

动态复杂环境下的机器人路径规划蚂蚁预测算法

朱庆保

(南京师范大学计算机科学系 南京 210097)

摘 要 研究了一种新颖的动态复杂不确定环境下的机器人路径规划方法和动态避障蚂蚁预测算法. 该方法模拟蚂蚁的觅食行为, 由多组蚂蚁采用最近邻居搜索策略和趋近导向函数相互协作完成全局最优路径的搜索. 在此基础上用虚拟蚂蚁完成与动态障碍物碰撞的预测, 并用蚁群算法进行避障局部规划. 理论和仿真实验结果均表明, 即使在障碍物非常复杂的地理环境, 用文中算法也能迅速规划出优化路径, 且能安全避碰.

关键词 机器人; 路径规划; 移动障碍物; 蚂蚁预测; 蚁群优化算法

中图法分类号 TP24

Ants Predictive Algorithm for Path Planning of Robot in a Complex Dynamic Environment

ZH U Qing Bao

(Department of Computer Science, Nanjing Normal University, Nanjing 210097)

Abstract Based on Ant Colony Optimization (ACO), this paper presents a novel algorithm underlying the robot path planning and dynamic obstacle avoidance in a complex and unfamiliar environment. By mimicking the food hunting behavior of ant colony, this algorithm can search for the global optimal path by adopting the nearest neighbor search strategy combining an approximating direction function used by multiple ant groups. The virtual ants, based on this algorithm, are able to predict their potential collision with the moving obstacles. The subsequent local plans for avoiding such collisions are then scheduled under the ACO. The analytical and computer experiment results demonstrate that this novel algorithm can plan an optimal path rapidly in a cluttered environment. The successful obstacle avoidance is achieved, and the model is robust and performs reliably.

Keywords robot; path planning; moving obstacles; ants predictive algorithm; ant colony optimization algorithm

1 引 言

移动机器人路径规划问题是指在有障碍物的工作环境中, 如何寻找一条从给定起始点到终止点的较优的运动路径, 使机器人在运动过程中能安全、无碰撞地绕过所有的障碍物, 且所走路径最短. 当机器

人具备全局环境信息时, 可用一次性的全局规划来得到一条自起点到终点的安全路径, 并能对运行过程中的某些性能指标进行优化, 这方面的研究已有广泛的报道, 例如用遗传算法、神经网络、随机树等方法对机器人路径进行规划^[1~4], 然而, 这些方法存在着搜索空间大、算法复杂、效率不高等问题, 特别是当障碍物的数目增加或地形障碍趋于复杂时, 这

些路径规划算法的复杂度将会大大增加, 甚至无法求解。

另一方面, 机器人的工作环境不少是动态不确定环境, 对这类环境中的动态障碍物, 机器人很难具有先验知识, 这种情况, 机器人只能根据实时探测到的环境信息进行避碰规划。为了解决这类规划问题, 有学者提出了一种全局环境未知时的静态环境下的滚动窗口规划算法^[5, 6], 在此原理基础上, 提出了对动态障碍物的避碰策略和方法^[7, 8], 有效地解决了动态障碍物环境下机器人行走过程中的安全避碰问题。然而该算法由于缺少全局信息, 加上机器人的视野有限, 滚动出的路径很难达到全局较优。

机器人路径规划是一类 NP Hard 问题, 传统算法复杂、速度慢且不易得到较优解。根据目前的研究现状和不足, 本文研究了一种在机器人固定工作空间中含有未知移动障碍物环境下的路径规划蚂蚁预测算法。该算法首先根据静态障碍物的信息由两个蚂蚁家族各 m 只蚂蚁协作完成机器人路径全局规划, 在此基础上, 进行动态碰撞预测, 据此进行避障局部规划, 不仅可安全避碰, 而且使机器人行走路径基本达最优。该算法是一种模拟蚂蚁觅食行为的仿生算法, 可有效解决这一类 NP Hard 问题。

2 问题描述

对于任意二维地形, 存在着有限个静态和动态障碍物, 由于静态障碍物的坐标极易测绘, 因而可视作是已知信息, 动态障碍物则是不确定的信息。规划的目的是使机器人由起点 g_{begin} , 安全避碰地且沿一条较短路径到达终点 g_{end} 。

设机器人 Rob 是直径 $\leq R_a$ 的圆形机器人。任意时刻, Rob 能探测到以当前位置为中心, r 为半径区域内的环境信息; Rob 的运动速率记为 v_R ; 环境探测及路径规划所需的时间忽略不计。

记 AS 为机器人 Rob 在二维平面上的凸多边形有限运动区域, 其内部分布着有限个静态障碍物 Sb_1, Sb_2, \dots, Sb_n 和有限个动态障碍物 Db_1, Db_2, \dots, Db_q ; 各动态障碍物之间的最近距离 $> 2r$; t 时刻 $Db_i (i = 1, 2, \dots, q)$ 的运动速率记为 $v_{di}^t(t)$, $v_{di}^t(t) \in [0, v_{dh}]$, 其中, v_{dh} 为一有限速率。 Db_i 的运动轨迹和方向属未知, 但其沿不闭合、不自相交的光滑平缓轨迹单向运动, 且在机器人传感器探测范围内近似为方向不变的直线运动。

设保证机器人能自由运动的活动范围为 $[0, R_a]$, 在 AS 中建立系统直角坐标系 Σ_0 , 以 AS 左上角为坐标 0 点, 以横向为 X 轴, 纵向为 Y 轴, 则有 AS 在 X, Y 方向的最大值分别为 x_{max} 和 y_{max} 。以 R_a 为步长将 X, Y 分别进行划分, 由此形成一个个栅格, 如图 1 所示。每行的栅格数 $N_x = x_{max} / R_a$; 每列的栅格数 $N_y = y_{max} / R_a$ (考虑 AS 为任意形状, 因此, 可在 AS 边界补以障碍栅格, 将其补为正方形或长方形), 其中 $Sb_i (i = 1, 2, \dots, n)$ 占一个或多个栅格, 当不满一个栅格时, 算一个栅格。该划分策略从实用出发, 使场景描述与实际环境严格相符, 规划出的路径保证机器人畅通无阻。

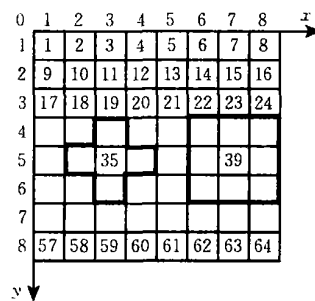


图 1 栅格坐标与序号关系

记 $g \in AS$ 为任意栅格, A 为 AS 中 g 的集合, 记 $OS = \{o_1, o_2, o_m\} \in A (m \geq n)$ 为静态障碍栅格集, $\forall Sb_i \in OS, Db_i \notin OS$. $\forall g \in A$ 在 Σ_0 都有确定的坐标 (x, y) , 记作 $g(x, y)$, $x = row, y = col$, 其中, row 为 g 所在的行号, col 为 g 所在的列号(为了和序号一致及方便, 定义左上角第一个栅格的坐标为 $(1, 1)$). 令 $C = \{1, 2, \dots, M\}$ 为栅格序号集, $g(1, 1)$ 序号为 1, $g(1, 2)$ 序号为 2, $\dots, g(2, 1)$ 序号为 $(N_x + 1)$, \dots , 如图 1 所示。

根据上述约定, $g_i \in A$ 的坐标 (x_i, y_i) 与序号 $i \in C$ 构成互为映射关系, 序号 i 的坐标可由式(1)确定:

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

式中, int 为舍余取整运算, \bmod 为求余运算。

Rob 在 AS 中的位置记为 P_R , $\forall P_R$ 在 Σ_0 都有确定的坐标 (x, y) , t 时刻的位置表示为 $P_R(t)$, 其坐标为 $(x_R(t), y_R(t))$ 。

规划起始位置为任意位置 $g_{begin} \in A$, 且 $g_{begin} \notin OS$, 终止点 $g_{end} \in A$, $g_{end} \notin OS$ 也为任意位置, 其它约束条件为: $begin, end \in C$, $end \neq begin$; g_{begin} 与任意动态障碍物的距离 $\geq r$ 。

3 蚁群算法基本原理及基于静态信息的全局规划算法

3.1 算法的基本原理与问题定义

经研究发现: 蚂蚁在觅食过程中能够在所经过的路径上留下一一种称为信息素的物质, 而且蚂蚁在觅食过程中能够感知这种物质的存在及其强度, 并以此指导自己的运动方向, 它们倾向于朝着该物质强度高的方向移动. 因此, 由大量蚂蚁组成的集体觅食行为便表现出一种信息正反馈现象: 某一路径越短, 该路径上走过的蚂蚁就越多, 则留下的信息素强度就越大, 后来者选择该路径的概率就越大. 蚂蚁个体之间就是通过这种信息交流来选择最短路径并达到搜索食物的目的. 蚁群算法就是模拟蚁群这一觅食行为的优化算法.

为了模拟实际蚂蚁的觅食行为, 设机器人出发点 g_{begin} 为蚁穴位置, 食物源则在最终目标点 g_{end} , 蚂蚁觅食过程就是从 g_{begin} 出发, 在 AS 范围寻找食物源的过程, 经过蚂蚁群体的反复寻食, 由于蚂蚁留下的信息素的正反馈作用, 最终绕开所有障碍物找到一条最短路径. 为了减少一开始的随机搜索时间, 在算法中增加了一个具有趋近引导作用的启发函数.

为了更好地适应复杂地形的路径规划, 提高搜索效率, 用两个蚂蚁家族共 $2m$ 只蚂蚁协作完成最优路径的搜索. 其中, 蚂蚁家族 ant_1 的 m 只蚂蚁以 g_{begin} 为蚁穴位置 (出发点), 以 g_{end} 为食物源; 蚂蚁家族 ant_2 的 m 只蚂蚁则以 g_{end} 为蚁穴位置 (出发点), 以 g_{begin} 为食物源, 两组蚂蚁相向搜索, 从而协作完成规划工作, 具体原理方法见下面的叙述.

本文方法是先基于静态环境规划出全局优化路径, 在此基础上, 再进行动态障碍物的避碰规划, 故本节暂不考虑动态障碍物.

为了叙述方便, 进一步作出如下约定和定义.

定义 1. $ant_i = \{1, 2, \dots, k, \dots, m\}$ 表示一个蚂蚁家族所有蚂蚁的集合 ($i=1, 2$), $k \in ant_i$ 表示某只蚂蚁, m 为第 i 蚂蚁家族的蚂蚁总数, $\tau_{ij}(t)$ 表示蚂蚁在 t 时刻残留在栅格 g_i, g_j 连线上的信息量.

定义 2. $\forall g \in A, g \notin OS$, 则称 g 为可行节点, 所有可行节点的集合称可行域, 记为 FS ; $\forall g \in A, g \in OS$, 则称 g 为禁入节点, 所有禁入节点的集合称禁入域, 记为 NFS .

显然, $FS = A \cap (OS)^c$; $NFS = A \cap OS$; 其中, 上标 c 表示补集.

定义 3. 蚂蚁 k 任意时刻所处的位置为 P ,

$\forall P$ 在 Σ_0 都有确定的坐标 (x, y) , 仅记 k 在 t_i 时刻处于某栅格的位置为 $P(x_i(t_i), (y_i(t_i)))$ 简记为 P_i 或 $P(t_i)$, 若它与 $g_j(x_j, y_j) \in A$ 的坐标相等, 则视 P_i 与 g_j 等价, 记作 $P_i \sim g_j$.

定义 4. 任意栅格间的距离指两栅格间的连线长度, 记作 $d(g_i, g_h)$ 或 $d(P_i, P_h)$, $i, h \in C$, 由式(2)计算. 特别地, 若有 $d(g_i, g_j)$, $i, j \in C$, 满足 $|j - i| = 1$ 或 $|j - i| = N_x$, (g_i, g_j) 在 AS 中的连线则称一个边 eg , 简称边 ij , $d(g_i, g_j)$ 则为边长, 记作 d_i .

$$d(g_i, g_h) = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2} \quad (2)$$

定义 5. $BR_i(g_i(x_i, y_i)) = \{g | g \in A, d(g, g_i) \leq 1\}$, $i \in C$ 称 g_i 的邻域或蚂蚁 k 在 g_i 处的视野域.

图 1 中的粗线十字范围示出了按本定义算出的 g_{35} 的邻域.

定义 6. 设 t_i 时刻, k 处于 g_i , $Wk_i(g_i(x_i, y_i)) = \{g | g \in BR_i(g_i(x_i, y_i)), g \notin OS, i \in C\}$ 称 t_i 时刻 k 在 BR_i 的可行域; 令 $tabu_k = \{P(t_0), P(t_1), \dots, P(t_i)\}$ 且有 $t_0 < t_1 < \dots < t_i$ 为 k 从 t_0 到 t_i 时刻已走栅格位置的集合, t_{i+1} 时刻, $\forall P(t_{i+1}) \in Wk_i$ 且 $\forall P(t_{i+1}) \notin tabu_k$, 则称 $\forall P(t_{i+1})$ 为 t_{i+1} 时刻的可行点, 可行点的集合用 Z 表示, 显然, $|Z| \leq |Wk_i|$.

$tabu_k$ 中各位置点在 AS 中的连线称 P_0 到 P_e 的通道, 记作 $path(P_0, P_e)$ (为简便, 也用 $path(P_0, P_e)$ 表示该通道上的节点集合), 通道的长度记作路程 L , 由式(3)计算.

很显然, $tabu_k$ 是第 k 只蚂蚁已走位置的集合, 它随着蚂蚁的行走动态调整. 按该定义, 这些位置不允许再走, 因此, 称 $tabu_k$ 为禁忌表.

根据该定义, 显然有 $Wk_i \cap OS = \emptyset$, $Wk_i = (BR_i - BR_i \cap OS)$.

$$L = \sum_{i=1}^e d_i, d_i = d(g_i, g_h), g_i, g_h \notin OS, i, h \in C \quad (3)$$

定义 7. $\eta_{1_i}(g_i) = D/d(g_i, g_{end})$, $\eta_{2_j}(g_j) = D/d(g_j, g_{begin})$, $\eta_{1_i}(g_i)$, $\eta_{2_j}(g_j)$ 分别称蚂蚁家族 1、蚂蚁家族 2 的第 k 只蚂蚁选择栅格 g_i 或 g_j 的启发函数, D 为权重常系数.

定义 8. $k_1 \in ant_1, k_2 \in ant_2$, k_1 从 g_{begin} 出发, k_2 从 g_{end} 出发, 经过 n 个时刻, k_1, k_2 的位置分别为 P_{k_1}, P_{k_2} , 若有 $|d(P_{k_1}, P_{k_2})| \leq 1$, 则称 k_1, k_2 相遇 (对仿真实验一节给出的走斜格方案, 该条件则为 $|d(P_{k_1}, P_{k_2})| \leq 1$ 或 $|d(P_{k_1}, P_{k_2})| \leq \sqrt{2}$).

根据以上原理和定义, 机器人路径规划的蚁群算法描述如下:

将两个蚂蚁家族的各 m 只蚂蚁分别放置在出发点 g_{begin} 和终止点 g_{end} , 对于两蚂蚁家族中的每一只蚂蚁, 以当前节点为中心, 按两族蚂蚁相向趋近原则和最近邻居选择策略选择并行走走到下一可行节点. 为了达到一开始就能逐步趋近的目的, 两族蚂蚁分别采用定义 7 给出的趋近导引启发函数. 由于两族蚂蚁除了采用的启发函数不同、出发点不同外, 算法完全相同, 为了减少变量的下标, 以下以蚂蚁家族 1 的搜索算法为例进行说明, 并将表示蚂蚁家族 1 的所有下标省略.

3.2 算法步骤

考察机器人路径规划问题, 若机器人 Rob 在时刻 t_i 处于 g_i 位置, 它下一步要走的栅格范围必然是 g_i 邻域 BR_i 内的可行域中的可行点. 因此, 蚂蚁从当前节点 g_i 选择下一节点时, 只要在 $Wk_i(g_i)$ 内选择 g_j 即可. 算法步骤如下:

1. 初始化: 将 m 只蚂蚁放置在出发点 g_{begin} , 并设置到禁忌表 $tabu_k$ 中 ($k=1, 2, \dots, m$). 令 $\tau_{ij}(0)=\tau_0$ (τ_0 为常数). 设置寻食代数计数器 $n=0$. 最大代数为 MAX, 其中, 令 $m \leq 4$.

2. $\forall k$, 以当前节点 $g_i \in FS$ 为中心, $i \in C$, 按两族蚂蚁相向趋近原则和最近邻居选择策略选择并行走走到下一节点 g_j , 且有 $g_j \in Wk_i(g_i)$, $g_j \notin tabu_k$, 节点选择如下:

2.1. 若有 $d(g_{begin}, g_j) \leq 1$, 即 $g_j \in BR_{begin}(g_{begin})$, ($\forall ant_1$, 若 $d(g_{end}, g_j) \leq 1$, $g_j \in BR_{end}(g_{end})$), 则 $\forall k$ 选择一个节点 $g_j^{k_i} \in Wk_{begin}(g_{begin})$, 且有 $g_j^{k_i} \neq g_j^{k_j} \in Wk_{begin}(g_{begin})$, $i, j=1, 2, \dots, m$. 其中, 上标 k_i, k_j 表示不同蚂蚁选择的不同节点.

2.2. 若 $g_i \notin BR_{begin}(g_{begin})$, $\forall k$ 根据式 (4) 或式 (5) 选择下一个节点 $g_j \in Wk_i(g_i)$, $g_j \notin tabu_k$

$$j = \begin{cases} \arg \max \{ [\tau_{ij}(n)]^{\beta} \eta_j(g_i) \}, & q \leq q_0 \\ S, & \text{否则} \end{cases} \quad (4)$$

$$p_{ij}^{k_j}(n) = \frac{[\tau_{ij}(n)]^{\beta} \eta_j(g_i)}{\sum_{q \in |Z|} [\tau_{iq}(n)]^{\beta} [\eta_q(g_i)]}, \quad j \notin tabu_k \quad (5)$$

式中, $j \in C$, 蚂蚁 k 所选 g_j 的节点序号, 在此省略了上标 k ; S 为由式 (5) 决定的随变量; q 为随机数 ($0 < q \leq 1$); q_0 为初始化时给定的阈值; $\eta_j(g_j)$ 为由定义 7 给出的启发信息; β 为在边 e_{ij} 上残留信息的重要程度; $p_{ij}^{k_j}(n)$ 为在 n 代, 蚂蚁 k 由节点 i 转移到节点 j 的概率, $i, j \in C$.

q 和 q_0 是为了防止出现停滞而设的随机搜索策略所需参数, 以增加搜索的多样性. 当 $q > q_0$ 时, 计算 $|Z|$ 个节点的转移概率 $p_{ij}^{k_j}$, 并根据赌轮盘规则选择节点 j . 需注意的是: 两族蚂蚁所用算法中, 蚂蚁所处的节点和所用的启发信息不同.

将 j 加入禁忌表 $tabu_k$.

3. 局部信息素更新

随着时间的推移, 以前留下的信息逐渐消逝, 用参数 $1-\rho$ 表示信息消逝程度, 每一只蚂蚁选择完一个节点即走完一个边后, 按式 (6) 进行局部信息更新.

$$\tau_{ij}(n+1) = (1-\rho)\tau_{ij}(n) + \rho \Delta \tau_{ij}^k \quad (6)$$

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q_1}{l_{jb}}, & \text{当蚂蚁 } k \text{ 走过边 } e_{ij} \\ 0 & \text{否则} \end{cases},$$

$$l_{jb} = \sum_{l=1}^w d_l.$$

当 $\tau_{ij}(n+1) < \tau_{min}$ 时, 令 $\tau_{ij}(n+1) = \tau_{min}$.

式中, Q_1 为常数, τ_{min} 是设定的最小值; d_l 是蚂蚁 k 已走过的边的边长, 按定义 4.5, $d_l \equiv 1$, 但当采用第 6 节仿真实验中允许走斜格方案时, d_l 必须由式 (2) 计算. w 是蚂蚁 k 在本次寻食中已走过的边数; l_{jb} 是 k 在本次寻食中到当前时刻为止已走过的路程 (边的累加总长).

4. $\forall k, k=1, 2, \dots, m$, 选择完第 j 个节点后, 按定义 8 定义的条件, 检查两族蚂蚁中的所有蚂蚁是否已有蚂蚁相遇, 若有则转步 5; 否则, 令 $i=j$ (即以新选的节点 j 为当前节点), 返步 2 开始选择下一个节点, 直到有蚂蚁相遇或所有节点选择完毕. 若重复多次仍无蚂蚁相遇, 则属于起、终点间无可行通道, 详细分析见第 5 节.

5. 当两族中的两只蚂蚁或多只蚂蚁满足定义 8 的相遇条件时, 将相遇蚂蚁所走通道连接, 并用式 (3) 计算其路程 L . 例如, $k_1 \in ant_1$, 所走的通道为 $Path(P_0, P_j)$, 所走的节点集合为 $tabu_{k_1} = \{1, 11, 12, 22, 23, 33, 43, 44\}$, $k_2 \in ant_2$ 所走的通道为 $Path(P_e, P_s)$, $|s-j| \leq 1$ 或 $|s-j| \leq N_s$, $s, j \in C$, 所走的节点集合为 $tabu_{k_2} = \{88, 78, 68, 58, 57, 56, 46, 45\}$, 节点 44 和 45 满足定义 8 相遇条件, 则将两通道连接为 $Path(P_0, P_e)$, 通道的节点集合则为 $tabu_{k_{12}} = \{1, 11, 12, 22, 23, 33, 43, 44, 45, 46, 56, 57, 58, 68, 78, 88\}$, 所有相遇蚂蚁所走通道连接完成后, 用式 (3) 计算各连接路程 L_k , 并保存最短路程 L_{kmin} ($L_{kmin} = \min L_k$, 若允许走斜格时, 各路程长度 L_k 可能不同).

将本次觅食得到的 L_{kmin} 与已得到的历史最优长度 L_d 比较, 若有 $L_{kmin} < L_d$ 则用 L_{kmin} 替换 L_d , 并记忆最佳通道的节点集合.

6. 全局信息素更新

本次觅食相遇并完成通道连接后, 将本次觅食最短通道上的信息素按式 (7) 调整:

$$\tau_{ij}^{new} = (1-\alpha)\tau_{ij}^{old} + \alpha \Delta \tau_{ij} \quad (7)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k,$$

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q_2}{L_{kmin}}, & ij \in global-best-tour \\ 0, & \text{其它} \end{cases}.$$

式中, Q_2 为常数; α 为全局信息素挥发系数; L_{kmin} 为本次寻食最佳通道的路程; $ij \in global-best-tour$ 表示蚂蚁 k 所走的边 ij 属于最佳通道;

7. 令寻食代数 n 加 1, 若不等于 MAX, 则清空禁忌表, 重复上述寻食过程, 直到 $n=MAX$ 为止. 最终记忆的最佳通道即为规划出的最优路径.

3.3 变更目的地时快速规划

变更目的地是指 Rob 在沿原规划路径行进途中, 根据需要或根据指令需转移到的新目标地址

g_{new} , 这时必须以当前位置 $P_R(x_R, y_R)$ 为起点, 以 g_{new} 为规划终点, 迅速规划出一条新的路径. 这种情况, 只需用 P_R 取代 g_{begin} , g_{new} 取代 g_{end} 即可用上述算法快速规划出新路径. 本算法能进行快速规划的重要原因是算法简单和高效. 仿真实验证明, 在超过数百平米环境中, 本算法用 Pentium 4 微机的规划时间小于 1s, 相对于机器人的移动而言, 该时间是完全可以忽略不计的, 因此, Rob 可以动态地改变其目的地, 快速地进行新路径规划, 快速地进行需要执行新任务.

4 动态避障蚂蚁预测算法及避障策略

定义 9. $\text{View}(P_R(t_i)) = \{P \mid P \in A, d(P, P_R(t_i)) \leq r\}$ 称为 Rob 在位置 $P_R(t_i)$ 处的视野域.

机器人 Rob 在第 3 节规划出的全局优化路径上行进的过程中, 会随机地遇到不知方向和运动轨迹的动态障碍物, 因此, Rob 每行进一步, 都需用传感器探测其视野域内的环境信息, 并分类出动态障碍物和静态障碍物, 若有动态障碍物, 则测出其方向、速度与运动轨迹, 据此进行动态避障局部规划.

动态避障蚂蚁预测算法的基本原理与方法如下所述.

1. 场景探测:

设 t_i 时刻探测到障碍物 $Xb_i \in \text{View}(P_R(t_i))$, 若 $Xb_i \notin \text{View}(P_R(t_i)) \cap OS$, $X = S, D$, 则 Db_i 为动态障碍物, 测其坐标位置为 $P_d(x_i(t_i), y_i(t_i))$, 经过一个较小的时刻 Δt 后, 其位置坐标为 $x_i(t_i + \Delta t)$, $y_i(t_i + \Delta t)$, 由此可知其方向, 并用式 (8) 估算其速度, 用式 (9) 估计 Db_i 在 View 内的近似的直线运动轨迹.

$$v_d = d(p_d(x(t + \Delta t), y(t + \Delta t)), p_d(x(t), y(t))) / \Delta t \quad (8)$$

上式中 $y = kx + G$ 其中,

$$\begin{cases} k = (y(t + \Delta t) - y(t)) / (x(t + \Delta t) - x(t)) \\ c = y(t) - x(t) / (x(t + \Delta t) - x(t)) \end{cases} \quad (9)$$

2. 机器人与障碍物相碰的预测

令蚂蚁 k 在 t_i 时刻从 Rob 当前位置以 Rob 的行走步长 δ 速度 V_R 沿 $\text{path}(p_0, p_e)$ 虚拟前进, t_i 时刻到达 View 的边缘, 令 $\Delta t = \delta / V_R$, 记 $\tau = \{t_i, t_i + \Delta t, t_i + 2\Delta t, \dots, t_v\}$, k 行走的位置集合记为 $\Pi(P(t_i), P(t_v))$, $\Pi(P(t_i), P(t_v)) \in \text{path}(p_0, p_e)$.

在 t_i 时刻, Rob 获知 Db_i 以速度 v_d 沿式 (9) 轨迹运动; 通过式 (9) 获得 Db_i 各位置及坐标 $p_d(x(t_i + k\Delta t), y(t_i + k\Delta t))$, $k = 1, 2, \dots, h$ (为简洁, 以下均略写坐标表示). 记 $D_i = \{p_d(t_i), p_d(t_i + \Delta t), p_d(t_i + 2\Delta t), \dots, p_d(t_v)\}$, 为 Db_i 的运动轨迹的位置集合, $D_i \in \text{View}$.

k 在前进过程中, $\forall \Delta t$ 都根据式 (10) 做相碰预测. 若满

足式 (10) 条件, 则称 $P(t_x)$ 为预测的第一个碰撞点, 简记作 P_c . 从 t_x 开始, 所有预测碰撞点的集合记作 CO .

$$\begin{cases} \forall P(t) \in \Pi(P(t_i), P(t_v)), \\ d(P(t), p_d(t)) > \epsilon, t < t_x, t \in \tau \\ d(P(t), p_d(t)) \leq \epsilon, t = t_x, t \in \tau \end{cases} \quad (10)$$

式中, ϵ 是考虑 Rob 和动态障碍物体积时的安全无碰撞距离阈值.

Rob 可根据 CO 情况, 进行如下避碰规划.

2.1. 若有 $|CO| = 1$, 即仅有一点相碰, 表示 Db_i 在 t_x 时刻穿越 Rob 的路径并恰与 Rob 相碰. 此种情况, Rob 只要在 $t_x - n\Delta t$ 时刻等待一段时间 t_w 即可, 其中, $n\Delta t = \eta(R_a/2 + R_d) / V_R$, $t_w = \rho(n\Delta t + 2R_d/v_d)$, 式中, R_d 为 Db_i 最大半径, η, ρ 为安全系数.

2.2. 若有 $|CO| > 1$, 表示 Db_i 至少有一部分时间贴近或在 Rob 路径上运行, Rob 应绕道前进. 此时, 可按下述方法进行绕道避碰局部路径实时规划.

3. 局部路径规划

局部路径规划的目的是为了避开动态障碍物而临时规划出的一条绕道路径, 使 Rob 绕开动态障碍物后再回到原来的全局优化路径上.

规划方法是以预测的第一个碰撞点 P_c 的前两个位置 $P_c(t_x - 2\Delta t)$ 或以 Rob 当前位置 $P_R(t_i)$ 为起点, 并选择 P_i 作规划终点, $P_i \in \text{path}(p_0, p_e)$, 用第 3 节蚁群算法可迅速规划出该绕道局部路径. 与第 3 节算法不同处仅为: 将 $CO' = CO \cup P_c(t_x - \Delta t) \cup P_m(t_m + \Delta t)$ 列为禁入域或视为障碍栅格. 其中, $P_m(t_m)$ 是预测的最后一个碰撞点. 将 CO 向前和向后各扩展一个位置是为了防止由于预测误差而引起碰撞, 保证规划出的路径安全可靠.

局部绕道路径终点 P_i 根据以下情况选取 ($P_i \notin CO'$):

(1) 若有 $d(P_R(t_v), g_{\text{end}}) < L/K$, 则取 g_{end} 位置作为 P_i , L 为总路程, K 为常系数.

(2) 若有 $d(P_R(t_v), g_{\text{end}}) \geq L/K$, 则以位置 $P_R(t_v)$ 为 P_i . 若在此区间该局部新路径不存在, 则取一新点 $P_j \in \text{path}(p_0, p_e)$ 且满足 $d(P_j, g_{\text{end}}) < d(P_R(t_v), g_{\text{end}})$ 作为规划终点, 直至找到新路径为止 (其极限情况是也取 g_{end} 位置作为规划终点).

该绕道避碰策略最终使 Rob 仍回到全局最优路径上, 既保证了安全, 又保证了其路径最优, 提高了 Rob 的工作效率.

5 可达性、安全性及信息存储量分析

由本文算法的原理可知, 蚂蚁所选择的节点总是那些没有障碍物的自由栅格, 因此, 得出的路径总是安全避碰的. 然而, 并不是所有情况下所有蚂蚁都能走出一个有效的安全路径, 有些地形情况, 随机给出的出发点和终止点之间可能根本不存在有效通道或通道被动态障碍物阻塞而又别无它路可走, 这时蚂蚁可能会被障碍阻挡而不能达到最终目标点. 在什么条件下才具有可达性呢? 下面将分别对静态环

境和动态环境下的可达性作出分析. 另一方面, 用这种栅格表示环境, 其信息存储是否会成为应用本算法的障碍呢? 以下也将作出分析.

5.1 静态环境下的可达性分析

定义 10. 令 $T = \{t_0, t_e\}$, $t_e > t_0$, 且 t_e 为有限时刻, $\forall P \in FS$, 若存在连续映射 $f: T \rightarrow AS$ 使得 $f(t_0) = P_0$, $f(t_e) = P_e$, 则称映射 f 是 AS 中客观存在的从 P_0 到 P_e 的一条可行路径, $f(t_0)$, $f(t_e)$ 分别为可行路径 f 的起点与终点. 像集 $f(T)$ 称为从 P_0 到 P_e 的一条可行通道, 记作 $Path(P_0, P_e)$. 显然, $Path(P_0, P_e)$ 构成了 AS 中连接点 P_0 到 P_e 的连续曲线, 为简便, 也用 $Path(P_0, P_e)$ 表示可行通道上的节点集合, 而可行路径则是指产生可行通道的连续映射. 很显然, 这里的 $Path$ 与定义 6 给出的 $Path$ 是等同的.

定义 11. $\forall P_0, P_e \in FS$, FS 中所有从 P_0 到 P_e 的可行路径的集合记为 Fp_{0e} .

定义 12. 设可行路径 $f_i \in Fp_{0e}$, 若 $\forall t_1, t_2 \in [t_0, t_e]$, $t_1 < t_2$, 都有 $d(f_i(t_2), P_e) < d(f_i(t_1), P_e)$, 则称 $f_i(t_2) = P(t_2)$ 是向 P_e 严格趋近的节点位置, f_i 是从 P_0 到 P_e 严格趋近可行路径, 其中, $P_e \sim g_{end}$ 或 $P_e \sim g_{begin}$.

引理 1. $\forall e_{ij} \in f_i$, 都有 $\tau_{ij} \leq \tau_{\max} = \max \left\{ \tau_m / \rho, \frac{\tau_m(1+1/(1+\alpha))}{\rho+\alpha/(1-\alpha)} \right\}$, $\forall e_{kl} \notin f_i$, 都有 $\tau_{kl} \leq \tau_{ij}$, 其中 $\tau_m = \max(\alpha \Delta \tau_{ij}, \rho \Delta \tau_{ij}^k)$.

证明. 设某代寻食得到的最优路径为 f_i , 根据算法步骤和式(4)、式(5)、式(6)、式(7), 可分以下三种情况:

(1) 对蚂蚁从未走过的边, 信息素只消散, 有 $\tau_{kl}(n) = (1-\rho)^n \tau_0$, $n \rightarrow \infty$, $\tau_{kl}(n) \rightarrow 0$, τ_{kl} 将被限制在 τ_{\min} ; 很显然, 这种情况 $e_{kl} \notin f_i$.

(2) $e_{ij} \in f_i$, $e_{ij} \notin f_i$, 有

$$\tau_{ij}(n) = (1-\rho)^n \tau_0 + \sum_{i=1}^n (1-\rho)^{n-i} \tau_m, \tau_m = \rho \tau_{ij}^k,$$

$$n \rightarrow \infty, \tau_{ij}(n) \rightarrow \frac{1}{\rho} \tau_m \text{ (注: 后项为一等比数列)},$$

(3) $e_{ij} \in f_i$, $e_{ij} \in f_i$, 则有

$$\tau_{ij}(n) = (1-\alpha)^n (1-\rho)^n \tau_0 + \sum_{i=1}^n (1-\alpha)^{n-i} (1-\rho)^{n-i} [(1-\alpha) \tau_m + \tau_m], t_s \rightarrow \infty, n \rightarrow \infty, \tau_{ij}(n) \rightarrow \frac{\tau_m(1+1/(1-\alpha))}{\rho+\alpha/(1-\alpha)}.$$

比较以上三种情况可见, $\forall \tau_{ij}$ 都有 $\tau_{ij} \leq \tau_{\max}$, 且 $\tau_{ij} \geq \tau_{kl}$. 证毕.

定理 1. 一定存在有限大小的 t_p , 使得 $P(t_j)$ 是向 P_e 严格趋近的, 其中 $t_j \leq t_p$.

证明. 设 k 的当前位置为 $P(t_i)$, $t_i < t_p$, $P(t_i) \sim g_i$, k 从 g_i 选择并行走到下一个节点 g_j , 根据定义 4~6, $\forall P(t_{i+1}) \in Wk_i(P(t_i))$, $\exists P(t_{i+1})$ 使得 $d(P(t_{i+1}), P_e)$ 为最小, 记为 $P(t_j)$, 且在有限次节点选择过程中, 必然存在一个严格趋近的节点位置 $P(t_j)$, 使得 $d(P(t_j), P_e) < d(P(t_{j-1}), P_e)$, $t_j \leq t_p$. 设 k 从 $P(t_{j-1})$ 选择 $P(t_j)$ 的概率为 p , 从算法步骤可知 $p = qp_1 + (1-q)p_2$, 其中, p_1 是恒用式(4)选择 $P(t_j)$ 的概率, p_2 是恒用式(5)选择 $P(t_j)$ 的概率. 使用式(4)或式(5)的概率由 q 确定. 由式(4)和引理 1 可知, $p_1 \equiv 1$, 在应用中, 一般取 $q_0 \geq 0.8$, 因而有 $qp_1 \geq 0.8$; 下面估算 p_2 的值. 由定义 4~6, $|Wk_i| \leq 5$, 可行点 $|Z| \leq 3$, 按平均概率, $p_2 = 1/3$; 又知 $\forall P(t_{i+1}) \in Wk_i(g_i)$, $(\max\{d(P(t_{i+1}), P_e)\} - \min\{d(P(t_{i+1}), P_e)\}) \in [\sqrt{2}, 2]$, 取最小值 1.414, 则有 $p_2 = 1.414 \times 1/3 \approx 0.47$. 由此可得 $p \approx 0.894$. 以上计算把各边上的信息素认为相等, 根据引理 1, 趋近节点所在边上的信息素大于非趋近节点边上的信息素, 因此, 实际上 $p > 0.894$. 故 k 在 s 次寻食中, 选择一个趋近的节点的概率为 $p = 1 - (1 - 0.894)^s$, 当 s 较大时, $p \approx 1$, 例如 $s = 10$ 时, $p \approx 0.9999999998$. 由此可得, k 在有限次寻食过程中选择一个严格趋近节点的概率基本为 1, 本定理由此得证. 证毕.

定理 2. $\forall k \in anti, i = 1, 2, k = 1, 2, \dots, m$, 其当前位置为 $P_k(t_i)$, $\exists k$ 选择一个趋近节点, 使得 $d(P_k(t_{i+1}), P_e) < d(P_k(t_i), P_e)$, 即 $P_k(t_i) \in f_i(T)$, 其中, $P_e \sim (g_{end} \text{ 或 } g_{begin})$.

证明. 根据算法, 每族由 4 只蚂蚁组成, 由定理 2 的证明可知, $\forall k$ 选择一个严格趋近的节点位置的概率 ≥ 0.894 , 两族 8 只蚂蚁选择一个严格趋近的节点的概率 $= 1 - (1 - 0.894)^8 \approx 1$. 证毕.

定理 3. 设存在满足定义 10 的映射 f , 一定存在有限大小的 t_m , 使得 $k_i \in anti_1, k_j \in anti_2, i, j = 1, 2, \dots, m$, 在 $t \leq t_m$ 时刻相遇.

由定理 1 和定理 2 可证, 证明从略.

定理 4. $\forall P_0, P_e \in FS$ 且 $P_0 \sim g_{begin}, P_e \sim g_{end}$, 若存在符合定义 10 的映射 f , 一定存在一个有限时刻 t_s , 使得 k 在 t_s 内从 g_{begin} 走到终点 g_{end} .

证明.

证法 1: 由定理 1~3 可证, 证明从略.

证法 2: 设 $P_j \in Path(P_0, P_e)$, 令 $q = |Path(P_0, P_e)|$, 为可行通道的节点数, 由式(5), k 从 P_i 选择 P_j 的最小下限概率为

$$p_{\min} = \frac{[\tau_{\min}]^{\beta} \eta_j(g_i)}{(|Z|-1)\tau_{\max}^{\beta} + \tau_{\min}^{\beta}} > 0 \quad (11)$$

k 找到有效路径的最小下限概率则为 p_{\min}^q , 令 $p(n)$ 为 t_s 时刻内经 n 次寻食后找到有效路径的概率, 则有 $p(n) = 1 - (1 - p_{\min}^q)^n$, $t_s \rightarrow \infty, n \rightarrow \infty, p(n) \rightarrow 1$ (12)

由定理 2 的证明可知, $p_{\min} \geq 0.894$, q 为有限个节点, 因此, 当 n 为一个不大的有限值时, $(1 - p_{\min}^q)^n$ 即为一个可忽略的极小值 ϵ , 使 $p(n) \approx 1$. 例如, 在 p_{\min} 取最保守的极限最小值的情况下, 当 $q = 20, n = 50$ 时, $p(n) \geq 0.999 \approx 1$. 由此本定理得证. 证毕.

本定理说明, 只要 AS 中客观存在可行通道, 必有一组蚂蚁在有限次寻食中找到对应的路径.

5.2 有动态障碍物时的安全性与可达性分析

根据本文方法可知, 有动态障碍物时仅影响避障局部路径规划, 且在局部规划过程是把预测碰撞点及邻域栅格作为静态障碍栅格处理的, 因此, 上述对静态环境的分析结论完全适于动态避障局部规划, 在此基础上, 进一步作出动态环境安全性分析.

引理 2. $\forall t \geq 0$, Rob 在 $View(P_R(t))$ 内最多只能探测到一个动态障碍物.

根据第 2 节环境约束条件, 有

$$d(p_d^1(t), p_d^2(t)) > 2r, \quad \forall p_d^1(t) \in D_i, \quad \forall p_d^2(t) \in D_j,$$

其中, $t \geq 0, i \neq j$, 易知引理成立.

根据该引理, 每次局部规划仅需绕开一个动态障碍物即可.

把上述所有定义、定理中的 OS 扩充为 OS' , $OS' = OS \cup CO'$, 则有如下定理.

定理 5. $\forall t \geq 0$, 若 $\forall CO'$ 使得 $Fp_{oe} \neq \emptyset$, 本次必能规划出局部避障路径.

本定理根据引理 2 和定理 4 可证.

定理 5 说明, 只要预测的碰撞点及其在运动轨迹上的邻域不把唯一可行的路径阻塞, 就一定可以规划出局部避障路径.

若考虑动态障碍物变速或突然停止等情况, 仅需令 $OS' = OS \cup D_i, D_i \in View$, 即能可靠地解决此类问题.

推论 1. 若每一次都能规划出局部避障路径, Rob 必能安全避碰地达到终点.

这一推论的成立是显然的, 栅格是以 Rob 的有效活动范围划分的, 而在碰撞预测中, 充分考虑了安全距离 ϵ , 且为了避免因计算误差而引起碰撞, 把预测的碰撞点及其在轨迹上的邻域都作为禁入域, 因此, 只要能规划出有效路径, 其行走总是安全的. 另一方面, 根据引理 2, 每次规划只需绕开一个障碍

物, 若每次都能绕过, 最终达到终点是很显然的.

这一推论主要说明, 只要动态障碍物不把唯一的可行通道阻塞, Rob 一定能安全避碰地到达终点.

5.3 信息存储策略及环境信息存储可行性分析

若机器人工作环境的范围很大, 一般不可能是微型机器人, 因此, 机器人完全可以配置辅存或大容量半导体盘, 环境信息可用数据文件或表格等数据结构存储. 因规划过程只使用与当前栅格邻域有关的极少量信息, 可用文件指针或目录页等方法实现快速存取, 因此, 即使工作范围在数十平方公里以上, 也不会存在信息存取的困难, 更不会成为算法的障碍. 对于微小型机器人, 其工作范围往往限于一个小的局部环境, 例如一个车间等. 设其工作场地范围纵横各 300m, 机器人的活动直径 $Ra = 0.3m$, 则每行 1000 个栅格, 共 1000 行. 用一个二进制位的 0 或 1 表示栅格是否为障碍栅格, 则每行用 125 字节, 全局信息共用 125KB. 这对于目前的半导体存储技术和微型计算机技术而言, 是不存在任何技术问题的, 即使用单片机系统, 扩展数百 KB 甚至更大容量半导体存储器也已是很成熟的技术, 而且本文算法非常简单, 即使用简单的单片机也能实现, 因此, 实现本算法在技术上是完全可行的.

6 仿真实验

为了验证算法的效果, 作者进行了大量的仿真实验, 结果都十分令人满意. 因动态避碰最终归为静态局部规划, 因此, 先进行静态环境下规划实验, 再仿真动态障碍物的避碰规划.

静态环境下的实验方法是取不同的栅格数, 随机生成障碍物并随机生成起始点和终止点, 用本算法进行路径规划. 实验表明, 只要有客观存在的可行路径, 一般情况, 蚂蚁只需寻食一至两次即可找到最优路径. 图 2 示出了在某障碍物环境下, 不同起点和终点时, 规划出的几条优化路径的示例^①. 图 2 结果显示, 不管路径有多复杂, 只要有通路存在, 本算法必能迅速规划出优化路径, 而且作者所做的其它实验结果表明, 即使栅格规模大幅增加, 也不影响这一结论.

若将定义 5 改为 $BR_i(g_i(x_i, y_i)) = \{g | g \in A, d(g, g_i) \leq \sqrt{2}\}$, 则 g_i 的邻域共有 9 个栅格 (图 1 中的粗线田字格示出了 g_{39} 的邻域), 这时, 将允许蚂蚁

① 由于原始实验记录图较大较多, 也为了说明问题, 本文对所有实验原始图进行了缩小、合并、加坐标及加标注等处理.

走斜线, 得到的仿真结果如图 3 中的线 1 所示. 但这时规划出的路径会从障碍格的角边穿过, 若不允许这种情况发生, 只需将可行域进行约束, 约束条件如下: $g_i(x_j, y_j) \in BR_i$, 若有 $g_j(x_j, y_j) \in OS$, 且有 $y_i = y_j, |x_i - x_j| = 1$, 或者 $x_i = x_j, |y_i - y_j| = 1$, 则 $g_l(x_l, y_l)$ 为禁入点, 其中, g_l 为 g_j 的邻格, 且满足条件: $y_l = y_j, |x_l - x_j| = 1$, 或者 $x_l = x_j, |y_l - y_j| = 1$ 且有 $|x_i - x_l| = 1, |y_i - y_l| = 1$. 按该约束条件进行

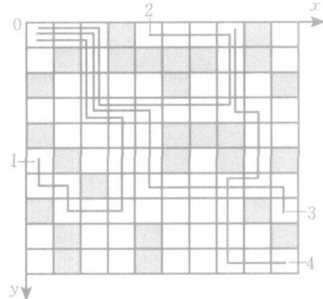


图 2 仿真实验结果 1

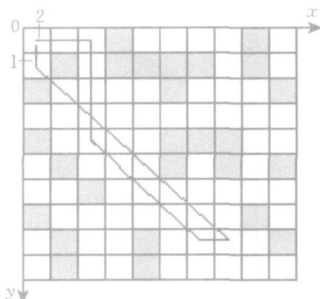


图 3 仿真实验结果 2

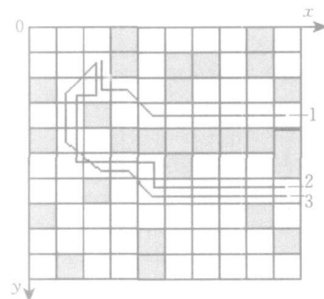
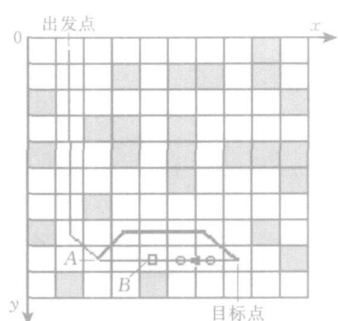


图 4 比对实验结果

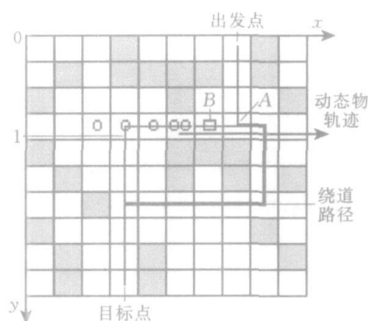
图 5 是仿真有动态障碍物情况下的避碰规划实验的例, 其中的圆圈表示动态障碍物, 细线为原规划出的全局优化路径, 粗线表示规划出的避障绕道路径. 在图 5 中, A 指向的点为 Rob 位置, B 指向的点为预测的机器人与动态物碰撞点; 在图(c)中, 粗线 1 是第一次预测碰撞而规划出的避障绕道路径, C 点是在该绕道路径中再次预测的碰撞点, 粗线 2 是

在绕道路径中又遇动态障碍物而第二次规划出的避障绕道路径. 实验表明, 碰撞点若不在 C 点而是 3 指向的位置, 规划出的路径为 4 所示的避障路径(设 d 点不为碰撞点).

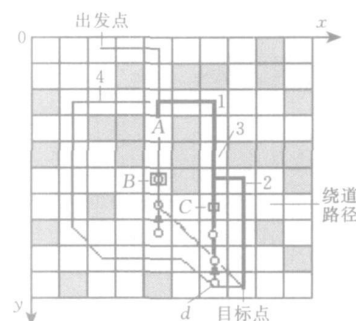
从实验结果可见, 即使是非常复杂的障碍情况, 只要有通路存在, 总能规划出有效路径且基本为最优优化路径.



(a) 在视野域附近绕道



(b) 大绕道情况



(c) 在绕道途中再遇动态物

图 5 动态避障规划实验结果

7 结 语

算法用两组各 m 只蚂蚁协作完成基于静态全局信息的路径优化规划, 其中, 一组蚂蚁置于机器人出发点, 另一组蚂蚁置于目标点, 两组蚂蚁相向并行搜索, 由两组共 $2m$ 只蚂蚁共同完成最优路径的搜索, 加上设有随机搜索策略, 保证了搜索的多样性, 使搜索不易陷于停滞. 只要有可行通道客观存在, 即使在非常复杂的地形环境, 也能迅速找到一条优化

路径. 由于采用的启发函数具有趋近导向作用和蚂蚁信息素的正反馈作用, 加上 $2m$ 只蚂蚁搜索的并行性, 因而规划效率极高. 在此基础上, 实时地进行动态障碍物运动轨迹预测及碰撞预测, 在机器人视野域范围, 将该动态物轨迹上预测碰撞点及附近的栅格视为静态障碍物, 从而用蚁群算法实时地作出动态避障局部规划, 绕过动态障碍物后再回到全局优化路径上或从当前位置重新规划到目的地的优化路径, 使得不仅安全避碰, 而且使机器人行走路径基本达最优. 理论和仿真结果表明, 该算法具有简单、

快速、高效、安全、实用的特点,即使在复杂的地形环境中也可迅速规划出优化路径,特别适于复杂地形环境下的机器人路径规划,并且能在机器人实时改变目标位置的情况下,迅速用本算法做出新路径的实时规划.

参 考 文 献

- 1 Wang Chun Miao, Soh Y. C., Wang Han, Wang Hui. A hierarchical genetic algorithm for path planning in a static environment with obstacles. In: Proceedings of the IEEE CCECE 2002, 2002, 3: 1652~1657
- 2 D'Amico A., Ippoliti G., Longhi S. A.. Radial basis function networks approach for the tracking problem of mobile robots. In: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2001, 1: 498~503
- 3 Weerayuth N., Chaiyaratana N.. Closed loop time optimal path planning using a multi objective diversity control oriented genetic algorithm. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2002, 6: 7
- 4 Bruce J., Veloso M.. Real time randomized path planning for robot navigation. In: Proceedings for the IEEE/RSJ International Conference on Intelligent Robots and System, 2002, 3: 2383~

2388

- 5 Zhang Chun Gang, Xi Yu Geng. Path planning of robots in environment without global information based on rolling windows. Science of China, Series E, 2001, 31(1): 51~58(in Chinese)
(张纯刚, 席裕庚. 全局环境未知时基于滚动窗口的机器人路径规划. 中国科学(E 辑), 2001, 31(1): 51~58)
- 6 Zhang Chun Gang, Xi Yu Geng. Analysis of less optimization of rolling path of moving robot. Science of China, Series E, 2002, 31(5): 713~720(in Chinese)
(张纯刚, 席裕庚. 移动机器人滚动路径规划的次优性分析. 中国科学(E 辑), 2002, 31(5): 713~720)
- 7 Xi Yu Geng, Zhang Chun Gang. Rolling path planning of robots in a kind of dynamic uncertain environment. Acta Automatica Sinica, 2002, 28(2): 161~174(in Chinese)
(席裕庚, 张纯刚. 一类动态不确定环境下机器人的滚动路径规划. 自动化学报, 2002, 28(2): 161~174)
- 8 Zhang Chun Gang, Xi Yu Geng. Rolling path planning of moving robots in a dynamic and uncertain environment and analysis of its safety. Theory and Application of Control, 2003, 20(1): 37~44(in Chinese)
(张纯刚, 席裕庚. 动态未知环境中移动机器人的滚动路径规划及安全性分析. 控制理论与应用, 2003, 20(1): 37~44)



ZHU Qing Bao professor. His research interests include artificial intelligence and intelligent control.

Background

This project is funded by a research funding of "211 Engineering" under the title "The Bionic Intelligence Algorithms and their Applications in Robot Control". In this project there are two major research contents: (a) The robot gait pattern control based on Central Pattern Generator (CPG). (b) Robot path planning and navigation based on Ant Colony Optimization. The second topic includes robot path planning and navigation in a familiar and unfamiliar surrounding consisting of moving objects, or an environment with multiple robots. The major achievements of this paper is that it pro-

vides a solution to the robot path planning, virtual ant collision prediction algorithms and strategies for dynamic collision avoidance, under a complex, static environment involving moving objects. The proposed approach can effectively solve issues of robot path planning in a cluttered environment. The research conducted by the authors in this field has led to various publications in Chinese Journal of Computer, Journal of Software, Control Theory and Applications, including more than 10 papers indexed by Engineering Index (EI).