

# 全局未知环境下多机器人运动蚂蚁导航算法\*

朱庆保<sup>+</sup>

(南京师范大学 计算机科学系,江苏 南京 210097)

## Ant Algorithm for Navigation of Multi-Robot Movement in Unknown Environment

ZHU Qing-Bao<sup>+</sup>

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

+ Corresponding author: Phn: +86-25-83064939, Fax: +86-25-83598636, E-mail: qbzhu@email.njnu.edu.cn

**Zhu QB. Ant algorithm for navigation of multi-robot movement in unknown environment. *Journal of Software*, 2006,17(9):1890–1898. <http://www.jos.org.cn/1000-9825/17/1890.htm>**

**Abstract:** The navigation problem of multi-robot movement in a complex and unknown environment is studied in the paper. A new algorithm, ants navigation algorithm, is presented. At the start the method maps the global targets onto the area near the border of the robot's eyeshot, and takes them as the local targets. Then two groups of ants will be cooperating to complete the search for the local optimal path in the robot's eyeshot. Based on these configurations, the algorithm can predict possible collision with other robots and execute subsequent avoidance plans. The local search will be executed by the algorithm repetitively whenever the robot progresses a step. So, the path of the robot will be altered dynamically, which makes the robot move on the global optimal path to the ending node. The simulation results indicate that the optimal path, which the robot moves on, can lead the robot to reach the end safely even in complicated geographical environment. The effect is very satisfactory.

**Key words:** multi-robot; navigation; path planning; unknown environment; ant algorithm

**摘要:** 研究了全局未知静态复杂环境下多机器人运动的导航问题,提出了一种新颖的蚂蚁导航算法.该方法将全局目标点映射到机器人视野域边界附近作为局部导航子目标,再由两组蚂蚁相互协作完成机器人视野域内局部最优路径的搜索,在此基础上进行与其他机器人的碰撞预测与避碰规划.机器人每前进一步都重复上述过程.因此,机器人前进路径不断地动态修改,从而在每条局部优化路径引导下,使机器人沿一条全局优化的路径到达目标点.仿真实验结果表明,即使在障碍物非常复杂的地理环境下,算法也能沿一条全局优化路径导航,且能安全避碰,效果十分令人满意.

**关键词:** 多机器人;导航;路径规划;未知环境;蚂蚁算法

中图法分类号: TP24 文献标识码: A

已有不少学者提出了解决未知环境下机器人(包括多机器人)运动导航或路径规划的方法和策略,诸如滚动窗口规划方法、遗传算法、各类定位、导航方法等<sup>[1-4]</sup>,取得了很多研究成果.然而,由于这类路径规划或导航属于 NP-Hard 问题,因此,不断寻求更佳的导航或规划算法就成为国内外相关领域的一个研究热点.特别是如何在

\* Supported by the National Science Foundation of Jiangsu Province of China under Grant No.BK2006218 (江苏省自然科学基金)

Received 2005-07-25; Accepted 2005-11-08

复杂环境下避免死锁和振荡,如何使机器人所走路径全局最优或较优,仍是有待解决的问题.

蚁群优化算法<sup>[5]</sup>是一种仿生算法,它的基本思想是模拟蚂蚁群体的觅食运动过程来实现寻优.常规滚动规划算法<sup>[1]</sup>需要通过复杂的逻辑判断来决定直线滚动方向,而蚂蚁算法可以集群体智能的作用,用很简单的算法即可找到一条曲折而复杂的导航路径.因此近年来已有学者探索性地将蚁群优化算法用于静态环境下的机器人路径规划<sup>[6]</sup>,也有学者研究了不考虑环境信息时的多机器人协作蚂蚁算法<sup>[7]</sup>.但是,在未知复杂环境下,机器人(包括多机器人)运动导航或路径规划蚁群算法仍是一个全新的课题.

根据目前的研究现状和不足,本文研究了一种全局未知静态复杂环境中有多多个移动机器人运动时的导航问题(并设各机器人间无通信功能),提出了一种基于目标点直线映射的滚动窗口蚂蚁算法和动态局部优化路径导航方法及机器人相互间的避碰策略等.算法首先将每个机器人的目标点映射到该机器人视野域附近,再由两组蚂蚁采用最近邻居搜索策略相互协作完成该机器人视野域内的最优路径的搜索,在此基础上进行与其他机器人之间的碰撞预测与避碰规划.机器人每前进一步,都要重复这一过程.因此,机器人前进路径不断地动态修改,在每条局部优化路径的导航下,可使机器人沿一条全局优化的路径到达终点.

### 1 环境描述

设有  $q$  个机器人  $Rob_i(i=1,2,\dots,q)$ ,每个  $Rob$  的运行速度为  $v_R$ ,行走步长为  $\delta$ .记  $AS$  为  $Rob$ s 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未知的静态障碍物  $Sb_1, Sb_2, \dots, Sb_n$ .导航的目的是使每个机器人由任意起点  $g_{begin}$  安全避碰地沿一条较短路径到达任意目标点  $g_{end}$ .

在任意时刻,各个  $Rob$  能探测到以当前位置为中心、 $r$  为半径区域内的环境信息,该区域称为该  $Rob$  的视野域  $View$ ;为了简便说明问题,设任意时刻任意两个机器人之间的最近距离  $>2r$ ,且在  $View$  范围近似为直线运动.

在  $AS$  中建立系统直角坐标系  $\Sigma_0$ ,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  $\delta$  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  $N_x=x_{max}/\delta$ ;每列的栅格数  $N_y=y_{max}/\delta$ .记  $g \in AS$  为任意栅格,  $A$  为  $AS$  中  $g$  的集合,记  $OS=\{o_1, o_2, \dots, o_m\} \subseteq A(m \geq n)$  为静态障碍栅格集,  $\forall Sb_i \in OS. \forall g \in A$  在  $\Sigma_0$  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.令  $C=\{1, 2, 3, \dots, M\}$  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.

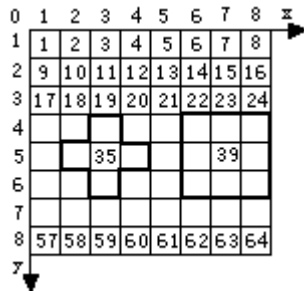


Fig.1 Relation between rating coordinate and serial numbers

图 1 栅格坐标与序号关系

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:

$$x_i = ((i-1) \bmod N_x) + 1, y_i = (\text{int})((i-1)/N_x) + 1 \tag{1}$$

其中,  $\text{int}$  为舍余取整运算,  $\bmod$  为求余运算.

$Rob_i$  在  $AS$  中的位置为记为  $P_R^i, \forall P_R^i$  在  $\Sigma_0$  都有确定的坐标  $(x, y), t$  时刻的位置表示为  $P_R^i(t)$ ,其坐标为  $(x_R(t), y_R(t))$ (为简便起见,省略了上标  $i$ ,以下相同).

## 2 多机器人运动蚂蚁导航算法

算法首先将全局目标点映射到 *Rob* 的视野域 *View* 的边界外作为局部导航子目标  $g_{sub}$ , 一组蚂蚁以 *Rob* 当前位置  $P_R(t)$  为蚁穴,  $g_{sub}$  为食物源, 另一组蚂蚁反之. 最终蚂蚁通过寻食绕开所有障碍物找到一条 *Rob* 在视野域内的最短路径. 在每条局部优化路径的导航下, 可使机器人沿一条全局优化的路径到达终点.

由于每一个机器人导航的方法是相同的, 因此, 下面以其中的一个机器人的导航方法为例, 并省去了所有表示机器人序号的下标及其所在位置上标. 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  $i$  个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为可行域, 记作  $FS$ ;  $\forall g \in A, g \in OS$ , 则称  $g$  为禁入节点, 所有禁入节点的集合称为禁入域, 记为  $NFS$ .

定义 3.  $k$  任意时刻所处的位置为  $P$ ,  $\forall P$  在  $S_0$  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  $g_j(x_j, y_j)$  的坐标相等, 则视  $P_i$  与  $g_j$  等价, 记作  $P_i \sim g_j$ .

定义 4. 任意栅格间的距离是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  $AS$  中的连线则称为一个边  $e_{ij}$ , 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.  $BR_i(g_i(x_i, y_i)) = \{g | g \in A, d(g, g_i) \leq S\}, i \in C$  称为  $g_i$  的邻域或蚂蚁  $k$  在  $g_i$  处的视野域. 其中,  $S = \{1, \sqrt{2}\}$ . 当只能走直线时,  $S=1$ ; 允许走斜格时,  $S=\sqrt{2}$ . 图 1 中的粗线范围示出了两种情况的邻域.

定义 6. 设  $t_i$  时刻  $k$  处于  $g_i, Wk_i(g_i(x_i, y_i)) = \{g | g \in BR_i(g_i(x_i, y_i)), g \notin OS, i \in C\}$  称为  $t_i$  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,  $|Z| < |Wk_i|$ .  $tabu_k$  中各位置点的连线称为  $P_R(t_i)$  到局部子目标点  $g_{sub}$  的通道, 记作  $Path(P_R(t_i), g_{sub})$ . 为简便起见, 也用  $Path(P_R(t_i), g_{sub})$  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.

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

定义 7.  $\eta_{1_i}(g_i) = D/d(g_i, g_{sub}), \eta_{2_j}(g_j) = D/d(g_j, g_R), \eta_{1_i}(g_i), \eta_{2_j}(g_j)$  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,  $D$  为权重常数,  $g_R$  为  $k$  所在栅格.

定义 8.  $k_1 \in ant_1, k_2 \in ant_2, k_1$  从  $g_R$  出发,  $k_2$  从  $g_{sub}$  出发. 经过  $n$  个时刻,  $k_1, k_2$  的位置为分别  $P_{k_1}, P_{k_2}$ , 若有  $d(P_{k_1}, P_{k_2}) \leq S$ , 则称  $k_1, k_2$  相遇.  $S$  的定义同定义 5.

### 2.1 全局导航算法步骤

根据以上的定义和约定, 多机器人运动导航蚂蚁算法的步骤描述如下:

Step 1. 产生任意的起点  $g_{begin}$  和终点  $g_{end}$ , 并对有关参数进行初始化;

Step 2. 将  $g_{end}$  目标点映射到 *Rob* 视野域 *View* 边界外作为局部导航子目标  $g_{sub}$  (映射算法见第 2.2 节). 若有  $g_{sub}$  与  $g_{end}$  相同, 导航结束; 否则, 转 Step 3;

Step 3. *Rob* 在  $P_R(t)$  进行环境探测, 测取  $Sb_i$  的坐标, 其中  $P_R(t) \sim g_R$ , 并探测有无其他机器人闯入;

Step 4. *Rob* 根据探测到的  $Sb_i, i=1, 2, \dots, n$ , 用多蚂蚁协作算法迅速规划出不考虑其他机器人闯入时的局部优化路径  $F_i$  (见第 2.3 节). 该路径对应的结点集合为  $Path(p_R, p_b)$ , 其中,  $p_b \sim g_{sub}, p_R \sim P_R(t)$ ;

Step 5. 若无其他机器人  $Rob_j$  闯入, 转 Step 7; 否则, 判定本机器人和  $Rob_j$  的优先级 (见第 2.4 节): 若  $Rob_j$  优先级低, 转 Step 7; 否则, 测出  $Rob_j$  的方向、运动轨迹并预测出 *Rob* 沿 Step 4 规划出的  $F_i$  向  $g_{sub}$  前进时与  $Rob_j$  的碰撞点, 碰撞点的集合记作  $C_0$ ;

Step 6. 根据  $C_0$  情况和避碰策略进行避碰处理 (见第 2.4 节). 当碰撞点不能回避时, 将  $C_0$  作为静态障碍物,

再用多蚂蚁协作算法重新规划避碰局部导航优化路径 LFS;

Step 7. 机器人沿着规划出的局部最优路径向  $g_{sub}$  前进一步;

Step 8. 返回 Step 2 重复上述过程.

### 2.2 局部导航子目标的映射算法

在  $t_i$  时刻,Rob 的视野域为  $View(P_R(t_i))$ ,若有  $d(P_R(x_i,y_i),g_{end}(x_e,y_e))=r+\delta$ ,则取  $g_{sub}=g_{end}$ ,此时,即为  $g_{sub}$  与  $g_{end}$  相同.否则,令

$$d(P_R(x_i,y_i),g_{sub}(x_j,y_j))=r+\delta \text{ 且满足 } \min d(g_{sub}(x_j,y_j),g_{end}(x_e,y_e)) \quad (4)$$

其中, $r$  和  $\delta$  均为已知.因为  $g_{sub},g_{end}$  间的直线距离最短,因此,满足上述条件的  $g_{sub}$  点必定在过  $P_R$  与  $g_{end}$  点的直线上.由此直线方程并根据式(4)可求出  $g_{end}$  映射到子目标点  $g_{sub}$  的坐标  $x_j,y_j$ .很显然,得到的  $g_{sub} \notin View(P_R(t_i))$ .为了在  $View$  边界  $g_{sub}$  方向上障碍物很多,的情况下,蚂蚁仍能找到一条导向子目标点的路径,假定  $g_{sub}$  任意  $View$  边界间的任何栅格均无障碍物,且规定  $g_{sub}$  与  $View$  任意边界的距离为 1(1 个单位代表  $\delta$ ),如图 2 所示.

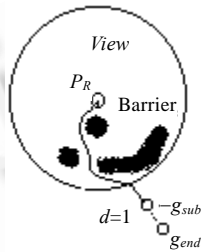


Fig.2 Local navigation path  
图 2 局部导航路径示意图

### 2.3 局部导航路径多蚂蚁协作规划算法

由于两组蚂蚁除了采用的启发函数、出发点不同外,算法完全相同,为了减少变量的下标,下面以蚂蚁家族 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.算法步骤如下:

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

Step 2.  $\forall k$ ,以当前节点  $g_i \in FS$  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  $g_j \in Wk_i(g_i)$ ,  $g_j \notin tabu_k$ ,节点选择算法如下:

Case 1. 若有  $g_i \in BR_R(g_R)$ ,则  $\forall k$  选择一个节点  $g_j^k \in Wk_R(g_R)$ ,且令  $g_j^k \neq g_j^k \in Wk_R(g_R)$ ,  $i,j=1,2,\dots,m$ ,其中,上标  $k_i,k_j$  表示不同蚂蚁选择的不同节点.该约束是为了保证搜索多样性;

Case 2. 若  $g_i \notin BR_R(g_R)$ , $\forall k$  根据式(5)或式(6)选择下一个节点  $g_j \in Wk_i(g_i)$ ,  $g_j \notin tabu_k$

$$j = \begin{cases} \arg \max \{ [\tau_{ij}(n)]^\beta \eta_j(g_i) \} & \text{if } q \leq q_0 \\ S & \text{else} \end{cases} \quad (5)$$

$$p_{ij}^k(n) = \frac{[\tau_{ij}(n)]^\beta \eta_j(g_i)}{\sum_{q \in |Z|} [\tau_{iq}(n)]^\beta [\eta_q(g_i)]} \quad j \notin tabu_k \quad (6)$$

式中: $j \in C,k$  所选  $g_j$  的节点序号,在此省略了上标  $k$ ;  $S$  是由式(6)决定的节点号; $q$  是随机数( $0 < q \leq 1$ ); $q_0$  是为了防止出现停滞而设的随机搜索策略所需参数; $\eta_j(g_i)$  是由定义 7 给出的启发信息; $\beta$  是在边  $e_{ij}$  上残留信息的重要程度; $p_{ij}^k(n)$  是  $k$  在  $n$  代寻食过程中由节点  $i$  转移到节点  $j$  的概率, $i,j \in C$ .

将  $j$  加入禁忌表  $tabu_k$ .

Step 3. 每一只蚂蚁选择完一个节点即走完一个边后,按式(7)进行局部信息更新;

$$\tau_{ij}(n+1) = (1-\rho)\tau_{ij}(n) + \rho \Delta \tau_{ij}^k \quad (7)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q_1}{l_{jb}}, & \text{当蚂蚁}k\text{走过边}e_{ij}; l_{jb} = \sum_{l=1}^w d_l, \\ 0, & \text{否则} \end{cases}$$

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

式中,  $Q_1$  为常数;  $\rho$  表示信息素消散系数;  $\tau_{\min}$  是设定的最小值;  $w$  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.

Step 4.  $\forall k, k=1, 2, \dots, m$ , 选择第  $j$  个节点后, 按照定义 8 定义的条件, 检查是否已有蚂蚁相遇: 若有, 则转 Step 5; 否则, 返回到 Step 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.

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

Step 6. 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(8)调整:

$$\tau_{ij}^{new} = (1-\alpha)\tau_{ij}^{old} + \alpha\Delta\tau_{ij}; \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \tag{8}$$

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

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

Step 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.

### 2.4 与其他机器人的碰撞预测与避碰策略

设  $t_i$  时刻  $Rob$  探测到障碍物  $Xb_i \in \text{View}(P_R(t))$ , 若  $Xb_i \notin \text{View}(P_R(t)) \cap OS$ , 则为其他  $Rob_i$ , 测其坐标位置为  $P_R^i(x_i(t), y_i(t))$ , 简记为  $R(x(t), y(t))$ , 经过一个较小的时刻  $\Delta t$  后, 其位置坐标为  $x_i(t_i + \Delta t), y_i(t_i + \Delta t)$ , 由此可知其运动方向, 可用式(9)估算其速度, 用式(10)估计  $Rob_i$  在  $\text{View}$  内的近似的直线运动轨迹.

$$v_R^i = d(R(x(t + \Delta t), y(t + \Delta t)), R(x(t), y(t))) / \Delta t \tag{9}$$

$$y = kx + c \tag{10}$$

其中,  $k = (y(t + \Delta t) - y(t)) / (x(t + \Delta t) - x(t))$ ,  $c = y(t) - x(t) / (x(t + \Delta t) - x(t))$ .

当  $Rob$  与  $Rob_i$  互相探测到时, 仅需其中的一个进行避碰处理即可. 为了达到这一目的, 为各个机器人设定动态优先级, 优先级按机器人所在方位确定: 其  $x$  坐标最大者, 优先级最高; 当  $x$  坐标相等时, 则  $y$  坐标最大者, 优先级最高.  $Rob$  与  $Rob_i$  比较其坐标值, 由优先级低者进行碰撞预测与避碰处理.

令  $Rob$  在  $t_i$  时刻从  $P_R(t_i)$  以步长  $\delta$ 、速度  $V_R$  沿  $\text{Path}(p_R, p_b)$  虚拟前进,  $t_v$  时刻到达  $g_{sub}$ , 令  $\Delta t = \delta / V_R$ , 记  $\tau = \{t_i, t_i + \Delta t, t_i + 2\Delta t + \dots + t_v\}$ , 其位置集合记为  $\Gamma(P(t_i), P(t_v)), \Gamma(P(t_i), P(t_v)) \subseteq \text{Path}(p_R, p_b)$ .

在  $t_i$  时刻,  $Rob$  获知  $Rob_i$  以速度  $v_R^i$  沿式(10)轨迹运动; 通过式(10)获得  $Rob_i$  各位置及坐标  $R(x(t_i + k\Delta t), y(t_i + k\Delta t)), k=1, 2, \dots, h$ . 记  $B_I = \{R(t_i), R(t_i + \Delta t), R(t_i + 2\Delta t), \dots, R(t_v)\}$  为  $Rob_i$  运动轨迹的位置集合,  $B_i \in \text{View}$ .  $Rob_i$  与  $Rob$  的相对位置可分解为以下几种情况,  $Rob$  可据此判断是否相碰:

情况 1.  $B_i \cap \Gamma(P_R(t_i), P(t_v)) = \emptyset$ , 且满足  $\forall t$  都有  $d(R(t), P_R(t)) \geq \varepsilon, t \in \tau$ , 其中,  $\varepsilon$  是实用中根据  $Rob_i$  和  $Rob$  体积大小而设定的安全阈值;

情况 2.  $B_i \cap \Gamma(P_R(t_i), P(t_v)) = \emptyset$ , 且满足  $\begin{cases} d(R(t), P_R(t)) \geq \varepsilon, & \text{当 } t \neq t_j, t, t_j \in \tau; \\ d(R(t), P_R(t)) < \varepsilon, & \text{当 } t = t_j, t, t_j \in \tau; \end{cases}$

情况 3.  $B_i \cap \Gamma(P_R(t_i), P(t_v)) = \emptyset$ , 且满足  $\begin{cases} \forall d(R(t), P_R(t)) \geq \varepsilon, & \text{当 } t \in \tau_1, \tau_1 \in \tau \\ \forall d(R(t), P_R(t)) < \varepsilon, & \text{当 } t \in \tau_2, \tau_2 \in \tau \end{cases}$ , 其中,  $\tau_1 + \tau_2 = \tau$ ;

情况 4.  $B_i \cap \Gamma(P_R(t_i), P(t_v)) = \emptyset$ , 且满足  $\begin{cases} d(R(t), P_R(t)) \geq \varepsilon, & \text{当 } t \in \tau_3, \tau_3 \in \tau \\ d(R(t), P_R(t)) < \varepsilon, & \text{当 } t \in \tau_4, \tau_4 \in \tau \end{cases}$ , 其中,  $\tau_3 + \tau_4 = \tau$ .

以上 4 种情况可根据式(9)、式(10)以及  $V_R, v_R^i, I$  集合中的已知坐标值进行判定.

对于情况 1:  $Rob_i$  的路径与  $Rob$  路径互不相干,  $Rob$  可按原路径继续前进.

情况 2 表示  $Rob_i$  在  $t_j$  时刻穿越  $Rob$  的路径. 此种情况,  $Rob$  只要在  $t_j - n\Delta t$  时刻等待一段时间  $t_w$  即可, 其中,

$$n\Delta t = \eta(\Delta/2 + R_d)/V_R, t_w = \rho(n\Delta t + 2R_d/v_d^i).$$

式中,  $R_d$  为  $Rob_i$  最大半径;  $\eta, \rho$  为安全系数.

对于情况 3:  $Rob_i$  至少有一部分时间( $\tau_2$ )贴近  $Rob$  路径运行, 考虑两者体积时, 将有碰撞可能.

情况 4 表示  $Rob_i$  至少有一部分时间将在  $Rob$  的路径上运行.

对情况 3 或情况 4, 从  $t_x$  开始, 记录所有预测碰撞点, 记其集合为  $C_o$ . 将  $C_o$  作为禁入集合, 重新用第 2.3 节算法进行局部路径规划,  $Rob$  按新规划的路径前进并结合避碰策略即可安全避碰.

### 3 收敛性分析

定义 9. 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  $f: T \rightarrow View$ , 使得  $f(t_0) = P_0, f(t_e) = P_e$ , 则称  $f$  是  $View$  中客观存在的一条可行路径,  $f(t_0), f(t_e)$  分别为可行路径  $f$  的起点与终点.

定义 10.  $\forall P_0, P_e \in FS$ ,  $FS$  中所有从  $P_0$  到  $P_e$  的可行路径的集合记为  $F$ .

定义 11. 设局部路径  $f_i \in F$ , 若  $\forall t_1, t_2 \in [t_0, t_n], t_2 = t_1 + \Delta t$ , 都有  $d(P(t_1), P(t_2)) = \min\{d(P(t_1), P, P \in View)\}$  且满足  $L(P(t_2), P_e) < L(P(t_1), P_e)$ , 则称  $f_i(t_2) = P(t_2)$  是向  $P_e$  严格趋近的节点位置. 其中,  $\Delta t$  为  $k$  从当前节点走到下一个节点所需时间,  $P_e \sim g_{sub}$  或  $P_e \sim g_R$ .

引理 1. 对  $\forall e_{ij}$ , 都有  $\tau_{\min} \leq \tau_{ij} \leq \tau_{\max}$ , 其中,  $\tau_{\max} = \max\left\{\frac{\tau_m}{\rho}, \frac{\tau_m}{\rho} \left[\frac{(1-\alpha)\rho + \rho}{(1-\alpha)\rho + \alpha}\right]\right\}, \tau_m = \max(\rho\Delta\tau_{ij}, \alpha\Delta\tau_{ij})$ .

证明: 设某代寻食得到最优路径为  $f_o$ , 根据算法步骤和式(5)~式(8), 可分为以下 3 种情况:

(1) 边  $e_{ij}$  蚂蚁  $k$  从未走过, 该边上的信息素只消散, 不增加, 根据式(7)有

$$\tau_{ij}^{\min}(t) = (1-\rho)^t \tau_0, t \rightarrow \infty, \tau_{ij} \rightarrow 0, \tau_{ij} = \tau_{\min}(\text{被强制到 } \tau_{\min}) \quad (11)$$

(2)  $k$  走过  $e_{ij}$ , 但  $e_{ij} \notin f_o$ , 此时, 该边上的信息素只进行局部更新, 根据式(7)有

$$\tau_{ij}^{\max}(t) = (1-\rho)^t \tau_0 + \sum_{i=1}^t (1-\rho)^{t-i} \tau_m, \tau_m = \rho\Delta\tau_{ij}, t \rightarrow \infty, \tau_{ij}^{\max}(t) \rightarrow \frac{1}{\rho} \tau_m \quad (12)$$

(3)  $k$  走过  $e_{ij}$  且  $e_{ij} \in f_o$ , 此时, 该边上的信息素进行局部和全局更新, 根据式(7)、式(8)有

$$\tau_{ij}^{\max}(t) = (1-\rho)^t (1-\alpha)^t \tau_0 + \sum_{i=1}^t (1-\rho)^{t-i} (1-\alpha)^{t-i} [(1-\alpha)\tau_m + \tau_m], t \rightarrow \infty, \tau_{ij}^{\max} \rightarrow \frac{\tau_m}{\rho} \left[\frac{(1-\alpha)\rho + \rho}{(1-\alpha)\rho + \alpha}\right] \quad (13)$$

其中,  $\tau_m = \max(\rho\Delta\tau_{ij}, \alpha\Delta\tau_{ij}^k)$ .

比较以上 3 种情况可见: 对  $\forall e_{ij}$ , 都有  $\tau_{\min} \leq \tau_{ij} \leq \tau_{\max}$ .

定理 1. 若在机器人视野域内存在满足定义 9 的映射  $f$ , 当  $0 < q_0 < 1$  时, 在局部路径规划中, 经过一个有限时间  $t_m$ , 算法能收敛到最优解.

证明: 若存在满足定义 9 的映射  $f$ , 必然存在可行解, 设任意一个一般解  $f_i \in F$  包括  $x$  个趋近的节点、 $y$  个非趋近的节点. 设选择一个任意趋近和任意非趋近节点的概率分别为  $R$  和  $r$ , 下面估算  $R$  和  $r$ .

根据算法, 当  $q \leq q_0$  时,  $k$  根据式(5)选择节点; 否则, 根据式(6)选择节点. 这些节点既可以是趋近节点也可能是非趋近节点. 因此, 设  $p_5^x, p_6^x$  分别是  $q \leq q_0$  和  $q > q_0$  时分别选择一个趋近节点的概率,  $p_5^y$  和  $p_6^y$  分别是  $q \leq q_0$  和  $q > q_0$  选择一个非趋近节点的概率, 根据算法有

$$R = q_0 p_5^x + (1 - q_0) p_6^x; r = q_0 p_5^y + (1 - q_0) p_6^y.$$

由引理 1、定义 7 和式(5)可知,  $p_5^y = 0, p_5^x = 1$ . 根据引理 1,  $\forall \tau > \tau_{\min} > 0$ ; 根据定义 7,  $\eta_{\min} > 0$ . 因此, 由式(6)计算出的选择解  $f_i$  上任意节点的极限最小概率为

$$p_{\min} \geq \frac{\tau_{\min}^{\beta} \eta_{\min}}{\sum_{h \in \{Z\}^{-1}} \tau_{\max}^{\beta} \eta_{\max} + \tau_{\min}^{\beta} \eta_{\min}} > 0.$$

因而有  $p_6^x > 0, p_6^y > 0$ , 由此推知  $R > 0, r > 0$ .  $k$  经过  $t$  代寻食至少有一次选择一个最优解的概率为

$$P(t) \geq 1 - (1 - R^x r^y)^t, \text{ 当 } t \rightarrow \infty \text{ 时, } P(t) \rightarrow 1 \quad (14)$$

即当  $t = t_m$  是一个较大的值时有:  $P(t) = 1 - \varepsilon \approx 1$ .

本定理说明, 当  $0 < q_0 < 1$  时, 无论信息素如何变化, 经过一个充分长的时间, 算法总能够收敛到最优解. 可以证明, 当考虑信息素时, 能使收敛速度有较大提高(篇幅所限, 证明从略).

**定理 2.** 设存在满足定义 9 的映射  $f$ , 一定存在有限大小的  $t_s$ , 使  $k_i \in \text{ant}_1, k_j \in \text{ant}_2 (i, j = 1, 2, \dots, m)$  在  $t \leq t_s$  时刻相遇. 根据定理 1, 这一结论是显然的.

**推论 1.** 用两组蚂蚁进行相向搜索, 可使搜索速度大幅提高.

**证明:** 参考定理 1 的证明过程, 每组蚂蚁只需搜索  $x/2$  和  $y/2$  个节点. 因此, 用  $x/2$  和  $y/2$  替换式(14)中的指数  $x, y$ , 很显然, 得到优化解的概率将有较大提高.

**引理 2.**  $\forall t \geq 0$ ,  $Rob$  在  $View(P(t))$  内最多只能探测到一个其他机器人.

根据第 2 节环境约束条件, 有  $d(p_i^x(t), p_j^x(t)) > 2r$ , 其中,  $t \geq 0, i \neq j$ . 易知引理成立.

**推论 2.** 若每一次都能规划出局部优化避碰路径, 则  $Rob$  必能安全避碰地沿一条优化路径达到目标点.

因为  $Rob$  每走一步都要规划局部导航路径, 使前进路径动态修改, 若每次都能绕开其他机器人和障碍物规划出局部优化路径, 则  $Rob$  能沿一条优化的路径达到目标点是很显然的, 其直观结果可参见第 4 节.

下面粗略估算一下收敛速度. 为方便且不失一般性, 将  $View$  等价于一个  $n \times n$  的方形区域. 根据定义 6、 $View$  半径和第 2.3 节算法, 用式(14)计算每组蚂蚁寻优概率的总节点数为  $((n+1-2)/2)/2$ . 由于有启发函数的引导加上有明确的前进目标, 因此, 一条优化路径上的节点多数为趋近节点. 在计算中, 通常取  $q_0 \geq 0.85$ , 当  $n=9, t=5$  时, 用式(5)选择节点的概率取最保守值 0.1, 根据式(14), 由两组蚂蚁选择一个优化路径的概率  $p(t) \approx 0.999 \approx 1$ . 在这种最保守的估算且在未考虑  $2m$  只蚂蚁并行搜索及信息素作用的情况下, 也只需搜索几代即可收敛到最优解. 仿真实验也证明了这一结论.

#### 4 计算机仿真实验

不失一般性, 假定任意两相邻栅格的距离为 1 并设机器人的视野域为  $9 \times 9$  的方形. 为了实验算法的效果, 设计了如图 1 所示的复杂地形, 其中黑色为障碍格. 图 3 给出了用本文算法得到的一条路径. 很显然, 这条路径不仅安全避碰, 且基本上全局最优. 图 4 示出了在相同条件下, 用滚动规划方法未能达到终点.

图 5\* 中的细线显示出了  $Rob$  与其他机器人无碰撞时规划出的路径, 粗线则显示出了与另一个机器人有碰撞时改走的路径. 图 5 中, 小圆形表示另一个机器人, 箭头表示其方向, 小方形表示预测的碰撞点.

为了进一步分析本文算法能得到全局优化的导航路径的原理, 图 6 给出了图 3 实验过程中的分解子过程, 其中, 粗线框表示  $Rob$  视野域范围. 图 6(a)~图 6(d) 分别示出了  $Rob$  在起点和从起点分别前进 1 步、2 步、6 步后规划出的局部优化路径. 根据算法假定, 子目标点映射在视野域边界外, 且假定子目标点与视野域任意边界间为无障碍物. 因此, 子目标可以映射到视野域外的障碍格如图 6(b) 所示, 路径也可以穿越视野域外的障碍格, 如图 6(c) 所示. 从图 6 可以看出,  $Rob$  仅沿每次规划的局部路径走一步, 但它构成了全局的一种导航趋势, 从而使得不仅能避碰, 且所走路径全局最优或较优. 与滚动规划方法相比, 滚动规划每次仅能沿直线滚动, 在复杂的环境中

\* 为了直观和压缩篇幅, 图 5 由两幅实验图合成, 并加注了碰撞点、机器人位置及箭头.

必须进行繁琐的逻辑判断,否则易引起死锁和振荡,且很难达到全局最优.本文方法蚂蚁每次不是只走一步,也不是计算 Rob 当前位置到子目标点的直线距离,而是在复杂的障碍环境中找到一条通向子目标点的非常曲折的绕道局部避碰优化引导路径,因而有效地解决了滚动规划存在的一些问题.

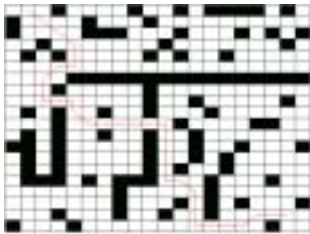


Fig.3 Global optimal path is from our algorithm  
图3 本算法得到的优化路径

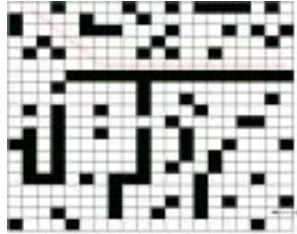


Fig.4 Global path is from rolling algorithm  
图4 滚动规划得到的路径

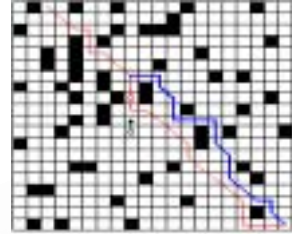
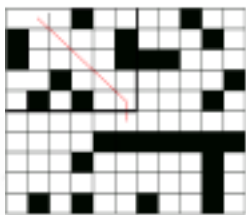
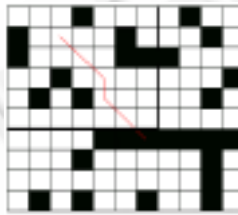


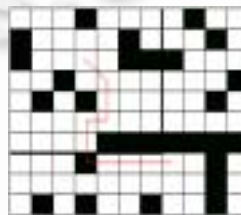
Fig.5 Optimal path is from forecasted collision  
图5 避碰优化路径



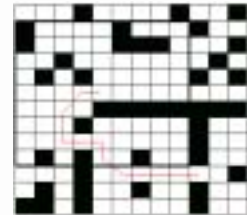
(a) First local path  
(a) 第1个局部路径



(b) Second local path  
(b) 第2个局部路径



(c) Third local path  
(c) 第3个局部路径



(d) Sixth local path  
(d) 第6个局部路径

Fig.6 Local path resolving drawing  
图6 局部路径分解图

## 5 结 语

全局环境未知时的多机器人运动导航的难点在于很难保证机器人所走路径全局最优或较优,甚至易引起死锁和震荡等.本文将最终目标点映射到机器人视野域外部作为局部导航子目标,并将该子目标与视野域边界视为无障碍物且其距离为 1,然后用两组各  $m$  只蚂蚁分别置于机器人和子目标位置相向并行搜索,实时规划出一条可能是曲折复杂的局部趋势导航路径.机器人每前进一步,都要进行场景探测,预测与其他机器人的碰撞,并重新进行局部导航路径的规划,因而其前进路径动态修改.在每个局部最优路径的引导下,使算法不仅能免于死锁、震荡,且能使机器人所走路径全局最优或较优.同时,算法还设计了优先级,由机器人自动识别,由优先级低的机器人进行避碰规划,可使多个机器人在同一空间安全地各自工作或协作工作.实验结果表明,本算法具有简单、速度快、效果好等特点,非常适于异常复杂环境的多机器人运动导航.作者已在做的进一步工作是研究动态不确定环境下多机器人运动蚂蚁导航问题,相信在这一更为复杂的环境中,仍然可以取得很好的导航效果.

## References:

- [1] Zhang CG, Xi YG. Mobile robot path planning based on rolling windows in the unknown environment. *Science in China (Series E)*, 2001,31(1):51-58 (in Chinese with English abstract).
- [2] Zhang CG, Xi YG. Robot rolling path planning based on locally detected information. *ACTA Automatica Sinica*, 2003,29(1):38-44.
- [3] Bruce J, Veloso M. Real-Time randomized path planning for robot navigation. In: *Proc. of the Intelligent Robots and System*, 2002. IEEE/RSJ Int'l Conf., Vol 3. 2002. 2383-2388.
- [4] Sun SD, Lin M. Path planning of multi-mobile robots using genetic algorithms. *ACTA Automatica Sinica*, 2000,26(5):673-676 (in Chinese with English abstract).
- [5] Dorigo M, Di Caro G. Ant colony optimization: A new meta-heuristic. In: *Evolutionary Computation*, 1999 (CEC 99). *Proc. of the 1999 Congress*, Vol 2. Washington: IEEE Press, 1999. 1474-1477.



- [6] Fan XP, Luo X, Yi S, Yang SY, Zhang H. Optimal path planning for mobile robots based on intensified ant colony optimization algorithm. In: Proc. of the Int'l Conf. on Robotics, Intelligent Systems and Signal Processing 2003. IEEE, 2003. 131-136.
- [7] Ding YY, He Y, Jiang JP. Multi-Robot cooperation method based on the ant algorithm. In: Proc. of the Swarm Intelligence Symposium, 2003 (SIS 2003). IEEE, 2003. 14-18.

附中文参考文献:

- [1] 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.
- [4] 孙树栋,林茂.基于遗传算法的多移动机器人协调路径规划.自动化学报,2000,26(5):673-676.



朱庆保(1955 - ),男,山东沂源人,教授,主要研究领域为人工智能,智能控制。

## 第 20 届国际科技数据委员会(CODATA)国际学术会议通知

第 20 届 CODATA 国际学术会议将于 2006 年 10 月 23 日至 25 日在北京国际会议中心召开,会议的主题是“信息社会的科学数据与知识”。本次大会旨在推动数据应用和知识提升,讨论如何利用信息社会的福祉把握机遇、迎接挑战。

本次大会由中国科学院院长路甬祥院士和国际 CODATA 主席 Shuichi Iwata 教授共同担任会议指导委员会主席,由国际科技数据委员会(CODATA)、中国科学院、中国科学技术协会、中国社会科学院和国际科学院组织(IAP)联合主办,由中华人民共和国科学技术部和中国国家自然科学基金委员会协办,由 CODATA 中国全国委员会、中国科学院计算机网络信息中心承办。国际科学数据委员会(CODATA)是国际科学联合会(ICSU)组建的跨学科的科学委员会,其宗旨是提高重要科技数据的质量、管理水平和共享程度。2006 年适逢 CODATA 成立 40 周年,预计届时全球 600 余名专家学者将齐聚北京,使此次会议成为 CODATA 历史上一次里程碑式的盛会。届时科技部部长徐冠华院士、医科院院长刘德培院士、ICSU 前主席 Jane Lubchenco 教授以及微软公司技术计算副总裁 Tony Hey 教授等近 10 位国内外著名学者将作大会主题报告。

第 20 届 CODATA 国际学术会议涉及的部分前沿问题包括:数字鸿沟、数据归档和存储、数据共享政策和法规、数据可视化和多媒体技术、数据挖掘和知识发现、e-science 和网格、数字图书馆、元数据和互操作、科学数据与科普、eGY 和 IPY(电子地球物理年和国际极年)、科学数据在教育、经济、环境和社会领域的应用、科学数据在科学研究中的应用服务。

会议在全球范围内征集论文。征文内容涉及数据归档、存储、处理、传输、应用和共享服务的技术和规范,与数据领域相关的信息技术,科学数据在科学研究和创新中的应用以及科学数据在自然科学(包括:天文学和气象学、能源和环境、物理学和化学、地球科学、材料学、灾害减轻和预防、天然气水合物、神经信息学、生物信息学、生物多样性、医药卫生等)和社会科学中的应用。

详情请见会议网站:<http://www.codata.cn/2006/index.html>。

本次大会将为全球数据领域的专家提供一个共享和交流的平台。