素浓度，使其聚焦于潜在最优解或近似最优解周围的区域，从而提高解的质量和收敛速度。本文在不同地图尺度下，将 PFACO 与多种基准算法进行比较，以评估其性能。实验结果表明

，PFACO 的表现始终优于其他对比算法，尤其是在较小的地图上，其解的质量超过了确定性路径规划算法。此外，本文验证了 ADPI、PSPRS 和 LTOS 的有效性。这些发现表明，PFACO 为基于蚁群优化的方法增添了有价值的新成员。然而，与其他基于蚁群的算法一样，PFACO 在大尺度地图上的计算成本较高。未来工作将着重提升其在动态复杂环境中的可扩展性和效率，以缓解这些局限，并将所提出的方法拓展至更广泛的应用领域 [23], [24]。

## 致谢

本研究部分获得上海市科技重大项目（编号：2021SHZDZX0103）资助。

## 参考文献

[1] M. Dorigo 和 T. St¨utzle，《蚁群优化：概述与最新进展》。Springer，2019年。[2] L. Liu、X. Wang、X. Yang、H. Liu、J. Li 和 P. Wang，《移动机器人路径规划技术：综述与展望》，*Expert Systems withApplications*，第227卷，文章编号120254，2023年。[3] H. Cao、Q. Yang、X.-D. Gao、P.-L. Xu、Z.-Y. Lu 和 J. Zhang，《基于随机两两竞争的蚁群优化中蚂蚁选择与信息素更新》，载于*2023 IEEE*系统、人和控制论国际会议（*SMC*）。IEEE，2023年，第1455–1460页。[4] B. Bullnheimer、R. F. Hartl 和 C. Strauss，《蚁群系统的全新基于排序的版本——一项计算研究》。*Central European Journal of Operations Research*，第7卷，第1期，1999年。[5] T. St¨utzle 和 H. H. Hoos，《最大最小蚁群系统》，*Future generation computer systems*，第16卷，第8期，第889–914页，2000年。[6] H. Zhang、Z. Gan、Y. Liu、W. Lin、B. Liu、C. Meng 和 C. Ouyang，《基于启发式算法的递归图编码与卷积神经网络模型联合优化》，载于*2024 IEEE*系统、人和控制论国际会议（*SMC*）。IEEE，2024年，第2513–2519页。[7] Y. Guan、Y. Chen、Y. Liu、H. Zhang、Z. Zhou、W. Ding、Z. Zou、L. Xu、C. Ouyang 和 Z. Gan，《面向动态供需的多智能体协同物流实时调度框架》，*IEEE*工业信息学汇刊，2025年。[8] W. Li、L. Xia、Y. Huang 和 S. Mahmoodi，《采用自适应贪心策略优化路径问题的蚁群优化算法》，*Journal of Ambient Intelligence and Humanized Computing*，第1–15页，2022年。[9] R. Skinderowicz，《提升蚁群优化效率以求解大规模旅行商问题实例》，*Applied Soft Computing*，第120卷，文章编号108653，2022年。[10] C. Liu、L. Wu、W. Xiao、G. Li、D. Xu、J. Guo 和 W. Li，《用于求解路径规划的改进启发式机制蚁群优化算法》，*Knowledge-based systems*，第271卷，文章编号110540，2023年。[11] C. Meng、W. Lin、B. Liu、H. Zhang、Z. Gan 和 C. Ouyang，《Rts-vit：用于图像分类的实时共享视觉变换器》，*IEEE*生物医学与健康信息学期刊，第1–12页，2025年。[12] X. Zhou、H. Ma、J. Gu、H. Chen 和 W. Deng，《基于参数自适应的动态混合机制蚁群优化》，*Engineering Applicationsof Artificial Intelligence*，第114卷，105139，2022年，第。[13] Z. Chen、S. Zhou 和 J. Luo，《用于连续函数的鲁棒蚁群优化》，*Expert Systems with Applications*，第81卷，第309–320页，2017年。[14] M. Dorigo、V. Maniezzo 和 A. Colorni，《蚁群系统：通过协作智能体群体进行优化》，*IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*，第26卷，第1期，第29–41页，1996年。[15] M. Bhardwaj、S. Choudhury 和 S. Scherer，《通过模仿学习启发式搜索》，载于机器人学习会议，PMLR。PMLR，2017年，第271–280页。

[16] R. Yonetani、T. Taniai、M. Barekatain、M. Nishimura 和 A. Kanezaki，《使用神经网络 A* 搜索进行路径规划》，载于国际机器学习会议，PMLR。PMLR，2021年，第12 029–12 039页。[17] Y. Liu、J. Qiu、E. Hart、Y. Yu、Z. Gan 和 W. Li，《基于学习的神经网络蚁群优化》，载于遗传与进化计算会议论文集，2023年，第47–55页。[18] Q. Luo、H. Wang、Y. Zheng 和 J. He，《基于改进蚁群算法的移动机器人路径规划研究》，神经计算与应用，第32卷，第6期，第1555–1566页，2020年。[19] P. E. Hart、N. J. Nilsson 和 B. Raphael，《最小成本路径启发式确定的形式化基础》，*IEEE*系统科学与控制论汇刊，第4卷，第2期，第100–107页，1968年。[20] X. Pu、X. Song、L. Tan 和 Y. Zhang，《具有多目标的单机器人和多机器人路径规划中的改进蚁群算法》，进化智能，第17卷，1326，2024年，第3期，第1313页。[21] J. Cui、L. Wu、X. Huang、D. Xu、C. Liu 和 W. Xiao，《多策略自适应蚁群优化算法及其在机器人路径规划中的应用》，*Knowledge-BasedSystems*，第288卷，111459，2024年，第。[22] S. Wu、A. Dong、Q. Li、W. Wei、Y. Zhang 和 Z. Ye，《最远点优化与多目标策略蚁群优化算法在机器人路径规划中的应用》，*Applied Soft Computing*，第167卷，文章编号112433，2024年。[23] X. Guo、F. Guo、L. Qi、J. Wang、S. Liu、S. Qin 和 W. Wang，《具有资源共享的多产品人机协作混合拆卸线平衡建模与优化》，*IEEE*计算社会系统汇刊，2025年。[24] T. Wei、X. Guo、M. Zhou、J. Wang、S. Liu、S. Qin 和 Y. Tang，《考虑人为因素的拆卸线平衡问题多目标离散和声搜索优化器》，*IEEE*人机系统汇刊，2025年。