

DOI:10.3969/j.issn.1001-4551.2019.04.020

基于模糊推理技术 PSO 算法的 机器人路径规划研究

刘彩霞

(包头职业技术学院 数控技术系, 内蒙古 包头 014030)

摘要:针对移动机器人路径规划质量不高的问题,对环境建模、适应度函数建立、算法选择等方面进行了研究归纳,提出了一种基于模糊推理技术 PSO 算法的路径规划方法。首先,对障碍物进行了扩展处理,通过坐标系转换建立了简化的环境模型;其次,在分析传统 PSO 算法采用定值的惯性因子 ω 和学习因子 c_1 、 c_2 取值对算法性能影响的基础上,提出了改进的 PSO 算法,采用模糊推理技术自适应地动态调整 c_1 、 c_2 ;最后,将改进 PSO 算法应用于提高移动机器人路径规划中。研究结果表明:相对于传统 PSO 与 APSO 算法路径规划,改进算法在复杂环境下路径长度、平滑度、运行时间方面分别最少提高了 17%、14%、7%,验证了算法在路径规划方面的可行性与高效性。

关键词:移动机器人;粒子群算法;模糊推理;路径规划

中图分类号:TP242

文献标志码:A

文章编号:1001-4551(2019)04-0445-06

Robot path planning based on PSO algorithm with fuzzy inference technology

LIU Cai-xia

(Baotou Vocational & Technical College, Computer Numerical Control
Technology Department, Baotou 014030, China)

Abstract: Aiming at the problem of poor quality of path planning for mobile robots, a path planning method of PSO algorithm based on fuzzy reasoning technology was proposed, which focuses on environment modeling, fitness function establishment and algorithm selection. Firstly, the obstacles were extended and the simplified environment model was established by coordinate transformation. Secondly, based on the analysis of the influence of the fixed inertia factor ω and the learning factor c_1 and c_2 on the performance of the traditional PSO algorithm, an improved PSO algorithm was proposed, c_1 and c_2 were adaptively and dynamically adjusted by using the fuzzy reasoning technology. Finally, the improved PSO algorithm was applied to improve the path planning of mobile robot. The results indicate that, compared with the traditional PSO and APSO algorithm, the improved algorithm improves the path length, smoothness and running time at least by 17%, 14% and 7% respectively in complex environment, which verifies the feasibility and efficiency of the algorithm in path planning.

Key words: mobile robot; particle swarm algorithm; fuzzy Inference; path planning

0 引 言

机器人可以将人类从高风险、高体力劳动的工作中解放出来^[1]。而路径规划问题是机器人研究的热门问题之一。

根据其环境建模方式不同,机器人路径规划方法可分为:可视图法^[2]、栅格法^[3]、自由空间法^[4]、voronoi 法^[5]等;按照路径搜索地方式可分为:A*算法^[6]、Dijkstra 算法^[7]、Floyd 算法^[8]等。目前常用的移动机器人路径规划算法主要有粒子群(particle swarm

optimization, PSO)算法、神经网络^[9]、遗传算法^[10]、蚁群算法^[11]、模糊控制^[12]等。相较于其他算法,PSO 算法具有收敛速度快、参数设置简单等优点,近年来在路径规划领域更是得到广泛的关注。但随着算法研究的深入,PSO 算法逐步暴露出局部探索能力差、规则制定不够完善等缺点。当前,国内外学者主要围绕着参数自适应、新衍生规则制定、与其他算法融合等方面进行了改进,算法性能有所提升但并不完美^[13-15]。

为此,本文提出基于模糊推理技术的 PSO 算法来解决机器人路径规划问题。

收稿日期:2018-10-16

作者简介:刘彩霞(1977-),女,内蒙古鄂尔多斯人,硕士,副教授,主要从事数控技术、机电一体化技术方面的研究。E-mail:404105253@qq.com

1 问题描述和适应度函数建立

1.1 问题描述与环境建模

本文研究的是静态环境下机器人的全局路径规划问题,因此将机器人视为质点,不考虑高度信息,设其在二维空间中运动。笔者用多边形描述障碍物轮廓及环境边界,并对各种障碍物的边界进行扩展,每条边界向外扩展了半个机器人的宽度。

距离判断示意图如图 1 所示。

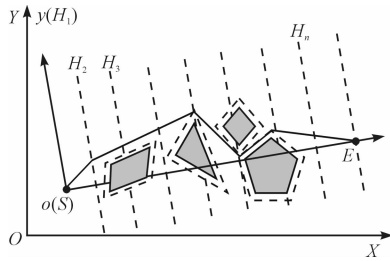


图 1 距离判断示意图

在全局环境场景用笛卡尔坐标系 XOY 表示,虚线表示经边界扩展后的障碍物边缘信息,起始点为 $S(X_s, Y_s)$,目标点为 $E(X_e, Y_e)$ 。为了简化计算,本文采用坐标变换,连接 S 和 E 的直线作为 x 轴,经过起始点 S 并垂直于 x 轴的直线为 y 轴,得到新环境坐标系 xoy ,则任一点 $P(X, Y)$ 在 XOY 下的坐标可通过坐标变换,求得其在坐标系 xoy 下的对应坐标,即:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} X \\ Y \end{bmatrix} - \begin{bmatrix} x_s \\ y_s \end{bmatrix} \right) \quad (1)$$

$$\theta = \arcsin \frac{y_e - y_s}{\sqrt{(x_e - x_s)^2 + (y_e - y_s)^2}} \quad (2)$$

式中: θ —初始 X 轴与 SE 之间的夹角; (x, y) — (X, Y) 在新坐标系 xoy 中的坐标。

使用若干条直线簇 $H_1, H_3, \dots, H_n$ 将线段 SE 垂直等分(其中 H_1 和 H_2 分别经过 S, E),每条直线簇上取一点作为一个路径节点,于是这些直线簇上的纵坐标(在转换后的坐标系下) $y = (y_1, y_2, \dots, y_{n-1}, y_n)$ (其中 $y_1 = y_n = 0$) 这些中间节点便构成一条从起点到终点的路径,这样就将原来空间中的二维问题转换为简便的一维问题。

规划路径与障碍物的碰撞检测,可通过各路径段

$$d_{ij} = \frac{|(y_{i+1} - y_i) \cdot x_{oj} + (x_{i+1} - x_i) \cdot y_{oj} + (y_i \cdot x_{i+1} - y_{i+1} \cdot x_i)|}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}} \quad (7)$$

则路径的安全性为:

$$f_3 = \sum_{i=1}^{n-1} \sum_{j=1}^m d_{ij} \quad (8)$$

式中: m, n —障碍物顶点个数与路径的节点数。

是否与障碍物相交进行判断。对于多边形障碍物,判断路径段是否与障碍物相交,即是判断路径段是否与多边形的各个边相交。假设某一路径段为 $p_i p_{i+1}$,多边形某条边为 $o_j o_{j+1}$,可使用向量的混合积来判断两条线段是否相交。若某一路径段为 $p_i p_{i+1}$ 与某多边形某条边为 $o_j o_{j+1}$ 相交,则路径段 $p_i p_{i+1}$ 不通;若不相交,则 $p_i p_{i+1}$ 为可行路径段。

1.2 适应度函数的建立

设任意一条路径中有 n 个结点(包括起始点和终点),假设每条路径的形式为:

$$L = (p_1, p_2, \dots, p_i, \dots, p_n), (1 \leq i \leq n, n \geq 2) \quad (3)$$

式中: $p_1 = s$ —起点; $p_n = e$ —终点; p_i —中间节点, $p_i = (x_i, y_i)$ 。

则实现的 3 个目标定义如下:

(1) 路径总长度评价函数 f_1 的建立。总的路径长度是由相邻结点间的路径长度求和得来的,那么相邻结点间的路径长度如下:

$$d_i = \begin{cases} |p_i p_{i+1}|, & p_i p_{i+1} \text{ 可行} \\ +\infty, & p_i p_{i+1} \text{ 不可行} \end{cases} \quad (3)$$

可用向量的混合积来判断两条线段是否相交,以此确定路径段是否可行。设某一路径段为 $p_i p_{i+1}$ 与多边形任意边为 $o_j o_{j+1}$ 相交,则路径段 $p_i p_{i+1}$ 不通;若不相交,则 $p_i p_{i+1}$ 为可行路径段。则总的路径长度为:

$$f_1 = \sum_{i=1}^{n-1} d_i \quad (4)$$

(2) 路径平滑度评价函数 f_2 的建立。若相邻的两条路径段之间的夹角越小,则表示路径越平滑,机器人转弯的压力也就越小。本文使用路径段之间夹角的余弦来之和描述路径的平滑度,即:

$$f_2 = \sum_{i=2}^{n-1} \cos(i) \quad (5)$$

其中,路径段 $p_{i-1} p_i$ 与路径段 $p_i p_{i+1}$ 之间的夹角为:

$$\cos(i) = \frac{|p_{i-1} p_i|^2 + |p_i p_{i+1}|^2 - |p_{i-1} p_{i+1}|^2}{2 |p_{i-1} p_i| \cdot |p_i p_{i+1}|} \quad (6)$$

(3) 路径的安全性 f_3 。为了描述路径的安全性,要计算出每个障碍物的各个顶点到整条路径中每一段路径的距离。障碍物顶点 $o_j(x_{oj}, y_{oj})$ 到直线 $p_i p_{i+1}$ 的距离 d_{ij} 为:

本文采用权重系数法来求解多目标路径规划问题,则路径 P 对应的总目标函数如下:

$$f = \omega_1 f_1 + \omega_2 f_2 + \omega_3 f_3 \quad (9)$$

式中: $\omega_1, \omega_2, \omega_3$ —路径长度、平滑度、安全度的权值系数。

2 粒子群算法及其改进

2.1 基本粒子群算法

PSO 算法是由 KENNEDY 和 EBERHART^[15] 于 1995 年提出。PSO 算法速度和位置更新规则如下:

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r[pbest_{ij}(t) - x_{ij}(t)] + c_2 r[gbest_{ij}(t) - x_{ij}(t)] \quad (10)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (11)$$

式中: i — 初始种群中粒子数量, $i = 1, 2, \dots, NP$; j — 粒子维度 D 的第 j 维; t — 当前算法迭代次数; $rand$ — 介于 $(0, 1)$ 之间的随机数; $pbest_{ij}(t)$ — t 代时粒子 i 的个体最优解; $gbest_{ij}(t)$ — t 代时整个粒子群最优解; $v_{ij}(t)$ — t 代时粒子 i 的第 j 维速度; $x_{ij}(t)$ — t 代时粒子 i 的第 j 维位置; ω — 惯性因子, 取值非负; c_1, c_2 — 学习因子, 取值为 $[0.5, 2.5]$ 之间。

通过分析可看出, ω, c_1, c_2 参数的取值对算法的性能有着较大的影响。而在传统 PSO 算法通常取定值, 导致算法全局搜索能力和局部搜索能力不能够得到很好均衡, 当种群多样性降低时往往易陷入局部最优出现“早熟”现象。对此, 本文通过对 3 个参数采用自适应动态调整机制来提高算法性能。对于惯性因子 ω 采用比较经典的线性递减权值策略, 即:

$$\omega(t) = \frac{(\omega_{\max} - \omega_{\min})(t_m - t)}{t_m} + \omega_{\min} \quad (12)$$

式中: $\omega(t)$ — t 代时惯性因子; $\omega_{\max}, \omega_{\min}$ — 惯性因子最大最小值; t_m — 算法最大迭代次数。

2.2 基于模糊推理技术 PSO 算法

本文采用模糊推理技术自适应调整 PSO 算法 c_1, c_2 参数, 需要考虑算法运行过程中哪些变量影响 c_1, c_2 的取值。

因此, 本文通过下式定义了迭代因子 δ , 即:

$$\delta(t) = \frac{t}{t_m} \times 100\% \quad (13)$$

通过下式定义了种群多样性因子 σ , 即:

$$\sigma(t) = \frac{1}{NP} \sum_{i=1}^{NP} \sqrt{\sum_{j=1}^D (x_{ij}(t) - \bar{x}_j(t))^2} \quad (14)$$

通过下式定义了群体适应度差异因子 $Error$, 即:

$$Error = \frac{1}{NP} \sum_{i=1}^{NP} (Fitness(x_i) - \text{Min}F) \quad (15)$$

由此, 构建的以迭代因子 δ 、种群多样性因子 σ 、群体适应度差异因子 $Error$ 为输入变量, c_1 和 c_2 为输出变量的模糊推理系统, 如图 2 所示。

对于模糊系统的输入, 迭代因子 δ 本身有一个可能取值范围。而对于种群多样性因子 σ 、群体适应度差异因子 $Error$, 需要对 σ 进行归一化操作, 即:

$$\sigma' = \frac{\sigma - \text{Min}\sigma}{\text{Max}\sigma - \text{Min}\sigma} \quad (16)$$

同理, 对 $Error$ 进行归一化操作:

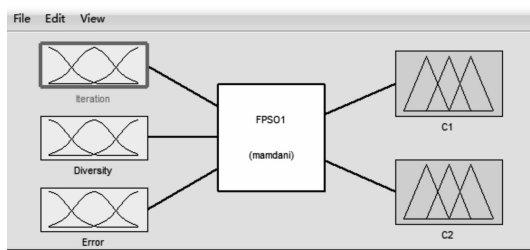


图2 自适应调整 c_1, c_2 的模糊推理系统

$$Error' = \begin{cases} 1, & \text{if } \text{Min}F = \text{Max}F \\ \frac{Error - \text{Min}F}{\text{Max}F - \text{Min}F}, & \text{otherwise} \end{cases} \quad (17)$$

式中: σ' — 种群多样性 σ 经过归一化操作后映射到 $[0 - 1]$ 之间的值; $\text{Max}\sigma, \text{Min}\sigma$ — σ 值的最大、最小值; $Error'$ — 误差 $Error$ 经过归一化操作后映射到 $[0 - 1]$ 之间的值; $\text{Max}F, \text{Min}F$ — 种群中粒子适应度最大、最小值。

由式 (17) 可以看出: 当最小适应度等于最大适应度时, 误差将 1, 此时粒子很接近; 第 2 个条件是关于不同适应度的情况。

另外, 根据模糊控制技术的定义, 需要对输入输出变量模糊化。对输入变量迭代因子 δ 的模糊语言以及隶属度函数的设计如图 3 所示 ($\sigma, Error$ 的模糊语言以及隶属度函数的设计与其相同)。

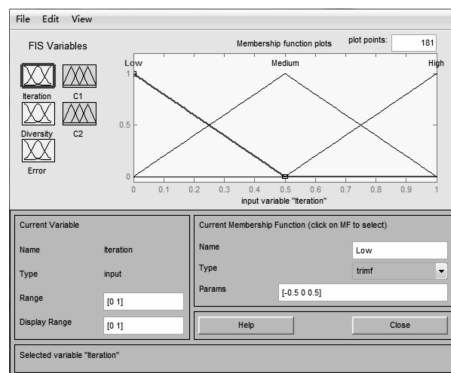


图3 δ 隶属度函数

由图 3 可以看出, 对于输入变量来说模糊语言选择的为 $\{Low, Medium, High\}$; 由图 4 可以看出输出变量 c_1 和 c_2 模糊语言选择的为 $\{Low, MediumLow, Medium, MediumHigh, High\}$ 。两者隶属度函数都取三角形函数。

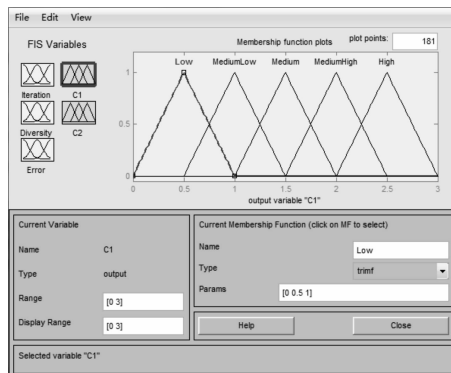


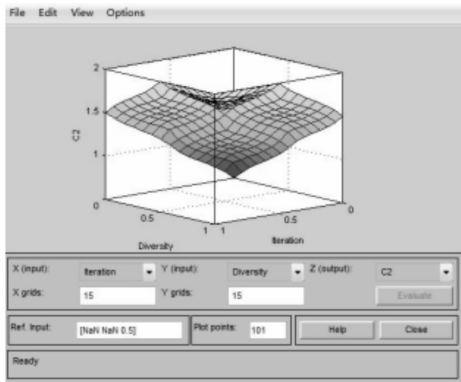
图4 c_1 隶属度函数图

在将输入输出变量模糊化后,需要进一步确定模糊规则来进行推理。本研究制定出 27 条模糊规则,如图 5 所示。

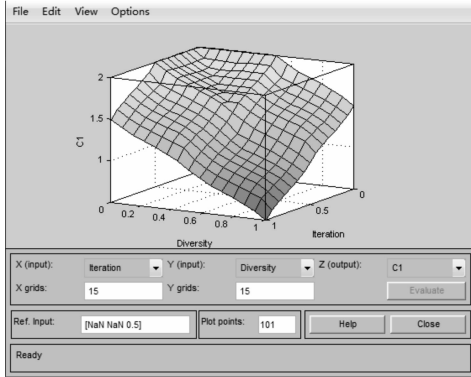
1. If (Iteration is Low) and (Diversity is Low) and (Error is Low) then (C1 is MediumLow)(C2 is Low) (1)
2. If (Iteration is Low) and (Diversity is Low) and (Error is Medium) then (C1 is MediumHigh)(C2 is Low) (1)
3. If (Iteration is Low) and (Diversity is Low) and (Error is High) then (C1 is High)(C2 is Low) (1)
4. If (Iteration is Low) and (Diversity is Medium) and (Error is Low) then (C1 is Medium)(C2 is Medium) (1)
5. If (Iteration is Low) and (Diversity is Medium) and (Error is Medium) then (C1 is MediumHigh)(C2 is MediumLow) (1)
6. If (Iteration is Low) and (Diversity is Medium) and (Error is High) then (C1 is MediumHigh)(C2 is Low) (1)
7. If (Iteration is Low) and (Diversity is High) and (Error is Low) then (C1 is Low)(C2 is MediumLow) (1)
8. If (Iteration is Low) and (Diversity is High) and (Error is Medium) then (C1 is Medium)(C2 is Medium) (1)
9. If (Iteration is Low) and (Diversity is High) and (Error is High) then (C1 is MediumLow)(C2 is Low) (1)
10. If (Iteration is Medium) and (Diversity is Low) and (Error is Low) then (C1 is Medium)(C2 is Medium) (1)
11. If (Iteration is Medium) and (Diversity is Low) and (Error is Medium) then (C1 is MediumHigh)(C2 is MediumLow) (1)
12. If (Iteration is Medium) and (Diversity is Low) and (Error is High) then (C1 is MediumHigh)(C2 is Low) (1)
13. If (Iteration is Medium) and (Diversity is Medium) and (Error is Low) then (C1 is MediumLow)(C2 is MediumHigh) (1)
14. If (Iteration is Medium) and (Diversity is Medium) and (Error is Medium) then (C1 is Medium)(C2 is Medium) (1)
15. If (Iteration is Medium) and (Diversity is Medium) and (Error is High) then (C1 is MediumHigh)(C2 is MediumLow) (1)
16. If (Iteration is Medium) and (Diversity is High) and (Error is Low) then (C1 is Low)(C2 is MediumHigh) (1)
17. If (Iteration is Medium) and (Diversity is High) and (Error is Medium) then (C1 is MediumLow)(C2 is MediumHigh) (1)
18. If (Iteration is Medium) and (Diversity is High) and (Error is High) then (C1 is Medium)(C2 is Medium) (1)
19. If (Iteration is High) and (Diversity is Low) and (Error is Low) then (C1 is Low)(C2 is MediumLow) (1)
20. If (Iteration is High) and (Diversity is Low) and (Error is Medium) then (C1 is Medium)(C2 is Medium) (1)
21. If (Iteration is High) and (Diversity is Low) and (Error is High) then (C1 is MediumLow)(C2 is Low) (1)
22. If (Iteration is High) and (Diversity is Medium) and (Error is Low) then (C1 is Low)(C2 is MediumHigh) (1)
23. If (Iteration is High) and (Diversity is Medium) and (Error is Medium) then (C1 is MediumLow)(C2 is MediumHigh) (1)
24. If (Iteration is High) and (Diversity is Medium) and (Error is High) then (C1 is Medium)(C2 is Medium) (1)

图 5 模糊控制规则

模糊规则输出曲面观测到输出变量 c_1 和 c_2 的变化情况如图 6 所示。



(a) c_2 输出曲面



(b) c_1 输出曲面

图 6 输出变量的输出曲面

另外,由于实际输出变量与模糊推理出的变量论域相同,此处省去了模糊量精确化这部分内容。

3 基于模糊推理技术 PSO 算法的路径规划实现

本研究将粒子的各维位置分量对应于平行直线簇 $(H_1, H_2, \dots, H_{n-1})$ 上的点,这样粒子的维度就可以根据直线簇的数量来确定,因此粒子的维度是 $n-1$ 。则 $x_{ij}(k)$ 表示第 k 次迭代时粒子 i 在平行线簇 H_j 上的取值 y_j 。

基于模糊推理技术 PSO 算法的路径规划步骤可描述如下:

(1) 根据路径的起始点和终止点建立新的笛卡尔坐标系,并进行坐标系转换;

(2) 对所有的障碍物进行扩展处理,每条边界向外扩展了半个机器人的宽度,计算扩展后各障碍物的边界在新坐标系下的位置坐标;

(3) 使用直线簇 $(H_1, H_2, \dots, H_{n-1})$ 将起点与终点之间的线段垂直平分,根据直线簇与机器人工作区域的交点确定粒子在各维位置的上下界 $x_j^{\min}, x_j^{\max}$;

(4) 初始化速度向量 $v_{ij}(0)$ 与位置向量 $x_{ij}(0)$, 计算每个粒子的适应度值 $f_i(0)$;

(5) 根据式(12—15) 分别计算其惯性因子 ω 、迭代因子 δ 、种群多样性因子 σ 和群体适应度差异因子 $Error$,并根据模糊推理技术自适应调整式(10) 中的学习因子 c_1, c_2 ;

(6) 根据式(10) 更新粒子在各维的速度 $v_{ij}(t)$, 根据式(11) 更新粒子在各维的位置 $x_{ij}(t)$;

(7) 根据式(9) 求其适应度 $f_i(t)$,更新个体最优值 $pbest_{ij}(t)$ 与种群最优值;

(8) 如果适应度满足精度要求或者迭代次数达到上限,则算法结束,否则转至步骤(5) 继续迭代,直到满足要求;

(9) 迭代结束后输出粒子在各维的位置,并将其从 xoy 坐标系下转换到 XOY 坐标系,即为最优路径的各个节点在原始坐标系 XOY 下的位置坐标。

算法流程图如图 7 所示。

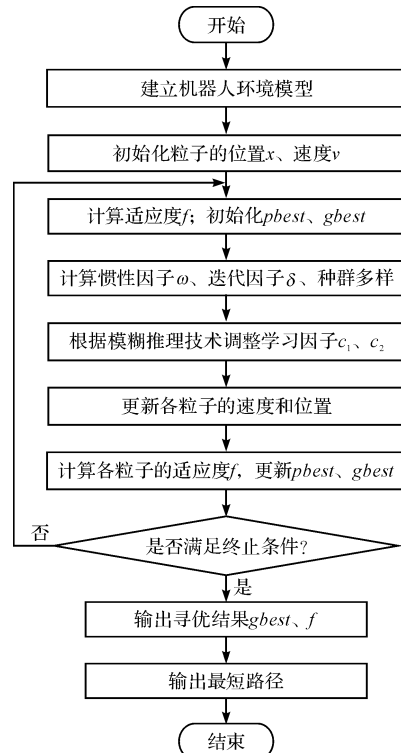


图 7 Fuzzy-PSO 算法流程图

4 实验及结果分析

笔者采用 Matlab, 分别使用标准 PSO 算法、APSO 算法与 Fuzzy-PSO, 对该路径规划进行仿真验证。参数: 最大迭代次数为 $t_m = 1\ 000$; 种群规模 $NP = 50$; 维度 $N = 30$; 粒子最大速度 $v_{\max} = 15$; 标准 PSO 算法的固定惯性因子 $\omega = 0.6$, 固定学习因子 $c_1 = c_2 = 2$; Fuzzy-PSO 的最大惯性因子 $\omega_{\max} = 0.9$ 、最小惯性因子 $\omega_{\min} = 0.1$ 。本文在建立适应度函数时, 选择路径长度、平滑度、安全度的权值系数分别为: $\omega_1 = 0.5$ 、 $\omega_2 = 0.3$ 、 $\omega_3 = 0.2$ 。而 APSO 算法参数参考其文献。

设置两种障碍物复杂度不同的环境, 两种环境下机器人的工作空间区域大小均为 100×100 。环境一的起点和终点坐标为: $S(45, 179)$, $E(158, 47)$; 环境二的起点和终点坐标为: $S(22, 189)$, $E(140, 42)$ 。

两种环境下标准 PSO 与 Fuzzy-PSO 的规划结果如图(8,9)所示。

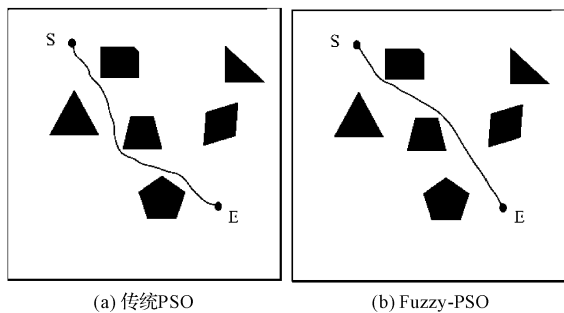


图8 环境一的路径规划结果

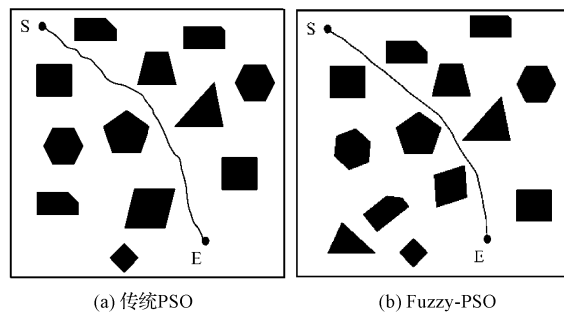


图9 环境二下的路径规划结果

由图8 仿真路径图可以看出: 采用 Fuzzy-PSO 和传统 PSO 算法均能规划出一条从起点到终点的无碰撞路径, 而 Fuzzy-PSO 算法规划出来的路径更短且更平滑。

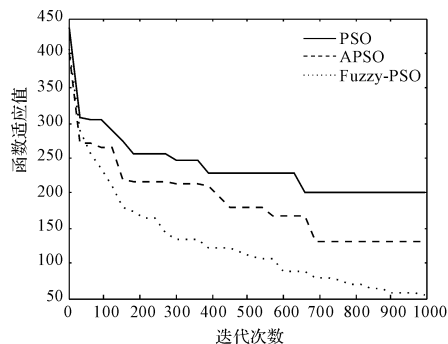
在两种环境下使用3种算法规划结果的相关数据统计如表1所示。

由表1可以看出: 相较于传统 PSO 与 APSO 算法路径规划, 经过 Fuzzy-PSO 算法处理的路径规划问题在路径长度、平滑度、安全度和运行时间上均优于传统算法。

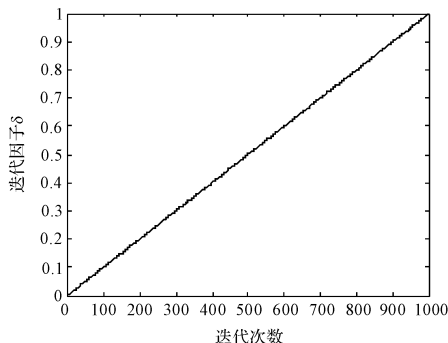
表1 算法结果对比

环境	算法	最优适应度 f	路径长度 f_1	平滑度 f_2	安全度 f_3	仿真时间 t
环境1	PSO	103.71	196.41	18.19	0.21	11.22
	APSO	100.10	192.46	16.68	0.19	11.13
	Fuzzy-PSO	99.51	189.193	16.25	0.16	10.88
环境2	PSO	212.96	303.12	28.11	0.30	12.13
	APSO	140.99	283.25	24.26	0.20	11.99
	Fuzzy-PSO	60.75	232.71	20.85	0.18	11.11

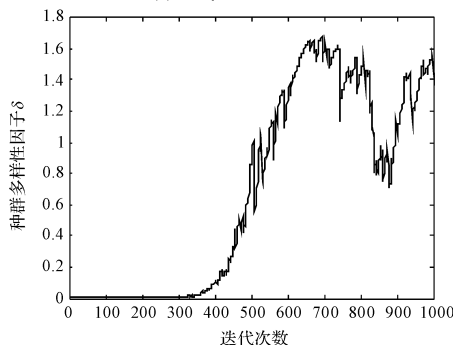
环境二下各参数对比曲线如图10所示。



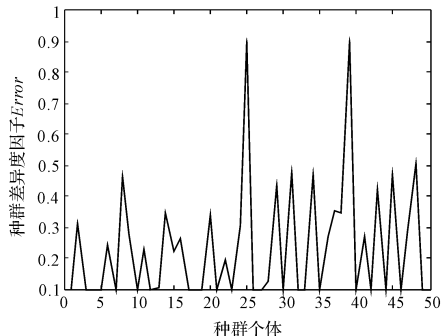
(a) 各算法适应度函数变化曲线



(b) Fuzzy-PSO的 δ 变化曲线



(c) Fuzzy-PSO的 δ 变化曲线



(d) Fuzzy-PSO的Error变化曲线

图10 环境二下各参数对比曲线

由图 10 可看出:(1)复杂环境二对于传统 PSO 与 APSO 算法在算法迭代末期陷入局部最优,而 Fuzzy-PSO 算法在保证算法效率的同时又增加了全局搜索能力。而这同样在输入变量迭代因子 δ 、种群多样性因子 σ 、群体适应度差异因子 Error 的变化曲线中得到体现。(2)随着迭代因子 δ 不断增加。(3)相应的种群多样性因子 σ 始终保持在充足水平。(4)群体适应度差异因子 Error 依据不同个体差异性进行取值。将这些量依据 Fuzzy-PSO 算法的模糊规则对学习因子 c_1 、 c_2 的取值使得算法全局搜索能力和局部搜索能力得到很好均衡,杜绝了种群多样性降低时易陷入局部最优出现“早熟”现象。

对比 APSO 算法,采用比较经典的线性递减权值策略自适应调整惯性因子 ω ,本文算法运用在路径规划领域更具有价值。

5 结束语

本文使用模糊推理技术动态地自适应调整学习因子 c_1 、 c_2 ,对比经典的线性递减权值策略自适应调整惯性因子 ω ,有效地均衡了算法的全局搜索能力和局部搜索能力,提高了算法的性能。

仿真结果表明:在简单环境下,本文提出的改进算法规划出的路径在路径长度、路径平滑度与时间效率上相比于标准 PSO 与 APSO 算法分别最少提高了 1.7%、2.5%、2.3%;在复杂情况下,改进算法分别最少提高 17%、14%、7%。由此,可以验证了该算法在解决路径规划,尤其对于复杂路径规划问题的可行性的与高效性。

参考文献 (References):

- [1] 王志中. 基于改进粒子群算法的机器人路径规划[J]. 制造技术与机床, 2018, 36(2): 150-154.
- [2] 陈超,唐坚. 基于可视图法的水面无人艇路径规划设计[J]. 中国造船, 2013, 54(1): 129-135.
- [3] 朱爱斌,刘洋洋,何大勇,等. 解决路径规划局部极小问题

- 的势场栅格法[J]. 机械设计与研究, 2017, 33(5): 46-50.
- [4] 关英姿,宋春林,董惠娟. 空间自由漂浮机器人对运动目标抓捕的路径规划[J]. 机器人, 2017, 39(6): 803-811.
- [5] SHI L, KODAGODA S. Towards generalization of semi-supervised place classification over generalized Voronoi graph[J]. **Robotics & Autonomous Systems**, 2013, 61(8): 785-796.
- [6] BAYILI S, POLAT F. Limited-Damage A*: A path search algorithm that considers damage as a feasibility criterion[J]. **Knowledge-Based Systems**, 2011, 24(5): 501 - 512.
- [7] WU Han. Research on Shortest Path Search of Improved Dijkstra Algorithm in GIS Navigation Application[J]. **Computer Science**, 2012, 39(10): 245-234.
- [8] 李胜文,张敏,李慧蓬,等. 基于 Floyd 算法的山西省电气化铁路电能质量测试方案[J]. 现代制造, 2017, 12(3): 7-8.
- [9] 王薇,魏世民,杨月巧,等. 基于神经网络的移动机器人路径规划[J]. 北京工业大学学报, 2010, 36(9): 1287-1291.
- [10] 赵大兴,余明进,许万. 基于高适应度值遗传算法的 AGV 最优路径规划[J]. 计算机工程与设计, 2017, 38(6): 1635-1641.
- [11] 王辉,王景良,朱龙彪,等. 基于改进蚁群算法的泊车系统路径规划[J]. 控制工程, 2018, 25(2): 253-258.
- [12] 李晶,李文斌. 基于模糊控制的移动机器人路径规划[J]. 中国农机化学报, 2015, 36(1): 272-278.
- [13] 魏勇,赵开新,王东署. 改进粒子群算法在移动机器人路径规划中的应用[J]. 火力与指挥控制, 2018, 43(2): 41-43.
- [14] 侯仰强,王天琪,李亮玉,等. 基于蚁群算法-粒子群算法的白车身侧围点焊机器人路径规划[J]. 中国机械工程, 2017, 28(24): 2990-2994.
- [15] 司艳伟,陈洪立. 六自由度果蔬采摘机器人关节空间轨迹规划[J]. 轻工机械, 2017, 35(4): 15-19.
- [16] RAHIMI A, KUMAR K D, ALIGHANBARI H. Particle swarm optimization applied to spacecraft reentry trajectory[J]. **Journal of Guidance Control Dynamics**, 2013, 36(1): 307-310.

[编辑:李辉]

本文引用格式:

刘彩霞. 基于模糊推理技术 PSO 算法的机器人路径规划研究[J]. 机电工程, 2019, 36(4): 445 - 450.

LIU Cai-xia. Robot path planning based on PSO algorithm with fuzzy inference technology[J]. *Journal of Mechanical & Electrical Engineering*, 2019, 36(4): 445 - 450.

《机电工程》杂志: <http://www.meem.com.cn>