

# 动态环境下改进蚁群算法的多 Agent 路径规划\*

郑延斌<sup>1,2</sup>, 王林林<sup>1</sup>, 席鹏雪<sup>1</sup>, 樊文鑫<sup>1</sup>, 韩梦云<sup>1</sup>

(1. 河南师范大学计算机与信息工程学院, 河南 新乡 453007;

2. 智慧商务与物联网技术河南省工程实验室, 河南 新乡 453007)

**摘要:**针对动态环境下的多 Agent 路径规划问题, 提出了一种改进的蚁群算法与烟花算法相结合的动态路径规划方法。通过自适应信息素强度值及信息素缩减因子来加快算法的迭代速度, 并利用烟花算法来解决路径规划过程中的死锁问题, 避免陷入局部最优。在多 Agent 动态避碰过程中, 根据动态障碍物与多 Agent 之间的运行轨迹是否相交制定相应的避碰策略, 并利用路径转变函数解决多 Agent 的正面碰撞问题。仿真实验表明, 该方法优于经典蚁群算法, 能够有效解决多 Agent 路径规划中的碰撞问题, 从而快速找到最优无碰路径。

**关键词:**蚁群算法; 动态环境; 烟花算法; 避碰策略; 路径规划

**中图分类号:**TP24

**文献标志码:**A

**doi:**10.3969/j.issn.1007-130X.2019.06.017

## An improved ant colony algorithm for multi-agent path planning in dynamic environments

ZHENG Yan-bin<sup>1,2</sup>, WANG Lin-lin<sup>1</sup>, XI Peng-xue<sup>1</sup>, FAN Wen-xin<sup>1</sup>, HAN Meng-yun<sup>1</sup>

(1. College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007;

2. Henan Engineering Laboratory of Intellectual Business and Internet of Things Technologies, Xinxiang 453007, China)

**Abstract:** Aiming at the problem of multi-agent path planning in dynamic environments, we propose an improved dynamic path planning method by combining the ant colony algorithm and fireworks algorithm. This method accelerates the iteration speed of the algorithm by adapting the pheromone intensity value and the pheromone reduction factor, and uses the fireworks algorithm to solve the deadlock problem in the path planning process and avoid falling into a local optimum. In the process of multi-agent dynamic collision avoidance, corresponding collision avoidance strategies are made according to whether the motion trajectory between dynamic obstacles and multi-agent intersects, and the path collision function is used to solve the multi-agent frontal collision problem. Simulation results show that the proposed algorithm is superior to the traditional ant colony algorithm. It can effectively solve the collision problem in multi-agent path planning, and quickly find the optimal collision-free path.

**Key words:** ant colony algorithm; dynamic environment; fireworks algorithm; collision avoidance strategy; path planning

## 1 引言

路径规划问题是 MAS(Multi-Agent System)

研究的核心问题之一<sup>[1,2]</sup>。Agent 路径规划是指在已知或者未知的环境中找出一条从起点到终点的最优无碰路径<sup>[3]</sup>。蚁群算法<sup>[4]</sup>是一种基于仿生学的全局优化启发式正反馈算法, 能够成功应用于众

\* 收稿日期:2018-07-06; 修回日期:2018-09-26

基金项目:河南省科技攻关项目(142300410349, 132102210538); 河南省软科学项目(142400411001); 河南师范大学青年基金(2017QK20)

通信地址:453007 河南省新乡市河南师范大学计算机与信息工程学院

Address: College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, Henan, P. R. China

多优化问题中,尤其是在机器人路径规划问题中应用效果显著<sup>[5-7]</sup>。针对传统蚁群算法收敛速度慢、易陷入局部最优的缺陷,研究者作出了一些改进。Seng 等<sup>[8]</sup>通过改进信息素更新规则及状态转移概率来加快蚁群算法的迭代速度,从而快速找到最短路径;Yu 等<sup>[9]</sup>通过自适应地调整信息素更新规则来加快蚁群算法的迭代速度,避免陷入局部最优;柳长安等<sup>[10]</sup>提出利用狼群分配原则的方法对蚁群算法中最差路径上释放的信息素进行删除,从而加快了找到最短路径的速度;Yen 等<sup>[11]</sup>提出模糊蚁群优化算法,使用模糊控制最小蚁群优化算法的迭代学习误差来快速找到最短路径;屈鸿等<sup>[12]</sup>根据限定信息素强度的上下界,并使陷入死锁的蚂蚁采取回退策略来解决死锁问题,然后根据动态障碍物的运动方向来给出避碰策略,从而完成机器人路径规划。在多 Agent 环境中,受资源约束的影响,Agent 在运动过程中相互之间可能发生碰撞。但是上述方法对于多 Agent 之间的避障问题考虑得较少。

烟花算法 FWA(FireWorks Algorithm)<sup>[13]</sup>是一种模拟烟花在空中爆炸的现象来进行多点同时爆炸式搜索的算法,具有局部和全局搜索能力,并且种群具有多样性。

烟花算法是一种跳出局部搜索区域的邻域随机搜索算法,能够解决易陷入局部最优的问题,但是利用烟花算法的优势来解决蚁群算法易陷入局部最优问题的相关文献并不多见。

本文提出了一种改进的蚁群算法与烟花算法相结合的动态路径规划方法。该方法在 Agent 路径规划方面首先利用动态调节信息素强度  $Q$  值以及缩减因子来加快算法的收敛速度;其次,对于陷入死锁的 Agent 采取在当前节点利用烟花算法的爆炸算子来增大搜索空间,快速找到待选节点;最后在多 Agent 避障转移过程中制定出多 Agent 之间的路径转变函数,完成多 Agent 在动态环境下的路径规划。

## 2 相关知识

### 2.1 蚁群算法原理

蚁群算法<sup>[14]</sup>是模拟蚂蚁觅食原理而设计出的一种新型的随机搜索算法,整个路径规划过程可以看成是蚂蚁在地图中寻找食物的过程。蚂蚁在其走过的路径上释放一种信息素,在一定时间内,

其他蚂蚁可以感知这种信息素,并倾向于朝信息素浓度高的方向行走,由此形成的正反馈作用导致蚂蚁在搜寻最短路径的过程中走过的路径长度越短,留在路径上的信息素就会越多,影响其他蚂蚁选择该路径的概率就越大;反之,影响力就越小。蚂蚁之间就是通过这样的相互沟通、协调最终避开障碍物找到一条最优或者次优路径。

### 2.2 蚁群算法的基本模型

#### 2.2.1 栅格间转移概率

在蚁群算法中,蚂蚁  $k$  会根据路径上的残留信息素浓度的大小来选择下一步要走的栅格,在时刻  $t$  蚂蚁  $k$  从节点  $i$  转移到节点  $j$  的转移概率公式如式(1)所示:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & \text{if } j \in J_k(i) \\ 0, & \text{else} \end{cases} \quad (1)$$

其中,  $\eta_{ij}(t) = 1/(d_{ij} + d_{jk})$  表示启发式信息,  $d_{ij}$  表示从节点  $i$  到节点  $j$  之间的欧氏距离,  $d_{jk}$  表示下一节点到目标点之间的欧氏距离;  $J_k(i)$  表示待选路径所组成的集合;路径上的信息素浓度用  $\tau_{ij}(t)$  表示;  $\alpha$  表示信息素浓度的启发因子;  $\beta$  为启发因子的期望值。

#### 2.2.2 信息素更新规则

当迭代完成之后路径上的信息素需要更新,更新值由历史留下的信息素值与将要留下的信息素值决定,更新公式为:

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

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3)$$

其中,  $\rho \in (0,1)$  表示信息素,  $\tau_{ij}(t)$  表示路径上的信息素浓度,  $1-\rho$  为信息素残留因子;  $m$  为蚂蚁数量,  $\Delta\tau_{ij}(t)$  为从节点  $i$  到节点  $j$  的信息素增量值;  $\Delta\tau_{ij}^k(t)$  则表示残留信息素值。

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k, & \text{if 蚂蚁 } k \text{ 从节点 } i \text{ 到节点 } j \\ 0, & \text{else} \end{cases} \quad (4)$$

其中,  $Q$  为常数,表示信息素强度值;  $L^k$  表示蚂蚁  $k$  搜索到的路径长度。

### 2.3 烟花算法

烟花算法<sup>[13]</sup>是由 Tan 等于 2010 年提出的一种新颖优化算法。其具有局部和全局搜索能力,并且种群具有多样性。每个烟花即为解空间中的一

个可行解,其邻域搜索的过程为烟花爆炸产生一定数量火花的过程。适应度值好的烟花爆炸半径小,具备局部搜索能力,适应度值差的烟花爆炸半径大,具备很好的全局搜索能力,因此整个种群很好地兼顾了全局搜索能力和局部搜索能力<sup>[15]</sup>。爆炸算子的公式如式(5)和式(6)所示:

$$A_i = A^* \frac{f(x_i) - Y_{\min} + \epsilon}{\sum_{i=1}^N (f(x_i) - Y_{\min}) + \epsilon} \quad (5)$$

$$S_i = n \frac{Y_{\max} - f(x_i) + \epsilon}{\sum_{i=1}^N (Y_{\max} - f(x_i)) + \epsilon} \quad (6)$$

其中,  $A_i$  为第  $i$  个烟花的爆炸半径;  $N$  表示初始烟花的种群数量;  $A^*$  是用来控制爆炸半径的常数;  $Y_{\min}$  为当前种群中的最佳适应度值;  $S_i$  为第  $i$  个烟花产生的火花个数;  $n$  是常数,用来限制产生的火花总数;  $Y_{\max}$  是当前种群中适应度值最差的个体的适应度值;  $f(x_i)$  为烟花  $x_i$  的适应度值;  $\epsilon$  为一个极小的常数,以避免出现分母为零的情况。为避免烟花爆炸产生的火花数目过多或过少,给出了每个烟花产生火花数量的限制公式:

$$\hat{s}_i = \begin{cases} \text{round}(am), S < am \\ \text{round}(bm), S > bm \\ \text{round}(S_i), \text{otherwise} \end{cases}, a < b < 1 \quad (7)$$

其中,  $\text{round}(\cdot)$  为四舍五入函数;  $a$  和  $b$  为给定的常数。

对烟花的每一维进行位移操作,具体如下:

$$\Delta x_i^k = x_i^k + \text{round}(0, A_i) \quad (8)$$

其中,  $\text{round}(0, A_i)$  表示在幅度  $A_i$  内生成的均匀随机数。

### 3 改进蚁群算法

研究发现蚁群算法在路径寻优和自组织性方面存在很大的优势,但是在动态环境中传统蚁群算法存在易陷入局部最优以及迭代速度较慢的缺陷,因此本文针对这些缺陷对传统的蚁群算法做出以下改进。

#### 3.1 信息素更新规则

##### 3.1.1 自适应信息素强度值

蚂蚁探索初期,为了找到更好的路径,需要扩大搜索范围,此时信息素强度值必须设置得比较小,这样就能避免蚂蚁向一条路径继续行走而得到局部最优解。而在探索后期,路径已经基本探索完毕,较短路径也基本确定,这时就不需要大范围地

探索而需要稳定于较短路径。因此,  $Q$  值需要足够大,在这种情况下信息素强度值不再是常数,而是需要随着迭代次数的增加而不断改变,具体公式如下所示:

$$Q = \gamma \star \ln(k^*) \quad (9)$$

其中,  $k^*$  是当前迭代次数,  $\gamma$  是调控因子,取值在  $0 \sim 1$ 。

##### 3.1.2 信息素缩减因子

信息素更新可以分为实时更新和全局更新 2 个过程,蚂蚁在一轮迭代结束后更新全局信息素的时候,找到的最短路径上的信息素值会增加,本文以每轮结束后所有路径的信息素平均值为基准引入信息素缩减因子,具体公式如下所示:

$$v = \frac{L_{\text{best}}}{L_{\text{avg}}} \quad (10)$$

$$L_{\text{avg}} = \sum_{i=1}^k L^i / k^* \quad (11)$$

其中,  $L_{\text{best}}$  表示一轮搜索过程中找到的最短路径,  $L_{\text{avg}}$  表示一轮搜索过程中所有蚂蚁的平均路径长度。

从式(10)中可以看出,搜索到的路径越短表明本条路径上的信息素缩减因子越小,保留的信息素越多,反之则会越少。

所有蚂蚁都完成路径搜索进行信息素的全局更新,更新规则如下所示:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + (1-v)\Delta\tau_{ij}(t) \quad (12)$$

这种引入缩减因子的路径信息素更新方法不仅可以加快算法的收敛速度,也会大大增加最优路径的选择概率。

#### 3.2 解决死锁问题

对于死锁问题,文献[12]采取蚂蚁回退策略来解决,但是频繁地对陷入死锁的蚂蚁采取回退策略可能会相应地增加算法的运算时间,而本文采用的烟花算法具有较好的全局与局部搜索能力,烟花算法中的爆炸算子的局部搜索能力能够很好地解决蚁群算法易陷入局部最优的问题。本文采用在死锁边缘增加一种安全触发机制,当判断出下一个待选节点的相邻栅格都为障碍栅格时蚂蚁停止向前行走,此时 Agent 在当前节点利用烟花算法的爆炸算子以当前所找到的最短路径  $L_{\text{best}}$  为爆炸半径进行下一待选节点的寻找,信息素则在找到的最优值上进行更新。更新公式如式(13)~式(15)所示。

$$A_i = A^* \frac{L_{\text{best}} - Y_{\min} + \epsilon}{\sum_{i=1}^N (L_{\text{best}} - Y_{\min}) + \epsilon} \quad (13)$$

$$S_i = n \frac{Y_{\max} - L_{\text{best}} + \epsilon}{\sum_{i=1}^N (Y_{\max} - L_{\text{best}}) + \epsilon} \quad (14)$$

$$\Delta \tau_{ij}(t) = \min S_i \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (15)$$

烟花之间的差异性也使得蚂蚁在寻找最短路径的过程中同时具有开采性和勘探性,能够扩大蚂蚁的搜索范围,避免陷入死锁,加快找到最短路径。

### 3.3 改进蚁群算法路径规划

改进蚁群算法流程如下所示:

步骤 1 对智能体所处的环境进行栅格化处理,初始化蚁群算法中的参数,蚂蚁数量  $m$ 、迭代次数  $K_{\max}$ 、启发因子  $\alpha$ 、期望值  $\beta$ ;

步骤 2 设置蚂蚁  $k(k=1, 2, \dots, m)$  初始位置;

步骤 3 根据式(1)进行下一节点的选择;

步骤 4 判断蚂蚁是否到达局部最优值,如果到达则转步骤 5,否则继续行走;

步骤 5 当蚂蚁进入局部最优值时,依据式(13)~式(15)采用烟花算法的爆炸算子来扩大蚂蚁搜索范围进行下一节点的选择;

步骤 6 找到下一节点继续行走,如果没有则返回步骤 2;

步骤 7 对蚂蚁所寻路径和路径长度进行记录,另  $k=k+1$ ,若  $k < m$ ,则返回步骤 2 继续进行探索;

步骤 8 利用式(9)~式(12)对信息素进行更新;

步骤 9 判断迭代次数  $K$  是否到达最大值,如果没有则返回步骤 2 继续进行探索,否则结束算法;

步骤 10 判断是否找到最短路径,找到就结束算法,否则返回步骤 2;

步骤 11 结束算法。

## 4 避碰规划

多 Agent 根据规划好的路径行走的过程中,行走路线会因为动态障碍物的出现而改变,这时就需要提前预知障碍物的位置以便及时避障。本文给出 Agent 的视觉半径  $R$  和安全半径  $r$  来进行障碍物的风险预测,根据障碍物与 Agent 之间的距离大小分别给出精确避障层以及紧急避障层 2 个阶段的避碰策略。

### 4.1 精确避障层

当  $r < d_{io} \leq R$  时 Agent 处于精确避障层,  $d_{io}$  表示第  $i$  个 Agent 与障碍物之间的欧氏距离,首先

检测障碍物的当前位置与障碍物将要移动到的位置之间的连线,判断障碍物的运行轨迹线与 Agent 的运行轨迹是否有交点来进行动态避障。

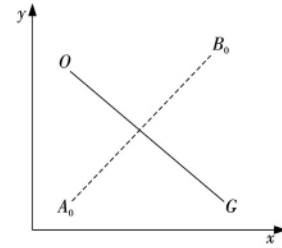


Figure 1 Trajectory lines of the agent and obstacle

图 1 Agent 与障碍物轨迹线

假设此时障碍物的位置为  $A_o(x_{o1}, y_{o1})$  向点  $B_o(x_{o2}, y_{o2})$  移动,  $OG$  为 Agent 的运行轨迹,此时点  $A_o$  与线段  $OG$  之间的距离为:

$$d_a = (Ax_{o1} + By_{o1} + C) / \sqrt{A^2 + B^2} \quad (16)$$

点  $B_o$  与线段  $OG$  之间的距离为:

$$d_b = (A'x_{o2} + B'y_{o2} + C') / \sqrt{A'^2 + B'^2} \quad (17)$$

如果  $d_a \cdot d_b \leq 0$ , 则障碍物运行轨迹与 Agent 的运行轨迹有交点;如果  $d_a \cdot d_b > 0$ , 则障碍物运行轨迹与 Agent 的运行轨迹不相交。障碍物的运行轨迹与 Agent 的运行轨迹没有交点,则说明 Agent 与障碍物不会发生碰撞,此时 Agent 按照原来规定好的路线行走;而当两者轨迹有交点时,说明在接下来的行走过程中会发生碰撞。下面分情况给出相应的避碰策略。

#### 4.1.1 侧面碰撞

若 Agent 与障碍物发生的是侧面碰撞,则使 Agent 在到达碰撞点之前进行等待,等待碰撞危险过去了再继续按照原来的路径行走。等待原则如下所示:

(1)以碰撞点  $P$  为基础,当 Agent 与碰撞点的欧氏距离  $d_{ip}$  与障碍物与碰撞点的欧氏距离  $d_{op}$  不相等时,距离碰撞点远的进行等待。

(2)当  $d_{ip} = d_{op}$  时,判断 Agent 到达目标点的欧氏距离  $d_{ig}$  与障碍物到达目标点的欧氏距离  $d_{og}$  的大小,距离目标点远的进行等待。

等待时间如式(18)所示:

$$\Delta T = \frac{d_{op}}{|v_i - v_o|} \quad (18)$$

其中,  $d_{op}$  为障碍物当前点与碰撞点之间的欧氏距离,  $v_i$  为 Agent 的速度,  $v_o$  为障碍物的速度。

#### 4.1.2 正面碰撞

如果 Agent 与动态障碍物即将发生正面碰撞,则 Agent 舍弃原来规划好的路径并转换方向

重新规划路径,重新规划的路径要能够保证与其他 Agent 不发生碰撞。Agent 在避碰过程中需要考虑相邻 Agent 的运动状态,通过考虑与相邻路径上 Agent 的相对运动状态来向相邻路径上转变运动轨迹,由于 Agent 之间也是存在差异性的,因此,对于相邻路径的选择也会产生一种决策作用。因此路径转变函数为:

$$\min T = \int_0^{s_0} [\xi w_s + \psi w_f + \kappa w_a + a^{*T} \cdot Z \cdot a^*] ds \quad (19)$$

其中,  $\min T$  是安全转移约束值,  $s$  表示节点  $i$  到节点  $j$  的路径长度,  $w_s$  表示风险程度代价值;  $w_f$  表示 Agent 的路径距离代价值,  $w_a$  表示 Agent 之间的距离代价值,  $\xi$  表示风险程度权重因子;  $\psi$  表示 Agent 路径代价权重因子,  $\kappa$  表示 Agent 之间代价权重因子;  $a^*$  为 Agent 的加速度;  $Z$  为 Agent 自身的影响因子。给出路径转变约束函数可以很好地解决与障碍物的避碰过程中与其它 Agent 的碰撞问题,从而快速到达目的地。

#### 4.2 紧急避障层

处于紧急避障层的 Agent 会遇到突发的动态障碍物的影响,此时需要以最快的反应进行避障,因此当 Agent 遇到突发情况时需要使行驶速度较慢的 Agent 快速地选择与动态障碍物相反的方向进行避碰,直到障碍物在安全半径  $r$  的区域外。

本文给出的策略能够很好地解决 Agent 在避障过程中重新规划路径时与其它 Agent 碰撞的问题,从而能够快速找到无碰最优路径。

#### 4.3 基于分层的避碰算法

综合考虑多 Agent 与障碍物的运动状态给出避碰算法如下所示:

**步骤 1** 初始化有关参数,包括起始点与目标点的确定,以及探测半径  $R$  和安全半径  $r$ ,用改进蚁群算法规划好每个 Agent 的初始路径;

**步骤 2** 在行走过程中判断动态障碍物是否到达 Agent 的探测范围之内,如果是则执行步骤 3,否则继续按照原定路线行走;

**步骤 3** 动态障碍物到达 Agent 的探测范围之内,此时进入到精确避障层,根据式(16)和式(17)计算 Agent 与动态障碍物的运行轨迹是否有交点,如果有则执行步骤 4,如果没有则返回步骤 2;

**步骤 4** 当 Agent 与动态障碍物的运行轨迹有交点时,若判断 Agent 与障碍物即将发生侧面碰撞则执行步骤 5;

**步骤 5** 针对即将与障碍物将发生侧面碰撞的 Agent,根据式(18)等待碰撞风险过去,然后继续按照原定路线行走;

**步骤 6** 当判断 Agent 与障碍物将发生正面碰撞时,则根据式(19)的路径转变函数重新规划路径,直到碰撞风险过去;

**步骤 7** 判断 Agent 是否遇到突发的危险,若遇到则进入紧急避障层,Agent 快速选择与动态障碍物相反的方向行走,否则执行步骤 8;

**步骤 8** Agent 根据避碰策略规划好的路径继续行走,判断是否到达目标点,如果到达则算法终止,否则返回步骤 2;

**步骤 9** 算法终止。

多 Agent 根据避碰算法能够有效地避免与动态障碍物及其它 Agent 的碰撞,从而能够快速找到最优无碰路径。

## 5 实验仿真与分析

### 5.1 仿真实验

设机器人的运行环境为二维平面环境,在机器人的运行环境中同时存在有限数量的静态和动态障碍物。静态障碍物的形状是任意的,且位置信息已知,动态障碍物有 4 个,可以利用传感器测出动态障碍物的运动状态及位置信息。机器人与障碍物的碰撞形式有 3 种:正面碰撞、侧面碰撞以及动态障碍物运动方向不确定的碰撞。机器人可以在任意状态下暂停运动或改变运动速度。

#### 5.1.1 仿真实验 1

为了验证本文所提算法对环境的适应性,随机生成一个  $20 \times 20$  的栅格环境将本文算法与传统蚁群算法即文献[4,9]中的算法在迭代次数和最短路径上进行对比验证。由于文献[9]中算法采用改进信息素挥发因子的方式来调控蚂蚁全局与局部搜索的能力,利用本文算法与其进行对比可以有效验证本文算法的高效性,由此可以得到如图 2 所示的 3 种算法的路径规划图。3 种算法的路径长度及迭代次数对比图如图 3 所示,实验中参数为  $K_{\max} = 200, m = 30, \alpha = 1, \beta = 7, \rho = 0.5$ 。

通过图 2 的对比实验可以看出,本文算法在寻找最优路径的过程中能够很好地避开障碍物的干扰,从而快速找到最优无碰路径。通过图 3 的对比结果可以看出,在最短路径的寻找过程中,本文算法优于文献[4,9]中所提的算法,因此本文所提算

法能够得到较好的结果,经过 100 次实验取得的平均值如表 1 所示。

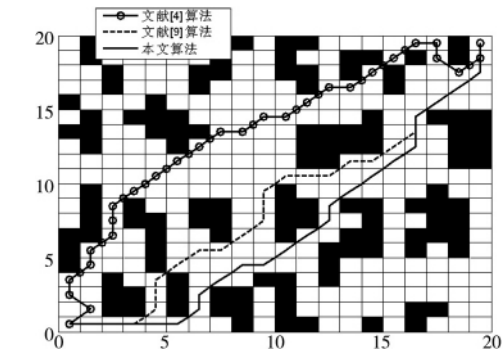


Figure 2 Shortest path comparison in experiment 1  
图 2 实验 1 中最短路径对比图

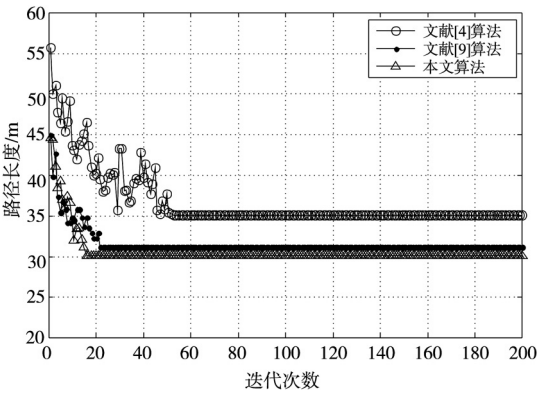


Figure 3 Comparison of iterations in experiment 1  
图 3 实验 1 中迭代次数对比图

通过表 1 中数据可以得知,本文提出的算法无论在迭代速度还是在最短路径方面都优于文献[4, 9]中的算法,经过多次实验验证了本文所提算法的稳定性。

Table 1 Performance comparison results in experiment 1  
表 1 实验 1 中性能对比结果

算法	迭代次数	最短路径/m
文献[4]算法	53	35.071 4
文献[9]算法	22	31.194 2
本文算法	16	30.102 1

5.1.2 仿真实验 2

利用本文算法与传统蚁群算法即文献[4, 12]改进蚁群算法在 20 \* 20 的栅格环境中进行对比,使用文献[12]中的参数  $K_{max} = 200, \rho = 0.16, \beta = 5.0, m = 50, \alpha = 1.5$ 。本文算法中信息素强度值是随着迭代次数的增加自适应变化的。在解决死锁问题时,文献[12]中频繁地对进入死锁的蚂蚁采取回退策略,可能会使算法的运行速度减慢,而本文使用烟花算法的爆炸算子避免蚂蚁陷入局部最优,缩短了算法找到最短路径的时间。经过实验验证

可以得到如图 4 所示的多 Agent 的路径规划图以及如图 5 所示的 3 种算法的迭代次数对比图,经过 100 次实验得到表 2 所示的 3 种算法的平均时耗及最短路径的对比结果。

通过图 4 可以看出,本文算法能较稳定地找到最优路径。通过图 5 可以看出,本文算法在迭代次数以及最优路径上都优于文献[4, 12]中的算法。

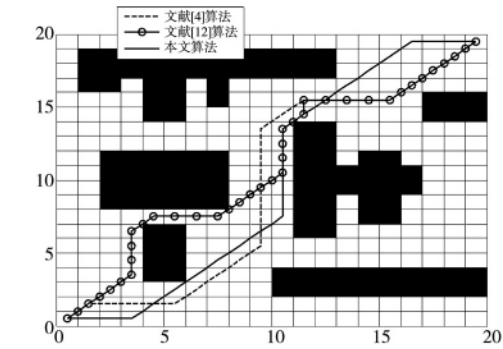


Figure 4 Shortest path comparison in experiment 2  
图 4 实验 2 中最短路径对比图

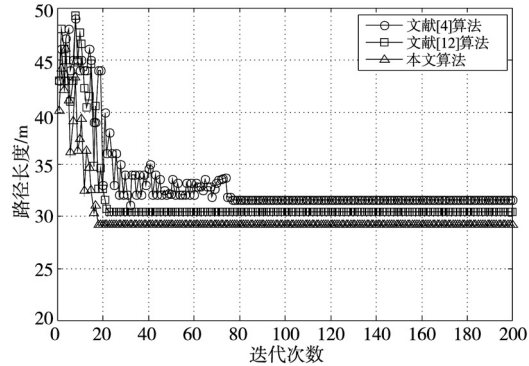


Figure 5 Comparison of iterations in experiment 2  
图 5 实验 2 中迭代次数对比图

通过表 2 中的数据可以看出,本文算法在加入烟花算法的爆炸算子的基础上可以避免蚂蚁陷入局部最优值,从而加快算法的迭代速度,从 3 种算法的对比结果中可以看出,本文算法无论在时耗还是最短路径方面都优于文献[4, 12]中的算法。

Table 2 Performance comparison results in experiment 2  
表 2 实验 2 中性能对比结果

性能	时耗/ms	最短路径
文献[4]算法	891	31.556 3
文献[12]算法	336	30.384 8
本文算法	330	29.152 5

5.2 动态环境下多 Agent 之间避碰分析

本文对动态环境中的多 Agent 之间的避碰给出实验验证分析,文献[12]中的 Agent 的速度是固定不变的,在与动态障碍物的避碰过程中对于 Agent 之间的碰撞没有给出明确的解决方案;而本



文中 Agent 的速度是不断变化的,在 Agent 之间的避碰过程中加入 Agent 的自我决策性进行多 Agent 之间的避碰效果验证。根据多次实验,Agent 的检测半径  $R=4$ ,安全半径  $r=2$ 。

Agent1 和 Agent2 从起点 S 处开始按照规定好的路线行走,在  $t_1$  时刻会遇到动态障碍物 D1,如图 6 所示,此时根据所提出的动态障碍物碰撞预测方法开始进行预测,检测到 D1 处在 Agent1 的探测范围之内;然后根据 D1 的运动方向可以知道如果 Agent1 不转变方向则会不可避免地与 D1 发生碰撞,因此 Agent1 需要转变方向,以进行动态避障。此时,由于 Agent2 处于 Agent1 转变方向后的安全碰撞范围之内,因此为了避免 Agent1 与 Agent2 发生碰撞 Agent2 也需要转变方向,最终 Agent1 与 Agent2 都成功避开障碍物各自按照重新规划的路线行走。

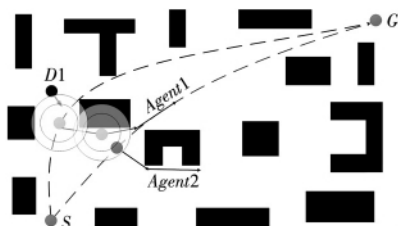


Figure 6 Agent1 steering diagram

图 6 Agent1 转向图

如图 7 所示,此时 Agent1 成功躲过动态障碍物 D1 以及 Agent1 与 Agent2 之间的相互影响,在重新规定好的路线上继续行走,到  $t_2$  时刻会遇到方向不确定的动态障碍物 D2 与方向垂直向下的动态障碍物 D3,此时检测到 D2 与 Agent1 不会发生碰撞,而 D3 则会影响 Agent2 的路线,二者会发生正面碰撞,Agent2 只需要调整运行路线的角度即可避免与 D3 的碰撞,结果如图 7 所示。

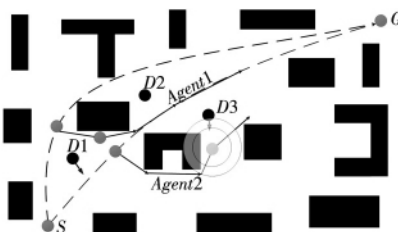


Figure 7 Agent2 dynamic obstacle avoidance diagram

图 7 Agent2 动态避障图

当 Agent1 与 Agent2 成功躲过动态障碍物 D3 时继续行走,而在  $t_3$  时刻 Agent1 检测到动态障碍物 D4 即将进入到危险检测区域,由于 D4 的行走方向是垂直向上的,此时经过计算可得 D4 朝与 Agent1 行走路线相反的方向行走,如图 8 所示

两者不会发生碰撞,因此避免了 Agent1 的路径转变。图 9 所示为本文所提算法中规划好的 Agent1 与 Agent2 的最优无碰路径。

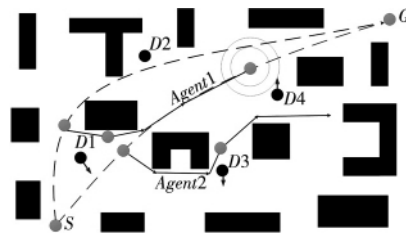


Figure 8 Agent1 dynamic obstacle avoidance diagram

图 8 Agent1 动态避障图

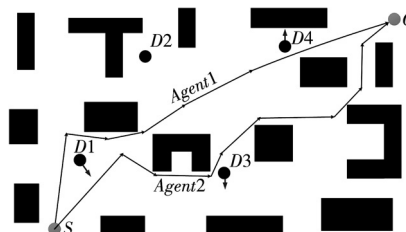


Figure 9 Final path diagram

图 9 最终路径图

## 6 结束语

本文研究了动态环境下多 Agent 的路径规划,首先通过改进蚁群算法进行路径规划,然后引入烟花算法的爆炸算子增加动态搜索性解决死锁问题,从而避免陷入局部最优,在多 Agent 的动态避障过程中给出多 Agent 之间的路径转变函数,保证在发生正面碰撞时路径转变的安全性,仿真实验验证了本文所提算法的有效性。

### 参考文献:

- [1] Gorbenko A, Popov V. Multi-agent path planning[J]. Applied Mathematical Sciences, 2012, 6(135): 6733-6737.
- [2] Chakraborty J, Mukhopadhyay S. A robust cooperative multi-robot path planning in noisy environment[C]// Proc of the 2010 International Conference on Industrial and Information Systems, 2010: 626-631.
- [3] Bennet D J, Mcinnes C R. Distributed control of multi-robot systems using bifurcating potential fields[J]. Robotics & Autonomous Systems, 2010, 58(3): 256-264.
- [4] Dorigo M, Maniezzo V, Colnini A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems Man & Cybernetics, Part B: Cybernetics A, 1996, 26(1): 29-41.
- [5] Kovács B, Szayer G, Tajti F, et al. A novel potential field method for path planning of mobile robots by adapting animal motion attributes [J]. Robotics & Autonomous Systems, 2016, 82(C): 24-34.

- [6] Pei Zhen-bing, Chen Xue-bo. Improved ant colony algorithm and its application in obstacle avoidance for robot[J]. CAAI Transactions on Intelligent Systems, 2015, 10(1): 90-96. (in Chinese)
- [7] Darwish A H, Joukhadar A, Kashkash M. Using the bees algorithm for wheeled mobile robot path planning in an indoor dynamic environment[J]. Cogent Engineering, 2018, 5(1): 1-23.
- [8] Seng De-wen, Tang Mei-xia, Wu Hao, et al. Multiple constrained dynamic path optimization based on improved ant colony algorithm[J]. International Journal of u-and e-Serv-ice, Science and Technology, 2014, 7(6): 117-130.
- [9] Yu L J, Wei Z H, Wang H, et al. Path planning for mobile robot based on fast convergence ant colony algorithm[C]//Proc of the 2017 IEEE International Conference on Mechatronics and Automation, 2017: 1493-1497.
- [10] Liu Chang-an, Yan Xiao-hu, Liu Chun-yang, et al. Dynamic path planning for mobile robot based on improved ant colony optimization algorithm[J]. Acta Electronica Sinica, 2011, 39(5): 1220-1224. (in Chinese)
- [11] Yen C T, Cheng M F. A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance[J]. Microsystem Technologies, 2018, 24(1): 125-135.
- [12] Qu Hong, Huang Li-wei, Ke Xing. Research of improved ant colony based robot path planning under dynamic environment[J]. Journal of University of Electronic Science and Technology of China, 2015, 44(2): 260-265. (in Chinese)
- [13] Tan Ying, Zhu Yuan-chun. Fireworks algorithm for optimization[C]//Proc of the 1st International Conference on Advances in Swarm Intelligence, 2010: 355-364.
- [14] Wan Xiao-feng, Hu Wei, Zheng Bo-jia, et al. Robot path planning method based on improved ant colony algorithm and Morphin algorithm[J]. Science & Technology Review, 2015, 33(3): 84-89. (in Chinese)
- [15] Bao Xiao-xiao, Ye Chun-ming, Ji Lei, et al. Improved chaotic firework algorithm for multi-objective scheduling optimization[J]. Application Research of Computers, 2016, 33(9): 2601-2605. (in Chinese)

#### 附中文参考文献:

- [6] 裴振兵, 陈雪波. 改进蚁群算法及其在机器人避障中的应用[J]. 智能系统学报, 2015, 10(1): 90-96.
- [10] 柳长安, 鄢小虎, 刘春阳, 等. 基于改进蚁群算法的移动机器人动态路径规划方法[J]. 电子学报, 2011, 39(5): 1220-1224.
- [12] 屈鸿, 黄利伟, 柯星. 动态环境下基于改进蚁群算法的机器

人路径规划研究[J]. 电子科技大学学报, 2015, 44(2): 260-265.

- [14] 万晓凤, 胡伟, 郑博嘉, 等. 基于改进蚁群算法与 Morphin 算法的机器人路径规划方法[J]. 科技导报, 2015, 33(3): 84-89.

- [15] 包晓晓, 叶春明, 计磊, 等. 改进混沌烟花算法的多目标调度优化研究[J]. 计算机应用研究, 2016, 33(9): 2601-2605.

#### 作者简介:



郑延斌(1964-), 男, 河南内乡人, 博士, 教授, 研究方向为虚拟现实和多智能体系统。E-mail: zybcgf@163.com

ZHENG Yan-bin, born in 1964, PhD, professor, his research interests include virtual reality, and multi-agent system.



王林林(1993-), 女, 河南淮阳人, 硕士生, 研究方向为虚拟现实和多智能体系统。E-mail: 1206789134@qq.com

WANG Lin-lin, born in 1993, MS candidate, her research interests include virtual reality, and multi-agent system.



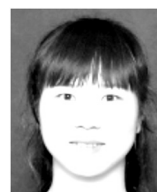
席鹏雪(1993-), 女, 河南延津人, 硕士生, 研究方向为虚拟现实和多智能体系统。E-mail: xipengxue@163.com

XI Peng-xue, born in 1993, MS candidate, her research interests include virtual reality, and multi-agent system.



樊文鑫(1994-), 男, 河南郑州人, 硕士生, 研究方向为虚拟现实和多智能体系统。E-mail: 525845701@qq.com

FAN Wen-xin, born in 1994, MS candidate, his research interests include virtual reality, and multi-agent system.



韩梦云(1993-), 女, 河南安阳人, 硕士生, 研究方向为虚拟现实和多智能体系统。E-mail: 1142767966@qq.com

HAN Meng-yun, born in 1993, MS candidate, her research interests include virtual reality, and multi-agent system.