1.算法基本原理与发展历程

在传统的蚁群算法中，下一个节点的概率选择采用轮盘赌方法

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{(\tau_{ij}(t))^{\alpha}(\eta_{ij}(t))^{\beta}}{\sum_{s\in allow_k}(\tau_{ij}(t))^{\alpha}(\eta_{ij}(t))^{\beta}}, s\in allow_k \\ 0, s\notin allow_k\end{cases} \tag{1}$$

$$\eta_{ij}(t)=\frac{1}{d_{ij}} \tag{2}$$

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \tag{3}$$

其中，τ_{ij}为路径 i 到 j 的信息素，η_{ij} 为路径 i 到 j 的启发式信息。α是决定信息素浓度对路径的相对影响的刺激因子。β是决定启发式信息相对影响的可见性因素。d_{ij}是节点 i 到节点 j 的距离(x_i,y_i)和(x_j,y_j)为网格 i 和网格 j 的坐标。$allow_k$是蚂蚁在网格 i 时下一步可以选择的网格集合(换句话说，它们是除了障碍网格和禁忌网格外的网格 i 周围的网格)。

更新策略: 传统蚁群算法通过轮盘赌轮法确定下一个节点，并不断重复，直到得到目标点。每次迭代完成后，信息素路径根据路径规划的长度进行更新。使用式(4)更新每个周期结束时每条路径上的信息素数量:

$$\begin{cases}\tau_{ij}=(1-\rho)\tau_{ij}+\Delta\tau_{ij} \\ \Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad 0<\rho<1\end{cases} \tag{4}$$

其中 m 为蚂蚁的数量。ρ是信息素的蒸发速率。$\Delta\tau_{ij}^{k}$表示蚂蚁 k 在网格 i 到网格 j 路径上留下的信息素值。本文采用循环系统模型，$\Delta\tau_{ij}^{k}$定义如下:

$$\Delta\tau_{ij}^{k}(k)=\begin{cases}\dfrac{Q_1}{L_k(t)} & \text{蚂蚁k经过了路径(i, j)} \\ 0 \quad otherwise\end{cases} \tag{5}$$

其中Q_1是一个常数。$L_k(t)$是蚂蚁 k 寻找的路径的长度。

然而，由于概率迁移的随机性和信息素强度更新的不当，传统蚁群算法容

易陷入局部最优，收敛性较差。为此，许多学者提出了各种改进的方法来解决信息素更新和路径搜索策略的问题。蚁群系统(ACS)算法[1]，通过更新每一代最优蚂蚁路径上的信息素来加快蚁群算法的收敛速度。通过自适应改变挥发速率和调整信息素更新公式，提高了蚁群的搜索能力和算法的收敛速度。

传统蚁群算法由于缺乏初始信息素和邻接网格之间的启发式值差异不明显，通常需要更长的搜索时间，导致收敛速度慢。当网格模型比较复杂时，机器人可能会陷入死锁状态，无法移动到周围的网格中。在网格图中，传统的路径规划可能有更多的弯曲次数和较大的累积弯曲角度。针对以上问题，本文做了以下改进。

2.算法实现步骤与说明

2.1 基于 A*算法的启发式信息

A*算法[2]是求解静态路网中最短路径最有效的直接搜索方法。它是在Dijkstra 算法的基础上发展起来的，避免了盲目搜索，提高了搜索效率。本文利用 A*算法作为路径搜索的启发式信息，提高了算法的收敛速度，从而获得更好的路径。同时在启发式信息中加入弯曲抑制算子，减少弯曲次数和累积弯曲角度。

A*算法的启发式代价用估计函数表示，估计函数方程 $f(n)$ 如下:

$$f(n)=g(n)+h(n) \tag{6}$$

$$h(n)=((n_x-g_x)^2+(n_y-g_y)^2)^{1/2} \tag{7}$$

$$g(n)=((n_x-s_x)^2+(n_y-s_y)^2)^{1/2} \tag{8}$$

其中 $g(n)$ 为从源节点到当前节点的最小代价。$h(n)$ 为从当前节点到目标节点的最小代价。n_x 和 n_y 是当前节点 n 的坐标。g_x 和 g_y 为目标节点 g 的坐标，s_x 为初始节点 s 的坐标，s_y 为初始节点 s 的坐标。

利用 A*算法的估计函数作为启发式信息在蚁群算法中搜索全局最优路径，并在蚁群算法的启发式值中添加弯曲抑制算子，以减少弯曲次数和较大的累积转弯角度。改进的启发式信息公式如下:

$$\eta_{ij}(t)=\frac{Q_2}{h(n)+g(n)+\cos t(bend)} \tag{9}$$

$$\cos t(bend)=\varphi\cdot turn+\psi\cdot thita \tag{10}$$

其中Q_2是一个大于 1 的常数。$\cos t(bend)$是一个弯曲抑制因子。*turn*是从节点 n−1(前一个节点)到节点 n+1(下一个节点)的路径弯折数。*thita*即节点 n−1 到节点 n(当前节点)的线段与节点 n 到节点 n + 1 的线段之间的夹角。其中 ϕ 是将转弯次数转换为网格长度的系数。ψ是角到栅格长度的转换系数。

2.2 最大最小蚂蚁系统

传统的蚁群算法可能会导致早熟和早熟现象，需要改进算法来解决这些问题。最大最小蚂蚁系统(MMAS)[1]可以很好地解决这些问题。

(1)信息素跟踪更新。传统蚁群算法在每次迭代试验后，将信息素提交到更新历史中。而在 MMAS 中，迭代完成后只更新最优网络的路径信息素。据此，提出了改进后的信息素轨迹更新规则

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}^{best} \tag{11}$$

$$\Delta\tau_{ij}^{best}=\frac{Q_1}{L^{best}}+\frac{Q_3}{C_{turn}^{best}} \tag{12}$$

$$C_{turn}^{best}=\omega_1 Cals(l)+\omega_2 Turns(l) \tag{13}$$

$$\omega_1=\frac{V_{uav}}{W_{uav}} \tag{14}$$

$$\omega_2=V_{uav}\times t_a \tag{15}$$

其中 Q3 是一个大于 1 的常数。L^{best}表示算法当前找到的最短路径。$Cals(l)$表示在最优路径上转弯的所有角度之和。$Turns(l)$是最优路径上的转弯数之和。ω_1和ω_2可以分别将转弯角度和转弯次数转换为网格长度。V_{uav}表示无人机的恒定速度。W_{uav}表示无人机转弯时的角速度。t_a为无人机转动一次加速和减速的时间。

(2)信息素限制。为了避免传统蚁群算法因信息素积累而陷入局部最优解而失去进一步搜索空间能力的情况，MMAS 的信息素被限制在上下限。$[Tau_{min},Tau_{max}]$。这个公式是:

$$Tau=\begin{cases}Tau_{\min},Tau\le Tau_{\min}\\ Tau,Tau_{\min}<Tau\le Tau_{\max}\\ Tau_{\max},Tau>Tau_{\max}\end{cases} \tag{16}$$

2.3 解决死锁问题

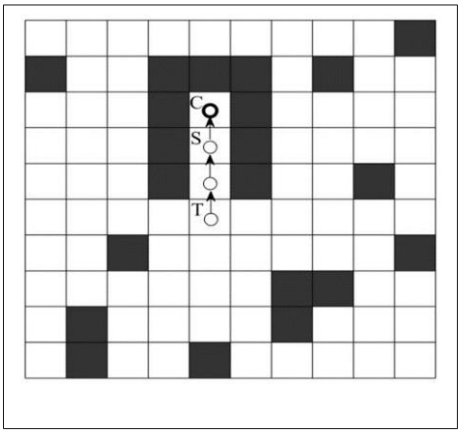

图 1 死锁状态图

当无人机环境更加复杂（特别是蚂蚁进入凹陷障碍的环境）时，由于禁忌表的存在，蚂蚁可能会陷入死锁状态，没有下一个网格移动。如图 1 所示，当蚂蚁从网格 T 传播到网格 S 时，下一个可选网格 C。此时，蚂蚁处于死锁状态，无法移动到周围的网格。

对于死锁问题，[3]采取了死亡策略，从蚂蚁群落中删除了陷入死锁状态的蚂蚁，并没有更新全球信息素。然而，当更多的蚂蚁陷入僵局状态时，能够达到目标的蚂蚁数量会显著减少，这导致解决方案的多样性减少，不利于为蚂蚁寻找最佳路径。本文认为，改进措施是蚂蚁在陷入死锁状态时采用缩回机制。已经进入网格的蚂蚁被困在死锁状态，改进的策略允许蚂蚁向后滚动一步并更新禁忌表信息。如果蚂蚁仍然陷入死锁状态，蚂蚁将继续回滚，直到网格 T。这时，蚂蚁逃离了死锁区。

2.4 算法步骤

Step1:采用网格法对工作环境建模，给出了移动机器人的起点起点和目标点目标。

Step2:初始化 ant 系统。设置蚂蚁数量 m，参数 α 决定信息素路径的相对影响，参数 β 决定启发式值，全局信息素挥发系数 ρ，信息素强度 Q1 等相关参数。

Step3:更新禁忌表。将蚂蚁 k (k = 1,2，•••，m)放置在当前节点上，并将当前节点加入到相应的禁忌表中。

Step4:进程死锁。根据禁忌表，判断蚂蚁是否陷入了死锁状态。如果蚂蚁处于死锁状态，则采用撤回机制，将死锁节点添加到禁忌表中。相反，它将判断蚂蚁是否到达目标点。如果蚂蚁到达目标点，它将转向 Step6，否则它将转向 Step5。

Step5:选择下一个网格。根据公式(6)计算启发式函数，根据公式(2)计算概率函数，最后使用轮盘赌方法选择下一个可行网格。如果蚂蚁到达目标网格，它将转到 Step6，否则将转到 Step3。

Step6:如果蚂蚁到达目标节点，则重复 Step3，直到每只蚂蚁在其迭代过程中完成搜索目标，然后转到 Step7。

Step7:更新信息素。每次迭代后，如果迭代次数满足不等式 N≤Nmax，则更新路径信息素，判断是否满足收敛条件。如果满足收敛条件，则退出。如果它不满足，它将转到 Step3。如果迭代次数满足不等式 N > Nmax，则不再计算。只要满足结束条件，就会输出最终结果。

3.数值仿真分析

3.1 地图模型

利用网格模型建立工作环境，将机器人的工作空间划分为 N * N 个正方形。如图 1 所示，黑色网格表示为障碍，白色网格表示为可以通行。为了识别障碍物，白色网格单元用 0 表示，灰色网格单元用 1 表示。网格方法简单、有效地创建和维护网格模型。此外，网格法对障碍物具有较强的适应性。该方法便于计算机存储和处理

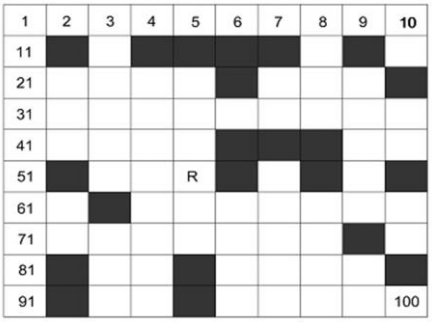

图 2 地图模型

将网格模型置于二维坐标系中。然后采用序列号法对每个网格进行标记。在 N*N 网格映射中，位置坐标(x,y)对应于号码为 R 的网格，转换关系如下式。

$$\begin{cases} x = \begin{cases} \mathrm{mod}(R,N) - 0.5 & if\ \mathrm{mod}(R,N)!=0 \\ N + \mathrm{mod}(R,N) - 0.5 & otherwise \end{cases} \\ y = N + 0.5 - ceil(\dfrac{R}{N}) \end{cases} \tag{17}$$

其中 mod 是取余操作，ceil 是向上取整操作。

3.2 传统简单蚁群算法与改进蚁群算法路径规划性能测试

尽量保持传统简单蚁群算法和改进蚁群算法的参数一致，比较两种算法的性能。蚁群数量为 50 只，信息素挥发系数 $\rho=0.3$，信息素刺激因子 $\alpha=1$，启发式信息影响因子 $\beta=7$。

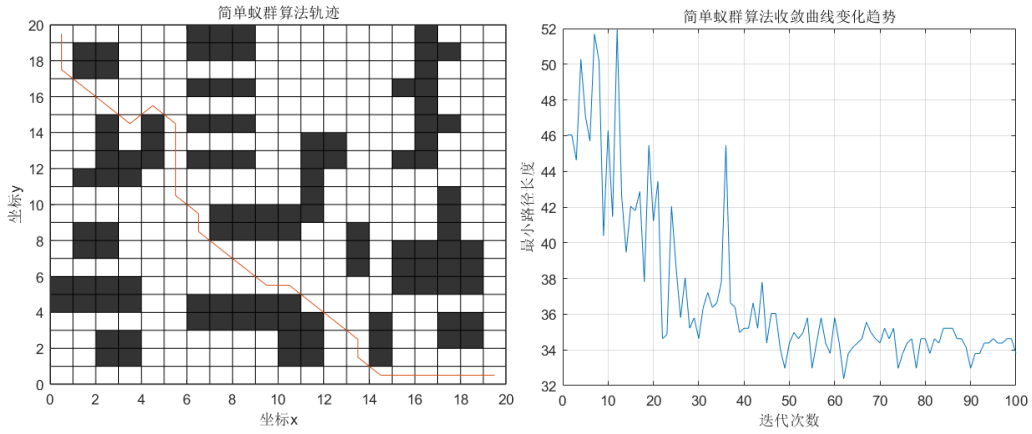

图 3 对于地图 1 简单蚁群算法规划的轨迹与收敛曲线

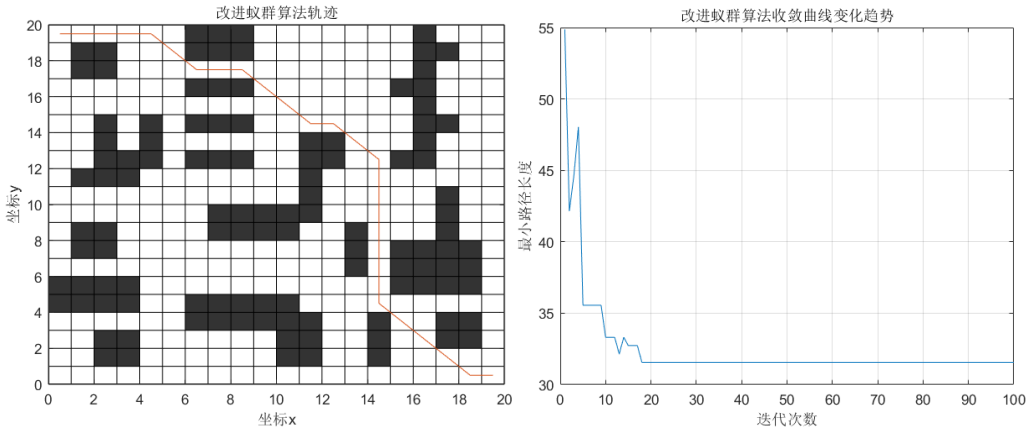

图 4 对于地图 1 改进蚁群算法规划的轨迹与收敛曲线

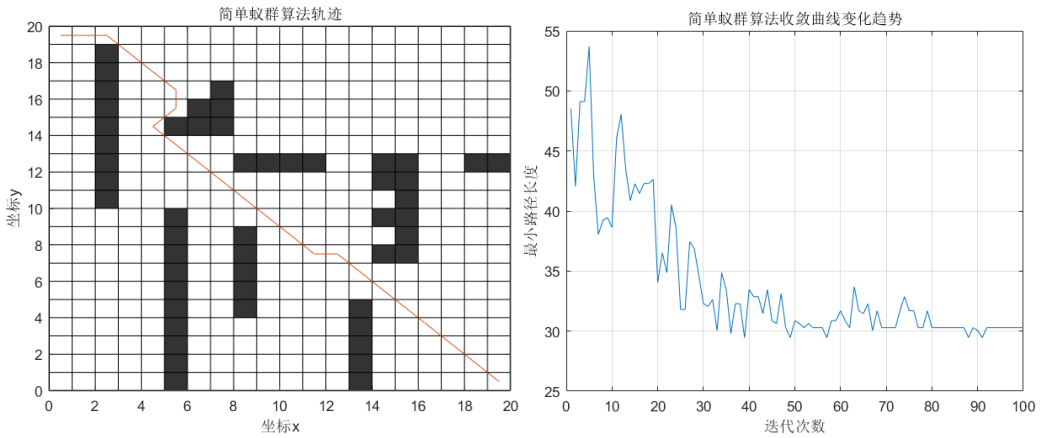

图 5 对于地图 2 简单蚁群算法规划的轨迹与收敛曲线

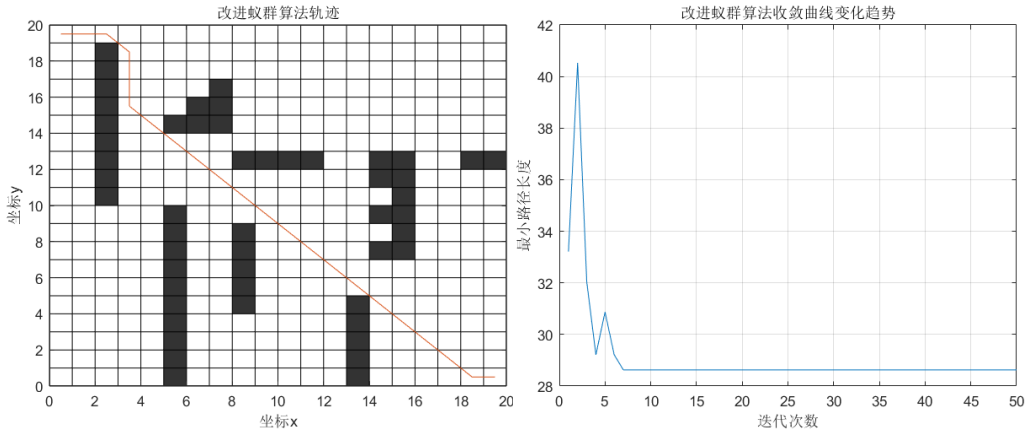

图 6 对于地图 2 改进蚁群算法规划的轨迹与收敛曲线

3.3 关于死亡机制与撤回机制的测试

为更好的比较死亡机制和撤回机制的优劣，将蚁群数量减少为 20 只。

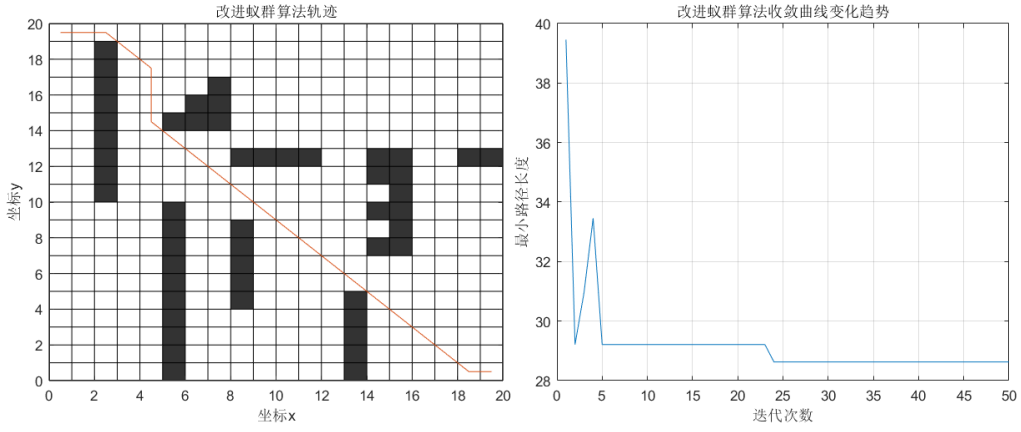

图 7 死亡策略的轨迹与收敛曲线

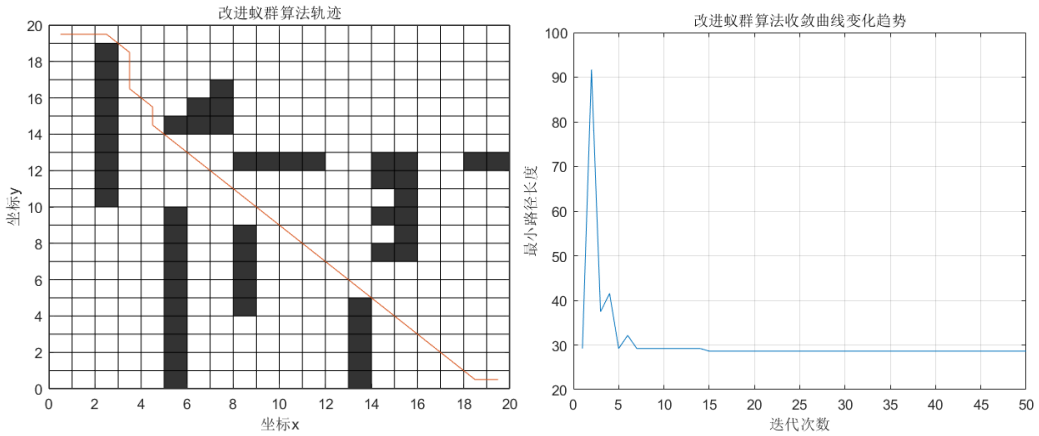

图 8 撤回机制的轨迹与收敛曲线

4.结论

通过简单蚁群算法的改进，实现了轨迹规划的收敛速度、最短路径长度和弯曲抑制效果的提高。使用 A*的估计函数作为启发式函数，以提高检索效率和路径的平滑度。通过采用撤回机制和改进的 MAX–MIN 蚂蚁系统方法，解决了蚂蚁死锁问题，提高了算法的全球搜索能力。

没有撤回机制。当蚂蚁陷入僵局状态时，撤回机制用于取代早期死亡策略，从而避免一次迭代中大量蚂蚁死亡。因此，每只蚂蚁都可以通过使用缩回机制获得路径，这增加了结果的多样性，有利于找到最佳路径，加快收敛速度。在算法的初始阶段收回的蚂蚁数量高于算法的中后期，缩回机制可以有效抑制蚂蚁数量的下降。

5.参考文献

[1]Stützle T, Hoos H H. MAX–MIN ant system[J]. Future generation computer systems, 2000, 16(8): 889-914.

[2]Duchoň F, Babinec A, Kajan M, et al. Path planning with modified a star algorithm for a mobile robot[J]. Procedia Engineering, 2014, 96: 59-69.

[3]Dong-Shu W, Hua-Fang Y. Path planning of mobile robot in dynamic environments[C]//2011 2nd International Conference on Intelligent Control and Information Processing. IEEE, 2011, 2: 691-696.

[4]Zhao J, Cheng D, Hao C. An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle[J]. Mathematical Problems in Engineering,

2016, 2016.

[5]Zhang W, Gong X, Han G, et al. An improved ant colony algorithm for path planning in one scenic area with many spots[J]. IEEE Access, 2017, 5: 13260-13269.