

基于二进制粒子群算法的移动机器人路径规划

王 艳, 朱庆保

(南京师范大学 计算机科学与技术学院, 江苏 南京 210097)

[摘要] 由于用 PSO 进行机器人路径规划的研究尚局限于用连续模型规划连续描述的环境中的路径, 使算法受到一定的局限性. 为此, 研究了一种全新的基于栅格法的机器人路径规划二进制粒子群算法. 首先用栅格法描述机器人工作环境, 在此基础上, 将机器人路径表示为粒子位置的二进制编码, 并以路径长度为适应值, 产生初始种群后, 再对粒子位置和速度进行更新, 经过多次迭代, 即可获得从起始点到目标点的一条全局最优路径. 该方法模型简单, 算法复杂度低, 收敛速度快, 计算机仿真实验证明了该方法的有效性和先进性.

[关键词] 移动机器人, 路径规划, 二进制编码粒子群算法, 栅格法

[中图分类号] TP 242 [文献标识码] A [文章编号] 1672-1292(2009)02-0072-07

Path Planning for Mobile Robot Based on Binary Particle Swarm Optimization

Wang Yan, Zhu Qingbao

(School of Computer Sciences, Nanjing Normal University, Nanjing 210097, China)

Abstract At present, the study of the robot path planning with Particle Swarm Optimization (PSO) is still confined to use the continuous model to plan the path with continuous method, so the algorithm has certain limitations. Then a discrete binary version of the PSO for path planning of robot based on grid method is proposed. The first step of the algorithm is to make the work environment of robot with grid method, based on which robot route is expressed as binary code of the particle location, and the length of the path is taken as the fitness value, and after the initial population is generated, the location and velocity of particles are updated. Then the discrete binary version of the PSO is introduced to obtain a optimized path between starting-point and goal-point through many iterations. This algorithm is simple in model, low in algorithm complexity and rapid in convergence. The simulation results are provided to verify effectiveness and practicability of the method.

Key words mobile robot, path planning, binary particle swarm optimization, grid method

移动机器人路径规划是移动机器人研究的基本问题之一,它是指在有障碍物的工作环境中,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,且所走路径较短.已有路径规划方法包括启发式图搜索法、人工势场法、神经网络法等,这些算法使问题的求解速度和规模有了不同程度的提高,但均存在一定的局限性.近年来,不少学者采用神经网络、蚁群算法^[1]等进行机器人路径规划,用仿生算法进行机器人路径规划成为明显的发展趋势.为了进一步探索新的机器人路径规划方法,近年也有学者尝试用粒子群算法(Particle Swarm Optimization, PSO)进行路径规划^[2-3],且能够在较短时间内获得全局最优或基本最优的一条路径,取得了显著的成果.然而该方法主要采用的是传统的粒子群算法,而传统的粒子群算法是一种用于连续优化的数学模型,这就限制了机器人的工作环境不能是离散模型而必须是连续模型,因而使该算法具有一定的局限性,即不适合基于离散环境模型下的机器人路径规划.

为弥补这一不足,受二进制编码粒子群算法(Binary Particle Swarm Optimization)^[4]的启发,并以此为基础,本文提出一种机器人工作环境为离散模型时的全新机器人路径规划的粒子群算法.该算法采用最简便的栅格法对机器人工作环境进行建模,在此基础上,将机器人路径表示为粒子位置的二进制编码,并以

收稿日期: 2008-04-03

基金项目: 国家自然科学基金(60673102)和江苏省自然科学基金(BK2006218)资助项目.

通讯联系人: 朱庆保,教授,博士生导师,研究方向: 人工智能与智能控制. E-mail: zhuqingbao@njnu.edu.cn

路径长度为适应值, 产生初始种群后, 根据粒子更新策略进行更新等运算, 经过多次迭代后即可获得从起始点到目标点的一条全局最优路径. 最后对机器人不同的工作环境进行了仿真实验, 并和相关算法进行了比较, 其效果令人满意.

1 问题描述和模型的建立

路径规划的第一步就是要建立适当的环境模型. 已有的建模方法有链接图法、顶点图像法等, 虽然在路径规划中也能得到精确解, 但模型的建立和更新计算量很大, 实际应用困难. 本文采用栅格法^[5], 该方法具有简单、直观、计算量小等特点.

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 b_1, b_2, \dots, b_n . 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. 在 AS 中以 AS 的左上角为坐标原点 a 以横向为 x 轴, 纵向为 y 轴建立系统直角坐标系, 如图 1 所示. 其中, 粗线表示 25 号栅格的邻域. 假设机器人在水平方向上的行走步长为 δ 并且 AS 在 x, y 方向的最大值分别为 x_{\max} 和 y_{\max} . 以 δ 为步长对环境进行划分形成一个栅格, 则每行的栅格数 $N_x = x_{\max} / \delta$ 每列的栅格数 $N_y = y_{\max} / \delta$ 其中 $b_i (i = 1, 2, \dots, n)$ 占一个或多个栅格, 当不满一个栅格时算一个栅格.

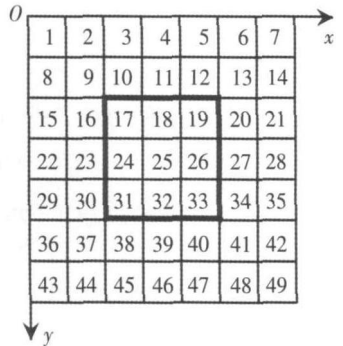


图 1 栅格坐标与序号关系

Fig.1 Relation of grid coordinates and the serial number

图 1 中的每一个数字代表环境中一个位置. 机器人运动轨迹由若干栅格序号组成. 记 $g \in AS$ 为任意栅格, A 为 AS 中 g 的集合, 记 $S = \{b_1, b_2, \dots, b_n\} \subseteq A$ 为静态障碍集. $\forall g \in A$ 在坐标系中都有确定的坐标 (x, y) , 记做 $g(x, y)$, x 为 g 所在的行号, y 为 g 所在的列号. 令 $C = \{1, 2, 3, \dots, M\}$ 为栅格序号集, $g(1, 1)$ 的序号为 1, $g(1, 2)$ 序号为 2, $g(2, 1)$ 序号为 $N_x + 1, \dots, g_i \in A$ 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:

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

式中, int 为舍余取整运算, \bmod 为求余运算.

对于任意二维凸多边形地形, 规划的目的是使机器人由任意起点 G_{begin} 安全地沿一条较短路径到达任意终点 G_{end} , 且 $G_{\text{begin}}, G_{\text{end}} \notin S$, 其它约束条件为: $\text{begin}, \text{end} \in C, \text{begin} \neq \text{end}$

根据栅格环境模型, 移动机器人在二维有限空间中可以朝着 8 个方向中的任意一个前进, 如图 2 所示. 为方便且不失一般性, 假定: 在机器人运动过程中, 障碍物的几何形状、大小均不发生变化.

2 二进制编码的粒子群算法

二进制编码粒子群算法是 Eberhart^[4] 等提出的 PSO 的离散二进制版, 它通过优化可连续变化的二进制概率 (即二进制变量为 1 的概率) 达到间接优化二进制变量的目的. 本文提出的基于栅格法的机器人路径规划算法就是以二进制编码的粒子群算法为基础的.

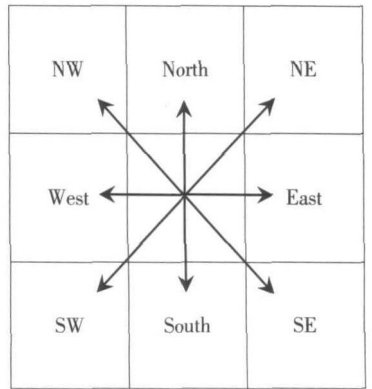


图 2 机器人运动方向示意图

Fig.2 Direction of the robot motion

标准粒子群算法如式 (2) 和式 (3) 所示. 在标准粒子群算法中, 给定一包含 M 个粒子的种群, 则第 i 个微粒的位置 $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$, 其中 D 表示粒子的维数. 它经历过的最好位置记为 $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$, 也称为 P_{best} 即历史最优解. 种群中的所有微粒所经历过的最好位置记为 $P_g = (P_{g1}, P_{g2}, \dots, P_{gD})$, 也称为 G_{best} 即全局最优解. 此外, 微粒 i 的速度用向量 $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})$ 表示. 对每一次迭代, 其第 j 维 ($1 \leq j \leq D$) 的位置和速度根据式 (2) 和式 (3) 进行更新:

$$V_{ij}(t+1) = \omega V_{ij}(t) + c_1 \text{rand}() (P_{ij}(t) - X_{ij}(t)) + c_2 \text{rand}() (P_{gj}(t) - X_{ij}(t)), \tag{2}$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t). \tag{3}$$

式中, ω 为惯性权重, c_1 和 c_2 是加速常数, $\text{rand}()$ 为在 $[0, 1]$ 范围内变化的随机函数.

此外, 微粒的速度 V_{ij} 被一个最大速度 V_{max} 所限制. 如果当前对微粒的加速导致它在 d 维的速度 V_{ij} 超过该维的最大速度 V_{max}^d , 则该维的速度被限制为 V_{max}^d .

在二进制编码的粒子群算法模型中, 将每一维 X_{ij} 和 P_{ij} 限制为 1 或者 0 而速度 V_{ij} 不作这种限制. 用速度来更新位置时, 如果 V_{ij} 高一些, 粒子的位置 X_{ij} 更有可能选 1, V_{ij} 低一点则选 0. 阈值在 $[0, 1]$ 之间, 而有这种特点的函数就是 Sigmoid 函数:

$$\text{Sig}(x) = (1 + \exp(-x))^{-1}.$$
 (4)

这样, 粒子的位置迭代公式可以变为:

$$X_{ij}(t+1) = \begin{cases} 0 & \text{若 } \rho \geq \text{Sig}(V_{ij}(t+1)); \\ 1 & \text{若 } \rho < \text{Sig}(V_{ij}(t+1)). \end{cases}$$
 (5)

其中, $\rho \in [0, 1]$ 为均匀分布的随机数. 通过式 (5), 便可将 $X_{ij}(t+1)$ 限制在集合 $\{0, 1\}$ 中.

在二进制编码的粒子群算法中, V_{ij} 仅表示一个概率, 即粒子的每一维分量的取值以 $\text{Sig}(V_{ij})$ 的概率取 1, 而以 $1 - \text{Sig}(V_{ij})$ 的概率取 0. 即如果粒子的某维分量是 0, 则其变为 1 的概率为 $\text{Sig}(V_{ij})$; 反之, 如果该分量为 1, 则其变为 0 的概率为 $1 - \text{Sig}(V_{ij})$. 这样, 可以定义每一位的改变概率为:

$$\rho(\Delta) = \text{Sig}(V_{ij}) * (1 - \text{Sig}(V_{ij})).$$
 (6)

由此看出, 标准 PSO 算法和二进制编码的 PSO 算法的区别主要在于速度的表示意义、位置向量的表示方法和更新方法, 这些区别都突出体现了这种改进 PSO 算法的二进制编码特点.

3 基于二进制粒子群算法的机器人路径规划

3.1 相关定义

为了叙述方便, 给出如下约束和定义:

定义 1 $\forall g \in A, g \notin S$ 则称 g 为可行栅格, 所有可行栅格组成的集合称为可行栅格集, 记为 FS ; $\forall g \in A, g \in S$ 则称 g 为障碍栅格, 所有障碍栅格组成的集合称为障碍栅格集, 记为 NFS . 即 $A = FS \cup NFS$.

定义 2 任意两栅格间的距离指两栅格间的连线长度, 记作 $\text{dist}(g_i(x_i, y_i), g_j(x_j, y_j))$, 其中 $g_i, g_j \in A, i, j \in C$. 由式 (7) 计算:

$$\text{dist}(g_i(x_i, y_i), g_j(x_j, y_j)) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$
 (7)

定义 3 $AR_i = \{g \mid g \in A, d(g, g_i) \leq L, i \in C\}$ 称为栅格 g_i 处的邻域. 其中 $L = \{1, \sqrt{2}\}$.

定义 4 $BR_i = \{g \mid g \in A, d(g, g_i) \leq L', i \in C\}$ 称为栅格 g_i 处的视野域. 其中 $L' = \{2, 2\sqrt{2}\}$.

定义 5 对于若干栅格组成的序列 $\{g_{i_1}, g_{i_2}, \dots, g_{i_n}\}$, 其中记集合 $SS = \{i_j \mid 1 \leq j \leq n, i_j \in C\}$, $g_k \in FS(k \in SS)$, 并且满足 $\text{dist}(g_{i_j}, g_{i_{j+1}}) = 1$ 或者 $\text{dist}(g_{i_j}, g_{i_{j+1}}) = \sqrt{2}$ 则此序列代表一条路径, 并称 g_k 为这条路径上的路径点.

定义 6 tabu_i^t 为第 i 个粒子在 t 时刻已走栅格位置的集合, 称为禁忌集合, 防止粒子重复走某一个栅格. 其中 tabu_i^0 表示初始时刻第 i 个粒子已走栅格位置的集合, 即 tabu_i^0 为空集.

定义 7 $\overline{g_i g_j} (i, j \in C, g_i, g_j \in FS)$ 称为栅格 g_i 和 g_j 的中心连线.

3.2 粒子位置编码方法

在粒子群算法中, 第 i 个微粒的位置 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$. 在本文中, 假定栅格 g_i 处的视野域为 BR_i , 其中的栅格数为 m . 此视野域中存在某一条路径 $\{g_{i_1}, g_{i_2}, \dots, g_{i_n}\}$, 其中 $n \leq m, g_k \in FS(k \in SS)$. 则编码方法为 $\{X_{ij} \mid 1 \leq j \leq m, \text{且当 } j \in SS, X_{ij} = 1 \text{ 否则 } X_{ij} = 0\}$. 如图 3 所示 (其中阴影部分表示障碍栅格), 假设某一条局部路径为 $\{1, 7, 8, 13, 18, 19, 24, 25\}$, 则此条路径对应的粒子编码为 $X = 1000001100001000011000011$. 这样, 每一位置向量都表示了一条或者多条路径. 因此, 粒子的维数由某时刻栅格 g_i 处视野域 BR_i 中栅格总数 m 决定, 因此整个路径规划过程中粒子群的维数是可变的. 使用这种方式进行编

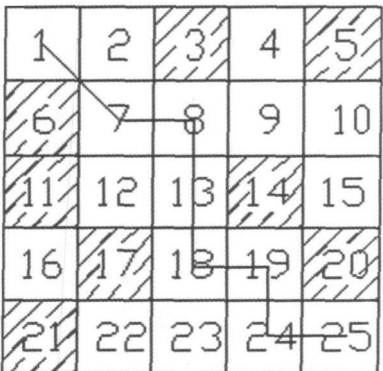


图 3 粒子群编码方法示例

Fig.3 Example of particle position encoding method

码, 保证了每一时刻所有的粒子都是同维的, 因而粒子群的位置和速度向量均是定长的.

3.3 初始种群的产生

机器人的初始路径(初始位置)产生即每次由当前位置找到下一个栅格位置, 最后将这些栅格连起来形成一条初始路径的过程^[6]. 因此, 初始种群产生的关键在于如何找到一个栅格.

从栅格 $g_i (g_i \in FS, i \in C)$ 出发, 设其邻域为 AR_i , 视野域为 BR_i , 那么可以选择的下一个节点 $g_j (j \in C)$ 必须满足条件 $g_j \in AR_i$ 且 $g_j \in FS$ 同时排除如图 4 所示的各种情况(其中阴影部分表示障碍栅格), 即机器人不可从两个障碍物之间穿过. 假设所选择的下一个节点 $g_j (j \in C)$ 与 g_i 的中心连线从两个障碍栅格之间穿过, 即出现了图 4 中所列的情况之一, 将选择的节点 g_j 记为 NS , 那么可选为下一个节点的栅格集合用式 (8) 表示. 本文利用赌轮盘法选择下一个路径点, 其中概率由式 (9) 计算得到:

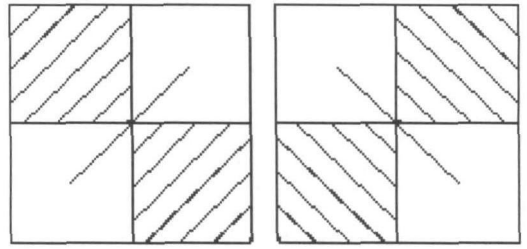


图 4 路径排除情况

Fig.4 Example of refused path

$$NGS = \{g_j \mid j \in C, g_j \in AR_i, g_j \in FS, g_j \neq NS\}, \quad (8)$$

$$prob = 1/|NGS|. \quad (9)$$

第 k 个粒子初始位置产生算法的具体步骤如下:

Step 1 从当前栅格 $g_i (g_i \in FS, i \in C)$ 出发, 由前定义得此时 $tabu_k^0$ 为空集.

Step 2 由式 (8) 所确定的下一个节点选择范围, 利用赌轮盘法根据式 (9) 计算概率的方法在当前位置 g_i 的视野域 BR_i 范围内, 选取栅格 g_w 作为下一个路径点, 即 g_w 必须满足 $w \in C, g_w \in NGS$ 且 $g_w \notin tabu_k$.

Step 3 更新当前点 g_i 为此次选取的栅格 g_w , 将 g_w 加入禁忌集合 $tabu_k$ 中.

Step 4 根据前面的视野域定义进行越界检查. 假定栅格 g_i 对应的坐标为 (x_i, y_i) , 若满足 $x_i = 1$ 或者 $x_i = 5$ 或者 $y_i = 1$ 或者 $y_i = 5$ 则栅格 g_i 为边界栅格, 转 Step 5 否则转 Step 2

Step 5 产生了机器人的初始路径集 $\{g_{i_1}, g_{i_2}, \dots, g_{i_n}\}$, 其中 $n \leq m$. 再利用 3.2 的编码方法得到其对应的位置向量 X_k .

对种群中所有粒子的初始位置均由上述算法产生, 最后得到了种群初始位置.

3.4 适应度函数值的确定

第 i 个粒子的位置分量 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$, 其对应的路径为 $\{g_{i_1}, g_{i_2}, \dots, g_{i_n}\}$, 则第 i 个粒子的适应度函数值为:

$$F_i = \sum_{j=1}^{n-1} \text{dist}(g_{i_j}, g_{i_{j+1}}), \quad (10)$$

式中, $g_{i_j} (1 \leq j \leq n)$ 均为路径点, $\text{dist}(g_{i_j}, g_{i_{j+1}})$ 由式 (7) 计算得出.

3.5 参数的选择

本文加速常数 c_1, c_2 均取 2 ω 则随迭代次数线性减小, 即:

$$\omega = \omega_{\max} - \text{iter}^* (\omega_{\max} - \omega_{\min}) / \text{iter}_{\max}, \quad (11)$$

式中, iter 为当前迭代次数, iter_{\max} 为总的迭代次数, $\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$

在二进制编码的粒子群算法中, 也存在 V_{\max} , 但它表示算法允许的概率范围. 通过数值计算, 可以发现, 如果 $V_{\max} > 10$ 则 Sig 函数将趋于 0 因此, 根据文献[4] 本文取为 6.0 此时 Sig 函数的范围为 0.0025 ~ 0.9975 这表明随着 V_j 增加, Sig 函数将逐渐减少, 但最低将达到 0.0025 从而保证算法仍然能够有能力发生变化, 避免陷入局部最小值.

3.6 粒子位置和速度更新策略

粒子速度更新仍然采用标准 PSO 算法中的速度更新公式, 即式 (2).

粒子位置更新问题是机器人路径规划二进制 PSO 算法的最重要的部分. 本文中的粒子位置的更新以式 (5) 为基础. 以下为本文采用的粒子位置更新策略:

由二进制 PSO 算法本身的特点, 本文中对粒子的位置更新实际上是对粒子位置向量中的某一分量的更新, 假设第 i 个粒子的位置 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$, 则更新后的位置 $X'_i = (X'_{i1}, X'_{i2}, \dots, X'_{id})$. 从 X_i 到

X'_i 的区别就在于其中某分量由原来的 0 变为 1 或者由原来的 1 变为 0 这样的改变有可能使 $X'_i = (X'_{i1}, X'_{i2}, \dots, X'_{iD})$ 对应的路径不能安全避障. 因此, 在改变粒子的位置 $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ 中的某一维分量的值时, 均要进行避障检查和越界检查, 检查改变后的位置向量 $X'_i = (X'_{i1}, X'_{i2}, \dots, X'_{iD})$ 对应的路径是否与障碍物相碰或是否超过视野域范围. 若安全避障且未超过视野域范围, 则允许将 X_i 更新到 X'_i . 否则不允许更新.

设粒子更新后的位置向量 $X'_i = (X'_{i1}, X'_{i2}, \dots, X'_{iD})$, 该向量只记录了某一视野域中所走过的可行栅格, 但是其中并未反映经过这些栅格的先后顺序. 如由图 3 所示, 其中位置向量 $X = 1000001100001000011000011$ 则可能的路径为 $1 \rightarrow 7 \rightarrow 8 \rightarrow 13 \rightarrow 18 \rightarrow 19 \rightarrow 24 \rightarrow 25$ 或者 $1 \rightarrow 7 \rightarrow 8 \rightarrow 13 \rightarrow 19 \rightarrow 18 \rightarrow 24 \rightarrow 25$ 在实际算法中, 当位置向量可以代表多条路径时, 只能选择其中之一.

这样, 粒子位置更新的关键在于如何对更新后的位置向量 $X'_i = (X'_{i1}, X'_{i2}, \dots, X'_{iD})$ 进行避障检查和越界检查. 本文采用的检查策略为将 $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ 中为 1 的栅格序号提取出来构成集合 $CGS = \{u \mid u \in C, X_{iu} = 1\}$, 进行全排列 $P_{|CGS|}^{|CGS|}$ (由前面的视野域定义可得其排列的计算代价较小), 对 $P_{|CGS|}^{|CGS|}$ 中的每一个排列 $u_1, u_2, \dots, u_{|CGS|}$ 进行边界和避障检查: 由定义 7 得 $\overline{u_i u_{i+1}}$ ($1 \leq i \leq |CGS| - 1$) 表示栅格序号为 u_i, u_{i+1} 的栅格中心的连线, 则避障必须满足 $\overline{u_i u_{i+1}}$ 不会穿过任何障碍栅格, 不会出现图 4 所示的情况, 且不会超过当前视野域的任何一边. 若至少存在一个排列符合要求, 则允许更新, 否则不可以更新.

3.7 算法步骤

假设种群规模为 N 个粒子, 维数为 D . 根据以上二进制编码的粒子群算法原理以及与机器人路径规划问题的结合方法, 机器人路径规划的二进制编码的粒子群算法步骤描述如下:

Step 1 建立基于栅格的机器人工作环境, 设置初始起始点 $G_{begin}(x_{begin}, y_{begin})$ 和目标点 $G_{end}(x_{end}, y_{end})$, 初始起始点 G_{begin} 为机器人的当前位置. 设置最大迭代次数 $iter_{max}$, 变量 $iter$ 保存当前迭代次数, 初始化粒子群算法中各个参数值 $\omega, c_1, c_2, V_{max}, \omega_{max}, \omega_{min}$. 定义一个路径表 $Path$ 将结果用此路径向量表示;

Step 2 根据机器人当前位置和视野域信息采用 3.3 的方法初始化种群中各粒子位置向量 X_i 和速度向量 V_i ($1 \leq i \leq N$), 若初始化失败则退出程序;

Step 3 由式 (10) 计算每一个粒子的适应值 F_i (即目标函数值);

Step 4 由式 (12) 更新每一个粒子 i ($1 \leq i \leq N$) 的历史最优解 P_{besti} :

$$P_{besti}(t+1) = \begin{cases} P_{besti}(t), & \text{若 } F_i(X_i(t+1)) \geq F_i(P_{besti}(t)); \\ X_i(t+1), & \text{若 } F_i(X_i(t+1)) < F_i(P_{besti}(t)). \end{cases}$$

(12)

Step 5 由式 (13) 更新种群的全局最优解 G_{best} 其中 $i = 1, 2, \dots, N$;

$$G_{best}(t+1) = \begin{cases} G_{best}(t), & \text{若 } G_{best}(t) \leq \min F_i(P_{besti}(t+1)); \\ P_{best \arg \min F_i(P_{besti}(t+1))}(t+1), & \text{若 } G_{best}(t) > \min F_i(P_{besti}(t+1)). \end{cases}$$

(13)

Step 6 由式 (2) 更新每一个粒子的速度, 若更新后的速度 $V_{ij}(t+1)$ 大于 $+V_{max}$ 或者小于 $-V_{max}$ 则需要调整 $V_{ij}(t+1)$, 保证更新后速度不小于 $-V_{max}$ 且不大于 $+V_{max}$ 调整方法如式 (14):

$$V_{ij}(t+1) = \begin{cases} -V_{max}, & \text{若 } V_{ij}(t+1) < -V_{max}; \\ +V_{max}, & \text{若 } V_{ij}(t+1) > +V_{max} \end{cases}$$

(14)

Step 7 由式 (5) 采用 3.6 的方法更新每一个粒子的位置. 当前迭代次数 $iter$ 加 1 并由式 (11) 更新 ω 的值;

Step 8 若 $iter < iter_{max}$ 则返回 Step 3 否则继续;

Step 9 将得到的此视野域中的最优路径追加到路径表 $Path$ 中;

Step 10 若到达目标点 G_{end} 或到达栅格 $g(x_g, y_g)$ 满足 $g \in AR_{G_{end}}$ 且 gG_{end} 不会出现图 4 情况, 转 Step

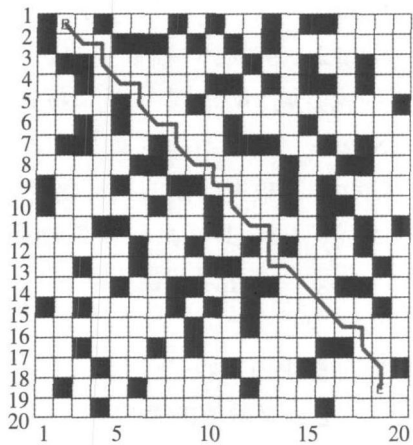


图 5 20x20 环境下得到的最优路径图

Fig.5 Optimal path under 20x20 environment

11, 否则更新机器人的当前位置为此局部路径终点, 转 Step 2

Step 11 路径表 Path保存的为从起始点到目标点的最优路径, 算法结束.

4 仿真实验

为了验证该算法的效果, 本文进行了实验, 实验环境为: P4 2.40 GHz, 内存为 1 024M, 编译工具 VC++6.0 粒子总数 $N = 20$ 最大迭代次数 $iter_{max} = 20$ 在 20×20 环境下对算法进行了验证, 能够获得较好路径, 证明了此算法的有效性和正确性.

为了进一步说明本文算法的有效性和先进性, 与相近算法进行了实验比较. 由于已有用粒子群算法进行机器人路径规划的研究均为连续环境, 因而与相近算法文献 [7]结果进行了比较. 图 6~图 11 示出了在 10×10 与 20×20 的栅格环境下本文算法与文献 [7]算法得到的路径图, 定量实验数据的对比如表 1所示.

表 1 算法性能对比表 (栅格单位长度为 20)

Table 1 Algorithm performance reference table

算法		10×10环境 (一)	10×10环境 (二)	20×20环境
问题最优解长度 (单位: 栅格数)	本文算法	18	17	36
	文献 [7]算法			
	(ACO-grid)	19	19	39
最优解路径长度	本文算法	348.28	353.12	774.52
	文献 [7]算法			
	(ACO-grid)	360	360	760
平均消耗时间 /s	本文算法	1.572	1.703	5.36
	文献 [7]算法			
	(ACO-grid)	< 0.1	< 0.1	12.6

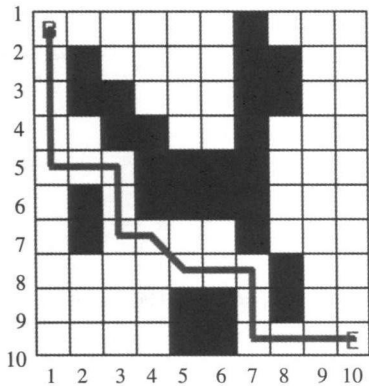


图 6 环境 1 下本文算法得到的路径

Fig.6 Optimal path using the algorithm proposed in this paper under the first kind of the environment

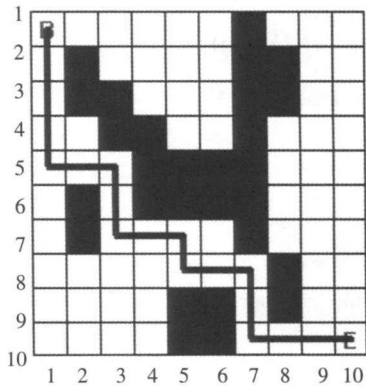


图 7 环境 1 下文献[7]算法得到的路径

Fig.7 Optimal path using the algorithm in the seventh reference under the first kind of the environment

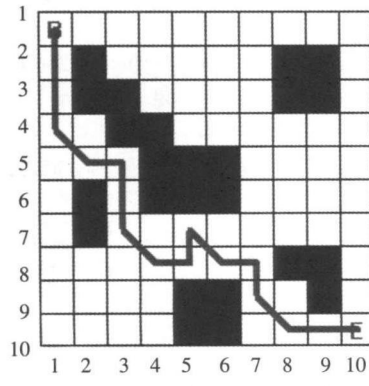


图 8 环境 2 下本文算法得到的路径

Fig.8 Optimal path using the algorithm proposed in this paper under the second kind of the environment

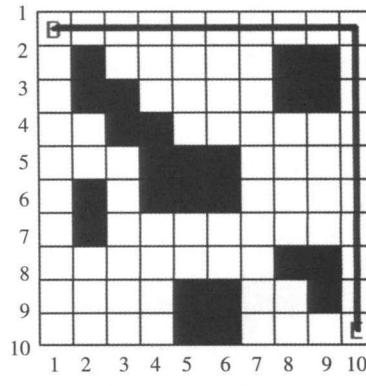


图 9 环境 2 下文献[7]算法得到的路径

Fig.9 Optimal path using the algorithm in the seventh reference under the second kind of the environment

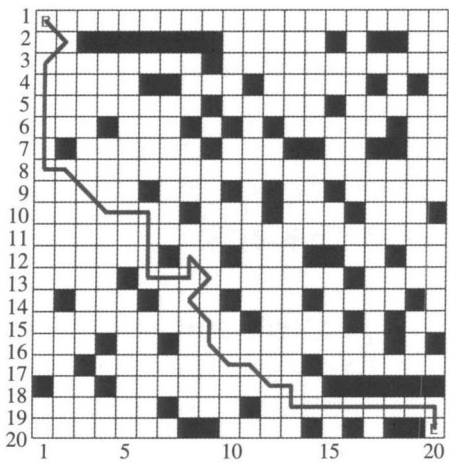


图 10 环境 3 下本文算法得到的路径

Fig.10 Optimal path using the algorithm proposed in this paper under the third kind of the environment

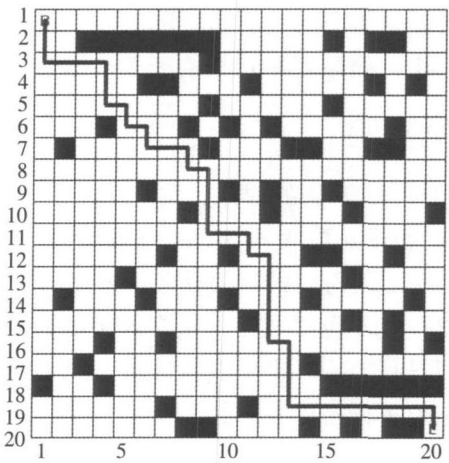


图 11 环境 3 下文献[7]算法得到的路径

Fig.11 Optimal path using the algorithm in the reference [7] under the third kind of the environment

由以上实验结果可见, 本文得到的路径长度多数优于文献 [7] 算法, 环境范围较小时, 本文平均消耗时间劣于文献 [7] 算法, 但环境范围加大一倍后, 本文算法平均消耗时间则远优于文献 [7], 结果表明了本文算法的有效性. 若综合考虑路径的长度和得到最优路径需要的时间两个方面, 本文算法具有一定的优越性.

5 结 论

本文首次提出了一种基于栅格环境的移动机器人路径规划二进制 PSO 算法, 该算法适用于基于栅格法表示的机器人环境地图的离散环境, 即用 PSO 算法解决具有离散特征的问题对象, 这为今后离散优化问题的研究提供了一种新方法和新思路. 实验结果表明, 这种以二进制 PSO 算法为基础的机器人路径规划算法是可行的、有效的, 并有一定的优越性.

[参考文献] (References)

[1] 朱庆保. 复杂环境下的机器人路径规划蚂蚁算法 [J]. 自动化学报, 2006 32(4): 586-593.
Zhu Qingbao Ant algorithm for path planning of mobile robot in a complex environment [J]. Acta Automatica Sinica 2006 32 (4): 586-593 (in Chinese)

[2] 秦元庆, 孙德宝, 李宁, 等. 基于粒子群算法的移动机器人路径规划 [J]. 机器人, 2004, 26(3): 222-225
Qin Yuanqing Sun Debao Li Ning et al Path planning of mobile robot based on particle swarm optimization algorithm [J]. Robot 2004, 26(3): 222-225 (in Chinese)

[3] 孙波, 陈卫东, 席裕庚. 基于粒子群优化算法的移动机器人全局路径规划 [J]. 控制与决策, 2005 20(9): 1 052-1 055
Sun Bo Chen Weidong Xi Yugeng Particle swarm optimization based global path planning for mobile robots [J]. Control and Decision 2005, 20(9): 1 052-1 055 (in Chinese)

[4] Kennedy J Eberhart R C. A discrete binary version of the particle swarm algorithm [C] // Proceedings of the 1997 Conference on Systems, Man, and Cybernetics. Piscataway, NJ: IEEE Service Center, 1997: 4 104-4 109.

[5] 马兆青, 袁曾任. 基于栅格的移动机器人实时导航和避障 [J]. 机器人, 1996, 18(6): 344-348
Ma Zhaoqing Yuan Cengren Real-time obstacle avoidance and navigation based on grid method [J]. Robot 1996 18(6): 344-348 (in Chinese)

[6] 周兰凤, 洪炳熔. 用基于知识的遗传算法实现移动机器人路径规划 [J]. 电子学报, 2006 34(5): 911-914
Zhou Lanfeng Hong Bingrong A knowledge based genetic algorithm for path planning of a mobile robot [J]. Acta Electronica Sinica 2006 34(5): 911-914 (in Chinese)

[7] 张美玉, 黄翰, 郝志峰, 等. 基于蚁群算法的机器人路径规划 [J]. 计算机工程与应用, 2005, 41(9): 34-37.
Zhang Meiyu Huang Han Hao Zhifeng et al Path planning for robots based on ant colony algorithm [J]. Computer Engineering and Applications 2005, 41(9): 34-37. (in Chinese)

[责任编辑: 严海琳]