

文章编号:1671-6833(2020)04-0046-06

# 基于自适应粒子群算法的机器人路径规划

高岳林<sup>1</sup>, 武少华<sup>2</sup>

(1.北方民族大学 宁夏智能信息与大数据处理重点实验室,宁夏 银川 750021; 2.宁夏大学 数学统计学院,宁夏 银川 750021)

**摘要:**针对粒子群算法在解决机器人路径规划中存在的路径易陷入局部最优、路径搜索后期收敛速度慢以及路径不平滑的问题,提出了一种基于模拟退火的改进自适应粒子群算法,该算法结合了模拟退火算法和粒子群算法的优点,路径搜索前期路径搜索速度快,路径搜索过程中路径具有概率突跳的能力,能够有效地避免陷入局部最优路径,而且利用3次样条插值使路径平滑,路径搜索后期路径收敛精度也很高。仿真结果表明,该算法在不同障碍物模型中均能够快速找到最短的平滑路径,而且效果优于传统方法。

**关键词:**粒子群算法;模拟退火算法;机器人路径规划;三次样条插值

**中图分类号:** TP391.9 **文献标志码:** A **doi:**10.13705/j.issn.1671-6833.2020.01.004

## 0 引言

机器人技术是近年来兴起的一种现代科学技术,它的发展结合了许多高新技术学科,主要有机械电子、信息学、控制论、软件开发、人工智能等。移动机器人路径的研究是机器人学科中一个重要的研究领域,研究的问题是从工作空间中寻找一个从开始位置到目标位置的一个最优路径。本质上属于路径规划问题。智能算法是近年来提出用来解决机器人路径规划问题的算法,主要使用的方法有遗传算法、蚁群算法、粒子群算法、神经网络等等。

国内外学者对机器人路径规划也作了许多研究。Willms等<sup>[1]</sup>提出了一种高效的动态实时机器人路径规划系统,进行实时机器人路径规划。Fujimura等<sup>[2]</sup>提出了一种基于人工势场的机器人路径规划。Saska等<sup>[3]</sup>提出了一种采用Ferguson样条的粒子群优化算法,该方法得到的路径比传统方法得到的路径光滑,而且易于执行。刘广瑞等<sup>[4]</sup>利用蚁群算法来进行最优路径的搜索,并对算法进行收敛性分析,从而提高了算法的收敛效果。赵开新等<sup>[5]</sup>通过梯度计算、节点探测、路径评

估3种方式对机器人路径规划进行研究。陈志军等<sup>[6]</sup>将机器人路径规划是建立在三维空间里,建立了三维路径规划的评价指标和优化函数,提出了一种基于模糊神经网络和遗传算法的机器人路径规划。强宁等<sup>[7]</sup>采用了三次样条插值和粒子群算法来进行多机器人全局路径规划。

粒子群算法在解决机器人路径规划问题时主要存在两个缺陷:①算法迭代到后期,粒子群多样性下降,当路径陷入局部较差路径时,很难自动跳出,出现“早熟”现象。②算法的收敛性差,算法迭代后期,许多粒子适应度相差不大,导致粒子搜索进入停滞状态,路径寻优也会停滞,造成了路径的精度低。因此对算法进行改进十分必要。粒子群优化算法(PSO)和模拟退火算法(SA)是两种不同的智能优化算法,粒子群优化算法通过追随当前搜索到的最优解来寻找全局最优,模拟退火算法的机制便是以一定的概率来接受一个比当前解要差的解,从而可能跳出这个局部的最优解,达到全局的最优解。针对以上问题,笔者提出一种基于模拟退火的改进粒子群算法(APSO-SA),通过充分发挥粒子群算法和模拟退火算法的优势从而实现机器人全局最优路径规划。

**收稿日期:**2019-12-28;**修订日期:**2020-02-06

**基金项目:**国家自然科学基金资助项目(61561001);北方民族大学重大专项(2019MSP003);宁夏高等教育一流学科建设资助项目(NXYLXK2017B09)

**作者简介:**高岳林(1963—),男,陕西榆林人,北方民族大学教授,博士生导师,主要研究方向为智能计算与智能信息处理,E-mail:gaoyuelin@263.net。

## 1 标准粒子群算法

粒子群算法(PSO)是一种启发式随机算法。鸟群中每一个鸟相当于粒子群算法中的一个粒子。每个粒子好比寻找食物的鸟都有速度和位置,通过自身和社会两种学习方式,粒子在搜索空间中运动,从而得到全局最优解。假设种群有  $N$  个粒子在  $D$  维空间搜索,标准粒子群算法更新公式为:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(p_{ij}^t - x_{ij}^t) + c_2r_2(p_{gj}^t - x_{ij}^t); \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}, \quad (2)$$

式中:  $t$  为迭代次数;  $v_{ij}$  表示第  $i$  个粒子  $j$  维的速度大小;  $x_{ij}$  表示第  $i$  个粒子  $j$  维的位置大小;  $w$  表示惯性权重;  $c_1$ 、 $c_2$  表示学习因子;  $r_1$ 、 $r_2$  表示随机数。粒子群算法虽然简单,但是算法迭代后期,由于粒子多样性差导致粒子容易陷入局部最优解,所以要粒子群算法进行改进。

## 2 粒子群算法的改进

### 2.1 带压缩因子的粒子群算法

为了有效地控制粒子飞行速度使算法全局搜索和局部搜索平衡,Clerc 等<sup>[8]</sup>提出了带压缩因子的 PSO 算法,其速度更新公式如下:

$$v_{ij}^{t+1} = \varphi \{v_{ij}^t + c_1r_1(p_{ij}^t - x_{ij}^t) + c_2r_2(p_{gj}^t - x_{ij}^t)\}, \quad (3)$$

其中搜索因子:

$$\varphi = \frac{2}{|2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)}|}。$$

带压缩因子的 PSO 算法相当于线性递减权重的 PSO 算法,该算法的优势在于去掉了惯性权重  $w$ , 减少了算法参数,只有两个学习因子参数,所以只需对两个学习因子进行自适应调节。其调节公式如下:

当  $f(x_i) \leq f_{\text{avg}}$ , 粒子  $x_i$  需要进行局部搜索,学习因子取:

$$\begin{cases} c_1 = c_{\min} + \frac{(c_{\max} - c_{\min})(f(x_i) - f_{\min})}{f_{\text{avg}} - f_{\min}}; \\ c_2 = c_{\max} \end{cases}; \quad (4)$$

当  $f(x_i) > f_{\text{avg}}$ , 粒子  $x_i$  要进行全局搜索,所以学习因子取:

$$\begin{cases} c_1 = c_{\max}; \\ c_2 = c_{\min} + \frac{(c_{\max} - c_{\min})(f(x_i) - f_{\min})}{f_{\text{avg}} - f_{\min}}, \end{cases} \quad (5)$$

式中:  $c_{\min} = 1.5$ ;  $c_{\max} = 2.5$ ;  $f(x_i)$  表示粒子  $x_i$  的适应度大小;  $f_{\text{avg}}$  表示所有粒子的平均适应度。

### 2.2 模拟退火算子

模拟退火算法是由 Metropolis 等于 1953 年提出的,该算法在搜索过程中具有概率突跳的能力,能够在一定程度上避免搜索过程中陷入局部最优解,由于粒子群算法对  $p_g$  全局最优粒子依赖性很强,所以为了避免粒子搜索过程陷入  $p_g$ ,所以对  $p_g$  引入模拟退火操作,其中 Metropolis 准则如下:

$$TF(p_i) = \begin{cases} 1, & f(p_i) < f(p_g); \\ \frac{e^{-(f(p_i) - f(p_g))}}{\sum_{i=1}^N e^{-(f(p_i) - f(p_g))}}, & f(p_i) \geq f(p_g), \end{cases} \quad (6)$$

式中:  $f(p_i)$  表示  $p_i$  粒子的适应度;  $f(p_g)$  表示当前粒子群算法迭代的全局最优适应度。采用轮盘赌策略从  $p_i$  中确定全局最优的某个个体代替  $p_g$ , 初温和退温公式为:

$$\begin{cases} t^{k+1} = \lambda t^k; \\ t^0 = f(p_g) / \ln 5. \end{cases} \quad (7)$$

以上为 APSO-SA 算法的基本思想,标准粒子群算法的时间复杂度为  $O(M, D, N)$ , 其中  $M$  为最大迭代次数,  $D$  为粒子维数,  $N$  为粒子个数。由于该算法是建立在压缩因子的粒子群算法的基础上,压缩因子的粒子群算法时间复杂度为  $O(M, D, N)$ , 学习因子的自适应调节策略和模拟退火操作的时间复杂度均为  $O(N)$ , 所以 APSO-SA 算法的时间复杂度为  $O(M, D, N)$ 。

## 3 机器人路径规划模型建立

### 3.1 三次样条插值编码设计

两点法对机器人路径进行规划<sup>[9]</sup>会造成路径平滑性差,机器人运动不平稳,频繁转向会造成能源的极大浪费,本文机器人路径规划是建立在二维平面上,提出的 APSO-SA 算法在二维平面进行三次样条插值编码,根据文献[3]提出的 Ferguson 三次样条插值对路径进行编码优化,其原理如下。

假设有以下节点  $(x_i, y_i)$ , 其中  $x_i < x_j, \forall i < j$ , 三次样条插值曲线  $S(x)$  满足以下条件:

(1) 每个分段区间  $[x_i, x_{i+1}]$ ,  $S(x) = S_i(x)$  为三次多项式。

(2) 每一个端点满足  $S(x_i) = y_i, i \in N \cap i \leq n$ 。

(3)  $S(x)$  的 1、2 阶导数为  $S'(x)$ 、 $S''(x)$ , 而且在  $[a, b]$  区间内连续, 即  $S_i(x)$  可以写成:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3;$$

式中:  $i \in N \cap i < n$ 。

机器人在搜索空间搜索最优路径可以看作在

三次样条空间中搜索最优样条,APSO-SA 算法中的每一组粒子表示为一组路径的节点,每一组路径节点个数便为三次样条曲线的个数,由于三次样条插值的性质,每一组路径节点的个数体现了路径可以转向的最大次数。粒子编码的路径节点个数会随着工作环境的复杂而增加,实际生活中机器人转向 3~5 次便可以绕过所有的障碍物,本文的障碍物模型设置转向节点为 3。

3.2 路径初始化

路径初始化决定着算法的好坏,PSO 算法机器人路径规划均是在机器人工作空间随机取粒子,但是导致算法收敛性差,于是对粒子的初始化进行改进,提出基于高斯分布的随机初始化,初始化公式如下:

$$x = (x_s + x_t) / 2 + r_3;$$

$$y = U(y_{min}, y_{max}, 1, n),$$

式中:  $x_s$ 、 $x_t$  分别为开始位置和目标位置的横坐标;  $r_3$  为高斯分布产生的随机数;  $n$  表示产生粒子的个数;  $y_{max}$ 、 $y_{min}$  分别为粒子初始化纵坐标的上下界;  $U$  表示在给定上下界中均匀产生  $n$  个粒子纵坐标的值。

3.3 障碍物处理

障碍物对机器人路径规划有很大的影响,若没有障碍物,机器人在二维空间中从开始位置到目标位置运动,直线运动最短,无须进行路径规划,然而现实机器人工作时,障碍物是客观存在的,笔者研究的是静态工作环境下的机器人路径规划,由于实际工作环境中障碍物形状千差万别,导致建模十分困难,为了方便统一研究,笔者对障碍物进行预处理,在二维空间中,障碍物均进行膨胀处理,即用原障碍物图形的外接圆表示该障碍物,具体见图 1。

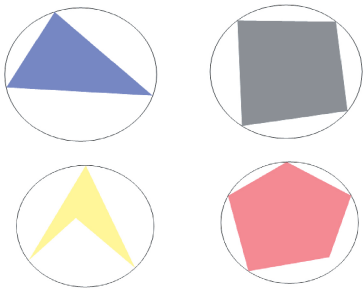


图 1 不同形状障碍物膨胀处理

Figure 1 Different obstacle with different shapes expansion

3.4 适应度函数

适应度函数的值可以看作路径的长度,粒子通过三次样条编码,不断寻找最优适应度,而

有些路径与障碍物相交会成为非法路径,如果直接去掉这些路径,那么存在的可行路径就会变得非常少,路径的多样性就会减弱,所以构造带有罚函数的适应度函数,通过附加较大的惩罚值,使路径在优化过程中自动淘汰非法路径。文献[8]中使用了一种带有罚函数的适应度函数,笔者对其做了一些简单的修改,得出适应度函数如下:

$$z = L(1 + beta \cdot V),$$

式中:  $L$  为机器人工作运行路径的长度;  $beta$  为惩罚值,一般设置为 100;  $V$  为路径的非法度,当  $V = 0$  时,  $z = L$ , 此时适应度的大小就为路径的大小。

4 算法流程

APSO-SA 算法机器人路径规划与粒子群算法机器人路径规划流程相似,具体算法流程如下:

步骤 1 创建障碍物模型,初始化机器人开始位置以及机器人目标位置,初始化机器人工作空间。

步骤 2 初始化 APSO-SA 算法的参数,包括粒子个数、学习因子、最大迭代次数、退火系数、路径节点个数。

步骤 3 按照式(8)、(9)初始化每组粒子的位置,以及对粒子进行三次样条编码。

步骤 4 按照式(4)、(5)自适应处理学习因子,同时按照式(3)更新粒子速度,按照式(2)更新粒子的位置。

步骤 5 按照式(10)计算每组粒子当前的适应度,并根据式(6)和式(7)对当前适应度最好的粒子进行退火处理。

步骤 6 判断算法是否达到最大函数评价次数,若满足,则停止搜索,输出结果,否则返回步骤 4 继续迭代。

5 仿真模拟

为了验证 APSO-SA 算法在机器人路径寻优中的优越性,笔者设计了以下实验,在 3 种障碍物模型中对 APSO-SA 算法与 PSO 算法进行比较测试。其中 3 种障碍物模型的最短路径长度通过几何计算得到,保留小数点后 2 位。见表 1。算法比较采用固定粒子个数和最大函数迭代次数,试验在 windows 7 系统 MATLAB 2012a 中完成。电脑配置为 Intel(R) Core(TM) i5-3317U CPU@ 1.70 GHz。

5.1 参数设置

笔者建立的 3 个障碍物模型参数如表 1~3 所示。

表 1 3 个模型基本参数

模型	开始位置坐标	目标位置坐标	障碍物个数	最优路径长度/m
单障碍物模型	(0,2)	(0,6)	1	7.39
多个均匀障碍物模型	(0,2)	(0,6)	12	6.13
多个不均匀障碍物模型	(0,0)	(4,6)	3	7.45

表 2 3 个模型圆心及其半径

模型	圆心及其半径
模型 1	(2,3); $r=2$
模型 2	(0,2),(0,4),(1,1),(1,3),(1,5),(2,2),(2,4),(3,1),(3,3),(3,5),(4,2),(4,4); $r=0.5$
模型 3	(1.5,4.5); $r1=1.5$ , (4,3); $r2=1$ , (1.2,1.5); $r3=0.8$

表 3 PSO 和 APSO-SA 算法参数

算法	粒子数	学习因子	函数评价次数
PSO	100	$c_1=2.05,c_2=2.05$	10 100
APSO-SA	100	$c_{min}=1.5,c_{max}=2.5$	10 100

5.2 数值实验

5.2.1 单障碍物模型

两种算法独立运行 30 次,数值结果用 SPSS 进行统计分析,结果如表 4、5 所示。

表 4 单障碍物模型数值结果比较

算法	最小路径长度/m	平均路径长度/m	标准差	运行时间/s
PSO	7.43	7.46	0.20	11.46
APSO-SA	7.42	7.41	0.03	11.50

表 5 单障碍物模型单因素方差分析结果比较

类别	平方和	df	均方	F	显著性
组间	0.03	1	0.03	132.32	小于 0.01
组内	0.03	58	小于 0.01	2	
总计	0.04	59			

图 2 为单障碍物模型两种算法比较结果图,从图 2 可以看出,PSO 算法和 APSO-SA 算法都能够在该模型中规划出一条无碰路径。PSO 算法搜

索出的最优路径长度为 7.47 m,而 APSO-SA 算法得出的最优路径长度为 7.41 m,同时 APSO-SA 路径搜索范围广,路径搜索前期变化比较大,10 代以后趋于平稳,表 4、5 也表明 APSO-SA 算法的路径寻优能力优于 PSO 算法,通过方差同质性检验后进行单因素方差分析,结果表明,两种算法具有显著性差异。

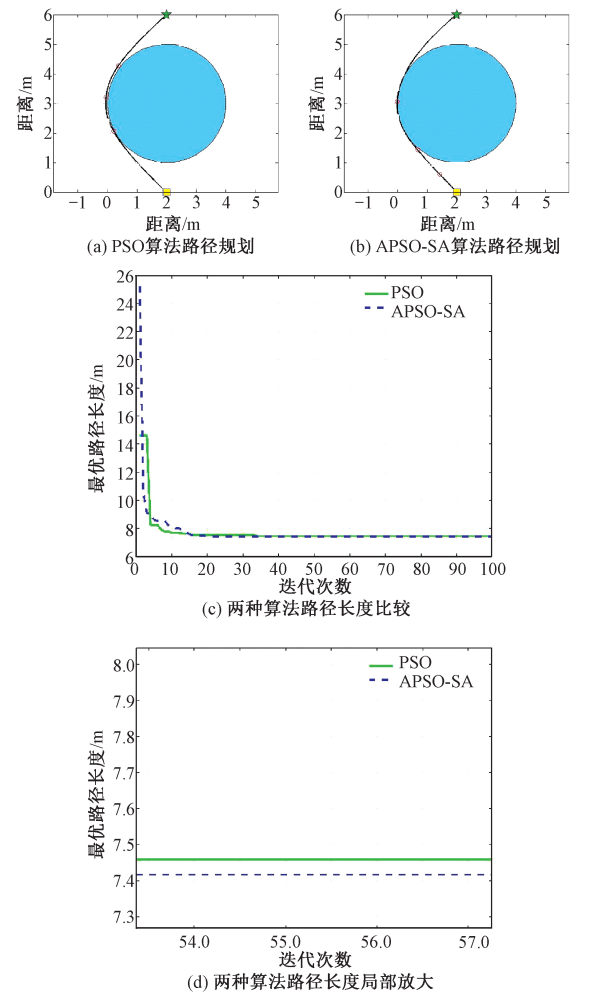


图 2 单障碍物模型两种算法比较结果

Figure 2 The results of the two algorithms of single obstacle model

5.2.2 多个均匀障碍物模型

图 3 分别给出了多个均匀障碍物模型两种算法得出的最优路径和最优路径长度比较。从实验结果可以看出,APSO-SA 算法的路径平滑度明显优于 PSO 算法。下面两种算法各自独立运行 30 次,运行结果及统计分析记录如表 6、7 所示。

由于该模型存在许多局部最优路径,所以获得的路径为局部最优路径不可避免。从表 6 可以看出,APSO-SA 有很好地跳出局部最优路径的性能,得到的路径为局部最优路径的概率仅为 20%。从时间上来看两种算法相差不大,仅比传统的

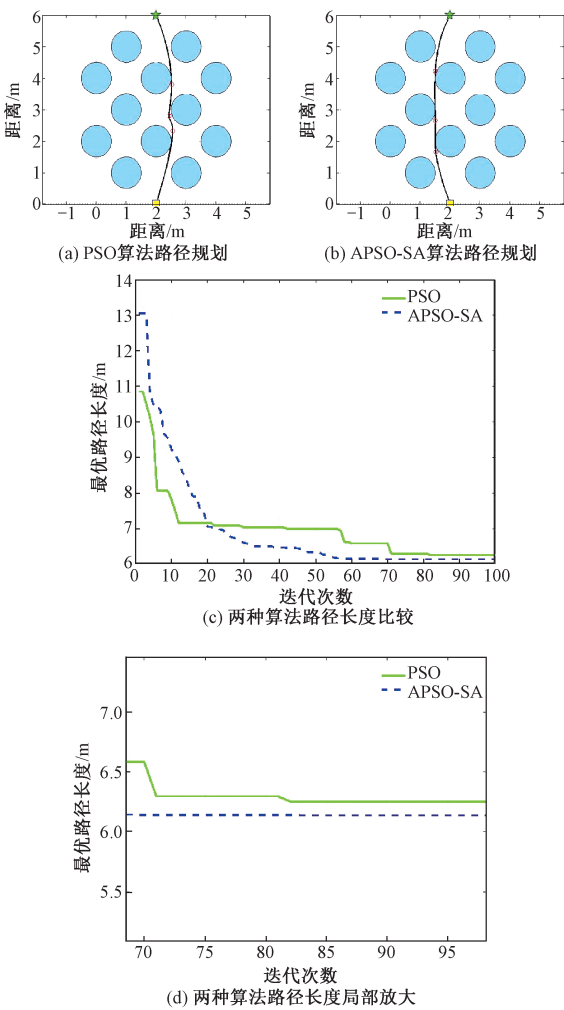


图 3 多个均匀障碍物模型两种算法比较结果

Figure 3 The results of the two algorithms of multiple uniform obstacle models

表 6 多个均匀障碍物模型数值结果比较

Table 6 Comparison of numerical results of multiple uniform obstacle models

算法	最小路径 长度/m	平均路径 长度/m	标准差	运行 时间/s	陷入局 部最优 次数
PSO	6.15	6.59	0.50	22.43	12
APSO-SA	6.14	6.32	0.20	22.47	6

表 7 多个均匀障碍物模型单因素方差分析结果比较

Table 7 One-way ANOVA results of multiple uniform obstacle models

类别	平方和	df	均方	F	显著性
组间	1.090	1	1.090	7.598	0.01
组内	8.297	58	0.143		
总计	9.384	59			

PSO 算法高了 0.04 s, 总体路径平均长度却减少了 0.27 m。这充分体现了 APSO-SA 算法多样化

策略的优势。单因素方差分析也可以得到两种算法具有显著性差异。

5.2.3 多个不均匀障碍物模型

从图 4 和表 8 可以看出, 在多个不均匀障碍物模型中, APSO-SA 算法的路径寻优能力也优于 PSO, 这充分说明了在复杂的工作环境中 PSO 算法中的粒子在搜索过程中极易从可行域飞到不可行域, 导致种群中大量的粒子进行约束处理后不能按最优方向去搜索最优值, 最终影响搜索效果。

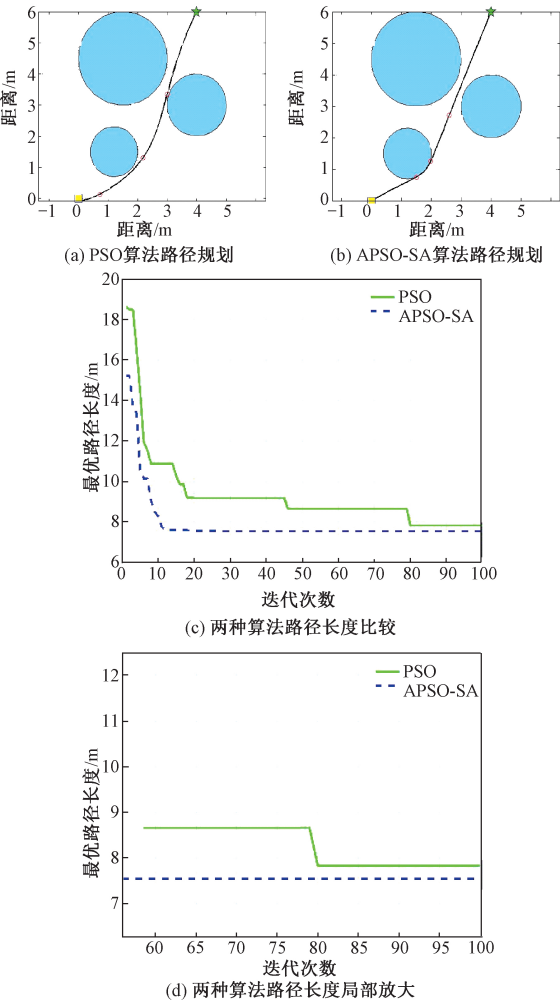


图 4 多个不均匀障碍物模型两种算法比较结果

Figure 4 The results of the two algorithms of multiple inhomogeneous obstacle models

表 8 多个不均匀障碍物模型数值结果比较

Table 8 Comparison of numerical results of multiple inhomogeneous obstacle models

算法	最小路径 长度/m	平均路径 长度/m	标准差	运行 时间/s	陷入局 部最优 次数
PSO	7.59	8.16	0.73	20.48	3
APSO-SA	7.55	7.60	0.30	20.56	1

表 9 多个不均匀障碍物模型单因素方差分析结果比较

Table 9 One-way ANOVA results of multiple inhomogeneous obstacle models

类别	平方和	df	均方	F	显著性
组之间	4.72	1	4.715	15.25	小于 0.01
组内	17.93	58	0.309		
总计	22.65	59			

6 结论

笔者提出了一种基于模拟退火与自适应粒子群混合的算法,用来解决机器人路径规划问题。其中采用高斯分布和三次样条插值法来初始化并编码粒子,采用退火策略和罚函数策略进一步提高了算法的运行效率,使机器人路径更加合理化。通过对适应度函数的调整,APSO-SA 算法也可以解决不同的应用问题。将来有可能应用于动态实时路径规划问题中。

参考文献:

[1] WILLMS A R, YANG S X. An efficient dynamic system for real-time robot-path planning [J]. IEEE transactions on systems man and cybernetics, 2006, 36(4):755-766.

[2] FUJIMURA K, SAMET H. A hierarchical strategy for path planning among moving obstacles (mobile robot)

[J]. IEEE transactions on robotics and automation, 1989, 5(1):61-69.

[3] SASKA M, MACAS M, PREUCIL L, et al. Robot path planning using particle swarm optimization of Ferguson splines [C] //IEEE Conference on Emerging Technologies and Factory Automation. Prague; IEEE, 2006;833-839.

[4] 刘广瑞,刘军,孔令云. 移动机器人路径规划蚁群算法及其收敛性分析[J]. 郑州大学学报(工学版), 2012, 33(2):24-27.

[5] 赵开新,王东署. 未知环境中自主机器人的路径规划研究[J]. 郑州大学学报(工学版), 2013, 34(5):74-79.

[6] 陈志军,曾蒸. 基于模糊神经网络和遗传算法的机器人三维路径规划[J]. 重庆师范大学学报(自然科学版), 2018, 35(1):93-99.

[7] 强宁,高洁,康凤举. 基于 PSO 和三次样条插值的多机器人全局路径规划[J]. 系统仿真学报, 2017, 29(7):1397-1404.

[8] CLERC M, KENNEDY J. The particle swarm-explosion, stability and convergence in a multi-dimensional complex space[J]. IEEE transactions on evolutionary computation, 2002, 6(1):58-73.

[9] 宋莉,李彩虹,朱宝艳,等. 基于两点法的移动机器人局部路径规划算法[J]. 山东理工大学学报(自然科学版), 2018, 32(1):10-14.

Robot Path Planning Based on Adaptive Particle Swarm Optimization

GAO Yuelin<sup>1</sup>, WU Shaohua<sup>2</sup>

(1.Ningxia Key Laboratory of Intelligent Information and Big Data Processing, North Minzu University, Yinchuan 750021, China; 2.School of Mathematics and Statistics, Ningxia University, Yinchuan 750021, China)

**Abstract:** Particle swarm optimization algorithm was easy fall into local optimum, the convergence speed was slow in the late path search, and the path was not smooth in the robot path planning. An improve simulated annealing adaptive particle swarm optimization algorithm was proposed. The algorithm combined the advantages of simulated annealing and particle swarm optimization. In the early stage of the algorithm route search was fast, and the algorithm had the ability of sudden jump in the path search process, which could effectively avoid falling into the local optimal path. Using cubic spline interpolation smooth the path, and the convergence precision of the late search path was high.The simulation results showed that the algorithm could quickly find the shortest smooth path in different obstacle models, and the path effect was better than the traditional method.

**Key words:** particle swarm algorithm; simulated annealing algorithm; robot path planning; cubic spline interpolation