

# 改进粒子群算法的三维空间路径规划研究

杨超杰, 裴以建, 刘 朋

云南大学 信息学院, 昆明 650500

**摘 要:**提出一种自适应混沌粒子群优化算法(SACPSO)用于三维空间路径规划。首先进行三维空间环境建模,并考虑使用路径长度、障碍物危险程度和路径平滑度三个评价函数来制定适应度函数;然后对算法中的三个控制参数提出了一种新的自适应更新策略,以此来动态调整算法的全局探索和局部开发能力;最后当种群陷入局部极值时,利用提出的自适应 Logistic 混沌映射对全局最优粒子进行混沌优化,引导种群跳出局部极值点。将该算法与其他改进的粒子群算法比较,结果表明,该算法在收敛到全局最优解时所用迭代次数更少,生成路径质量更高,有效地提高了粒子群算法应用于三维空间路径规划时的计算效率和可靠性。

**关键词:**路径规划;三维空间;粒子群;自适应;Logistic 混沌映射

**文献标志码:**A **中图分类号:**TP301 **doi:**10.3778/j.issn.1002-8331.1808-0245

杨超杰,裴以建,刘朋.改进粒子群算法的三维空间路径规划研究.计算机工程与应用,2019,55(11):117-122.

YANG Chaojie, PEI Yijian, LIU Peng. Research on three-dimensional space path planning based on improved particle swarm optimization algorithm. Computer Engineering and Applications, 2019, 55(11): 117-122.

## Research on Three-Dimensional Space Path Planning Based on Improved Particle Swarm Optimization Algorithm

YANG Chaojie, PEI Yijian, LIU Peng

Institute of Information, Yunnan University, Kunming 650500, China

**Abstract:** An adaptive chaotic particle swarm optimization algorithm(SACPSO) is proposed for three-dimensional space path planning. Firstly, the three-dimensional space environment modeling is carried out, and considers the three evaluation functions of path length, obstacle risk degree and path smoothness to formulate the fitness function. Then a new adaptive update strategy is proposed for the three control parameters in the algorithm, so as to dynamically adjust the global exploration and local exploitation capabilities of the algorithm. Finally, when the population falls into the local extremum, the proposed adaptive logistic chaotic map is used to optimize the global optimal particle and guide the population to jump out of the local extremum point. Comparing the algorithm with other improved particle swarm optimization algorithms, the results show that the algorithm uses fewer iterations when converging to the global optimal solution, and the quality of the generated path is higher, which effectively improves the computational efficiency and reliability of particle swarm optimization used in path planning problem in three-dimensional space.

**Key words:** path planning; three-dimensional space; particle swarm; self-adaptive; Logistic chaotic map

## 1 引言

三维空间路径规划是指在具有障碍物的三维环境中,按照预定义的评价标准,寻找一条从起始位置到目标位置的无碰撞最优(次优)路径<sup>[1]</sup>。目前,常用的三维路径规划算法主要有遗传算法<sup>[2]</sup>、D\*算法<sup>[3]</sup>、粒子群算法(PSO)<sup>[4]</sup>和蚁群算法等<sup>[5]</sup>,其中PSO算法作为一种新的群

智能优化算法,因具有操作简单和易于实现等优点,近年来,被广泛应用于三维路径规划领域<sup>[6]</sup>。

基本PSO算法应用于三维空间路径规划时建模简单、算法计算难度低,但存在容易陷入局部极值的缺陷,该缺陷使得规划出的路径不一定是全局最优路径。为了避免PSO算法在搜索过程中因早熟收敛而陷入局部

**基金项目:**云南大学服务云南行动计划项目(No.KS161012)。

**作者简介:**杨超杰(1993—),男,硕士生,研究方向为机器人技术及应用,E-mail:ynu\_ycj@163.com;裴以建(1957—),男,教授,研究方向为智能机器人的软、硬件开发;刘朋,男,硕士生,研究方向为机器学习。

**收稿日期:**2018-08-13 **修回日期:**2018-09-30 **文章编号:**1002-8331(2019)11-0117-06

**CNKI网络出版:**2018-12-21, <http://kns.cnki.net/kcms/detail/11.2127.TP.20181219.1050.010.html>

极值,国内外学者提出了诸多方法对PSO算法进行改进,如文献[7]和文献[8]在速度更新公式的基础上分别添加了有界随机扰动变量和邻近粒子信息,这两种方法是通过粒子速度更新引入变异机制来保持粒子寻优的多样性,防止粒子过早收敛,但粒子多样性的提高一定程度上影响了PSO算法的收敛速度。文献[9]对基本PSO算法中的惯性权重提出了“阶梯式”的调整策略,对不同时期的种群粒子设置不同大小的惯性权重,该方法在某种程度上较好地解决了PSO算法早熟收敛问题,但对参数的选择需要进行大量的数据实验,而且参数也容易受到环境规模的影响,算法灵活性不高。文献[10]和文献[11]将PSO算法中分别融入模拟退火法(SA)和差分进化法(DE),利用SA算法的概率突跳能力和DE算法的交叉与变异对已陷入局部极值的粒子采取引导措施,使粒子摆脱陷入局部极值的困境,这类融入其他先进算法并采取引导措施的方法虽然能够使种群快速跳出局部极值点,但对算法的整体寻优效率仍没有提升。

基于以上研究,为了能够使粒子群算法在避免陷入局部极值的同时还具有较高的寻优效率,提出一种自适应混沌粒子群优化算法(SACPSO)用于三维路径规划。该算法在开始迭代之前,首先进行环境建模,并考虑建立一种新的适应度函数模型用来评价生成路径的质量;然后在迭代过程中,为了提高算法在搜索空间内的寻优效率,对算法中的三个控制参数采用一种新的自适应更新策略,用于动态调整算法的全局探索和局部开发能力;最后,当种群陷入局部极值时,利用提出的自适应Logistic混沌映射引导种群跳出局部极值点,恢复算法的寻优能力,从而得到一条从起始位置到目标位置的较优路径。

## 2 路径规划数学建模及粒子适应度函数

路径规划数学建模结合文献[7]中的方法,分为两部分进行。第一部分如图1所示,首先确定路径规划的起始位置 $S$ 和目标位置 $G$ ;然后,为了方便计算,设环境中存在的障碍物为球形障碍物,并分别用式 $\{O_k(o_{x_k}, o_{y_k}, o_{z_k}), r_k | k=1, 2, \dots, m\}$ 表示,其中 $(o_{x_k}, o_{y_k}, o_{z_k})$ 为第 $k$ 个球形障碍物的球心坐标,  $r_k$ 为其半径。第二部分表示对环境模型进行分割,如图2所示,首先建立以点 $S$ 为坐标原点、 $SG$ 为 $Z'$ 轴正方向的局部坐标系 $O'-X'Y'Z'$ ;然后将 $SG$ 进行 $d+1$ 等分,过每个等分点作垂直于 $Z'$ 轴的 $d$ 个平面 $L_1, L_2, \dots, L_d$ ;最后,根据球形障碍物在 $X'O'Y'$ 平面上的投影作一个立方体 $ABCD-EFHM$ ,其中面 $ABCD$ 在 $X'O'Y'$ 平面上,是一个矩形平面,  $a$ 为障碍物边界向外扩展的距离,立方体的高 $AM=SG$ 。

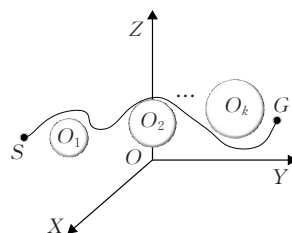


图1 环境模型 $O-XYZ$ 坐标系

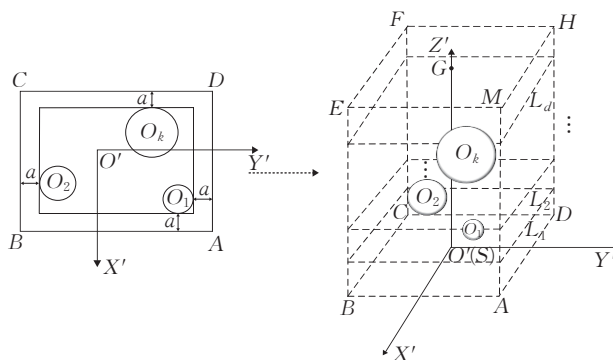


图2 环境模型 $O'-X'Y'Z'$ 坐标系

坐标系 $O-XYZ$ 与 $O'-X'Y'Z'$ 之间的转化关系为:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \alpha_x & \cos \alpha_y & \cos \alpha_z \\ \cos \beta_x & \cos \beta_y & \cos \beta_z \\ \cos \theta_x & \cos \theta_y & \cos \theta_z \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (1)$$

其中 $\cos \alpha_x$ 、 $\cos \alpha_y$ 和 $\cos \alpha_z$ 分别为 $X$ 轴与 $X'$ 、 $Y'$ 和 $Z'$ 轴的夹角,  $\cos \beta_x$ 、 $\cos \beta_y$ 和 $\cos \beta_z$ 分别为 $Y$ 轴与 $X'$ 、 $Y'$ 和 $Z'$ 轴的夹角,  $\cos \theta_x$ 、 $\cos \theta_y$ 和 $\cos \theta_z$ 分别为 $Z$ 轴与 $X'$ 、 $Y'$ 和 $Z'$ 轴的夹角。

起始位置 $S(P_0)$ 到目标位置 $G(P_{d+1})$ 之间的路径为:从点 $S$ 出发,首先到达平面 $L_1$ 上的某一点 $P_1(x'_1, y'_1, z'_1)$ ,然后从点 $P_1(x'_1, y'_1, z'_1)$ 出发到达平面 $L_2$ 上的某一点 $P_2(x'_2, y'_2, z'_2)$ ,最后到达平面 $L_d$ 上的某一点 $P_d(x'_d, y'_d, z'_d)$ ,连接 $P_d(x'_d, y'_d, z'_d)$ 与点 $G$ ,构成一条从起始位置到目标位置的路径。

粒子适应度函数是评价生成路径质量的唯一依据,因此,为了更好地对路径质量进行判断,综合考虑路径长度、障碍物危险程度和路径平滑度三个评价函数来制定适应度函数。

(1)规划出的路径尽可能短,路径长度评价函数可由下式表示:

$$T_{SG} = \sum_{j=0}^d \sqrt{(x'_j - x'_{j+1})^2 + (y'_j - y'_{j+1})^2 + (z'_j - z'_{j+1})^2} = \sum_{j=0}^d \sqrt{(x'_j - x'_{j+1})^2 + (y'_j - y'_{j+1})^2 + \left(\frac{h}{d+1}\right)^2} \quad (2)$$

式中,  $T_{SG}$ 表示一个粒子中所有相邻节点之间的欧式距离总和,  $h$ 表示立方体 $ABCD-EFHM$ 的高。

(2)引入障碍物危险程度,使路径和障碍物之间保持一定的安全距离。障碍物危险程度计算方法为:从第

$k(k=1,2,\cdots,m)$  个球形障碍物的球心  $O_k$  向路径段作垂线,分别求出球心和球体距离路径的最短垂线距离  $l(O_k)$  和  $l_k=l(O_k)-r_k$ ,若存在任意一个  $l_k<0$ ,则表示障碍物与路径相交,该路径为无效路径,代表该路径的粒子继续迭代优化;若所有的  $l_k\geq 0$ ,则表示生成路径成功避开障碍物,由公式(3)计算障碍物危险程度评价函数  $dan_m$ 。

$$dan_m = \sum_{k=1}^m dan_k$$

$$dan_k = \begin{cases} 1, & l_k \leq l^{\min} \\ \frac{l^{\max} - l_k}{l^{\max} - l^{\min}}, & l^{\max} > l_k > l^{\min} \\ 0, & l_k \geq l^{\max} \end{cases} \quad (3)$$

式中,  $l^{\max}$  表示障碍物可能产生危害影响的最大作用距离,  $l^{\min}$  表示障碍物可能产生危害影响的最小作用距离,  $m$  表示环境中球形障碍物的总数。

(3) 引入路径平滑度,避免规划出的路径多为折线。路径平滑度计算方法为:对一条路径中所有连接三个相邻节点的两条线段之间的偏转角  $\lambda$  求和来近似评估路径的平滑度,路径平滑度评价函数如公式(4)所示:

$$L_{\text{smooth}} = \sum_{j=1}^d \lambda_j =$$

$$\sum_{j=1}^d \arccos \left[ \frac{((x'_j - x'_{j-1}) \cdot (x'_{j+1} - x'_j) + (y'_j - y'_{j-1}) \cdot (y'_{j+1} - y'_j) + (z'_j - z'_{j-1}) \cdot (z'_{j+1} - z'_j))}{l(P_{j-1}, P_j) \cdot l(P_j, P_{j+1})} \right] \quad (4)$$

式中,  $\lambda \in [0, \pi]$ ,  $l(P_{j-1}, P_j)$ 、 $l(P_j, P_{j+1})$  表示相邻两节点之间的欧式距离。

综合  $T_{SG}$ 、 $dan_k$ 、 $L_{\text{smooth}}$  所得粒子适应度函数为:

$$f = \begin{cases} \emptyset, & dan_m = \emptyset \\ \kappa_1 \cdot T_{SG} + \kappa_2 \cdot dan_m + \kappa_3 \cdot \frac{1}{L_{\text{smooth}}}, & dan_m \neq \emptyset \end{cases} \quad (5)$$

式中,  $\kappa_1$ 、 $\kappa_2$  和  $\kappa_3$  是 0~1 之间的权重系数<sup>[12]</sup>,用来灵活配置  $T_{SG}$ 、 $dan_k$  和  $L_{\text{smooth}}$  之间的关系;  $dan_m = \emptyset$  表示粒子所代表的路径不能避开障碍物;  $f = \emptyset$  表示粒子适应度函数为空解。

### 3 改进 PSO 算法实现三维路径规划

#### 3.1 基本 PSO 算法

PSO 算法通常被称为基于群体的随机优化算法,是从自然界中鸟类捕食行为得到启发的<sup>[12]</sup>。在 PSO 算法中,每一个粒子都有记忆,可以追踪上一次迭代时的最佳位置:粒子个体最佳位置  $X_{pbest}$  和粒子全局最佳位置  $X_{gbest}$ 。假设有  $n$  个粒子存在于  $d$  维搜索空间中,则第  $i$  个粒子在第  $t$  次迭代时的位置和飞行速度分别表示为

向量  $X_i^t = (x_{i,1}^t, x_{i,2}^t, \cdots, x_{i,d}^t)$  和向量  $V_i^t = (v_{i,1}^t, v_{i,2}^t, \cdots, v_{i,d}^t)$ ,  $i=1,2,\cdots,n$ 。第  $i$  个粒子在第  $t+1$  次迭代时速度和位置更新方法为:

$$V_i^{t+1} = \omega \cdot V_i^t + c_1 r_1 (X_{ipbest}^t - X_i^t) + c_2 r_2 (X_{gbest}^t - X_i^t) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

其中  $\omega$  为惯性权重因子,  $r_1$  和  $r_2$  为服从均匀分布  $U(0,1)$  的随机变量,  $c_1$  和  $c_2$  是加速系数。

#### 3.2 PSO 参数的调整策略

##### 3.2.1 惯性权重因子 $\omega$ 的调整策略

定义  $f_{\max}$ 、 $f_{\min}$  和  $\bar{f}$  分别为当代粒子适应度函数的最大值、最小值和平均值,  $\bar{f}_{\text{left}}$  为当代所有比  $\bar{f}$  小的粒子适应度函数平均值,  $\bar{f}_{\text{right}}$  为当代所有比  $\bar{f}$  大的粒子适应度函数平均值,  $f_i$  为当代第  $i$  个粒子的适应度函数值。由于在迭代过程中适应度函数值最低的粒子为每次迭代的最优解,所以:

(1) 当  $f_i < \bar{f}_{\text{left}}$  时,该粒子解的质量较优,为了保护该粒子,惯性权重因子  $\omega$  应该赋予一个较小的值,表达式如下:

$$\omega = \omega_{\text{ave}} - \frac{\bar{f}_{\text{left}} - f_i}{\bar{f} - f_{\min}} \cdot (\omega_{\text{ave}} - \omega_{\min}) \quad (8)$$

$$\omega_{\text{ave}} = \frac{\omega_{\max} + \omega_{\min}}{2} \quad (9)$$

其中  $\omega_{\max}$  和  $\omega_{\min}$  分别是惯性权重的最大值和最小值。

(2) 当  $f_i > \bar{f}_{\text{right}}$  时,该粒子解的质量较差,为了使该粒子移动到更好的搜索区域得到较优解,惯性权重因子  $\omega$  应该赋予一个较大的值,表达式如下:

$$\omega = \omega_{\text{ave}} + \frac{f_i - \bar{f}_{\text{right}}}{f_{\max} - \bar{f}} \cdot (\omega_{\text{ave}} - \omega_{\min}) \quad (10)$$

(3) 当  $\bar{f}_{\text{left}} < f_i < \bar{f}_{\text{right}}$  时,该粒子解的质量处于(1)和(2)之间,惯性权重  $\omega$  采用线性递减策略,表达式如下:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot \frac{t}{T} \quad (11)$$

其中  $t$  和  $T$  分别为当前迭代次数和最大迭代次数。

##### 3.2.2 加速系数 $c_1$ 、 $c_2$ 的调整策略

在 PSO 算法中,加速系数  $c_1$  (认知部分)和  $c_2$  (社会部分)分别赋予粒子自我总结能力和向全局最优粒子学习的能力,是决定 PSO 性能的另外两个重要参数<sup>[13]</sup>。在上部分惯性权重自适应更新的基础上,根据每次迭代中每个粒子不同的惯性权重值大小来计算加速系数,该方法表达式及推导过程如下所示:

设  $c_1 = k_1 \cdot \omega + b_1$ ,  $c_2 = k_2 \cdot \omega + b_2$ ,考虑让认知部分  $c_1$  与粒子的惯性权重  $\omega$  成正比,社会部分  $c_2$  与粒子的惯性权重  $\omega$  成反比。

$$\text{由} \begin{cases} c_{1f} = k_1 \cdot \omega_{\max} + b_1 \\ c_{1f} = k_1 \cdot \omega_{\min} + b_1 \end{cases} \rightarrow \begin{cases} 2.5 = 0.9k_1 + b_1 \\ 0.5 = 0.4k_1 + b_1 \end{cases}$$



$$\begin{cases} c_{2i} = k_2 \cdot \omega_{\max} + b_2 \\ c_{2f} = k_2 \cdot \omega_{\min} + b_2 \end{cases} \rightarrow \begin{cases} 0.5 = 0.9k_2 + b_2 \\ 2.5 = 0.4k_2 + b_2 \end{cases} \text{可得:}$$

$$c_1 = 4\omega - 1.1 \quad (12)$$

$$c_2 = -4\omega + 4.1 \quad (13)$$

其中  $c_{1i} = 2.5$  和  $c_{1f} = 0.5$  分别是认知部分的初始值和最终值,  $c_{2i} = 0.5$  和  $c_{2f} = 2.5$  分别是社会部分的初始值的最终值。  $\omega_{\max} = 0.9$ ,  $\omega_{\min} = 0.4$ ,  $\omega$  是由式(8)~(11)计算所得,为某一粒子的惯性权重。

由公式(12)、(13)可以看出:当  $\omega$  的值较大,即当前该粒子的位置较差时,  $c_1$  较大,  $c_2$  较小,有利于扩大该粒子的搜索范围;当  $\omega$  的值较小,即当前该粒子的位置较优时,  $c_1$  较小,  $c_2$  较大,该粒子能在较优位置附近局部细化搜索,寻找更优的位置。

### 3.3 自适应 Logistic 混沌映射的引入

#### 3.3.1 Logistic 混沌映射

基本 Logistic 映射迭代公式为:

$$z^{\xi+1} = \mu z^{\xi} (1 - z^{\xi}) \quad (14)$$

其中  $\xi$  为迭代次数,  $\mu$  为控制参数,取  $\mu = 4$ , 初始值  $z^0 \in (0, 1)$ 。

使用基本 Logistic 映射对全局最优粒子的混沌优化过程如下<sup>[14]</sup>:

(1) 当种群陷入局部极值时,首先获取当前全局最优粒子  $X_g$ , 然后将  $X_g$  中每一维平面上选取到的坐标点  $(x'_j, y'_j)$  通过公式(15)映射到 Logistic 方程的定义域中,得到一组解  $\{(z_{jx}, z_{jy})\} (j = 1, 2, \dots, d)$ :

$$\begin{cases} z_{jx} = \frac{x'_j - x'_{\min}}{x'_{\max} - x'_{\min}} \\ z_{jy} = \frac{y'_j - y'_{\min}}{y'_{\max} - y'_{\min}} \end{cases} \quad (15)$$

其中  $x'_{\min}, y'_{\min}$  和  $x'_{\max}, y'_{\max}$  分别为每一维平面上  $X'$  轴和  $Y'$  轴的取值范围。

(2) 将  $\{(z_{jx}, z_{jy})\}$  作为 Logistic 方程的初始解代入式(14)中迭代,产生混沌序列  $\{(z_{jx}^C, z_{jy}^C)\} (C = 1, 2, \dots)$ , 再把产生的混沌序列由公式(16)通过逆映射返回到原解空间得  $X_g^C$ 。

$$\begin{cases} x'_{j,C} = z_{jx}^C \cdot (x'_{\max} - x'_{\min}) + x'_{\min} \\ y'_{j,C} = z_{jy}^C \cdot (y'_{\max} - y'_{\min}) + y'_{\min} \end{cases} \quad (16)$$

(3) 计算并比较  $X_g^C$  中每一个可行解的适应度函数值,用得到的最优解更新  $X_{pbest}$  和  $X_{gbest}$  的值。

#### 3.3.2 自适应 Logistic 混沌映射

直接使用基本 Logistic 映射混沌寻优时,是将产生的混沌序列直接由逆映射返回到原解空间,并没有考虑

此时粒子的搜索范围,这样计算了不必要的粒子适应值,增加了算法的计算量。

为了避免计算的盲目性,对混沌序列的搜索范围提出了一种自适应搜索策略。在混沌优化前,首先将当前迭代次数(第  $t$  次)中所有粒子的适应度函数值按从大到小排列;然后取前面 80% 的粒子(共有  $W$  个),并求得每个粒子在第  $j$  个平面上  $X'$  轴和  $Y'$  的最小值  $\tau_{x'}^{\min}(j, t)$ 、 $\tau_{y'}^{\min}(j, t)$  和最大值  $\zeta_{x'}^{\max}(j, t)$ 、 $\zeta_{y'}^{\max}(j, t)$ ;最后把每个平面上坐标轴的最小值和最大值作为逆映射到原解空间时粒子的搜索范围。逆映射过程中,任何平面上的坐标值超出范围,则直接舍去该粒子,这样不但能使算法快速逃出局部极值点,而且还减轻了计算量,提高了混沌寻优的搜索效率。

#### 3.4 算法实现

步骤 1 创建路径规划环境地图,确定路径规划的起始位置  $S$  和目标位置  $G$ 。

步骤 2 根据球形障碍物的空间分布和障碍物边界向外扩展距离的大小求出路径规划的环境空间模型  $ABCD-EFHM$ , 由空间模型求出粒子在平面上飞行的最大速度  $v^{\max}$ 。

$$v^{\max} = \begin{cases} v_{x'}^{\max} = (x'_{\max} - x'_{\min}) \times 0.1 \\ v_{y'}^{\max} = (y'_{\max} - y'_{\min}) \times 0.1 \end{cases} \quad (17)$$

步骤 3 设置粒子种群数目  $n$ 、最大迭代次数  $iter_{\max}$ 、自适应 Logistic 映射的最大混沌寻优次数  $Chaos_{\max}$ , 障碍物产生危害影响的最大(最小)作用距离  $l^{\max}(l^{\min})$ 。由公式(18)、(19)初始化每个粒子在坐标系  $O'-X'Y'Z'$  下的初始位置和初始速度。

$$\begin{cases} x'_{i,j} = (x'_{\max} - x'_{\min}) \cdot r + x'_{\min} \\ y'_{i,j} = (y'_{\max} - y'_{\min}) \cdot r + y'_{\min} \end{cases} \quad (18)$$

$$\begin{cases} v_{x'}^{i,j} = (2r - 1) \times v_{x'}^{\max} \\ v_{y'}^{i,j} = (2r - 1) \times v_{y'}^{\max} \end{cases} \quad (19)$$

其中  $r$  为  $(0, 1)$  上的随机数,  $i = 1, 2, \dots, n$  为种群粒子个数,  $j = 1, 2, \dots, d$  为粒子维数。

步骤 4 由公式(5)求出各粒子的初始适应度函数值  $f_i^0$ , 根据  $f_i^0$  的值更新  $X_{pbest}^0$  和  $X_{gbest}^0$ , 并求出每个粒子的初始惯性权重  $\omega_i^0$ , 再由  $\omega_i^0$  求出每个粒子的初始加速系数。

步骤 5 依据公式(6)和公式(7)更新每个粒子的速度和位置,如果粒子的速度和位置超出最大值和边界值,则按最值计算。

步骤 6 求出每个粒子在当前迭代次数下的适应度函数值  $f_i^t$ , 根据  $f_i^t$  的值,首先求出用于下一次迭代的惯性权重  $\omega_i^{t+1}$ , 并由  $\omega_i^{t+1}$  求出下一次迭代时的加速系数,然后用  $f_i^t$  的值更新粒子个体最佳位置  $X_{pbest}^t$  和粒子

全局最佳位置  $X'_{\text{gbest}}$ 。

步骤7 根据式(20)计算当前迭代时刻种群适应度方差  $\sigma^2$ ,将  $\sigma^2$  与预先设定的阈值进行比较,如果  $\sigma^2$  小于预先设定的阈值,则说明种群陷入局部极值,转入步骤8,否则转入步骤9。

$$\sigma^2 = \sum_{i=1}^n \left( \frac{f_i - \bar{f}}{f'} \right)^2$$
$$f' = \begin{cases} \max\{|f_i - \bar{f}|\}, & \max\{|f_i - \bar{f}|\} \geq 1 \\ 1, & \max\{|f_i - \bar{f}|\} < 1 \end{cases} \quad (20)$$

其中  $f_i$  为第  $i$  个粒子的适应度值,  $f'$  为归一化因子。

步骤8 采用自适应Logistic混沌映射对全局最优粒子进行混沌优化,引导种群跳出局部极值点。

步骤9 判断是否达到最大迭代次数  $iter_{\text{max}}$ ,如果没有,则转到步骤5。

4 实验结果与分析

实验是在 Windows7 系统、InterCore i5-3337U 处理器、主频为 1.80 GHz 的计算机上进行,实验平台为 Matlab9.1。

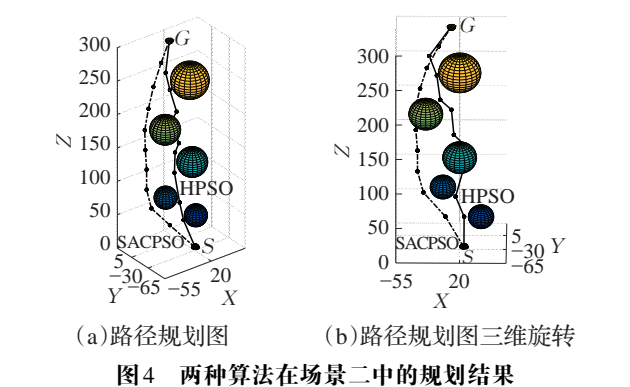
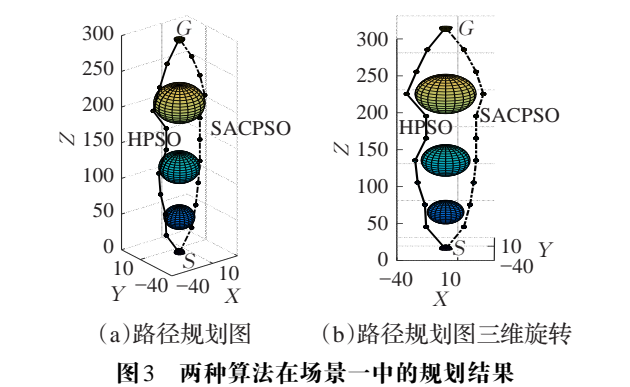
实验参数为:种群个数  $n=40$ ,最大迭代次数  $iter_{\text{max}}=150$ ,粒子维数  $d=10$ ,障碍物产生危害影响的最大(最小)作用距离  $l^{\text{max}}=5$  ( $l^{\text{min}}=3$ ),适应度函数  $f$  的加权系数  $\kappa_1=0.6$ 、 $\kappa_2=0.3$ 、 $\kappa_3=0.1$ ,种群适应度方差的阈值根据实验具体情况而定,混沌寻优次数  $Chaos_{\text{max}}=200$ ,障碍物边界扩展距离  $a=15$ 。

仿真实验在两个不同的场景下进行,分别为场景一和场景二。场景一中起始位置  $S$  和目标位置  $G$  分别为  $(0,0,0)$  和  $(0,0,300)$ ,该场景存在三个球形障碍物,球心坐标分别为  $(0,0,50)$ 、 $(0,0,120)$  和  $(0,0,210)$ ,其半径分别为 15、20 和 25。场景二中起始位置  $S$  和目标位置  $G$  分别为  $(25,-25,0)$  和  $(10,10,300)$ ,该场景有五个球形障碍物,球心坐标分别为:  $(45,0,30)$ 、 $(0,5,70)$ 、 $(20,-25,130)$ 、 $(-20,-20,190)$ 、 $(20,-20,250)$ ,其半径分别为 15、15、20、20 和 25。根据路径规划的起始位置坐标和障碍物空间分布,场景一中三维环境空间为  $[-40,40;-40,40;0,300]$ ,场景二中三维环境空间为  $[-55,75;-60,35;0,300]$ 。

为了验证改进算法在解决三维空间路径规划问题中的有效性,在相同环境模型下,将所提算法与文献[7]中含有固定值参数的 HPSO 算法进行对比实验。

在实验过程中,两种算法搜索到的最终路径是用 40 个粒子在两种不同场景下迭代 150 次后输出。实验结束后,图 3 和图 4 分别给出了两种算法在场景一和场景二中的规划结果,图 5 给出了两种算法在场景二中的最

优适应度值曲线,表 1 和表 2 总结了两种算法在迭代结束后产生的数据,数据主要包括迭代结束时花费的总时间  $AllTime$ 、输出路径的适应度函数值  $f$ 、开始收敛到最优解时的迭代次数  $Iter$  以及执行时间  $time$ 。



从场景一中规划的结果可以看出 HPSO 算法和 SACPSO 算法在简单环境中都能生成一条较优全局路径,但在场景二中,由于障碍物的分布比较集中,而 SACPSO 算法考虑到了障碍物危险程度和路径平滑度,导致算法最终生成的路径相比 HPSO 算法要更加平滑,路径解的质量更高。

分析表 1 和表 2 中的数据可以看出,SACPSO 算法由于引入了参数的自适应更新策略以及混沌寻优,导致算法的计算量增大,算法在迭代结束时所花费的时间最长,但 SACPSO 算法在开始收敛到全局最优解时所用的迭代次数  $Iter$  以及执行时间  $time$  比 HPSO 算法在场景一中分别少了 21 次和 3.15 s、在场景二中分别少了 26 次和 5.07 s,这说明,尽管 SACPSO 算法的计算量比 HPSO 算法大,但 SACPSO 算法在生成全局最优路径解的效率上要高于 HPSO 算法。

表1 场景一生成的数据

算法	$AllTime/s$	$f$	$Iter$	$time/s$
HPSO	29.46	286.32	38	7.46
SACPSO	36.41	284.79	17	4.31

表2 场景二生成的数据

算法	$AllTime/s$	$f$	$Iter$	$time/s$
HPSO	36.54	301.76	68	16.56
SACPSO	41.05	296.53	42	11.49

分析图5中两算法的最优适应度曲线不难发现,改进算法由于引入了参数的自适应调整策略以及全局最优粒子的自适应混沌寻优策略,导致算法在迭代过程中不仅加快了粒子群的收敛速度,而且算法在陷入局部极值时也能够快速地跳出局部极值点,恢复寻优能力。

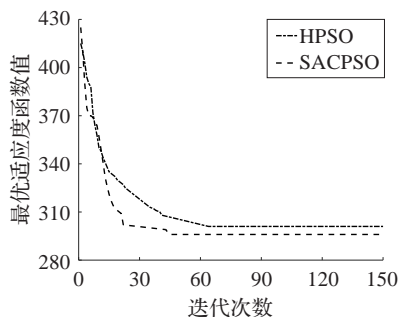


图5 场景二中两种算法的最优适应度曲线

## 5 结束语

提出了一种称为自适应混沌粒子群优化算法的改进粒子群算法用于三维空间路径规划。该改进算法主要有以下特点:(1)建立了一种新的适应度函数模型用于衡量生成路径的质量;(2)对算法中的三个控制参数提出了新的自适应更新策略;(3)当种群陷入局部极值时,利用提出的自适应 Logistic 混沌映射对全局最优粒子进行混沌优化,引导种群跳出局部极值点。通过相关的仿真实验,验证了所提算法能够进一步提高粒子群算法在三维空间中生成路径解的质量和效率。

## 参考文献:

- [1] 吴玉香,王超.一种改进的移动机器人三维路径规划方法[J].华南理工大学学报(自然科学版),2016,44(9):53-60.
- [2] Song B, Wang Z, Sheng L. A new genetic algorithm approach to smooth path planning for mobile robots[J]. Assembly Automation, 2016, 36(2): 138-145.
- [3] Majumder S, Prasad M S. Three dimensional D\* algorithm for incremental path planning in uncooperative environment[C]//International Conference on Signal Processing and Integrated Networks, 2016: 431-435.
- [4] Tang B, Zhu Z, Luo J. A convergence-guaranteed particle swarm optimization method for mobile robot global path planning[J]. Assembly Automation, 2017, 37(1): 114-129.
- [5] Chen J. Optimal path planning of robot based on ant colony algorithm[J]. Acta Technica Csav, 2017, 62(2): 219-227.
- [6] Nie Z, Yang X, Gao S, et al. Research on autonomous moving robot path planning based on improved particle swarm optimization[C]//Evolutionary Computation, 2016: 2532-2536.
- [7] 陈世明,谢竟,陈文栋,等.基于HPSO算法的三维空间路径规划[J].华中科技大学学报(自然科学版),2013,41(2): 109-113.
- [8] 熊保顺,崔浩林.快递机器人目标路径避障规划仿真研究[J].计算机仿真,2016,33(9):369-374.
- [9] 方群,徐青.基于改进粒子群算法的无人机三维航迹规划[J].西北工业大学学报,2017,35(1):66-73.
- [10] 唐汇禹,彭世蕤,孙经蛟,等.基于SAPSO算法的无人机三维航迹规划[J].战术导弹技术,2017(2):62-68.
- [11] Zeng N, Zhang H, Chen Y, et al. Path planning for intelligent robot based on switching local evolutionary PSO algorithm[J]. Assembly Automation, 2016, 36(2): 120-126.
- [12] 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): 1-13.
- [13] Lazzús J A, Rivera M, López-Caraballo C H. Parameter estimation of Lorenz chaotic system using a hybrid swarm intelligence algorithm[J]. Physics Letters A, 2016, 380(11/12): 1164-1171.
- [14] 李建美,高兴宝.基于自适应变异的混沌粒子群优化算法[J].计算机工程与应用,2016,52(10):44-49.
- [15] Khan S U, Yang S, Wang L, et al. A modified particle swarm optimization algorithm for global optimizations of inverse problems[J]. IEEE Transactions on Magnetics, 2016, 52(3): 1-4.