

动态障碍物环境下移动机器人的全局路径规划研究

李二超¹, 王慧莹¹, 杨秀平², 梁 波¹

(1. 兰州理工大学电气工程与信息工程学院, 甘肃 兰州 730050)

(2. 兰州理工大学经济管理学院, 甘肃 兰州 730050)

[摘要] 针对环境中存在动态障碍物时, 如何运用全局路径规划算法求解移动机器人的最佳路径, 设定动态障碍物的运动范围是已知的, 则危险程度是一个区间数. 定义一种 Pareto 概率支配公式, 求出不同区间数之间的占优概率, 由此得出哪条路径的安全程度更高. 对传统 NSGA-II 算法进行改进, 根据约束函数把所有的解区分为可行解与非可行解, 引入非可行解储备集储存好的非可行解, 引导可行解进化出更好的解. 建立环境模型, 用 Matlab 软件进行仿真, 仿真结果表明对不同的障碍物环境, 该方法均能规划出安全无碰的路径, 与传统算法进行对比, 改进后算法在求解动态障碍物环境下的机器人路径规划问题更加可行有效.

[关键词] 移动机器人, 路径规划, 动态障碍物, NSGA-II 算法

[中图分类号] TP273 [文献标志码] A [文章编号] 1001-4616(2017)03-0052-07

Research on Global Path Planning of Mobile Robot in Dynamic Obstacle Environment

Li Erchao¹, Wang Huiying¹, Yang Xiuping², Liang Bo¹

(1. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China)

(2. College of Economy and Management, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: For the dynamic obstacles in the environment, how to use the global path planning algorithm to solve the optimal path of mobile robot, assuming that the motion range of the dynamic obstacle is known, so the hazard level is an interval number. Define a Pareto probability dominance formula, find out the dominant probability between different interval numbers, which can be used to determine which path is more secure. The traditional NSGA-II algorithm is improved, and all solutions are classified into feasible and infeasible solutions according to the constraint function, the non feasible solution set is introduced to store the non feasible solution, guide feasible solutions to evolve better solutions. The environment model is established and simulated with MATLAB software. The simulation results show that for different obstacle environment, this method can be used to plan a safe path, compared with the traditional algorithm, the improved algorithm in solving the dynamic obstacle environment for robot path planning problem is more feasible and effective.

Key words: mobile robot, path planning, dynamic obstacles, NSGA-II algorithm

如今, 机器人已经大量应用在各种行业, 机器人不仅可以代替人们在有辐射、有粉尘、有毒等恶劣环境中作业, 还可以广泛应用于科学考察、地质勘探、灾难急救、制造业、农业生产等多个领域中^[1-2]. 60 年代, 美国研究的火星探索用移动机器人, 实现了在火星上软着陆后移动并采集数据. 近年来, 日本、美国等国家已研制出多种农业用机器人: 采摘机器人、耕耘机器人、除草机器人、饲喂机器人等.

不论是火星探索机器人还是工业机器人, 都要求能够在一定的范围内安全运动, 完成特定的任务, 所以, 路径规划和避障问题是首要解决的一个共同技术问题. 路径规划一般的分类方法是依据机器人对作业环境信息的掌握程度分为全局路径规划和局部路径规划, 其中全局路径规划在规划前已得到环境的整体信息^[3-5]. 而机器人实际工作的环境多是复杂多变的, 在移动过程中可能会遇到不确定的障碍物, 这时

收稿日期: 2017-03-18.

基金项目: 国家自然科学基金(61403175、41501597).

通讯联系人: 李二超, 博士, 副教授, 研究方向: 多目标优化、机器人控制. E-mail: lecstarr@163.com

机器人怎样在运动中避开障碍物就显得尤为重要^[6]. 本文考虑环境中存在不确定性障碍物, 给定其活动范围, 将 NSGA-II 算法引入机器人路径规划中, 并对传统的 NSGA-II 算法做出部分改进, 引入非可行储备集保留部分好的不可行解, 运行算法结果得到接近最优的路径^[7], 与传统 NSGA-II 算法进行对比, 改进后的算法搜索出的路径更加安全有效.

1 问题描述与建模

1.1 问题描述

本文要求解的问题可描述如下: 设环境中存在多个静态障碍物和不确定性障碍物, 不确定性障碍物也可称为危险源, 将机器人放入这样的环境中, 怎样规划出一条或多条从起始点到目标点并实现与障碍物无碰的路径, 权衡路径长度和安全程度两个相互冲突的目标, 筛选出若干条折中的最佳移动轨迹.

1.2 环境建模

机器人路径规划问题的第一步就是要建立一个适当的环境模型, 本文研究的是陆地移动机器人, 将机器人的移动空间用二维物理空间表示, 机器人用一个可以朝 360° 任意方向移动的点表示, 障碍物按机器人半径进行膨胀, 只考虑运动空间的几何约束, 不考虑机器人的运动学约束.

在全局坐标系 $O-XY$ 中, 起始点和目标点分别标记为 ST、TA, 用多边形表示静态障碍物, 隐形边框的圆形表示危险源的活动范围. 当起始点和目标点所在直线与 X 轴相交时, 对坐标进行如下变换: 以连接点 ST 和 TA 的直线作为 X' 轴, 以垂直于 X' 且经过 ST 点的直线为 Y' 轴^[8].

将线段 ST-TA 进行 $(n+1)$ 等分, 并在每个等分点处作垂线, 得到一组平行直线族 $\{l_1, l_2, \dots, l_n\}$. 该直线族与任一路径的交点将对应一个点序列 $(ph_1, ph_2, \dots, ph_n)$, 这些点序列加上起始点 ST 和 TA 就构成机器人的一条完整路径^[9], 可用如下所示的点集描述:

$$PH = \{ST, ph_1, ph_2, \dots, ph_n, TA\}. \quad (1)$$

1.2.1 约束违背函数

通过约束违背函数, 可以判断路径是否为可行路径. 只有当函数值为零时, 表明路径与环境中的所有静态障碍物无碰撞, 是可行的; 函数值不为零时, 表明路径与静态障碍物有一处或多处碰撞, 是不可行的, 函数公式如下:

$$cv(x_i) = \sum_{j=1}^{No_ob} cv_j(x_i),$$

$$cv_j(x_i) = \begin{cases} 1 & x_i \text{ 与障碍物 } OB_j \text{ 碰撞,} \\ 0 & x_i \text{ 不与障碍物 } OB_j \text{ 碰撞.} \end{cases} \quad (2)$$

1.2.2 路径总长度

将起始点 ST 设为 ph_0 , 目标点 TA 设为 ph_{n+1} , 路径 PH 总长度的计算公式为:

$$L(PH) = \sum_{j=0}^n d(ph_j, ph_{j+1}). \quad (3)$$

式中, n 表示路径段总数, $d(ph_j, ph_{j+1})$ 表示点 ph_j 与点 ph_{j+1} 间的距离, 在坐标系 $0-X'Y'$ 中, $d(ph_j, ph_{j+1})$ 的计算公式可表示为:

$$d(ph_j, ph_{j+1}) = \sqrt{(x_{ph_{j+1}} - x_{ph_j})^2 + (y_{ph_{j+1}} - y_{ph_j})^2} = \sqrt{\left(\frac{d(ph_0, ph_{n+1})}{n+1}\right)^2 + (y_{ph_{j+1}} - y_{ph_j})^2}. \quad (4)$$

1.2.3 路径安全程度

安全程度用来评价路径与不确定障碍物间发生碰撞的机率, 由于不确定障碍物给出的是活动范围, 具体位置未知, 所以决定了安全程度为一个区间数, 首先, 求出路径段与危险源之间的距离, 设危险源的活动范围为:

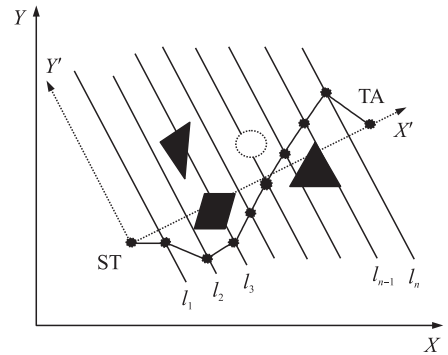


图 1 环境建模

Fig. 1 Environmental modeling

$$\Omega_i = \{ (x, y) \mid (x - x_{rc})^2 + (y - y_{rc})^2 \leq r_c^2 \}. \quad (5)$$

则路径段与危险源之间的最大与最小距离分别是:

$$d(\Omega_i)_{\max} = d(o_c) + r_c. \quad (6)$$

$$d(\Omega_i)_{\min} = \begin{cases} 0, & \text{PH} \cap \Omega_i \neq \emptyset; \\ d(o_c) - r_c, & \text{PH} \cap \Omega_i = \emptyset. \end{cases} \quad (7)$$

式中, (x_{rc}, y_{rc}) 为圆心, r_c 为圆的半径, $d(o_c)$ 为路径到圆心的距离. 定义安全度的率属度函数为:

$$\text{dan}_i = \begin{cases} 1, & d(Ds_i) \leq d_{\min}, \\ \frac{d_{\max} - d(Ds_i)}{d_{\max} - d_{\min}}, & d_{\min} < d(Ds_i) < d_{\max}, \\ 0, & d(Ds_i) \geq d_{\max}. \end{cases} \quad (8)$$

式中, $d(Ds_i)$ 为路径段与第 i 个危险源间的距离, d_{\max} 是危险源的最大活动半径, 当 $d(Ds_i) > d_{\max}$ 时, 路径是绝对安全的, 安全程度为 0; d_{\min} 为危险源的最小活动半径, 当 $d(Ds_i) < d_{\min}$ 时, 路径是绝对危险的, 安全程度

为 1. 当 $d(Ds_i)$ 的值介于 d_{\max} 与 d_{\min} 之间时, 由率属度公式 $\frac{d_{\max} - d(Ds_i)}{d_{\max} - d_{\min}}$ 求得安全程度的值.

由于路径安全程度测的是求得的路径与不确定危险源的碰撞概率, 显然其对于已知的静态障碍物是百分之百安全无碰撞的, 所以这里计算出的安全程度是局部的.

1.2.4 危险程度的比较

由(3)可知危险程度在一个区间取值, 所以其比较也就是区间数的比较. 设 x_i, x_j 两个目标的取值区间分别为 $\vec{F}(x_i) = [f(x_i)^-, f(x_i)^+]$ 和 $\vec{F}(x_j) = [f(x_j)^-, f(x_j)^+]$, 并假设 x_i 和 x_j 所对应真实目标函数值在区间上均服从均匀分布. $\vec{F}(x_i)$ 与 $\vec{F}(x_j)$ 的在一维坐标上的位置分为以下 3 种类型:

类型 1: 两区间互不相交, $f(x_i)^- \geq f(x_j)^+$;

说明选择所得 x_i 真实目标值一定好于 x_j , 因此, 有理由认为目标空间上可行解 x_i 以概率 1 优于 x_j , 而 x_j 优于 x_i 的概率为 0.

类型 2: 两区间部分相交, $f(x_i)^+ \geq f(x_j)^+ \geq f(x_i)^- \geq f(x_j)^-$;

说明真实目标值 x_i 同时存在大于、小于或等于 x_j 的可能. 先考虑区间 $[f(x_j)^+, f(x_i)^+]$, x_i 落在该区间的概率为 $\frac{f(x_i)^+ - f(x_j)^+}{f(x_i)^+ - f(x_i)^-}$, 在该区间 x_i 以概率 1 优于 x_j , 再考虑区间 $[f(x_i)^-, f(x_j)^+]$, x_i 落在该区间的概率为 $\frac{f(x_j)^+ - f(x_i)^-}{f(x_j)^+ - f(x_j)^-}$, 在该区间上 x_i 优于 x_j 的概率为 0.5, 所以, 在目标空间上 x_i 优于 x_j 的概率 $p(x_i < x_j)$ 为:

$$p(x_i < x_j) = \frac{f(x_i)^+ - f(x_j)^+}{f(x_i)^+ - f(x_i)^-} + \frac{f(x_j)^+ - f(x_i)^-}{f(x_i)^+ - f(x_i)^-} \left(0.5 \frac{f(x_j)^+ - f(x_i)^-}{f(x_j)^+ - f(x_j)^-} + \frac{f(x_i)^- - f(x_j)^-}{f(x_j)^+ - f(x_j)^-} \right). \quad (9)$$

类型 3: 一个区间内包含着另一个区间 $f(x_i)^+ \geq f(x_j)^+ \geq f(x_j)^- \geq f(x_i)^-$;

说明真实目标值 x_i 同时存在大于、小于或等于 x_j 的可能. 先考虑区间 $[f(x_j)^+, f(x_i)^+]$, x_i 落在该区间的概率为 $\frac{f(x_i)^+ - f(x_j)^+}{f(x_i)^+ - f(x_i)^-}$, 在该区间 x_i 以概率 1 优于 x_j , 再考虑区间 $[f(x_j)^-, f(x_j)^+]$, x_i 落在该区间的概率为 $\frac{f(x_j)^+ - f(x_j)^-}{f(x_i)^+ - f(x_i)^-}$, 在该区间 x_i 以 0.5 的概率优于 x_j , 所以, 在目标空间上 x_i 优于 x_j 的概率 $p(x_i < x_j)$ 为:

$$p(x_i < x_j) = \frac{f(x_i)^+ - f(x_j)^+}{f(x_i)^+ - f(x_i)^-} + 0.5 \frac{f(x_j)^+ - f(x_j)^-}{f(x_i)^+ - f(x_i)^-}. \quad (10)$$

以上的 $p(x_i < x_j)$ 满足:

(1) $0 \leq p(x_i < x_j) \leq 1$;

(2) $p(x_i < x_j) + p(x_j < x_i) = 1$.

$p(x_i < x_j) = q$ 表示 x_i 的目标值以概率 q 优于 x_j 目标值带来的效益,由 $p(x_i < x_j)$ 或 $p(x_j < x_i)$ 的值可以比较 x_i 与 x_j 目标值的大小.

2 基于改进 NSGA-II 算法的路径规划

2.1 传统 NSGA-II 算法

NSGA-II 是在 NSGA 的基础上改进得到的一种 MOEA^[10],其最突出的特点是提出了快速非支配排序法^[11-12],相比原始的 NSGA 降低了计算复杂度,由原来的 $O(mN_3)$ 降到 $O(mN_2)$,其中, m 为目标函数个数, N 为种群大小^[13]. 并且求出的解均匀分布,保持了种群的多样性. 引入了精英策略^[14],将父代种群与其产生的子代种群组合,优胜劣汰产生下一代种群,有利于保持父代中的优良个体进入下一代,并通过对种群中所有个体的分层排序,使得最佳个体不会丢失,提高了种群水平.

2.2 对传统 NSGA-II 算法的改进

在原来的算法中,所有不满足约束条件的解将会被舍去,但是这些非可行解中含有部分有价值的解,如路径长度短或分布性好的解,在改进的算法中,保留一定量的非可行解,设为 Na 个,通过约束违背函数可得每条路径与障碍物碰撞的路径段数,选取碰撞次数少的前 Q 个,若 $Q > Na$,再计算这 Q 个非可行解的拥挤距离,根据拥挤距离排序选取前 Na 个拥挤距离较大的解合并到可行解集合中,共同参与父代种群的筛选. 以后生成的每一代个体都将循环这样的步骤:分为可行解与不可行解——筛选 Na 个优秀的非可行解合并到可行解集合——进行非支配排序与拥挤距离计算——选取父代种群进行交叉变异.

2.3 算法步骤

- (1) 随机生成 N 个初始种群 P .
- (2) 根据路径约束违背函数将种群 P 划分为可行解与非可行解.
- (3) 选取碰撞次数少的前 Q 个非可行解,若 $Q > Na$,再计算这 Q 个非可行解的拥挤距离,根据拥挤距离排序选取前 Na 个拥挤距离较大的个体进入非可行储备集 Q_1 .
- (4) 将 Q_1 合并到可行解集合,作为备选染色体集合 A .
- (5) 根据非支配排序和拥挤距离对 Q_2 进行排序,采用锦标赛选择法选择 $N/2$ 个个体作为第一代父种群.
- (6) 对父种群进行自适应交叉变异操作生成子种群,将子种群和父种群合并成规模为 $2N$ 的新种群,转到步骤(2)将新种群划分为可行解与不可行解.
- (7) 同步步骤(3)选择非可行解中碰撞次数少、拥挤距离大的前 Na 个个体保存到非可行储备集中,再将可行解与此非可行储备集中的个体合并为集合 B .
- (8) 根据非支配排序和拥挤距离对 B 进行排序,采用精英策略筛选出前 N 个优秀个体作为下一代父种群.
- (9) 判断进化代数是否达到最大进化代数,若没有,跳转到步骤(6),否则根据非支配排序和拥挤距离从 N 个个体中选出 R 个优秀个体.
- (10) 算法结束.

3 实验结果与分析

3.1 实验参数设置

为了说明改进后算法的有效性,设计在给定的相同环境下,分别采用未做改进的 NSGA-II 算法和改进后的 NSGA-II 算法进行实验,对实验结果图和数据进行对比分析. 实验参数设置如表 1 所示.

表 1 参数设置

Table 1 Parameter setting

种群数目	迭代次数	输出个体数	η_c	η_m	圆半径	决策者容忍系数
50	800	10	20	20	2	0.6

3.2 实验结果与分析

实验一:机器人的活动环境设置为一个静态障碍物和一个动态障碍物(环境 1).

在给定的活动环境下,本文算法每运行 1 次可以有效地得到 50 个非支配路径解,取 R 为 10,筛选留下前 10 个最优解,图 2 是未经改进的 NSGA-II 算法在环境 1 下搜索的 10 条全局最优路径,图 3 是改进后的 NSGA-II 算法在环境 1 下搜索的 10 条全局最优路径.

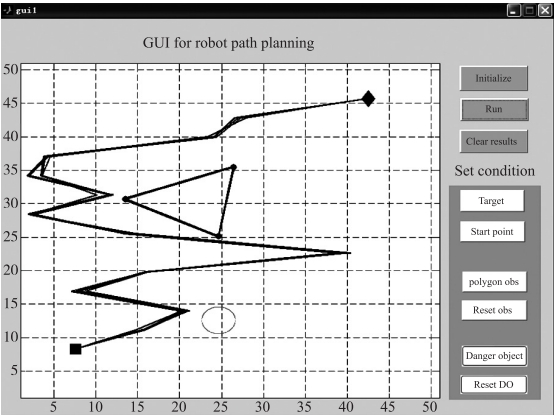


图 2 传统算法仿真结果图(实验一)
Fig. 2 Simulation results of traditional algorithm(Experiment 1)

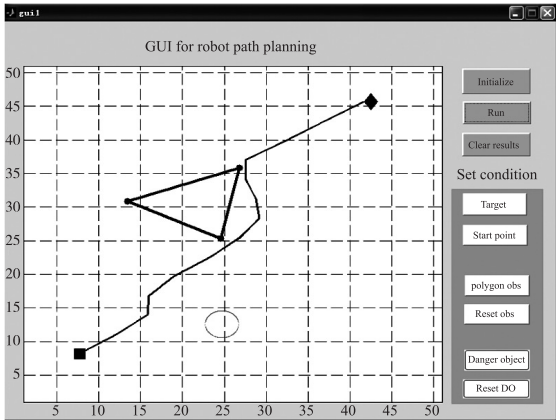


图 3 改进算法仿真结果图(实验一)
Fig. 3 Simulation results of the improved algorithm(Experiment 1)

表 2、表 3 分别是传统算法和改进算法在环境 1 中得到的 10 条路径的测试结果.

表 2 传统算法(实验一)

Table 2 Traditional algorithm(Experiment 1)			
路径序号	路径总长度	危险程度上限	危险程度下限
1	159.460 053 9	0.317 577 910	0.762 022 354
2	163.459 987 9	0.254 227 622	0.698 672 066
3	163.590 873 1	0.226 253 988	0.670 698 432
4	160.595 065 4	0.292 479 361	0.736 923 806
5	160.628 400 4	0.273 540 191	0.717 984 635
6	156.846 026 9	0.347 297 096	0.791 741 541
7	163.563 662 0	0.252 912 884	0.697 357 328
8	161.402 718 2	0.260 497 344	0.704 941 789
9	156.959 199 1	0.347 842 705	0.792 287 150
10	163.715 822 5	0.249 212 765	0.693 657 209
平均值	161.022 180 9	0.282 184 187	0.726 628 631

总耗时 43.688 s.

表 3 改进算法(实验一)

Table 3 The improved algorithm(Experiment 1)			
路径序号	路径总长度	危险程度上限	危险程度下限
1	55.976 346 97	0	0.189 652 596
2	55.978 468 83	0	0.189 726 971
3	55.980 017 09	0	0.189 667 513
4	55.978 711 25	0	0.189 838 276
5	55.984 029 00	0	0.190 387 387
6	55.985 150 03	0	0.190 373 391
7	55.993 922 21	0	0.191 101 944
8	56.123 535 09	0	0.191 676 425
9	56.002 376 60	0	0.199 565 048
10	56.123 798 29	0	0.191 226 094
平均值	56.012 635 54	0	0.191 321 564

总耗时 21.812 s.

图 2 中的路径虽然都是安全的,但是转弯多路径太长,明显达不到优化的效果,图 3 中的路径不仅绝对安全,而且出现折线少,很接近最短路径,实现了优化的效果. 由表 1 和表 2 中的数据可知传统算法得到的路径平均总长度是 161.022 180 9,改进算法得到的平均值是 56.012 635 54;传统算法得到的路径安全程度上限平均值是 0.282 184 187,下限平均值是 0.726 628 631,改进算法得到的路径安全程度上限平均值是 0,下

限平均值是 0.191 321 564;传统算法运行一次所需的时间是 43.688 s,改进算法运行一次所需的时间是 21.812 s. 很明显改进后的 NSGA-Ⅱ 算法在路径总长度、安全程度和算法执行耗时都远远优于未经改进的 NSGA-Ⅱ 算法.

实验二:在实验一的基础上改变迭代次数,设置为 600,其他参数保持不变,机器人的活动范围设置为两个静态障碍物和一个动态障碍物(环境 2).

图 4 是未经改进的 NSGA-Ⅱ 算法在环境 2 下搜索的 10 条全局最优路径,图 5 是改进后的 NSGA-Ⅱ 算法在环境 2 下搜索的 10 条全局最优路径.

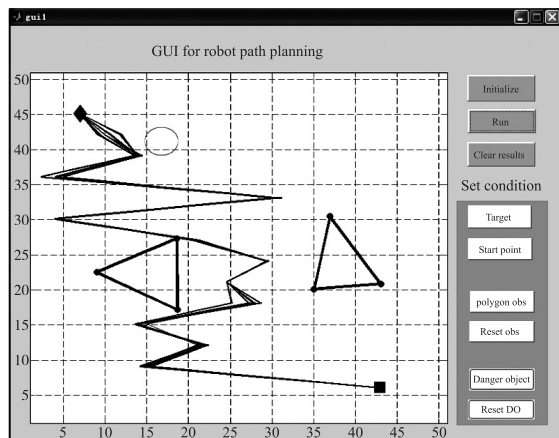


图 4 传统算法仿真结果图(实验二)

Fig. 4 Simulation results of traditional algorithm(Experiment 2)

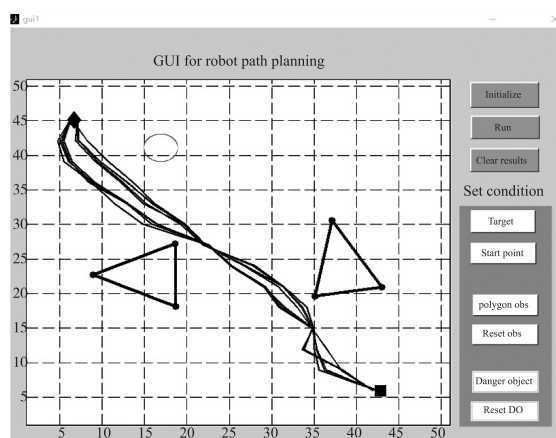


图 5 改进算法仿真结果图(实验二)

Fig. 5 Simulation results of the improved algorithm(Experiment 2)

表 4、表 5 分别是传统算法和改进算法在环境 2 中得到的 10 条路径的测试结果.

表 4 传统算法(实验二)

Table 4 Traditional algorithm(Experiment 2)

路径序号	路径总长度	危险程度上限	危险程度下限
1	167.295 895 4	0.329 107 232	0.773 551 676
2	172.518 267 7	0.314 319 278	0.758 763 723
3	172.547 201 2	0.311 225 400	0.755 669 844
4	163.786 632 7	0.407 452 687	0.851 897 132
5	167.640 646 4	0.316 263 380	0.760 707 824
6	164.516 302 5	0.423 092 341	0.867 536 786
7	172.563 093 1	0.312 042 546	0.756 486 990
8	167.901 426 2	0.347 897 522	0.792 341 966
9	167.790 102 6	0.397 949 672	0.842 394 117
10	167.336 970 1	0.417 099 646	0.861 544 090
平均值	168.389 653 8	0.357 644 970	0.802 089 415

总耗时 101.938 0 s.

表 5 改进算法(实验二)

Table 5 The improved algorithm(Experiment 2)

路径序号	路径总长度	危险程度上限	危险程度下限
1	56.259 412 61	0.334 821 77	0.779 266 215
2	62.499 984 83	0.319 595 181	0.764 039 625
3	56.283 560 52	0.330 712 892	0.775 157 336
4	56.328 971 53	0.327 386 344	0.771 830 789
5	57.392 770 29	0.326 777 735	0.771 222 18
6	56.264 753 40	0.332 765 303	0.777 209 747
7	57.795 325 80	0.324 756 433	0.769 200 878
8	58.001 869 51	0.324 110 603	0.768 555 048
9	56.263 978 10	0.333 821 110	0.778 265 554
10	59.802 184 22	0.322 918 977	0.767 363 421
平均值	57.689 281 08	0.327 766 635	0.772 211 079

总耗时 20.392 0 s.

由表 4 和表 5 可知,传统算法得到的路径平均总长度是 168.389 653 8,改进算法得到的平均值是 57.689 281 08;传统算法得到的路径安全程度上限平均值是 0.357 644 97,下限平均值是 0.802 089 415,改进算法得到的路径安全程度上限平均值是 0.327 766 635,下限平均值是 0.772 211 079;传统算法运行一次所需的时间是 101.938 0 s,改进算法运行一次所需的时间是 20.392 0 s. 相比环境 1 更复杂的环境 2,依然可以看出改进后的 NSGA-Ⅱ 算法在路径总长度、安全程度和算法执行耗时都要优于未经改进的 NSGA-Ⅱ 算法.

实验三:探讨障碍物的范围对算法性能是否存在影响. 按表 1 的参数设置,仅改变圆半径的大小,其他参数设置保存不变,分别测试圆半径取值为 4、6、8 时搜索出的路径图. 实验结果显示半径取值 4 和 6 时,算法仍能快速搜索出性能好的路径,仿真结果图为图 6 和图 7. 当半径值超过 8 时,算法难以搜索出合适的路径,所以建议参数半径值的设置范围为 2~6.

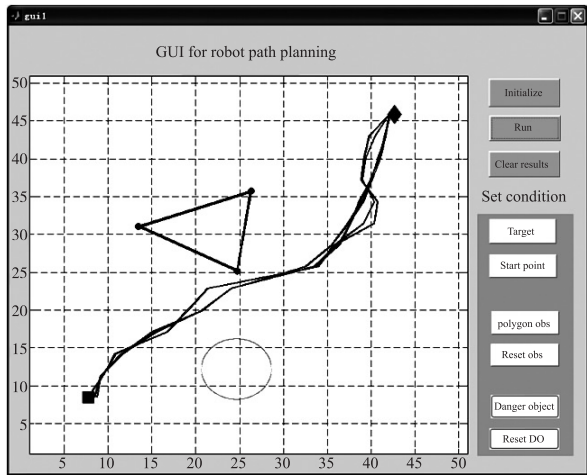


图 6 半径值为 4 的仿真结果

Fig. 6 Simulation results with a radius of 4

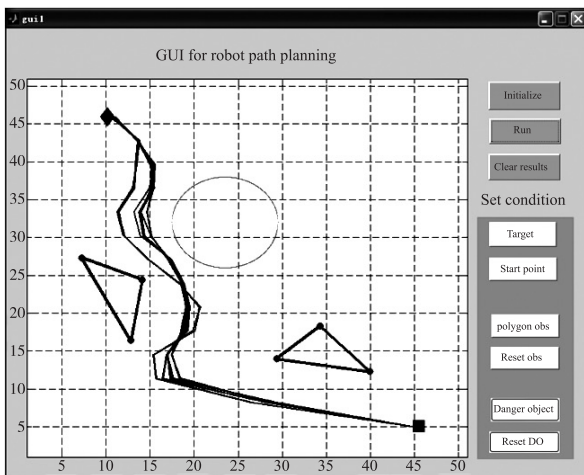


图 7 半径值为 6 时的仿真结果

Fig. 7 Simulation results with a radius of 6

4 结束语

针对环境中存在动态障碍物时,移动机器人如何搜索出一条安全无碰的最优路径,本文提出了一种改进的 NSGA-Ⅱ 算法,将好的不可行解考虑进采样空间,引导可行解向最优解收敛,从仿真结果图表中可以看出,在复杂程度不同的仿真环境中,改进后的 NSGA-Ⅱ 算法搜索出的路径在路径长度、安全程度和算法运算效率上都明显优于传统 NSGA-Ⅱ 算法,证明所提算法在解决机器人全局路径规划问题上有效可行.

[参考文献]

- [1] 周毅. 基于遗传算法的移动机器人路径规划研究[D].保定:河北工业大学,2015.
- [2] 侯占军. 移动机器人自主避障方法的研究[D].北京:北京工业大学,2008.
- [3] TANG B,ZHU Z,LUO J. Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning[J]. International journal of advanced robotic systems,2016,13(3):1-2.
- [4] 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.
- [5] 陈智. 基于栅格法多目标路径规划研究[D]. 武汉:华中科技大学,2015.
- [6] QU H,XING K,ALEXANDER T. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots[J]. Neurocomputing,2013,120(10):509-517.
- [7] 吴青松. 基于遗传算法的移动机器人路径规划研究[D]. 成都:电子科技大学,2005.
- [8] 张勇,巩敦卫,任永强,等. 用于约束优化的简洁多目标微粒群优化算法[J]. 电子学报,2011,39(6):1 436-1 440.
- [9] 张勇. 区间多目标优化问题的微粒群优化理论及应用[D]. 徐州:中国矿业大学,2009.
- [10] 申晓宁,郭毓,陈庆伟,等. 多目标遗传算法在机器人路径规划中的应用[J]. 南京理工大学学报(自然科学版),2006,30(6):659-663.

(下转第 66 页)

- processing letters, 2016, 44(2): 1–19.
- [19] MICHE Y, SORJAMAA A, BAS P, et al. OP-ELM: optimally pruned extreme learning machine[J]. IEEE transactions on neural networks, 2010, 21(1): 158–162.
- [20] HINTON G E, SRIVASTAVA N, KRIZHEVSKY A, et al. Improving neural networks by preventing co-adaptation of feature detectors[DB/OL]. <https://arxiv.org/abs/1207.0580>, 2012.
- [21] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of machine learning research, 2014, 15(1): 1 929–1 958.
- [22] BALDI P, SADOWSKI P. The dropout learning algorithm[J]. Artificial intelligence, 2014, 210(210): 78–122.
- [23] WAGER S, WANG S, LIANG P. Dropout training as adaptive regularization[C]//Advances in Neural Information Processing Systems, Laka Tahoe, Nevada, 2013: 351–359.
- [24] IOSIFIDIS A, TEFAS A, PITAS I. DropELM: fast neural network regularization with dropout and drop connect[J]. Neurocomputing, 2015, 162: 57–66.
- [25] BA L J, FREY B. Adaptive dropout for training deep neural networks[C]//Advances in Neural Information Processing Systems, Laka Tahoe, Nevada, 2013.
- [26] LI Z, GONG B, YANG T. Improved dropout for shallow and deep learning[C]//Advances in Neural Information Processing Systems, Barcelona, Spain, 2016: 1–10.
- [27] YANG W, JIN L, TAO D, et al. Drop sample: a new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition[J]. Pattern recognition, 2016, 58: 190–203.
- [28] KLEIN E B, STONE W N, HICKS M W, et al. Understanding Dropouts[J]. Advances in neural information processing systems, 2013, 26(2): 89–100.
- [29] FRANK A, ASUNCION A. UCI machine learning repository[DB/OL]. [<http://archive.ics.uci.edu/ml>], 2013.

[责任编辑: 顾晓天]

(上接第58页)

- [11] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271–289.
- [12] 王明昭, 王宇平, 王晓丽, 等. 一种均匀聚集距离的改进 NSGA-II 算法[J]. 西安电子科技大学学报(自然科学版), 2016, 43(3): 49–54.
- [13] DEB K, AGRAWAL S, PRATAP A, et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II[J]. Lecture notes in computer science, 2002, 1917: 849–858.
- [14] 陈婕, 熊盛武, 林婉如, 等. NSGA-II 算法的改进策略研究[J]. 计算机工程与应用, 2011, 47(19): 42–45.
- [15] AHMED F, DEB K. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms[J]. Soft computing, 2013, 17(7): 1283–1299, .
- [16] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part ii: handling constraints and extending to an adaptive approach[J]. IEEE transactions on evolutionary computation, 2014, 18(4): 602–622.

[责任编辑: 黄 敏]