

基于粒子群算法的移动机器人全局路径规划策略

李 擎 徐银梅 张德政 尹怡欣

北京科技大学信息工程学院, 北京 100083

摘 要 提出了一种基于保收敛粒子群优化算法的移动机器人全局路径规划策略, 为移动机器人在有限时间内找到一条避开障碍物的最短路径提供了一种解决方案. 首先建立环境地图模型, 将连接地图中起点和终点的路径编码成粒子, 然后根据障碍物位置规划出粒子的可活动区域, 在此区域内产生初始种群, 使粒子在受限的区域内寻找最优路径. 在搜索过程中, 粒子群优化算法的加速系数和惯性权重均随迭代次数自适应调节. 仿真实验表明算法可在起点与终点之间找到一条简单安全的最优路径. 与其他文献所提的方法进行了对比研究, 结果表明本文所提算法具有更快的搜索速度和更高的搜索质量.

关键词 移动机器人; 路径规划; 粒子群优化算法; 活动区域

分类号 TP 273⁺. 5; TP 242. 6

Global path planning method for mobile robots based on the particle swarm algorithm

LI Qing XU Yin mei ZHANG De-zheng YIN Yi-xin

School of Information Engineering University of Science and Technology Beijing Beijing 100083, China

ABSTRACT A global path planning method for mobile robots based on the guaranteed convergence particle swarm optimization algorithm is presented. A solution is provided for mobile robots to find the shortest path avoiding obstacles in a limited period of time. Firstly, an environmental map is set up and a path connecting the start point and the end point is coded as a particle. Then, a particular active region for particles is mapped out according to the location of obstacles. The initial particle population is generated within this region and particles fly in the active region to search for the optimum path. In the search process, both the acceleration coefficient and inertia weight of the particle swarm optimization algorithm are self-adaptively adjusted along with iteration processes. It is proved that the algorithm can plan out a simple and safe optimum path connecting the start point and the end point by simulation experiments. Comparative studies with a recently reported method show that the proposed algorithm has advantages such as faster search speed and higher search quality.

KEY WORDS mobile robot; path planning; particle swarm optimization; active region

路径规划是自主式移动机器人导航的基本环节之一. 所谓移动机器人的路径规划问题, 是指在其工作空间中找到一条从起始点到目标点的, 能避开动静障碍且能满足某个优化目标 (如行走路线最短、能量消耗最少) 的最优 (或次优) 路径. 基于环境模型的路径规划方法^[1]可以分为两种类型: 环境信息完全已知的全局路径规划和环境信息完全未知或部分未知的局部路径规划. 全局路径规划方法

通常可以找到最优解, 但需要预先知道准确的全局环境信息. 到目前为止, 对于全局路径规划问题已经有许多解决方法^[2-4], 但这些方法大都受到信息存储方式和规划时间的制约, 计算量大, 实时性差, 不能很好地适应于全局路径规划问题. 如何对全局路径规划方法做出改进, 使之在较短的有限时间内规划出最优路径, 是本文的研究目的所在.

由 Eberhart 和 Kennedy 提出的粒子群优化算法

收稿日期: 2009-01-20

基金项目: 国家自然科学基金资助项目 (No. 60374032); 第 36 批国家留学回国人员科研启动基金资助项目

作者简介: 李 擎 (1971-), 男, 教授, 博士, E-mail: liqing@ies.ustb.edu.cn

(particle swarm optimization algorithm, PSO)^[5]作为一种模拟鸟群飞行的仿生算法,具有算法简洁、易于实现和鲁棒性好等优点. 算法对种群大小不十分敏感,其收敛速度快的特点非常适合于对移动机器人路径规划这种实时性要求较高的复杂问题进行求解搜索. 文献 [6]首次将 PSO 引入移动机器人路径规划中,该方法首先使用 Dijkstra 算法获得链接图的最短路径,然后用 PSO 对所得路径中的节点位置进行二次优化以得到更优的路径,由于链接图并不能完全体现实际规划环境中的信息,二次优化后得到的路径不一定是全局最优路径,因此该方法限制了 PSO 的全局寻优能力. 文献 [7]提出了另一种基于 PSO 的全局路径规划算法,通过坐标变换将需规划路径的二维编码简化为一维编码,充分利用了 PSO 的全局寻优能力,但不能很好地解决早熟现象. 文献 [8]采用罚函数法表示粒子适应度函数,并加入碰撞能量测试点,极大地缩短了算法的执行时间,但没有给出碰撞和距离能量函数的权重比,并且该方法也不能保证寻得最优路径.

本文旨在深入研究粒子群算法在移动机器人路径规划问题中的应用. 对所提出的方法在简单和复杂环境下进行了仿真实验,并同文献 [7]中所提的方法进行了仿真对比. 仿真结果表明,无论是在路径长度还是在执行时间方面,本文所提的算法均优于文献 [7]中的方法.

1 粒子群算法

随机产生初始粒子群,而每一次迭代过程中粒子根据自身找到的最好解和整个群体目前找到的最好解来更新位置和速度^[9].

粒子 *i* 的速度和位置更新方程为:

$$v_{ij}^{k+1} = \omega v_{ij}^k + c_1 \text{rand}_1^k (p_{ij}^k - x_{ij}^k) + c_2 \text{rand}_2^k (g_j^k - x_{ij}^k) \tag{1}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \tag{2}$$

式中, v_{ij}^k 为粒子 *i* 在第 *k* 次迭代中第 *j* 维的速度; x_{ij}^{k+1} 为粒子 *i* 在 *k*+1 次迭代中第 *j* 维位置; c_1 、 c_2 为学习因子,通常取为 2^[5]; rand_1 、 rand_2 为 [0, 1] 区间内的伪随机数; ω 为惯性权重,决定了粒子对前面速度继承的多少; p 、 g 分别为粒子自身找到的最好解和整个群体目前找到的最好解.

对粒子 *i* 如果恰巧 $X_i = P_i = g$ 则速度更新仅仅依赖于 ωv_i 项,使得算法不能保证收敛,出现早熟现象,为此 Van den Bergh 和 Engelbrecht 提出了保收敛 PSO (guaranteed convergence particle swarm optimization GCPSO)^[10],对全局最好粒子用如下的更

新方程,如下面两式所示,而其他粒子仍用原方程 (1)、(2)更新:

$$v_{ij}^{k+1} = -x_{ij}^k + g_j^k + \omega v_{ij}^k + \rho^k (1 - 2 \times \text{rand}^k) \tag{3}$$

$$x_{ij}^{k+1} = g_j^k + \omega v_{ij}^k + \rho^k (1 - 2 \times \text{rand}^k) \tag{4}$$

式中,

$$\rho = 1, \rho^{k+1} = \begin{cases} 2\rho^k, & s > 15 \\ 0.5\rho^k, & f > 5 \\ \rho^k, & \text{其他} \end{cases}$$

f 为算法一次迭代过程中优化函数值连续保持不变的次数, s 为一次迭代过程中优化函数值连续减小的迭代次数.

2 模型描述

对机器人运动空间建模时作如下假定:机器人在二维有限空间中运动,空间中分布着有限个位置已知的静态障碍物,障碍物用多边形描述,且可以忽略其高度信息;为保证路径的安全性,这里把障碍物按照机器人半径尺寸膨胀,以便将机器人看作质点,忽略其尺寸大小,即按照点机器人来处理问题.

如图 1 所示,在全局坐标系 $O-X'Y'$ 中, S 为机器人的出发点, G 为终点. 图中黑色区域表示障碍物,机器人的路径规划就是在图中寻找一个点的集合 $P = \{S, y_1, \dots, y_j, \dots, y_D, G\}$,要求相邻点之间为直线连接且无障碍. 问题的解是一条最短可行路径,它对应优化算法中的一个粒子,点的坐标对应粒子的位置. 以机器人当前位置 S 作为原点,以 S 与目标点 G 的连线作为 X 轴,以垂直于 X 且经过 S 点的直线作为 Y 轴,建立局部坐标系 $O-XY$. 设 α 为 X 轴与 X' 轴的夹角,则相应的坐标变换公式为:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} X' \\ Y' \end{bmatrix} \tag{5}$$

把线段 SG 作 $D+1$ 等分,并在各个等分点上作垂线 l_j ,则垂线上点的 y 坐标即构成粒子 *i* 的位置编码 (对应于前面更新公式中的 x_{ij}^k). 等分点的个数 D 即为粒子的维数. 粒子的适应度取为路径长度:

$$f(Y_i) = \sum_{j=0}^D \sqrt{\left(\frac{L_{SG}}{D+1}\right)^2 + (y_{j+1} - y_j)^2} \tag{6}$$

式中, L_{SG} 为线段 SG 的长度.

应用优化算法求得局部坐标系中的路径点后,可以用反变换公式

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{7}$$

得到路径点在全局坐标系中的坐标.

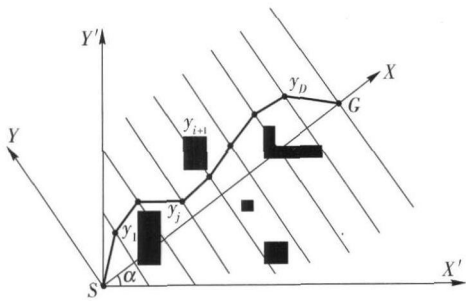


图 1 路径编码方法
Fig 1 Path coding method

3 基于 PSO 的路径规划

3.1 基本思想

初始化粒子时要满足粒子 i 第 j 维分量和第 $j+1$ 维分量所对应地图中的两个点之间直线连接且无障碍, 即不在障碍物边界范围以内. 为缩小粒子搜索空间, 提高搜索速度, 这里提出了“可活动区域”这一概念. 现举例说明如下: 首先确定与规划起点和规划终点连线 (SG 线) 相交的障碍物顶点, 并在 SG 线两侧找到与 SG 线距离最远的两个顶点, 如图 2 中的 e, f 两点, 然后通过 e, f 和 S 和 G 作矩形 $abcd$ 并在与所作矩形有交集的障碍物顶点中找到 SG 线两侧与 SG 线距离最远的两个顶点, 如图 2 中的 m, n 两点, 接着采用由 m, n, S 和 G 形成的矩形 $a'b'c'd'$ 代替原有矩形. 重复以上过程, 直到与所作矩形相交的障碍物不再存在时为止, 将所得矩形在 SG 线两侧各扩展一个较小的正数 ϵ , 这时得到的矩形范围即为本文所指的粒子“可活动区域”, 如图 2 中的灰色区域所示. 根据基本的几何知识可知, 最短路径一定存在于该区域内, 这样就变相地缩小了搜索空间.

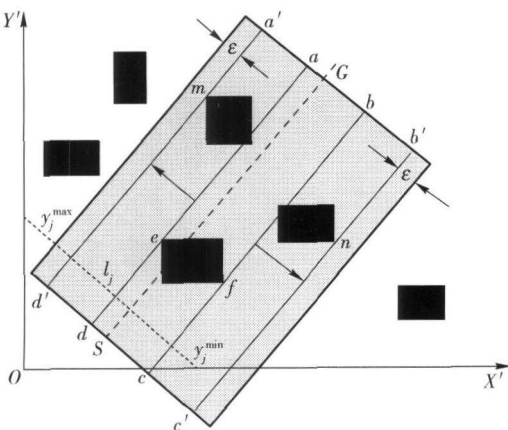


图 2 可活动区域图例
Fig 2 Illustration for the active region

粒子 i 第 j 维的初始位置由下式计算得到:

$$y_j = y_j^{min} + r \times (y_j^{max} - y_j^{min}) \quad (j=1, 2, \dots, D) \quad (8)$$

式中, r 为 $[0, 1]$ 上的随机数. y_j^{max} 和 y_j^{min} 为位置边界约束, 取值由各等分线 l_j 与“可活动区域”的交点决定, 如图 2 中的 y_j^{max} ; “可活动区域”在地图边界以外的部分, 则取为地图边界, 如图 2 中的 y_j^{min} . 速度极值 v_{jmax} 规定了粒子一次迭代中能够飞行的最大距离, 这里取为位置变化范围的 10%^[11], 即:

$$v_{jmax} = 0.1 \times (y_j^{max} - y_j^{min}) \quad (9)$$

粒子初始速度取为:

$$v_j^0 = (2r-1) \times v_{jmax} \quad (10)$$

式 (1)、(3) 和 (4) 中的惯性权重 ω 随迭代次数线性减小^[12], 可以使种群开始时大范围搜索新的解空间, 以后逐渐小范围搜索, 加强搜索现有解空间的能力, 即:

$$\omega = \omega_{max} - k \times \frac{\omega_{max} - \omega_{min}}{k_{max}} \quad (11)$$

式中, k 为当前迭代次数, k_{max} 为设定的最大迭代次数, ω_{max} 为最大惯性权重, ω_{min} 为最小惯性权重. 这里取为 $\omega_{max} = 0.9$, $\omega_{min} = 0.4$ ^[13].

为加快搜索速度, 本文中的加速系数也采用了自适应调节算法. 在搜索开始时采用较大的 α 和较小的 ϵ , 目的是使粒子飞遍整个搜索空间而不趋于个体极值点, 而在迭代后期则采用较小的 α 和较大的 ϵ , 以便使粒子趋于全局最优解, 自适应调节^[14]的公式如下所示:

$$\alpha = (\alpha_i - \alpha_f) \left[\frac{k_{max} - k}{k_{max}} \right] + \alpha_f \quad (12)$$

$$\epsilon = (\epsilon_i - \epsilon_f) \left[\frac{k_{max} - k}{k_{max}} \right] + \epsilon_f \quad (13)$$

式中, α_f 和 ϵ_f 分别为 α 和 ϵ 的最终值. $\alpha_i = 2.5$, $\epsilon_i = 0.5$, $\alpha_f = 0.5$, $\epsilon_f = 2.5$ ^[14].

在粒子搜索过程中, 每进行一次速度和位置的更新, 部分有效粒子会变为无效粒子, 这里的无效是相对有效而言的, 指粒子所代表的路径穿过障碍物, 即为有碰的. 有碰路径不能被接受, 必须被放弃. 如果对所有无效粒子都重新进行初始化必然要耗费大量的计算时间, 为此本文提出直接将部分有碰路径粒子的位置取为整个群体目前找到的最优位置, 在该位置附近细化搜索; 另一部分有碰路径粒子的位置取为邻域粒子所找到的最好解 L_r , 使得该无效粒子进入邻域粒子所在的局部最优区域搜索, 其中 L_r 由

$$L_r \in \{P_{i-p}, P_{i-p+1}, \dots, P_{i-1}, P_i, P_{i+1}, \dots, P_{i+p-1}, P_{i+p}\},$$

$$f(L_r) = \min \{f(P_{i-p}), f(P_{i-p+1}), \dots, f(P_{i-1}), f(P_i),$$

$$f(P_{i+1}), \dots, f(P_{i+1}), f(P_{i+1}) \} \quad (14)$$

确定. 这里的邻域仅与粒子序号有关, 而与粒子所处的空间位置无关. 该方法同重新初始化所有粒子相比, 既缩短了算法规划时间, 又加强了粒子群搜索多个局部最优的能力, 保证了多样性.

3.2 实现步骤

本文所提路径规划算法的具体实现步骤如下.

Step1 初始化 M 个粒子, 其具体过程为: 由式 (8) 初始化第 i 个粒子位置. 在“可活动区域”内随机产生粒子第 $j+1$ 维分量, 检测和第 j 维分量对应点的连线是否穿过障碍物, 不穿过则产生下一维分量, 穿过则重新产生. 如果对该维分量进行的若干次初始化尝试均失败, 则要从粒子第 1 维重新开始产生. 由式 (9)、式 (10) 初始化粒子速度. 计算所有粒子的适应度并将适应度最小的粒子位置设为全局极值点. 个体极值点则设为每个粒子的当前位置.

Step2 对全局最好粒子用式 (3) 更新粒子各维的速度 v_{ij} . 其他粒子由式 (1) 更新粒子各维的速度 v_{ij} . 同时注意边界约束. 即若 $v_{ij} > v_{jmax}$, 则 v_{ij} 取为 v_{jmax} ; 若 $v_{ij} < -v_{jmax}$, 则 v_{ij} 取为 $-v_{jmax}$.

Step3: 由式 (2) 更新粒子各维的位置 v_{ij} . 其更新流程既要考虑边界约束也要考虑障碍约束, 障碍约束的解决方法如算法基本思想中所述.

Step4: 对每个粒子由式 (6) 求其适应度并更新个体极值点 P 和全局极值点 g .

Step5: 转 Step2 进行迭代, 直到达到设定的最大迭代次数 k_{max} 或全局极值点适应度值连续 20 代保持在一定的范围内.

4 仿真研究

4.1 简单环境

假定机器人的工作空间大小为 100×100 , 障碍物的位置为顶点表示法 $Ob1[(10, 30), (30, 30), (30, 10), (10, 10)]$, $Ob2[(60, 80), (80, 80), (80, 50), (60, 50)]$, $S(0, 0)$ 为起始点, $G(100, 100)$ 为目标点, 如图 3 所示. 取粒子的维数为 4. 在简单环境中初始化 10 个粒子对应的路径如图 3 所示. 算法达到最大迭代次数 50 时, 经过 0.31 s 搜索得到的最优路径如图 4 所示, 其路径长度为 147.572.

4.2 复杂环境

图 5 为当机器人处于障碍物数目较多的环境空间中的初始种群, 实验中取种群大小 $M=10$, 粒子的维数 $D=8$, 最大迭代次数 $N=60$, 所得到的最优路径如图 6 中所示. 其最短路径长度值为 143.824.

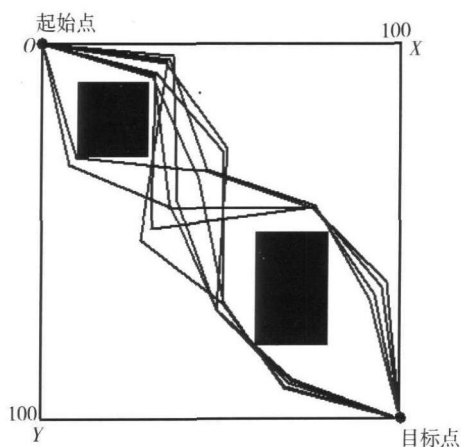


图 3 简单环境下粒子初始化

Fig 3 Initial path in simple environment

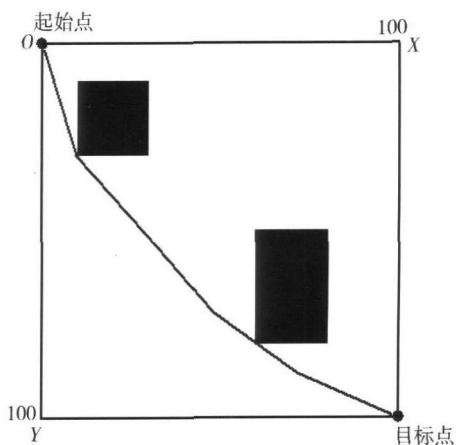


图 4 简单环境下最优路径

Fig 4 Optimum path in simple environment

执行时间为 0.24 s

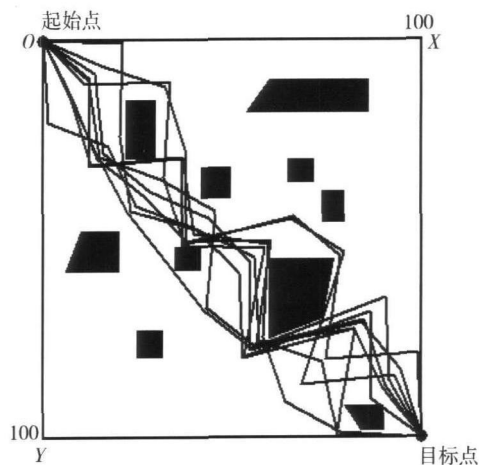


图 5 复杂环境下粒子初始化

Fig 5 Initial path in complex environment

4.3 对比研究

为了说明本文算法的优越性, 同文献 [7] 中所介绍的算法进行了对比仿真研究. 在 PC (Pentium processor 1 300MHz, 256M RAM) 上分别进行 20 次

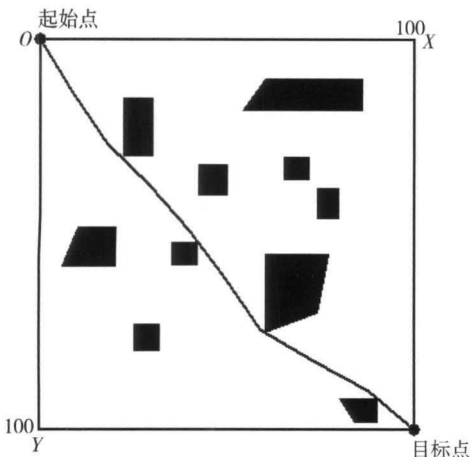


图 6 复杂环境下最优路径

Fig 6 Optimum path in complex environment

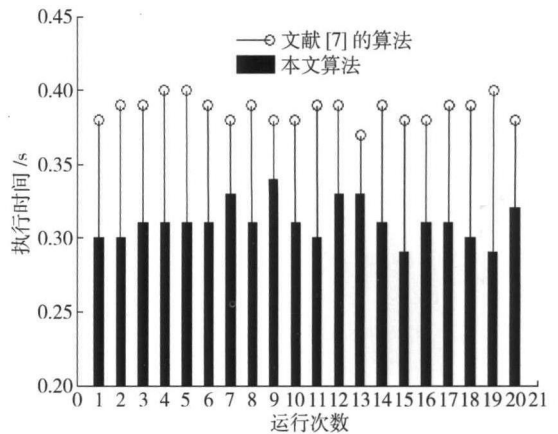


图 8 简单环境下执行时间对比

Fig. 8 Comparison of run time in simple environment

实验. 图 4 所示的简单环境中取种群大小 $M=80$, 粒子的维数 $D=4$, 迭代次数 $N=100$ 相同的条件下, 两种算法每次运行得到的最优路径长度和执行时间分别如图 7 和图 8 所示. 在如图 4 所示的简单环境和如图 6 所示的复杂环境下, 两种算法结果对比如表 1 所示, 其中复杂环境下参数取值均为种群大小 $M=50$, 粒子维数 $D=8$, 迭代次数 $N=80$. 可见改进后的算法求得最优路径长度和执行时间都显著缩短. “可活动区域”的引入使粒子群更容易搜索到最优路径, 改进后的粒子群优化算法加快了收敛速度.

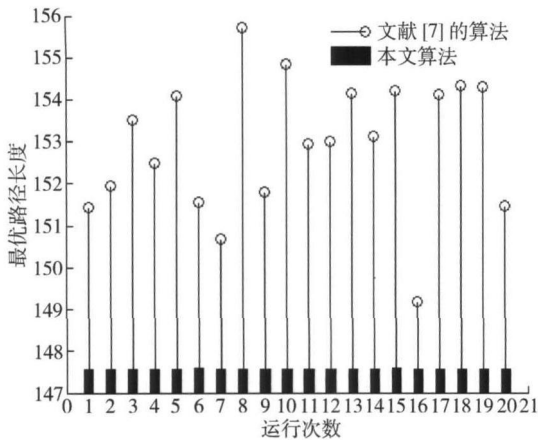


图 7 简单环境下最优路径长度对比

Fig 7 Comparison of the shortest length in simple environment

表 1 两种算法结果平均值对比

Table 1 Comparison of average values calculated by two algorithms

环境	算法	最优路径长度		执行时间 /s	
		均值	方差	均值	方差
简单	文献 [7] 算法	152.95	2.48	0.39	6.1×10^{-5}
	本文算法	147.56	1.1×10^{-4}	0.31	1.7×10^{-4}
复杂	文献 [7] 算法	160.12	16.86	1.71	4.10×10^{-3}
	本文算法	143.63	1.80×10^{-4}	1.43	2.04×10^{-4}

5 结语

仿真研究表明, 本文所提出的移动机器人路径规划算法可在起点与终点之间得到一条简单安全的最优路径. 其主要特点有: (1) 粒子群算法和其他优化算法相比有实现简单, 可调参数少的优点; (2) 加速系数和惯性权重均自适应调节, 收敛速度快; (3) 将迭代过程中的无效粒子直接取为全局极值点或邻域个体极值点的方案, 避免了重新初始化粒子, 从而缩短了规划时间; (4) “可活动区域”剔除了不必要的搜索区域, 提高了粒子群算法的搜索效率.

参考文献

- [1] Dai B, Xiao X M, Cai Z X. Current status and future development of mobile robot path planning technology. *Control Eng China*. 2005, 12(3): 198
(戴博, 肖晓明, 蔡自兴. 移动机器人路径规划技术的研究现状与展望. *控制工程*, 2005, 12(3): 198)
- [2] Lozano-perez T. Automatic planning of manipulator transfer movements. *IEEE Trans Syst Man Cybern*. 1981, 11 (10): 681
- [3] Takahashi O, Schilling R J. Motion planning in a plane using generalized voronoi diagrams. *IEEE Trans Rob Autom*. 1989, 5 (2): 143
- [4] Yu H B, Li X A. Fast path planning based on grid model of robot. *Microelectron Comput*. 2005, 22(6): 98
(于红斌, 李孝安. 基于栅格法的机器人快速路径规划. *微电子学与计算机*, 2005, 22(6): 98)
- [5] Kennedy J, Eberhart R C. Particle swarm optimization // *Proceedings of the 1995 IEEE International Conference on Neural Networks Perth*. 1995: 1942
- [6] Qin Y Q, Sun D B, Li N, et al. Path planning for mobile robot based on particle swarm optimization. *Robot*. 2004, 26(3): 222
(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. *机器人*, 2004, 26(3): 222)
- [7] Sun B, Chen W D, Xi Y G. Particle swarm optimization based glob-

- al path planning for mobile robots. *Control Decis* 2005, 20(9): 1052
(孙波, 陈卫东, 席裕庚. 基于粒子群优化算法的机器人全局路径规划. *控制与决策*, 2005, 20(9): 1052)
- [8] Zhao X Z, Chang H X, Zeng J F, et al. Path planning method for mobile robot based on particle swarm algorithm. *Appl Res Comput* 2007, 24(3): 181
(赵先章, 常红星, 曾隽芳, 等. 一种基于粒子群算法的机器人路径规划方法. *计算机应用研究*, 2007, 24(3): 181)
- [9] Yang W, Li Q Q. Survey on particle swarm optimization algorithm. *Eng Sci* 2004, 6(5): 87
(杨维, 李歧强. 粒子群优化算法综述. *中国工程科学*, 2004, 6(5): 87)
- [10] Van den Bergh F, Engelbrecht A P. A new locally convergent particle swarm optimizer // *Proceedings of IEEE International Conference on Systems Man and Cybernetics Hammamet* 2002; 96
- [11] Shi Y, Krohling R A. Co-evolutionary Particle Swarm Optimization to Solve Minimax Problems // *Proceedings of the IEEE Congress on Evolutionary Computation*. Honolulu, 2002; 1682
- [12] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization // *Proceedings of the 7th International Conference on Evolutionary Programming*. San Diego, 1998; 591
- [13] Shi Y, Eberhart R C. Empirical study of particle swarm optimization // *Proceedings of the 1999 Congress on Evolutionary Computation*. Washington D C, 1999; 1945
- [14] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficient. *IEEE Trans Evol Comput* 2004; 240

(上接第 369 页)

- (黄兆龙, 李隆盛, 湛渊源, 等. 沉泥轻质骨材制造与基础性能分析研究. 涌源工程股份有限公司, 1999)
- [3] Yang J C, Xie J N. Planning research of silt flowing into reservoir // *Reservoir Research Project Results Sustainable Management Symposium*. Taipei, 1998; 36
(杨锦川, 谢进南. 水库淤沙研究课题之规划 // 水库永续经营研究计划成果研讨会论文集. 台北, 1998; 36)
- [4] Peng Y C, Huang C L. Engineering properties of sintered waste sludge as lightweight aggregate in a densified concrete mixture. *J Chongqing Univ* 2009, 8(4): 231
- [5] Lin W M. Structural lightweight concrete properties. *Struct Eng* 1999, 26(3): 56
(林维明. 结构轻质混凝土土性质. *结构工程*, 1999, 26(3): 56)
- [6] AC 221.2-81 Standard Practice for Selecting Proportions for Structural Lightweight Concrete
- [7] AC 221.2-91 Standard Practice for Selecting Proportions for Normal Heavyweight and Mass Concrete
- [8] Liou J S. A Study on the Flexural Properties of Lightweight Concrete Beams [Dissertation]. Taipei: Taiwan University, 2007, 121; 53
- [9] Huang S C, Chang F C, Lob S L, et al. Production of lightweight aggregates from mining residues: heavy metal sludge and incinerator fly ash. *J Hazardous Mater* 2007, 144: 52
- [10] Jo B W, Park S K, Park J B. Properties of concrete made with alkali-activated fly ash lightweight aggregate (AFLA). *Constr Compos* 2007, 29: 128
- [11] Skuratova V A, Abu Alazm S M, Alyonov V A. Luminescence of aggregate centers in lithium fluoride irradiated with high energy heavy ions. *Nucl Instrum Methods Phys Res Sect B* 2002, 191: 251