

结合 B 样条优化的 UAV 多区域路径规划融合算法

汪瀚洋¹, 卢厚清¹, 陈亮², 赵小康¹, 杨柳¹

(1. 陆军工程大学 野战工程学院, 南京 210007; 2. 军事交通学院 汽车士官学校, 安徽 蚌埠 233011)

摘要:为解决无人机(UAV, unmanned aerial vehicle)在多个目标区域之间快速找到最佳遍历路径的类旅行商问题(TSP, travelling salesman problem),设计一种基于蚁群算法、A*算法以及三次B样条优化的融合规划算法;尽管蚁群算法相对其他优化算法在解决TSP问题上有较为良好的表现,但其规划路径处理时间长、生成路径转折多、路径质量和安全性较差;算法首先改进传统A*算法的节点扩展方式,快速生成两两目标区之间的局部路径,然后将蚁群算法和改进A*算法融合使用进行全局路径规划,最后结合改进三次B样条对路径进行平滑处理;基于栅格地图的仿真结果证明了该算法相比传统算法具有更好的高效性和稳定性。

关键词:多目标区域;无人机路径规划;TSP问题;A*算法;蚁群算法;三次B样条

Fused Algorithm for the Planning of UAV Path between Multiple Areas Combined with B-spline Optimization

WANG Hanyang¹, LU Houqing¹, CHEN Liang², ZHAO Xiaokang¹, YANG Liu¹

(1. College of Field Engineering, Army Engineering University, Nanjing 210007, China;

2. Automobile NCO School, Army Military Transportation University, Bengbu 233011, China)

Abstract: In order to solve the Traveling Salesman Problem (TSP) of Unmanned Aerial Vehicle (UAV) which wants to quickly find the best traversal path between multiple target areas, a fusion programming algorithm is designed by the ant colony algorithm, A* algorithm and cubic B-spline optimization; Although the ant colony algorithm has better performances than other optimization algorithms in solving TSP, its planning path processing time is long, the generated path turns much more, and the quality and security of path are poor; The algorithm firstly improves the node expansion mode of the traditional A* algorithm to quickly generate the local path between two target areas, then combines the ant colony algorithm and the improved A* algorithm for global path planning, and the path combined with the cubic B-spline is finally smoothed; The simulation results based on grid map show that the algorithm has better efficiency and stability than the traditional algorithm.

Keywords: multiple areas; the planning of UAV path; traveling salesman problem; A* algorithm; Ant colony algorithm; cubic B-spline

0 引言

无人机(UAV, unmanned aerial vehicle)作为新型智能化作战力量,具有较低的应用成本的同时,也具有较好的机动性,其在战场上被广泛使用于侦察任务。在实际复杂战场环境中,一般都有多个任务区域等待无人机去执行信息侦察任务。无人机需要规避战场中的威胁区域(这些区域中存在敌防空火力等威胁)对各个目标区域进行遍历侦察。如何快速规划出保证无人机遍历各个任务区域并安全返回原基地的最优路径是亟待解决的问题,该情形可被视为类旅行商问题(TSP, travelling salesman problem),其本质是无人机自主导航的需要。

当前国内外学者已提出了大量无人机路径规划算法。

在主要文献中提到的方法有:1)路图法,如A*、D*算法;2)概率法;3)势场法;4)智能优化算法,如蚁群算法等。需要指出的是,TSP问题因其本身的特性,目标点如果比较多,在进行穷举求解时,会出现组合爆炸的问题,所以近似求解是常被采取的方法。蚁群算法是众多智能优化算法中最经典的算法之一,其常被用于解决TSP问题,并取得了良好的计算效果。该算法具有良好的鲁棒性和稳定性,同时在与其它算法融合方面也有良好的表现,在路径规划、自主导航等问题上有着广泛应用。

A*算法是一种常见且应用广泛的搜索寻径算法,其特点在于对评价函数的定义上。在一般的有序路径搜索中,该算法总是选择综合代价值最小的路径节点作为扩展节点,优点是可实现路径的快速规划。文献[1]提出了一种使用圆

收稿日期:2022-04-26; 修回日期:2022-05-30。

基金项目:全军军事类研究生资助课题(JY2020C118)。

作者简介:汪瀚洋(1995-),男,安徽宁国人,硕士,主要从事军事运筹、进化优化和无人机任务规划方向的研究。

引用格式:汪瀚洋,卢厚清,陈亮,等.结合B样条优化的UAV多区域路径规划融合算法[J].计算机测量与控制,2022,30(9):193-200.

形邻域节点扩展法的 A* 算法改进, 提高了搜索效率, 但是没有解决生成路径贴近或者斜穿障碍物的问题, 给无人机的生存安全带来隐患。文献[2]提出了一种混合蚁群算法解决静态环境下的机器人路径规划问题, 避免了蚁群算法陷入局部最优, 但是凸显了蚁群算法收敛速度较慢的缺点。上述文章都体现出一个问题: 即单独使用蚁群算法或 A* 算法, 都不能很好解决本文情况下的 TSP 问题。蚁群算法易出现局部最优和陷入迟滞, 而 A* 算法会由于仅仅规划两两目标之间的局部最优路径, 以及组合爆炸的问题, 并不一定可以保证规划出全局最优遍历路径。本文通过蚁群算法融合改进 A* 算法来解决无人机多目标区域遍历路径规划的问题, 并结合改进三次 B 样条方法平滑规划路径。仿真结果表明, 融合算法能有效解决问题, 在算法规划速度、路径质量及安全度上有明显改善, 相比传统算法有明显优势。

1 路径规划问题描述与环境建模

1.1 问题描述

无人机多目标区域遍历问题本质是一个类 TSP 问题, 如图 1 所示, 无人机从基地出发前往各个目标区域执行侦察任务。在存在威胁区的侦察任务环境中, 无人机不仅需要规划两两目标区域之间的路径, 还需要规划到各个目的地的先后顺序, 在保证自身安全返回的同时力求全局路径的最优化, 以获取更高的侦察收益。

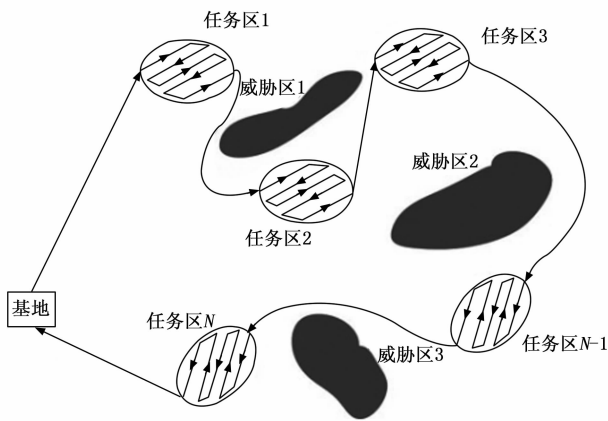


图 1 问题描述

为方便描述无人机在侦察任务中规避威胁区域并遍历各个任务区的问题, 将该问题简化描述如下:

- 1) 无人机可看作一个可操纵质点;
- 2) 执行侦察任务中, 无人机保持稳定飞行高度及速度, 环境为二维空间;^[3]
- 3) 基于栅格法建立地图模型, 全局静态威胁区域范围及任务区位置已知。无人机从给定起始基地位置出发, 规避环境中的威胁区并规划出一条遍历所有侦察任务区域并最后返回基地的安全航路。

1.2 环境建模

基于栅格地图的环境建模具有建模复杂性低、适应能

力强、易于实现与存储等优点。栅格地图将三维空间降维为二维平面, 并将整个平面划为大小相等的各个栅格。整个二维平面由栅格取值为二进制 0-1 矩阵所构成, 利用 0-1 矩阵进行判断栅格是否可通行。矩阵中数值为 0 代表无人机可以正常通行, 反之数值为 1 则为威胁区并不可通行^[4]。具体对应关系如图 2 所示。

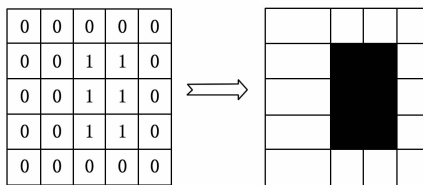


图 2 栅格设置

在实际环境中, 威胁区会呈现出不同的形状, 在栅格地图中构建威胁区时, 如按照威胁区的原形状进行建立, 如圆形、多边形甚至不规则形等, 将会导致障碍物边缘出现失真^[5], 从而导致搜索效率和路径安全性的下降。

在现有研究中, 栅格地图大多使用矩形外包原形状以避免失真情况的发生。因此, 本文中的威胁区统一使用“外包”进行处理, 即分别将原有规则或不规则形状统一进行栅格填充, 当威胁区不满足一个栅格大小, 将其填充为一个栅格。图 3 中分别为处理前后的威胁区形状, 这样只要保证规划路径不进入填充后的威胁区, 即可保证规划路径安全性。

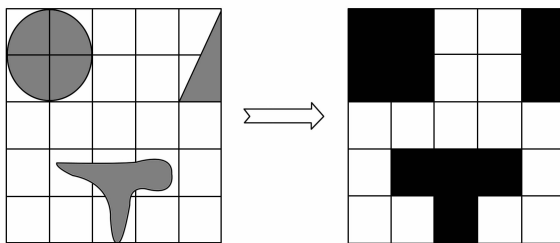


图 3 威胁区形变

无人机在栅格环境中运动, 图 4 显示了无人机的运动情况。各箭头所指方向为无人机初始可扩展方向, 即初始 8 邻域节点扩展。现在假设无人机在二维平面区域 S 内移动, 设 S 的左底角为原点 O, 以水平方向为 x 轴, 垂直方向为 y 轴, 建立平面直角坐标系 xOy, 如图 5 所示。对于该 25 个栅格集合, 其中任一栅格都有确定坐标和对应的序号, 则无人机在栅格地图中的地图坐标和序号位置对应关系如下:

$$\begin{cases} x_i = ((i-1) \bmod n_x) + 1 \\ y_i = n_y - \text{ceil}(i/n_x) + 1 \end{cases} \quad (1)$$

其中: mod 表示取余运算; ceil 表示求整运算; n_x 、 n_y 表示每行每列的栅格数; x_i 、 y_i 表示栅格 i 的中心点横坐标和纵坐标^[6]。

2 算法改进及融合

2.1 蚁群算法

蚂蚁在运动中, 会在其所经过的路径上留下一种被称

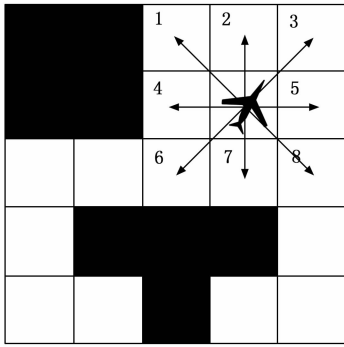


图 4 运动环境

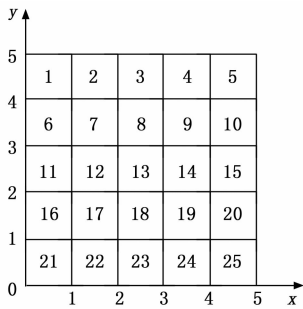


图 5 栅格坐标

为信息素的物质, 这种物质是各蚂蚁之间进行信息交流的载体。蚂蚁在运动中会感知到这种物质, 并循迹此物质进行运动, 同时爬行过程中同样释放信息素。某一路径上的信息素浓度越高, 蚂蚁将以更高的概率选择此路径进行运动, 使得该路径上的信息素浓度继续增高, 至此便出现一种信息正反馈现象。某路径上运动过的蚂蚁越多, 后来者选择该路径进行运动的可能性就越大^[7]。

蚁群算法的基本原理, 即基于蚂蚁根据路径上同类释放的信息素浓度进行路径选择, 距离短的路径在理论上会更多地被选择, 经过迭代和避免局部优化后, 可得到全局最优路径。蚁群算法可以有效求解 TSP 问题, 在本任务环境中, 算法数学模型如下:

在 t_0 初始时刻, 初始化各参数, 将 m 只蚂蚁放置于 n 个目标区域中。在 t 时刻区域 i 和 j 之间的路径 (长度为 d_{ij}) 上的信息素浓度为 $\tau_{ij}(t)$, 则有 $\tau_{ij}(t_0) = 0$ 。根据路径上信息素浓度, 蚂蚁将按轮盘赌的方法, 依照概率选择下一个目标区。以符号 $P_{ij}^k(t)$ 定义在 t 时刻蚂蚁 k 位于目标区 i 时, 选择目标区 j 的概率为:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in \alpha_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta}, & j \in \alpha_k \\ 0, & \text{其他} \end{cases} \quad (2)$$

其中: α_k 表示待侦查区域的集合, α 表示信息度启发因子, β 为启发函数的重要程度因子, $\eta_{ij} = 1/d_{ij}$ 是从目标区 i 到目标区 j 的启发式因子。

在所有蚂蚁选择完路径后, 路径上的信息素浓度更

新为:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \end{cases} \quad (3)$$

其中: $0 < \rho < 1$, 表示信息素挥发因子, $\Delta\tau_{ij}^k$ 为蚂蚁 k 在目标区 i 与目标区 j 之间路径上释放的信息素浓度, 则 $\Delta\tau_{ij}$ 表示 m 只蚂蚁在目标区 i 与目标区 j 之间路径上信息素浓度总和。

本文采用蚁周 (Ant-cycle) 模型, 利用全局信息更新路径上的信息素量, 则 $\Delta\tau_{ij}^k$ 信息素更新模型为:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & (i,j) \in path^k \\ 0, & (i,j) \notin path^k \end{cases} \quad (4)$$

其中: Q 为信息素常量, 表示 1 次循环中, 蚂蚁 k 所释放的信息素总量。Path^k 为蚂蚁 k 的路径集合, L_k 表示第 k 只蚂蚁本次循环中途径的路程总长^[8]。

所有蚂蚁完成一次路径选择后, 根据路径上的信息素浓度记录本次最佳路径, 更新路径信息量。经过指定次数循环后, 输出最终路径的优化结果。

2.2 A* 算法

A* 算法是一种启发式搜索算法, 其常被用来解决无人机路径规划问题。其是在 Dijkstra 算法的基础上引入评价函数机制, 将路径搜索代价进行综合考虑, 能有效解决全局静态环境下最短路径的搜索问题。

A* 算法通过评价函数来确定到目标节点搜索方向, 路径由当前节点到目标节点的评价函数 $f(n)$ 的最小值来确定。传统 A* 算法在进行路径规划时会设置两个列表, 一个是 Open list 表, 其用来保存准备搜索的节点, 另一个是 Closed list 表, 用来存放已经被搜索到的截至目前最小路径搜索代价的点。探索过程中, 先从 Open list 中找到路径搜索代价最小的点设为当前节点, 将其放入 Closed list, 然后对其进行扩展搜索, 将扩展搜索后得到的节点更新到 Open list 中, 再从 Open list 选取搜索代价最小的点设为当前节点, 重复过程, 直到找到目标点。

A* 算法的评估函数为:

$$f(n) = g(n) + h(n) \quad (5)$$

其中: n 为栅格中某节点; $g(n)$ 为起点到该点的最短路径的代价函数; $h(n)$ 为该点达到目标点最短路径代价函数。

在栅格地图中, 评价函数的选择在一般情况下根据移动体可扩展的方向而定。如果移动体只能向 4 个方向扩展, 这时曼哈顿距离是计算评价函数的最佳选择; 如果移动体可以向 8 个方向扩展, 这时评价函数使用切比雪夫距离计算更加合适; 如果移动体可以在任意方向移动, 则评价函数选择欧几里得距离计算比较有优越性。具体如图 6 所示。

考虑到无人机实际运行轨迹的任意性^[9], 本文中 A* 算法的评估函数中的 $h(n)$ 采用欧氏距离为代价值进行计算。设当前节点坐标为 (x_n, y_n) , 目标点坐标为 (x_g, y_g) , 欧

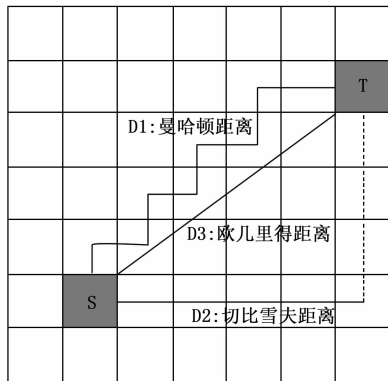


图 6 评价函数计算

几里得距离表示两坐标的最短距离，其公式为：

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \quad (6)$$

2.3 改进 A* 算法

A* 算法的规划路径存在斜穿威胁区栅格顶点、斜穿两个威胁区栅格相接点的现象，这是由于该路径的扩展代价最低，且符合节点扩展规责造成的，容易导致路径发生进入威胁区的风险，路径如图 7 所示。

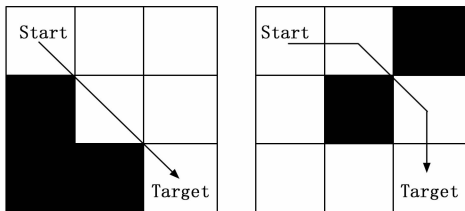


图 7 非法路径

针对该问题，本文对节点选择规责进行重新定义：


节点1	I 节点2	节点3
II 节点4		II 节点5
节点6	节点7 I	节点8

图 8 节点分类

1) 当威胁区位置在第 I 类节点时。其中，当威胁区位置在节点 2 时，则不选择节点 1、3；威胁区位置在节点 7 时，则不选择节点 6、8。

2) 当威胁区位置在第 II 类节点时。其中，当威胁区位置在节点 4 时，则不选择节点 1、6；威胁区位置在节点 5 时，则不选择子节点 3、8。

经过初步试验，上述优化路径如图 9 所示。改进后的路径可以有效避免无人机进入威胁区。

2.4 蚁群-A* 算法融合

蚁群算法在解决 TSP 问题时，是基于目标区的坐标来

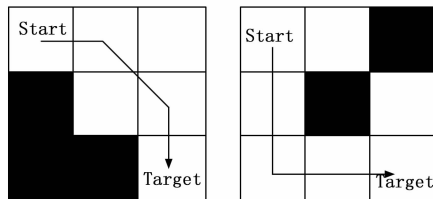


图 9 优化路径

计算的，即此时各目标区之间的距离是已知的。在本文栅格地图中，由于各任务区之间距离未确定、环境中存在威胁区不可通过等原因，蚁群算法在计算各任务区之间的距离时，需要反复迭代，且因为蚁群算法本身计算时间长、启发效果差和易陷入局部最优等原因，会导致全局规划路径耗时较长^[10]。

A* 算法同样不能很好地单独解决 TSP 问题。A* 算法通过找到任意两个目标区之间的最短距离的方法搜索全局路径，会导致结果不一定是全局范围中的最优路径。另外，不同于蚁群算法解决 TSP 问题属于近似计算的范畴，A* 算法属于精确计算，当节点数很少时，A* 算法进行路径穷举可以解决，但实际问题中，节点数往往很多。例如，对于一个仅有 16 个城市的旅行商问题，若采用 A* 算法进行穷举来求解问题的最优解，可行解共有 $15! / 2 = 653\ 837\ 184\ 000$ 个^[11]。在 1993 年，使用当时的工作站用穷举法求解此问题需时 92 小时。即使现在计算机速度快，面对复杂的问题，效率仍然不够。这就是所谓的“组合爆炸”，即所需要处理的数据因指数级的增长而变得难以处理。所以科学家逐步寻找近似算法或者启发式算法，目的是在合理的时间范围内找到可接受的最优解。

总之，蚁群算法的优势在于解决 TSP 问题的全局路径规划，而 A* 算法可以很好地解决 TSP 问题中两两目标区之间的局部路径规划^[12]。为了提高算法可行性和运行效率，本文考虑使用融合算法，即使用改进 A* 算法快速计算各任务区之间距离^[13]，然后使用蚁群算法规划全局路径^[14]。

3 基于改进三次 B 样条法的路径平滑处理

3.1 平滑策略

在栅格地图中，规划路径由栅格中心点连线组成，这导致 A* 算法生成的路径存在转折拐点多的问题，客观上给无人机运动增加了多余路径长度，且不符合无人机的动力学原理（无人机存在最大转弯角等）。本文选择改进的三次 B 样条方法对规划路径进行平滑处理，即先使用双向插点法处理路径，再使用三次 B 样条法平滑路径。

3.2 双向插点法

双向插点法核心思想是在 Floyd 算法的基础上，运用双向优化的理念，删除冗余节点，设置安全距离，以最大程度减少路径拐点。在优化中，设置安全距离，通过比较威胁区到路径的垂直距离与设置的安全距离的关系来判断路径的安全性。如图 10，设路径节点 S 坐标为 (x_s, y_s) ，T 坐标为 (x_t, y_t) 。威胁区几何中点 O 坐标 (x_o, y_o) ；设栅

格边长为 d ; 距离 OA 为 O 到线段 ST 之间的垂线距离, 记为 d_2 ; 线段 OB 为 O 到线段 ST 纵向距离, 记为 l ; 夹角 α 为线段 ST 与水平方向夹角; 记外接圆半径为 r 。

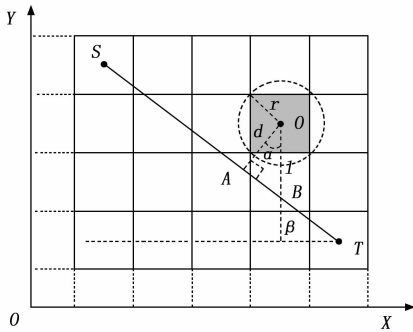


图 10 安全距离设置

则有:

$$\begin{cases} l = |y_o - (\frac{y_s - y_t}{x_s - x_t}(x_o - x_s) + y_s)| \\ \alpha = |\arctan \frac{y_t - y_s}{x_t - x_s}| \\ d_2 = l \cos \alpha \\ r = \frac{d}{\sqrt{2}} \end{cases} \quad (7)$$

设置安全距离为 D (D 为设置的以 O 为起点的线段长度, $D > r$), 以保证处理后的路径不进入威胁区, 即将 D 与 d_2 进行比较: 如 $d_2 \leq D$, 则该路径不可被选择。如 $d_2 > D$, 则该路径可被选择^[15]。

双向插点法具体步骤如图 11 所示。

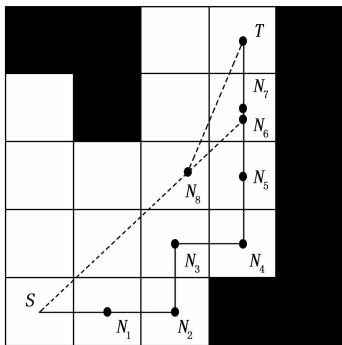


图 11 双向插点法

步骤 1: 对路径中同一线段上的冗余点进行删除, 只保持路径转折前的起点 S 、下一个目标点 T 。 $S \rightarrow N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow N_5 \rightarrow N_7 \rightarrow T$ 删除冗余点后的路径为 $S \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow T$ 。

步骤 2: 从路径起始点 S 起算, 在保留的 N_i 、 N_j 之间每步取一节点, 步长记为 q , 判断取得节点与上一节点之间是否存在威胁区域, 并通过公式 (7) 计算 d_2 , 判断 d_2 与 D 的大小; 如果威胁区域在安全距离之外, 则选择该节点为路径节点; 否则不予选择。如图所示, 处理之后的路径

为 $S \rightarrow N_6 \rightarrow T$ 。

步骤 3: 从目标点 T 方向出发, 反方向进行取点判断, 如图所示, 处理后得到 $S \rightarrow N_8 \rightarrow T$ 。

3.3 三次 B 样条曲线优化

B 样条曲线常用于路径规划的平滑, 其具有针对局部路径修改的特点, 并使用逼近多边形的方法从而获得曲线优化。结合前述双向插点法, B 样条曲线优化的路径会更为平滑^[17]。B 样条曲线实际是贝塞尔曲线的一种特例, 与贝塞尔曲线相比, B 样条曲线在保留贝塞尔曲线全部优点的同时, 又克服了其缺乏局部性质、连续性较差等缺点, 是路径平滑常用的数学工具。其核心思想是通过控制点对路径轨迹进行 n 次 B 样条曲线处理。

广义的样条函数定义为:

给定一组平面上顶点 (x_i, y_i) ($i=0, 1, \dots, n$), 并设在区间 $[a, b]$ 上的 $\Delta: a=x_0 < x_1 < \dots < x_{n-1} < x_n=b$, 那么在 $[a, b]$ 上的一个函数 $S(x)$ 称为 K 阶连续样条函数, 如果它满足下面两个条件:

1) 在每个小区间 $[x_{i-1}, x_i]$ ($i=1, 2, \dots, n$) 内, $S(x)$ 是具有 K 阶或 K 阶以上连续函数。

2) 在 x_i ($i=1, 2, \dots, n-1$) 处成立。

则对于三次样条函数, 现在假设在区间 $[a, b]$ 上给定一个分割 $\Delta: a=x_0 < x_1 < \dots < x_{n-1} < x_n=b$, 在 $[a, b]$ 上的一个函数 $S(x)$ 称为三次样条函数, 则满足下列条件:

1) 在每个小区间 $[x_{i-1}, x_i]$ ($i=1, 2, \dots, n$) 内, $S(x)$ 是三次多项式函数。

2) 在 x_i ($i=1, 2, \dots, n-1$) 处成立:

$$S^{(k)}(x_i - 0) = S^{(k)}(x_i + 0), k = 0, 1, 2 \quad (8)$$

即小区间上的三次多项式函数, 在拼接点处 x_i 具有二阶连续拼接。

3) 满足条件 $y_i = S(x_i), i=0, 1, \dots, n$

下面对 B 样条曲线进行定义, 给定 $m+n+1$ 个平面或空间顶点 (即控制点) P_i ($i=0, 1, \dots, m+n$), 称为 n 次参数曲线段:

$$P_{k,n}(t) = \sum_{i=0}^n P_{i+k} G_{i,n}(t), t \in [0, 1] \quad (9)$$

其顶点 P_i 所组成的多边形称为 B 样条曲线的特征多边形。其中, 基函数 $G_{i,n}(t)$ 定义为:

$$G_{i,n}(x) = \frac{1}{n!} \sum_{j=0}^{n-i} (-1)^j C_{n+1}^j (t+n-i-j)^n \quad (10)$$

其中: $t \in [0, 1], i=0, 1, \dots, n$ 。

取 $n=3$, 则有三次 B 样条曲线的基函数如下:

$$\begin{cases} G_{0,3}(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1) \\ G_{1,3}(t) = \frac{1}{6}(3t^3 - 6t^2 + 4) \\ G_{2,3}(t) = \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) \\ G_{3,3}(t) = \frac{1}{6}t^3 \end{cases}, t \in [0, 1] \quad (11)$$

将公式 (11) 代入式 (9) 中可得:

$$P(x) = \frac{[(-P_0 + 3P_1 - 3P_2 + P_3)x^3 + (3P_0 - 6P_1 + 3P_2)x^2 + (-3P_0 + 3P_2)x + (P_0 + 4P_1 + P_2)]}{6} \quad (12)$$

将公式 (12) 以矩阵形式表示, 则可得三次 B 样条曲线段 $P_{0,3}(t)$ 为:

$$P_{0,3}(t) = \frac{1}{6} [1 \quad t \quad t^2 \quad t^3] \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (13)$$

P_i 顶点位置定义为:

$$\begin{cases} P_{0,3}(0) = \frac{1}{6}(P_0 + 4P_1 + P_2) = \frac{1}{3}\left(\frac{P_0 + P_2}{2}\right) + \frac{2}{3}P_1 \\ P_{0,3}(1) = \frac{1}{6}(P_1 + 4P_2 + P_3) = \frac{1}{3}\left(\frac{P_1 + P_3}{2}\right) + \frac{2}{3}P_2 \end{cases} \quad (14)$$

$$\begin{cases} P_{0,3}'(0) = \frac{1}{2}(P_2 - P_0) \\ P_{0,3}'(1) = \frac{1}{2}(P_3 - P_1) \\ P_{0,3}''(0) = P_2 - 2P_1 + P_0 \\ P_{0,3}''(1) = P_3 - 2P_2 + P_1 \end{cases} \quad (15)$$

利用上述公式可得满足三次 B 样条的控制点及优化曲线。平滑结果如图 12 所示, 图中虚线即为处理后的实际轨迹。

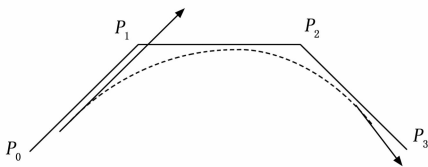


图 12 三次 B 样条曲线优化

4 算法流程

本文融合算法结构如图 13, 步骤如下。

步骤 1: 初始化栅格地图, 并设定 n 个目标区域。

步骤 2: 基于改进 A* 算法进行各目标区之间路径规划, 得到任意两个目标区之间的最短距离。

步骤 3: 初始化蚁群算法各参数。令时间 $t=0$ 和循环次数 $N_c=0$, 设置最大循环次数 N_{max} , 将 m 个蚂蚁置于 n 个目标区上, 令有向图上每条边 (i, j) 的初始信息素 $\tau_{ij}(t) = c$, 其中 c 表示常数, 且初始时刻 $\Delta\tau_{ij}(0) = 0$ 。

步骤 4: 循环次数 $N_c = N_c + 1$ 。

步骤 5: 蚂蚁禁忌表索引号 $k=1$ 。

步骤 6: 蚂蚁个体根据状态转移概率公式 (2) 计算的概率选择下一个目标区 j 并前进, $j \in a_k$ 。

步骤 7: 修改禁忌表指针, 即选择之后蚂蚁移动到新的

目标区, 并把该目标区移动该蚂蚁个体的禁忌表中^[18]。

步骤 8: 所有蚂蚁完成移动, 记录本次最佳路线。

步骤 9: 根据公式 (3) 和公 (4) 更新每条路径上的信息素。

步骤 10: 若满足结束条件, 即如果循环次数 $N_c \geq G$, 则循环结束并得到最佳遍历路径; 否则清空禁忌表转到步骤 4。

步骤 11: 使用三次 B 样条与插点法相结合的方法进行路径平滑, 最后输出最佳路径。

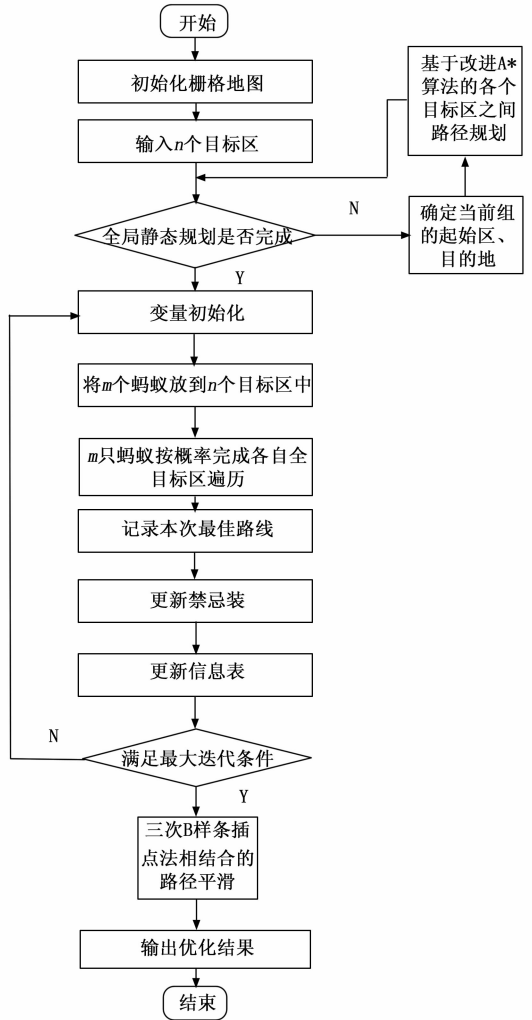


图 13 算法流程

5 仿真实验与分析

为验证算法的性能, 拟在 MATLAB2018b 环境中进行仿真实验。仿真选择基本蚁群算法 (ACO)、蚁群-A* 算法 (ACO-A*)、蚁群-A* 改进算法 (ACO-IA*) 和本文融合算法 (ACO-MA*)。首先建立 30×30 km 的栅格地图, 并在地图中设立 20 个目标区域进行实验, 目标区域如图中浅色区域所示。参数初始化设置如下: 蚂蚁数量 $m=30$, 目标区域数量 $n=20$, 最大迭代次数 200, 信息素启发因子 $\alpha=1$, 启发函数重要度因子 $\beta=5$, $\rho=0.3$, $Q=50$, $d=$

1 km, $D=0.8$ km, $q=0.1$ km。仿真结果如图 14 所示。

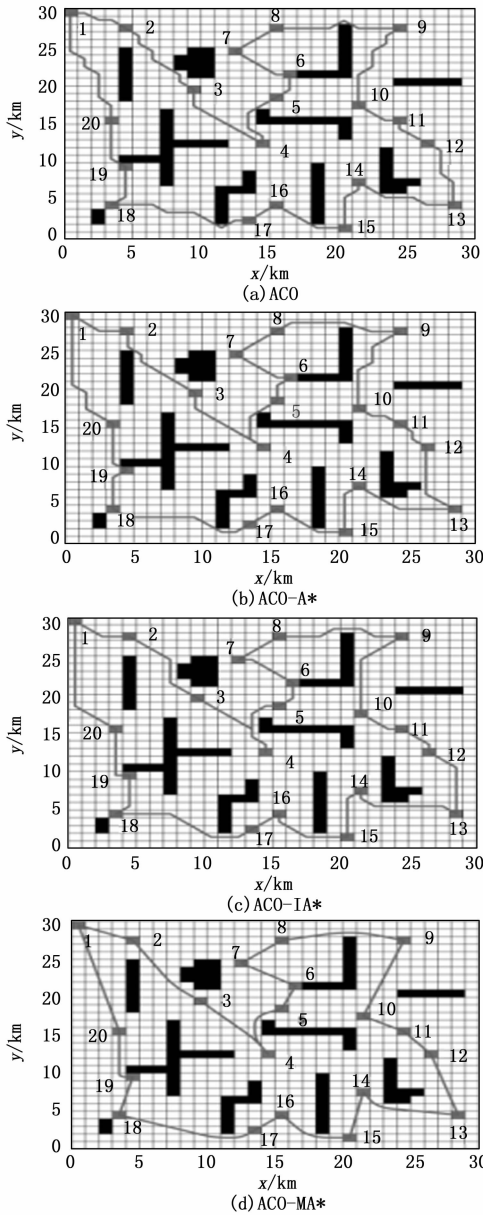


图 14 实验对比情况

从表 1 中可以看出, 基本蚁群算法 (ACO)、蚁群-A* 算法 (ACO-A*)、蚁群-A* 改进算法 (ACO-IA*) 和本文融合算法 (ACO-MA*) 都能顺利获得路径。但对比而言, ACO 算法处理时间最长, 且路径转角和路径靠近威胁区次数最多, 在实际应用中, 无人机将消耗更多燃料, 且无人机生存面临更大威胁。ACO-A* 算法相比 ACO 算法路径长度增加了 1 km, 但转角次数减少了 12 次, 占比 19.3%。处理时间显著减少了 24.33 秒, 占比 92.7%。路径靠近威胁区次数减少了 2 次, 占比 22.2%。ACO-IA* 算法生成的路径相比 ACO-A* 算法生成的路径, 路径长度、转角次数和处理时间几乎相同, 但有效避免了路径靠近威胁区, 显著增加了无人机路径安全性^[19]。而 ACO-MA* 算

法相比 ACO 算法路径长度减少了 6.13 km, 占比 4.2%。转角次数减少了 43 次, 占比 69.4%。处理时间减少了 24.3 秒, 占比 92.9%, 同时有效保证了路径的安全性。且 ACO-MA* 算法相比于 ACO-A* 算法和 ACO-IA* 算法, 除了处理时间有略微增加, 各项数据指标均有明显优化。考虑到本地图仅是 30×30 km 的栅格地图, 如果地图区域更大, 融合算法的性能将更有显著提升。如图 13 所示, 融合算法相比其他算法也更有良好的收敛性。

表 1 实验数据对比

算法	路径长度/km	转角次数/次	处理时间/s	路径靠近威胁区次数/次
ACO	144.12	62	26.25	9
ACO-A*	145.12	50	1.92	7
ACO-IA*	146.47	50	1.93	0
ACO-MA*	137.99	19	1.95	0

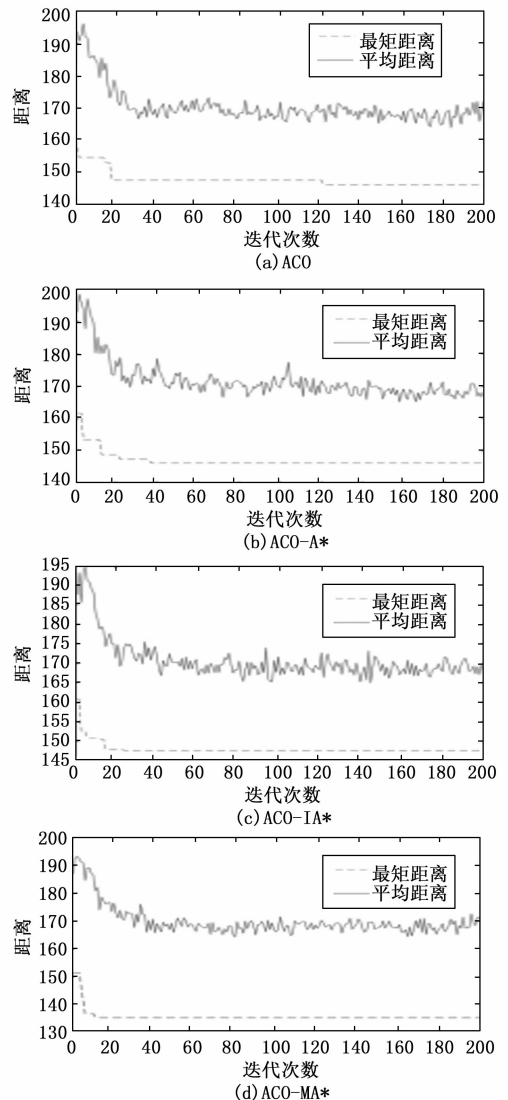


图 15 收敛性对比情况

6 结束语

本文针对无人机多目标区域遍历侦察问题,提出一种高效的遍历路径规划融合算法,对无人机多任务区遍历路径进行快速规划。针对传统的蚁群算法和 A* 算法在解决 TSP 路径规划问题上均有缺陷的情况,通过两种算法的改进与融合,使新算法相比于单个或其他融合算法有明显优势,证明了该算法的有效性和实用性。预计本文的研究成果将帮助智能无人机或决策者在复杂战场环境中快速和高效地作出决策,提高无人机生存能力和侦察效益,更好地发挥无人作战力量的优势^[20]。

参考文献:

- [1] 杨明亮, 李 宁. 改进 A* 算法的机器人路径规划 [J]. 机械科学与技术, 2021 (5): 1-7.
- [2] 李二超, 齐款款. B 样条曲线融合蚁群算法的机器人路径规划 [J]. 计算机应用, 2021 (12): 3558-3564.
- [3] 李孟锡, 何博侠, 周 昊. 基于 A~* 和蚁群算法的机器人多目标路径规划方法 [J]. 机械与电子, 2021 (6): 61-65, 69.
- [4] 杨芳清, 刘吉成. 融合改进 A~* 算法与动态窗口法的机器人路径规划 [J]. 工业控制计算机, 2021 (5): 106-108, 112.
- [5] 刘泽宇, 招启军, 张夏阳, 等. 倾转机翼无人倾转旋翼机飞行动力学稳定性分析 [J]. 飞行力学, 2021 (3): 1-7.
- [6] 王钦禾, 尹永鑫, 戴 丽, 等. 基于 Voronoi 图和蚁群算法的无人机航迹规划 [J]. 导航定位与授时, 2021 (2): 66-73.
- [7] 敖邦乾, 杨 莎, 叶振环. 改进蚁群算法水面无人艇平滑路径规划 [J]. 控制理论与应用, 2021 (7): 1006-1014.
- [8] 王红君, 叶 荣, 赵 辉, 等. 基于改进的烟花-蚁群算法和 B 样条曲线的农业机器人路径规划 [J]. 科学技术与工程, 2021 (7): 2730-2736.
- [9] 夏阳升, 石建迈, 陈 超, 等. 车机协同多区域覆盖侦察路径规划方法 [J]. 指挥与控制学报, 2020 (4): 372-380.
- [10] 杨海清, 芦 斌. 基于改进蚁群算法的水下无人机路径规划研究 [J]. 计算机测量与控制, 2020 (10): 216-220.
- [11] 余文凯, 章 政, 金 震, 等. 多目的地自主移动机器人的路径规划 [J]. 计算机工程与设计, 2020 (10): 2934-2941.
- [12] 段书用, 王启帆, 韩 旭, 等. 具有确保安全距离的 A~* 路径优化方法 [J]. 机械工程学报, 2020 (18): 205-215.
- [13] 余文凯, 章 政, 付雪画, 等. 基于地图预处理及改进 A~* 算法的路径规划 [J]. 高技术通讯, 2020 (4): 383-390.
- [14] 胡立坤, 王帅军, 吕智林, 等. 移动机器人多目标搜寻的 D~* 蚁群融合算法 [J]. 小型微型计算机系统, 2020 (3): 471-476.
- [15] 刘 昂, 蒋 近, 许迪文. 基于 A~* 和鸽群算法的快递无人机航路规划 [J]. 飞行力学, 2020 (3): 34-40, 51.
- [16] 向庭立, 王红军, 史英春. 区域覆盖的多机协同探测任务分配策略 [J]. 空军工程大学学报 (自然科学版), 2019 (6): 33-38, 71.
- [17] 曾德全, 余卓平, 张培志, 等. 三次 B 样条曲线的无人车避障轨迹规划 [J]. 同济大学学报 (自然科学版), 2019 (S1): 159-163.
- [18] 冯国强, 赵晓林, 高关根, 等. 基于 A~* 蚁群算法的无人机航路规划 [J]. 飞行力学, 2018 (5): 49-52, 57.
- [19] 任玉洁, 付丽霞, 张 勇, 等. 拓展搜索邻域的平滑 A* 算法机器人路径规划 [J]. 电子科技, 2018 (5): 33-36, 43.
- [20] 程 琪, 荆 涛, 于志游. 利用三次样条改进蚁群算法的无人机航路规划 [J]. 计算机测量与控制, 2016 (8): 272-274, 293.
- [21] 赵庶旭, 罗 庆, 王小龙. 基于改进 Mask R-CNN 的牙齿识别与分割 [J]. 中国医学物理学杂志, 2021, 38 (10): 1229-1236.
- [22] 王 涛. 基于卷积神经网络的零件回收抓取研究 [D]. 太原: 中北大学, 2021.
- [23] BOLYA D, ZHOU C, XIAO F, et al. YOLACT: Real-time instancesegmentation [J]. IEEE CVF International Conference on Computer Vision, 2019: 9157-9166.
- [24] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2016: 779-788.
- [25] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2017: 7263-7271.
- [26] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

(上接第 192 页)

- [16] 薛 腾, 刘文海, 潘震宇, 等. 基于视觉感知和触觉先验知识学习的机器人稳定抓取 [J]. 机器人, 2021, 43 (1): 1-8.
- [17] LI Y, QI H, DAI J, et al. Fully convolutional instance-aware semantic segmentation [C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2359-2367.
- [18] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks [J]. Advances in neural information processing systems, 2015, 28: 91-99.
- [19] HE K, GKIOXARI G, DOLLÁR P, et al. Mask r-cnn [C] // Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [20] 杨 攀, 郑积仕, 冯芝清, 等. 基于 Mask R-CNN 的密集木材检测分割方法 [J]. 林业工程学报, 2022, 7 (2): 135-142.
- [21] 赵庶旭, 罗 庆, 王小龙. 基于改进 Mask R-CNN 的牙齿识