

DOI: 10.13382/j.jemi.B2205754

基于区域搜索粒子群算法的机器人路径规划*

郝 琨 邓晁硕 赵 璐 刘永磊

(天津城建大学计算机与信息工程学院 天津 300384)

摘要:针对粒子群算法应用于移动机器人路径规划时存在的易早熟、易陷入局部最优等问题,提出一种基于区域搜索的自适应粒子群(region search-adaptive particle swarm optimization algorithm, RS-APSO)路径规划方法。首先,通过区域搜索算法对原始地图进行预处理,减少地图中的无效信息。其次,提出两种可变算子对惯性权重因子进行调节,对加速因子进行自适应改进,增强算法不同时期的搜索能力,利用新的加速因子使粒子快速摆脱较差区域。最后通过动态避障策略,使机器人可以安全规避移动障碍物。仿真结果表明,RS-APSO算法相较于PSO算法,平均运行时间降低了30.3%,平均迭代次数降低了43.9%,在动态环境中也能生成安全路径。

关键词: 路径规划;粒子群算法;区域搜索;可变算子;动态避障

中图分类号: TP242.6 **文献标识码:** A **国家标准学科分类代码:** 510.80

Robot path planning based on region search particle swarm optimization

Hao Kun Deng Chaoshuo Zhao Lu Liu Yonglei

(School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin 300384, China)

Abstract: Aiming at the problems of particle swarm optimization in mobile robot path planning, such as precocity and local optimum, a path planning method based on region search-adaptive particle swarm optimization algorithm (RS-APSO) is proposed. Firstly, the region search algorithm is used to preprocess the original map to reduce invalid information in the map. Secondly, two variable operators are proposed to adjust the inertia weight factor and improve the acceleration factor adaptively to enhance the search ability of the algorithm in different periods. The new acceleration factor is used to remove the bad region quickly from the particle. Finally, the robot can safely avoid moving obstacles through dynamic obstacle avoidance strategy. Simulation results show that compared with PSO algorithm, the average running time of RS-APSO algorithm is reduced by 30.3%, the average number of iterations is reduced by 43.9%, and can also generate safe path in dynamic environment.

Keywords: path planning; particle swarm optimization (PSO); regional search; variable operator; dynamic obstacle avoidance

0 引 言

随着机器人技术的发展,移动机器人被广泛应用在生活中各个领域^[1-2]。在移动机器人的应用中,路径规划是关键技术之一。路径规划是指移动机器人根据特定性能标准搜索从初始状态到目标状态的最优或次优路径^[3]。路径规划算法分为两类。一类是传统算法,包括RRT算法^[4]、A*算法^[5]、人工势场法^[6]、可视图法^[7]等,

另一类是智能算法,包括粒子群算法^[8]、遗传算法^[9]、蚁群算法^[10]、人工鱼群算法^[11]、神经网络算法^[12]等。与传统算法相比,智能算法在面对复杂地图时,通常可以更快的找到较优路径。粒子群算法是模仿鸟类觅食行为设计的一种智能算法。由于其具有参数简单、收敛速度快、计算复杂度低等优点,被广泛的应用在路径规划中。但该算法也存在易早熟、易陷入局部最优解、在复杂地图中寻优时间长等问题。针对这些问题,许多国内外学者做了深入的探索。

收稿日期: 2022-08-17 Received Date: 2022-08-17

* 基金项目: 国家自然科学基金(61902273)项目资助

Zhang 等^[13]采取精英策略对粒子进行更新,可以快速得到符合要求的路径。但精英策略限制了种群多样性。Shao 等^[14]采用基于混沌的 Logistic 映射来改善粒子初始分布,并通过突变策略,提高了算法的收敛速度。但该算法在运行时间上的提升有限。Mandava 等^[15]将粒子群算法与静止障碍物的势场法相结合,对路径平滑度进行处理。但生成路径的长度与平滑度并不理想。Tharwat 等^[16]提出了混沌粒子群算法来优化 Bezier 曲线的控制点,但该方法仅适用于静态环境。Cheng^[17]等结合混沌灰狼优化算法,提高算法全局搜索能力并尽快摆脱局部最优,但生成路径的平滑度有待提升。贾会群等^[18]提出自适应惯性权重加速因子粒子群算法,提高了粒子群算法的搜索效率。但该算法的适应度评价只考虑了路径长度,并未考虑其他因素。Song 等^[19]提出了一种新的多模态延迟粒子群优化算法,根据进化状态对速度更新公式进行切换。但该算法的延迟信息来源于随机选取的先前的局部粒子和全局粒子,降低了算法收敛速度。

综上,针对基本粒子群算法在路径规划中存在的初始地图探寻时间过长、易早熟、易陷入局部最优等问题,本文提出一种基于区域搜索的自适应粒子群路径规划方法。该方法采用区域搜索算法解决地图探寻时间过长的问題;通过两种可变算子与自适应加速因子,提高粒子全局搜索能力与局部搜索能力;利用新的加速因子避免粒子陷入较差区域;通过动态避障策略,使生成的路径可以安全规避移动障碍物。

1 环境建模

常用的环境建模方法有自由空间法、可视图法、链接图法、栅格法等。由于栅格法操作便捷、简单,本文采用栅格法构建环境模型。如图 1 所示,将二维工作空间采用栅格法划分为 20×20 的栅格地图。在栅格地图中,编号从左到右、从下到上依次为 $0, 1, 2, 3, \dots, 399$ 。白色栅格(即自由栅格)代表可行区域,黑色栅格代表障碍区域。

在栅格编号与栅格坐标的对应关系如式(1)和(2)所示:

$$\begin{cases} x = \text{mod}(p, N) + 1 \\ y = \text{fix}(p/N) + 1 \end{cases} \quad (1)$$

$$z = (x - 1) + (y - 1) \times N \quad (2)$$

式中: z 为栅格编号; (x, y) 表示栅格所对应的坐标点; N 表示每行的栅格数; mod 与 fix 分别表示求余与取整运算。

为保证路径安全,对地图进行如下预处理:

1) 用质量点来表示移动机器人,对障碍物进行膨胀,以此来保证路径安全;

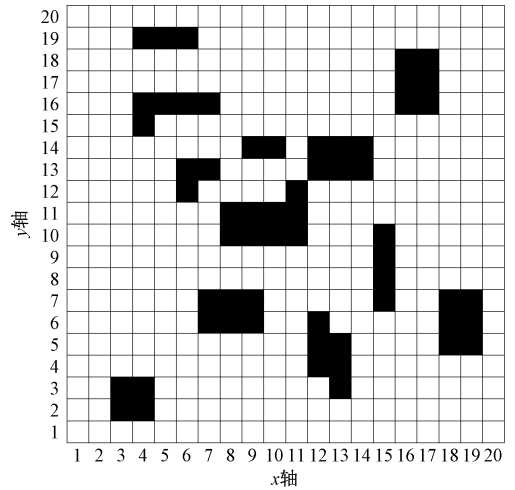


图 1 栅格地图

Fig. 1 Raster map

2) 若障碍物内存在部分栅格未完全占据的情况,则将该障碍物所占据的栅格全部视为障碍物栅格。

2 区域搜索算法

为解决基本粒子群算法在路径规划中存在初始地图探寻时间过长的问題,本文提出区域搜索算法。区域搜索算法分为两步,首先是进行障碍物边界角探寻,找出栅格地图中所有障碍物边界的栅格编号;其次通过区域搜索算法将栅格地图中需要进行搜索的区域标记出来,方便粒子群算法进行探索。

2.1 障碍物边界角探寻

障碍物边界角为障碍物边界之间的夹角或障碍物端点角。首先对地图中所有的障碍物进行障碍物边界角探寻,寻找符合条件的障碍物边界角栅格坐标。从编号为 0 的栅格开始遍历,判断其是否为黑色障碍物栅格,若是,则寻找其周围 8 个方向的栅格,判断是否为自由栅格。 $count$ 统计自由栅格的个数, $count1$ 统计左右两方向自由栅格的个数, $count2$ 统计上下两方向自由栅格的个数。假设第 1 个待探索障碍物栅格的坐标为 $R_i(x_i, y_i)$, 探索过程如式(3)所示:

$$\begin{cases} R(x_i + m, y_i + n) = 0, count = count + 1 \\ R(x_i - 1, y_i) = 0 \text{ or } R(x_i + 1, y_i) = 0, \\ count1 = count1 + 1 \\ R(x_i, y_i + 1) = 0 \text{ or } R(x_i, y_i - 1) = 0, \\ count2 = count2 + 1 \end{cases} \quad (3)$$

式中: m 和 n 取值为 $-1, 0, 1$, 通过 m 和 n 取值不同,对应上、下、左、右、左上、右上、左下、右下 8 个方向的探索。如向左上方向进行探索,则需判断 $R(x_i - 1, y_i + 1)$ 是否

为 0。当自由栅格总个数($count$) ≥ 4 时,具备成为障碍物边界角的必要条件,采用三元组 ($count, count1, count2$) 进行表示,判断条件如表 1 所示。

表 1 障碍物边界角
Table 1 Barrier boundary Angle

自由栅格个数 ($count$)	满足障碍物边界角条件 ($count, count1, count2$)
4	(4,1,1)
5	(5,1,1)、(5,1,2)、(5,2,1)
6	(6,1,1)、(6,1,2)、(6,2,1)
7	(7,1,2)、(7,2,1)

将符合条件的障碍物栅格视为障碍物边界角栅格,统一存入集合 M 中。

以栅格点 $p(7,6)$ 为例,如图 2 所示, p 点周围 8 个方向中,左上、左、左下、下、右下为自由栅格,即 (5,1,1),因此 p 点为障碍物边界角点。

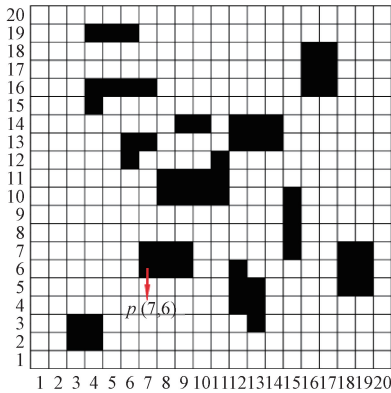


图 2 障碍物边界角探索示意图

Fig. 2 Schematic diagram of obstacle boundary angle exploration

对地图进行障碍物边界角探索后,将所得到的障碍物边界角进行标记,如图 3 所示,三角形标记栅格为地图中障碍物边界角。

2.2 区域搜索

区域搜索算法的基本思想为:假设障碍物栅格的坐标为 $q_i(x_i, y_i)$,在 $N \times N$ 的栅格地图中,使该点以一个栅格长度为步长向上下左右 4 个方向进行探索。将地图中所有的障碍物坐标存入集合 G 中,以向上探索为例,公式如下:

$$\begin{cases} q_u(x_i, y_i + 1) \in G, & \text{停止探索} \\ q_u(x_i, y_i + 1) \notin G \text{ and } 1 \leq x_i \leq N \text{ and } 1 \leq y_i + 1 \leq N, & \text{继续在该方向进行探索} \end{cases}$$

(4)

式中: q_u 为向上进行探索操作。若符合探索条件,则继

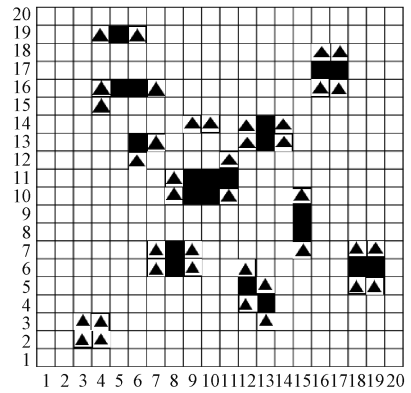


图 3 障碍物边界角

Fig. 3 Boundary angle of obstacle

续在该方向上以一个栅格长度为步长进行探索,否则停止探索。在下、左、右 3 个方向的探索,与上述探索方式相似。以栅格点 $p(7,6)$ 为例,如图 4 所示,根据式 (4),可以发现符合条件的只有阴影部分显示的区域。

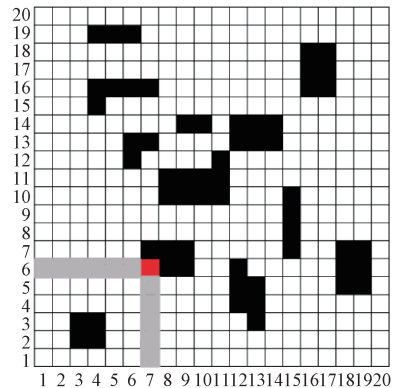


图 4 探索区域示意图

Fig. 4 Map of exploration area

对地图进行区域搜索后,将所得到的待搜索区域标记出来,如图 5 所示,菱形标记栅格为改进粒子群待搜索区域。

对于栅格地图确定搜索区域的过程如下:

1) 从编号为 0 的栅格开始,判断其是否为障碍物栅格,若是,通过式 (3) 对其周围 8 个方向的栅格进行判断,将其周围自由栅格进行标记,统计数量;

2) 当整个地图都遍历完全后,根据障碍物边界角判断标准,判断该障碍物是否为障碍物边界角栅格,将边界角栅格统一存入集合 M 中;

3) 对集合 M 中的每一个栅格,记为 $q_i(x_i, y_i)$,以式 (4) 为例对其上 ($x_i, y_i + 1$)、下 ($x_i, y_i - 1$)、左 ($x_i - 1, y_i$)、右 ($x_i + 1, y_i$) 4 个方向进行探索,将探索过的每个自由栅格都存入集合 H 中,遍历完整个集合 M 后,除去集合 H 中重复的自由栅格,保留的自由栅格即构成了新的

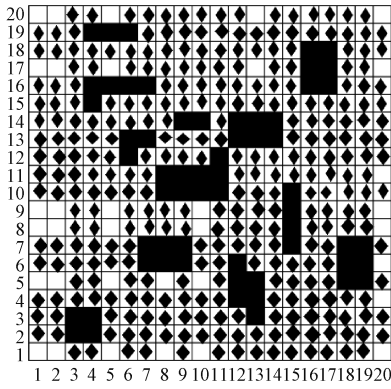


图 5 待搜索区域
Fig. 5 Area to be searched

探索区域。

3 RS-APSO 算法

3.1 RS-APSO 算法框架

本文基于传统粒子群算法,提出基于区域搜索的自适应粒子群算法(RS-APSO)。算法流程如图 6 所示。

首先通过区域搜索算法对地图进行预处理,剔除无效栅格点。然后初始化种群,提出两种可变算子对惯性权重因子进行优化,使其与通过自适应改进后的加速因子共同调节粒子的位置与速度,提高算法前期全局搜索能力与后期局部搜索能力。最后针对陷入较差区域的粒子,引入新的加速因子,重新更新该粒子的速度与位置,使粒子可以快速摆脱较差区域,生成较优路径。

3.2 传统粒子群算法

假设粒子群的种群规模为 n ,搜索区域维数为 D , $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 为粒子 i 当前的位置, $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 为粒子 i 当前的速度, $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ 为粒子 i 搜索的最优位置, $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$ 为整个粒子群搜索的最优位置。更新速度和位置公式如下所示:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 \text{rand}() (p_{id}^k - x_{id}^k) + c_2 \text{rand}() (p_{gd}^k - x_{id}^k) \quad (5)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (6)$$

式中: v_{id}^{k+1} 与 x_{id}^{k+1} 分别为粒子 i 在第 $k+1$ 次迭代时,速度与位置的第 d 维分量; p_{id}^k 为粒子 i 第 k 次迭代后最优位置解的第 d 维分量; p_{gd}^k 为粒子群第 k 次迭代后最优位置解的第 d 维分量; ω 为惯性权重; c_1 为认知因子; c_2 为社会因子; $\text{rand}()$ 为 $(0,1)$ 间的随机数。

3.3 RS-APSO 算法

从式(5)、(6)中可以看出,基本粒子群算法受惯性权重 ω 、认知因子 c_1 和社会因子 c_2 的约束。本文提出 Y

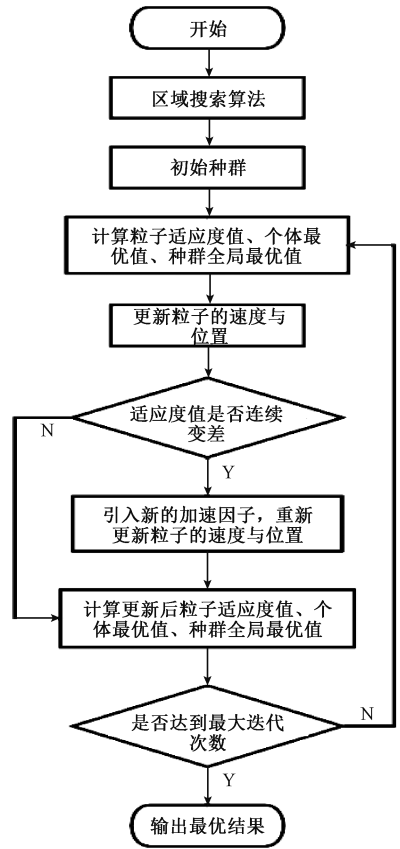


图 6 基于区域搜索的自适应粒子群算法流程
Fig. 6 Flow chart of adaptive particle swarm optimization algorithm based on region search

和 U 两个可变算子,分别在不同情况下,对惯性权重 ω 进行优化,从而提高粒子群算法前期全局搜索能力与后期局部搜索能力,公式如下所示:

$$Y = \frac{c_1 - c_2}{c_1 + c_2} \cdot \frac{k}{k_{\max}} \quad (7)$$

$$U = \frac{c_1}{c_2} \cdot \frac{k}{k_{\max}} \quad (8)$$

$$\omega = \begin{cases} \exp(Y) - \frac{1}{5}, & c_1 \geq c_2 \\ \frac{1}{3U}, & c_1 < c_2 \end{cases} \quad (9)$$

式中: k 为当前迭代次数; k_{\max} 为最终迭代次数, Y 算子与 U 算子可以通过认知因子 c_1 和社会因子 c_2 进行自适应调整。

在基本粒子群算法中,认知因子 c_1 控制个体粒子的学习能力,社会因子 c_2 控制粒子群的学习能力。针对粒子群算法的特殊性,在搜索前期增大 c_1 的值,提高粒子群中的单个粒子向自身最优的学习能力,使其进行更多的探寻;在搜索后期增大 c_2 的值,提高单个粒子向种群最优个体的学习能力,使其找到近似最优解。因此,对参数

c_1, c_2 进行自适应优化,公式如下所示:

$$c_1 = A \cdot \cos\left(\pi \frac{k}{k_{\max}}\right) + \alpha \quad (10)$$

$$c_2 = B \cdot \sin\left[\left(\frac{k}{k_{\max}} + \beta\right) \pi\right] + \beta \quad (11)$$

$$A = \cos\left(\frac{\pi}{2} \cdot \frac{k}{k_{\max}}\right), B = \sin\left(\frac{\pi}{2} \cdot \frac{k}{k_{\max}}\right) \quad (12)$$

式中: $\alpha = \beta = 1.5$ 。经过式(7) ~ (12)的优化,搜索前期 ω 的值较大,粒子群的全局搜索能力得到提升,同时认知因子 c_1 偏大,社会因子 c_2 偏小,可以进一步增强粒子向自身学习的能力;搜索后期 ω 的值较小,粒子群的局部搜索能力得到提升,同时认知因子 c_1 偏小,社会因子 c_2 偏大,可以进一步增强粒子向种群学习的能力。这将使得粒子群算法的搜索能力进一步提高。

3.4 适应度

适应度是用来评价个体优劣的标准,本文综合考虑了路径长度和路径平滑度两个因素,适应度函数公式如下:

$$fitness = \gamma \cdot f1 + \eta \cdot f2 \quad (13)$$

$$\begin{cases} \gamma + \eta = 1 \\ \frac{\gamma}{\eta} = I \end{cases} \quad (14)$$

式中: $fitness$ 为总适应度函数; $f1$ 与 $f2$ 分别代表路径长度与路径平滑度的适应度函数^[20]; I 为两个适应度函数权重比; γ 和 η 为两个适应度函数的权重系数。权重系数的确定是路径规划中非常关键的问题,由于路径长度的重要性高于路径平滑度,故将二者比值设为 1.4, 即 $\gamma = 0.58, \eta = 0.42$ 。

3.5 搜索过程优化

在粒子群进行搜索的过程中,粒子可能会陷入较差的区域^[18],即算法运行过程中适应度值连续多次变差,对算法的性能造成影响。针对该问题,本文采用重新更新当前粒子的速度和位置的方法,使粒子可以更快的跳出该区域。陷入较差区域的判定公式为:

$$fitness(i^k) < fitness(i^{k+1}) \quad (15)$$

在搜索过程中,若连续出现 3 次上述情况,则认定粒子陷入了较差区域。引入新的加速因子 c_3 ,对速度更新公式进行优化,公式如下:

$$\begin{cases} v_{id}^{k+1} = \omega v_{id}^k + c_1 rand() (p_{id}^k - x_{id}^k) + \\ c_2 rand() (p_{gd}^k - x_{id}^k) + c_3 rand() (p_{gd}^k - x_{id}^k) \\ c_3 = \sin\left[\left(\frac{k}{k_{\max}} + \beta\right) \pi\right] + \beta \end{cases} \quad (16)$$

式中: j 为当前粒子序号,被困在较差区域的粒子通过式(16)和(6)可以重新更新速度与位置,跳出当前区域。

3.6 动态避障策略

在实际环境中通常存在移动障碍物,对机器人的行进安全造成巨大威胁。因此本文提出一种新的动态避障策略:将机器人视为质点,为其设置一个碰撞检测区域,当移动障碍物进入碰撞检测区域时,则视为将要发生碰撞,机器人将调整下一路径点的位置进行跳跃避障,从而实现了对障碍物的有效规避。

碰撞检测区域的半径 $collision_radius$ 设定为:

$$\begin{cases} collision_radius = da \cdot Ugl + sa \cdot Ugl \\ da = \mu \cdot \frac{v_{rob}}{v_{obs}} \\ sa = \frac{v_{obs}}{v_{rob}} \end{cases} \quad (17)$$

式中: da 为危险系数、 sa 为安全系数、 Ugl 为单位栅格长度。其中 v_{rob} 为机器人行进速度, v_{obs} 为障碍物移动速度, μ 为比例系数,设 $dist$ 为机器人与障碍物之间的直线距离,则碰撞检测公式如下所示:

$$\begin{cases} dist > collision_radius, \text{未发生碰撞} \\ dist \leq collision_radius, \text{发生碰撞} \end{cases} \quad (18)$$

机器人按照静态规划路径行进时,若行进路径上出现移动障碍物且该障碍物进入碰撞检测区域,则进行跳跃避障:放弃原规划路径中下一路径点,在其同行或同列且靠近终点方向一侧随机选择若干点作为下一路径点的备选点,通过式(19),将 $fitness_jump$ 最大的备选点作为下一路径点。跳跃避障方式如图 7 所示。移动机器人附近环形区域为碰撞检测区域,机器人向终点行进过程中,移动障碍物进入碰撞检测区域,机器人通过选点策略在备选区域选定下一路径点进行跳跃避障,实线路径为动态避障路径。

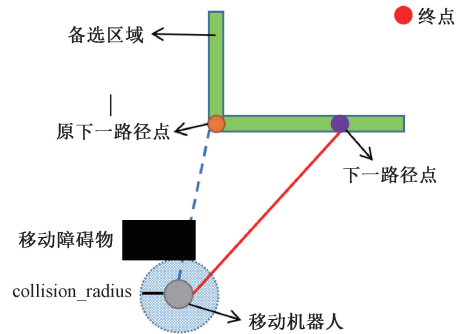


图 7 跳跃避障示意图

Fig. 7 Schematic diagram of jumping obstacle avoidance

$$fitness_jump = f1 \quad (19)$$

4 仿真结果与分析

4.1 仿真环境

为了验证 RS-APSO 在路径规划领域的性能。本文

在栅格地图环境下对 RS-APSO、文献 [18] 提出的 WCPSO、IC-VAINDIWPSO^[21]、基本粒子群算法 (PSO) 和基于区域搜索的基本粒子群算法 RS+PSO 的路径生成、迭代次数及算法运行时间等性能进行比较分析。仿真实验的配置如表 2 所示。

表 2 软硬件配置

Table 2 Hardware and software configuration

硬件	处理器	AMD Ryzen 7 5800H with
	RAM	Radeon Graphics 3.20 GHz 16.0 GB (15.9 GB 可用)
软件	操作系统	Windows 10(64 位操作系统)
	仿真工具	MATLAB r2018a

4.2 20×20 静态栅格地图仿真

在 20×20 栅格地图中,机器人起点栅格编号为 0,终点栅格编号为 399, $U_{gl} = 1.2$ m。根据障碍物数量与规模,将栅格地图分为简单栅格地图与复杂栅格地图,图 8 为 20×20 简单栅格地图,图 9 为 20×20 复杂栅格地图。

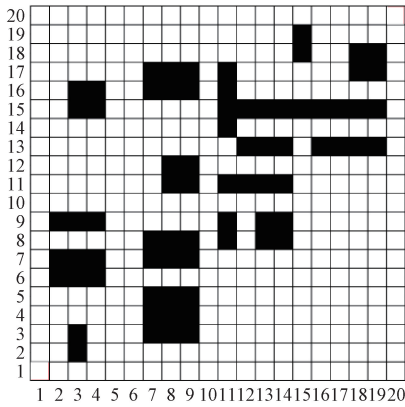


图 8 20×20 简单栅格地图

Fig. 8 20×20 Simple raster map

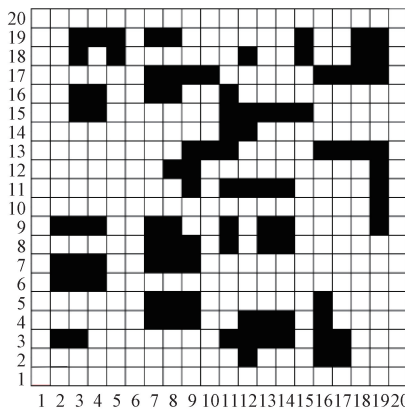


图 9 20×20 复杂栅格地图

Fig. 9 20×20 Complex raster maps

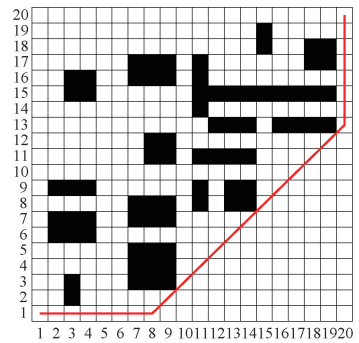
表 3 算法参数设置

Table 3 Algorithm parameter setting

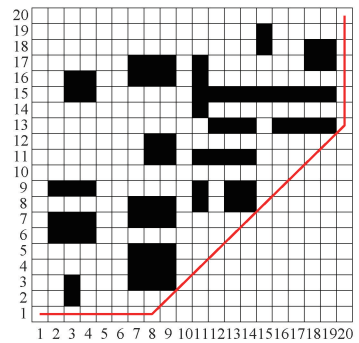
算法	RS-APSO	WCPSO	RS+PSO/	IC-VAINDIWPSO
	算法	算法	PSO 算法	算法
初始种群大小	60	60	60	60
迭代次数	100	100	100	100
认知因子 c_1	1.228~2.5	1.5~2.5	2	1.5
社会因子 c_2	1.228~2.5	1.5~2.5	2	1.5

5 种算法在 20×20 简单栅格地图中生成的路径如图 10 所示,其中图 10(a)、(b)、(c)、(d)、(e) 分别为 RS-APSO、IC-VAINDIWPSO、WCPSO、基于区域搜索的基本粒子群算法 (RS+PSO)、基本粒子群算法 (PSO) 所生成的路径。

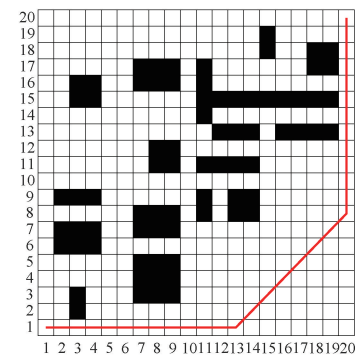
5 种算法在 20×20 复杂栅格地图中生成的路径如图



(a) 基于RS-APSO算法
(a) Based on RS-APSO algorithm

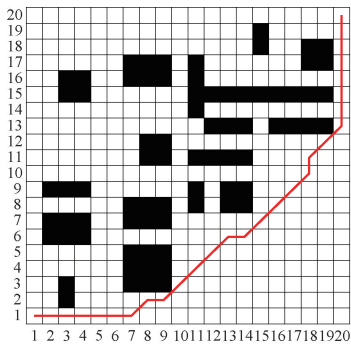


(b) 基于IC-VAINDIWPSO算法
(b) Based on IC-VAINDIWPSO algorithm

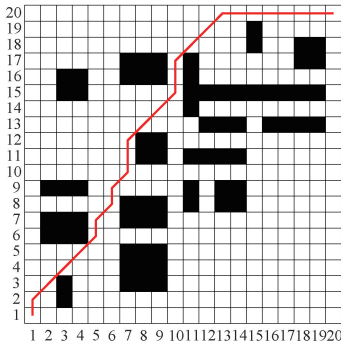


(c) 基于WCPSO算法
(c) Based on WCPSO algorithm

4 种算法的基本参数设置如表 3 所示。



(d) 基于RS+PSO算法
(d) Based on RS+PSO algorithm



(e) 基于PSO算法
(e) Based on PSO algorithm

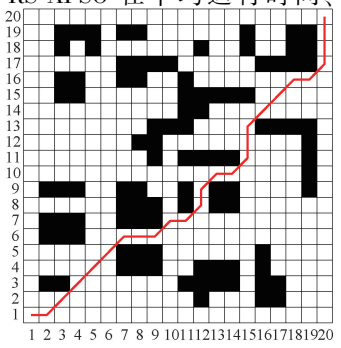
图 10 20×20 简单栅格地图下路径规划结果

Fig. 10 Result of path planning under 20×20 simple raster map

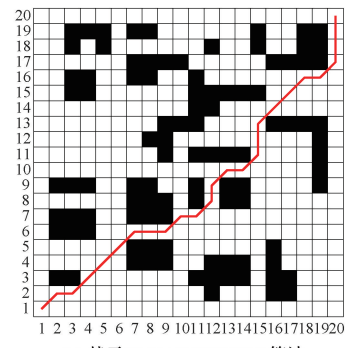
11 所示,其中图 11(a)、(b)、(c)、(d)、(e)分别为 RS-APSO、IC-VAINDIWPSO、WCPSO、基于区域搜索的基本粒子群算法(RS+PSO)、基本粒子群算法(PSO)所生成的路径。

由图 10 与 11 所示结果可明显看出 RS-APSO 所生成路径的长度与平滑度和 IC-VAINDIWPSO 基本一致,明显优于其他 3 种算法。这是因为通过引入新的加速因子 RS-APSO 可以很好的避免粒子陷入较差区域,在路径长度与平滑度上的比例分配也较为合理,在生成路径长度较短的同时也能很好的兼顾路径的平滑度。

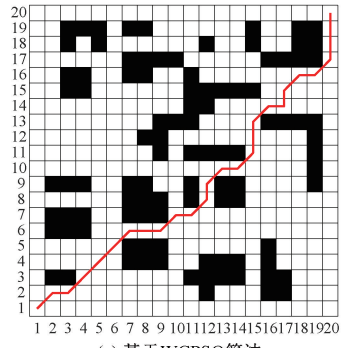
将每种算法仿真 20 次取平均值得到如表 4、5 所示数据。其中,RS-APSO 在平均运行时间、平均迭代次



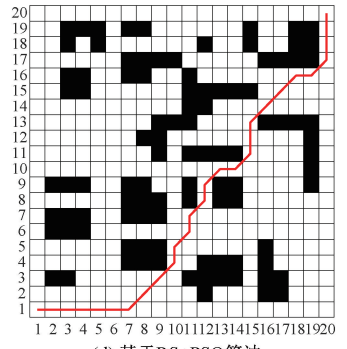
(a) 基于RS-APSO算法
(a) Based on RS-APSO algorithm



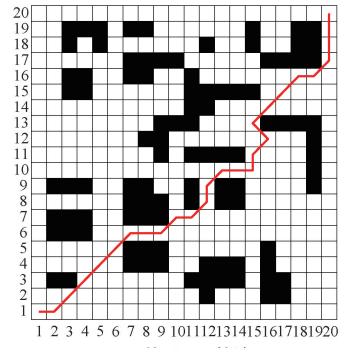
(b) 基于IC-VAINDIWPSO算法
(b) Based on IC-VAINDIWPSO algorithm



(c) 基于WCPSO算法
(c) Based on WCPSO algorithm



(d) 基于RS+PSO算法
(d) Based on RS+PSO algorithm



(e) 基于PSO算法
(e) Based on PSO algorithm

图 11 20×20 复杂栅格地图下路径规划结果

Fig. 11 Result of path planning under 20×20 complex raster map

数这两个性能指标上均优于其他 4 种算法。与 PSO 算

法相比,平均运行时间降低了 30.3%,平均迭代次数降低了 43.9%。这是由于使用区域搜索算法对地图进行了预处理,整个算法的执行速度都获得了提升;同时加速因子 c_3 的引入,不仅避免粒子陷入较差区域,也可以快速生成较优路径。

表 4 4 种算法在 20×20 简单栅格地图数据对比
Table 4 Four algorithms in 20×20 simple raster map data comparison table

算法	平均路径长度/m	平均运行时间/s	平均迭代次数
RS-APSO	37.164	27.04	31
IC-VAINDIWPSO	37.164	30.65	34
WCPSO	38.448	35.14	39
RS+PSO	37.872	30.69	37
PSO	39.936	47.52	52

表 5 4 种算法在 20×20 复杂栅格地图数据对比
Table 5 Four algorithms in 20×20 complex raster map data comparison table

算法	平均路径长度/m	平均运行时间/s	平均迭代次数
RS-APSO	36.456	28.37	30
IC-VAINDIWPSO	36.624	31.43	37
WCPSO	37.284	32.60	32
RS+PSO	37.800	38.10	36
PSO	38.820	52.74	57

5 种算法的最优个体适应度值如图 12 所示。RS-APSO 在运行时间与迭代次数上明显优于其他 4 种算法,这是因为经过区域搜索算法进行地图预处理后,无效栅格点被剔除,粒子可以直接在有效区域进行搜索。经过两种可变算子调节后的惯性权重因子 ω 可以加强认知因子 c_1 和社会因子 c_2 在前期与后期的搜索速度,算法整体的收敛速度加快。

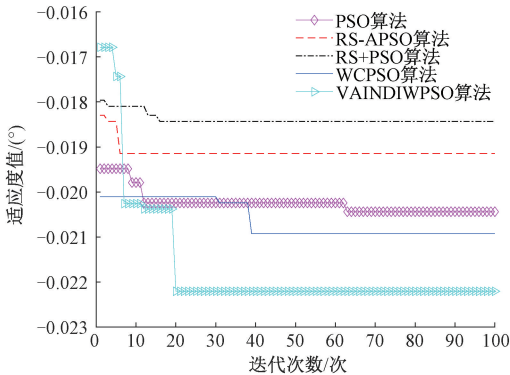


图 12 最优个体适应度对比

Fig. 12 Optimum individual fitness comparison

4.3 动态避障策略验证

在 20×20 栅格地图中,机器人起点栅格编号为 0,终

点栅格编号为 399, $U_{gl} = 1.2 \text{ m}$, $v_{rob} = 2 \text{ m/s}$, $v_{obs} = 1.2 \text{ m/s}$, $\mu = \sqrt{2}/4$ 。

当障碍物左右/上下移动时,动态避障路径如图 13、14 所示,其中虚线路径为初始静态规划路径,实线路径为实际动态避障路径。可以看出,在移动障碍物进入碰撞检测区域后,机器人提前做出判断,安全躲避了移动的障碍物。

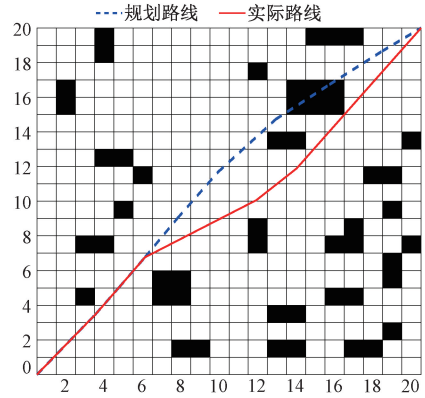


图 13 左右动态避障

Fig. 13 Left-right dynamic obstacle avoidance

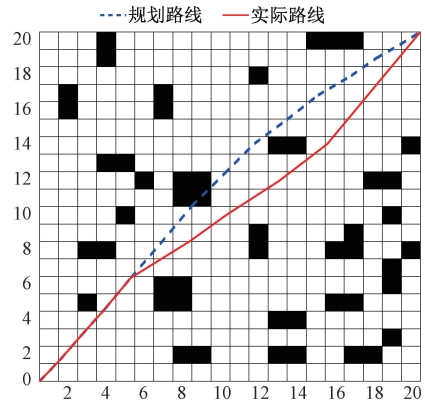


图 14 上下动态避障

Fig. 14 Up-down dynamic obstacle avoidance

当障碍物既有左右移动又有上下移动时,其运动方式为实线矩形框标注障碍物与虚线矩形框之间往复运动。动态避障路径如图 15 所示,当机器人行进到坐标点 (7,8) 时,上下移动的障碍物进入到碰撞检测区域,机器人重新选择坐标点 (9,9) 作为下一路径点进行避障;当机器人行进到坐标点 (17,15) 时,左右移动的障碍物进入到碰撞检测区域,机器人重新选择坐标点 (19,17) 作为下一路径点进行避障,采用动态避障策略后的机器人可以同时安全躲避左右移动的障碍物和上下移动的障碍物。

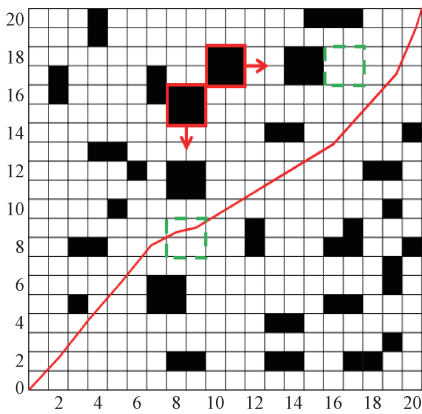


图 15 双向动态避障

Fig. 15 Bidirectional dynamic obstacle avoidance

5 结 论

针对移动机器人路径规划问题,本文提出了一种基于区域搜索的自适应粒子群算法(RS-APSO)。结合区域搜索算法对地图进行预处理,得到新的可探索区域,降低算法整体运行时间。通过对算法的自适应改进使算法性能得到进一步提升,新加速因子的引入也有效地避免粒子陷入较差区域。同时提出新的动态避障策略,使机器人可以安全规避不同方向的移动障碍物。仿真结果表明,RS-APSO 在静态环境中,优于其他粒子群优化算法,同时在动态环境中能够成功躲避移动障碍物,生成安全路径。目前 RS-APSO 主要是针对二维环境的路径规划,未来我们将在实体样机上对该算法进行测试,同时改进该算法以增强其在三维环境下的应用能力。

参考文献

- [1] FRAGAPANE G, KOSTER R, SCARBOSSA F, et al. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda[J]. *European Journal of Operational Research*, 2021, 294(2): 405-426.
- [2] PATLE B K, PANDEY A, PARHI D R K, et al. A review: On path planning strategies for navigation of mobile robot[J]. *Defence Technology*, 2019, 15(4): 582-606.
- [3] WAHAB M N A, NEFTI M S, ATYABI A. A comparative review on mobile robot path planning: Classical or meta-heuristic methods? [J]. *Annual Reviews in Control*, 2020, 50: 233-252.
- [4] 林依凡,陈彦杰,何炳蔚,等. 无碰撞检测 RRT* 的移动机器人运动规划方法[J]. *仪器仪表学报*, 2020, 41(10): 257-267.
- [5] LIN Y F, CHEN Y J, HE B W, et al. Motion planning method of mobile robot without collision detection RRT* [J]. *Chinese Journal of Scientific Instrument*, 2020, 41(10): 257-267.
- [5] MANDLOI D, ARYA R, VERMA A K. Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3D environment[J]. *International Journal of System Assurance Engineering and Management*, 2021, 12(5): 990-1000.
- [6] JIANG X, DENG Y. UAV track planning of electric tower pole inspection based on improved artificial potential field method [J]. *Journal of Applied Science and Engineering*, 2021, 24(2): 123-132.
- [7] DU C, TANG B. Novel unconventional-active-jamming recognition method for wideband radars based on visibility graphs[J]. *Sensors*, 2019, 19(10): 2344.
- [8] HOUSSEIN E H, GAD A G, HUSSAIN K, et al. Major advances in particle swarm optimization: Theory, analysis, and application [J]. *Swarm and Evolutionary Computation*, 2021, 63: 100868.
- [9] DAMOS M A, ZHU J, LI W, et al. A novel urban tourism path planning approach based on a multiobjective genetic algorithm [J]. *ISPRS International Journal of Geo-Information*, 2021, 10(8): 530.
- [10] 李志锟,黄宜庆,徐玉琼. 改进变步长蚁群算法的移动机器人路径规划[J]. *电子测量与仪器学报*, 2020, 34(8): 15-21.
- [10] LI ZH K, HUANG Y Q, XU Y Q. Path planning of mobile robot based on improved variable step ant colony algorithm [J]. *Journal of Electronic Measurement and Instrumentation*, 2020, 34(8): 15-21.
- [11] OUYANG F. Research on port logistics distribution route planning based on artificial fish swarm algorithm [J]. *Journal of Coastal Research*, 2020, 115(SI): 78-80.
- [12] SHIRI H, PARK J, BENNIS M. Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach [C]. 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6.
- [13] ZHANG X, ZHENG X, CHENG R, et al. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence [J]. *Information Sciences*, 2018, 427: 63-76.

- [14] SHAO S, PENG Y, HE C, et al. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization [J]. *ISA Transactions*, 2020, 97: 415-430.
- [15] MANDAVA R K, BONDADA S, VUNDAVILLI P R. An Optimized Path Planning for the Mobile Robot Using Potential Field Method and PSO Algorithm [M]. *Soft Computing for Problem Solving*. Springer, Singapore, 2019: 139-150.
- [16] THARWAT A, ELHOSENY M, HASSANIEN A E, et al. Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm [J]. *Cluster Computing*, 2019, 22(2): 4745-4766.
- [17] CHENG X, LI J, ZHENG C, et al. An improved PSO-GWO algorithm with chaos and adaptive inertial weight for robot path planning [J]. *Frontiers in Neurorobotics*, 2021, 15.
- [18] 贾会群, 魏仲慧, 何昕, 等. 基于改进粒子群算法的路径规划 [J]. *农业机械学报*, 2018, 49(12): 371-377.
JIA H Q, WEI ZH H, HE X, et al. Path planning based on improved particle swarm optimization algorithm [J]. *Transactions of the Chinese Society of Agricultural Machinery*, 2018, 49(12): 371-377.
- [19] SONG B, WANG Z, ZOU L. On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm [J]. *Cognitive Computation*, 2017, 9(1): 5-17.
- [20] HAO K, ZHAO J L, YU K C, et al. Path planning of mobile robots based on a multi-population migration genetic algorithm [J]. *Sensors*, 2020, 20(20): 5873.
- [21] CHU H, YI J, YANG F. Chaos particle swarm optimization enhancement algorithm for UAV safe path planning [J]. *Applied Sciences*, 2022, 12(18): 8977.

作者简介



郝琨, 2010 年于天津大学获得博士学位, 现为天津城建大学教授, 主要研究方向为水下传感器网络、计算机视觉等。

E-mail: kunhao@tcu.edu.cn

Hao Kun received her Ph. D. degree from Tianjin University in 2010. Now she is a professor at Tianjin Chengjian University. Her main research interests include underwater sensor networks, computer vision, etc.



邓晁硕, 2019 年于河南师范大学获得学士学位, 现为天津城建大学计算机与信息工程学院硕士研究生, 主要研究方向为路径规划等。

E-mail: dengchaoshuo0619@163.com

Deng Chaoshuo received his B. Sc. degree from Henan Normal University in 2019. Now he is a M. Sc. candidate at Tianjin Chengjian University. His main research interests include path planning of mobile robots, etc.



赵璐 (通信作者), 2014 年于哈尔滨工程大学获得博士学位, 现为天津城建大学讲师, 主要研究方向为机器学习、网络优化等。

E-mail: zhaolu6892@163.com

Zhao Lu (Corresponding author) received her Ph. D. degree from Harbin Engineering University in 2014. Now she is lecturer at Tianjin Chengjian University. Her main research interests include machine learning, network optimization, etc.



刘永磊, 2014 年于天津大学获得博士学位, 现为天津城建大学副教授, 主要研究方向为网络安全等。

E-mail: liuyonglei@tcu.edu.cn

Liu Yonglei received his Ph. D. degree from Tianjin University in 2014. Now he is an associate professor at Tianjin Chengjian University. His main research interests include network security, etc.