

基于改进粒子群算法的路径优化问题研究

梁 静<sup>1</sup>, 宋 慧<sup>1</sup>, 瞿博阳<sup>2</sup>, 毛晓波<sup>1</sup>

(1. 郑州大学 电气工程学院, 河南 郑州 450001; 2. 中原工学院 电子信息学院, 河南 郑州 450007)

**摘 要:**把动态约束机制及改进的粒子群优化算法(含交叉策略的动态多组群粒子群优化算法)与 Bezier 曲线结合来实现路径优化. 用三次 Bezier 曲线来描述路径,测试不同算法下得到的曲线在路径优化问题中的特性,通过比较得出不同算法的优劣性. 实验结果表明:动态多组群粒子群优化算法克服了标准粒子群优化算法易于早熟和陷入局部最优的特点,动态约束提高了交叉策略下算法的寻优性能,能够灵活地处理约束条件,克服了以往静态惩罚函数的缺点.

**关键词:**动态约束机制;动态多组群粒子群优化算法;Bezier 曲线

**中图分类号:** TP18      **文献标志码:** A      doi:10.3969/j.issn.1671-6833.2014.01.008

0 引言

近些年,随着科技的发展,机器人已经应用到工业、农业、医疗、航空航天等方面. 而路径优化则是机器人实施各种操作最基本的环节. 机器人路径优化的实质是按照一定的技术指标(例如:路径最短,时间最少,花费最小等),找到一条满足特定技术指标的从起点到终点和障碍物之间无碰撞的路径<sup>[1]</sup>. 目前,已有多种进化算法应用到路径优化中,其中文献[2-3]介绍了遗传算法实现的路径优化,但是其计算量大并且处理规模较小,文献[4-5]中,差分进化容易陷入最优,搜索能力差,信息反馈不完整,在运用时需要对其进行改进.

粒子群优化算法是一种模拟鸟群觅食行为的优化算法,由于粒子群优化算法在概念上容易理解,方便应用而且收敛迅速,跟其他的进化算法相比,它有一个突出的优势,就是可以用较少的个体进行搜索而获得较好的结果,存储量和计算量都相对较低<sup>[6]</sup>. 但是,粒子群优化算法由于收敛速度较快,易于陷入局部最优.

为了克服标准粒子群优化算法存在的缺点,笔者提出一种改进粒子群优化算法—动态多组群粒子群优化算法(DMS-PSO)的移动机器人路径规划方法. 用三次 Bezier 曲线来描述路径,可以将路径

规划问题转化为生成 Bezier 曲线的节点位置优化问题<sup>[7]</sup>. 以往的静态约束机制由于参数设置过大或者过小,都不易使粒子找到最优解,因此,采用一种新的动态约束机制来克服这种缺点. 实验结果表明:动态多组群粒子群优化算法增加了粒子的离散度,新的动态约束机制提高了算法的寻优能力.

1 曲线介绍

Bezier 曲线是由法国工程师 P. E Bezier 于 1962 年提出的一种新的曲线表示方式<sup>[8-10]</sup>. 目前 Bezier 曲线已广泛应用于计算机绘图.

一条  $n$  次 Bezier 曲线可以表示为

$$C(u) = \sum_{i=0}^n B_{i,n}(u)P_i, 0 \leq u \leq 1. \tag{1}$$

式中:  $\{P_i\}$  称为控制点;  $\{B_{i,n}(u)\}$  是著名的  $n$  次 Bernstein 多项式,其定义为

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!}u^i(1-u)^{n-i}, i = 0, 1, \cdots, n. \tag{2}$$

在 Bezier 曲线中,最常用的是三次 Bezier 曲线,要想获得三次 Bezier 曲线,应使  $n = 3$ ,由(1)和(2)可以得到三次 Bezier 的各点参数方程如下:

$$C(u) = P_0(1-u)^3 + 3P_1u(1-u)^2$$

收稿日期:2013-09-20;修订日期:2013-11-20  
基金项目:国家自然科学基金资助项目(211000531605);中国博士后科学基金特别资助项目(2012T50639);教育部高等学校博士学科点专项科研基金(20114101110005);河南省科技攻关项目(132102210521).  
作者简介:梁静(1981-),女,河南郑州人,郑州大学副教授,博士,主要研究领域为进化计算、模式识别、运筹学理论、群体智能的研究, E-mail: liangjing@zzu.edu.cn.

$$+ 3P_2u^2(1-u) + P_3u^3, u \in [0,1]. \quad (3)$$

文中用  $P_0, P_1, P_2, P_3$  定义一段三次 Bezier 曲线. 曲线起始于  $P_0$  走向  $P_1$ , 并从  $P_2$  的方向来到  $P_3$ , 曲线一般不会经过  $P_1$  或  $P_2$ . 两段或者多段 Bezier 曲线, 为保证曲线的连续性和在段与段之间的光滑性, 相邻两段曲线之间满足一定的条件. 现以两段为例说明这种情况, 假设  $P_0, P_1, P_2, P_3$  为第一段曲线上的 4 个点,  $P_4, P_5, P_6, P_7$  为第二段曲线上的 4 个点(其中  $P_4$  为该段的起点,  $P_7$  为该段的终点), 那么这两段上的点满足

$$P(u) = \begin{cases} P_0(1-u)^3 + 3P_1u(1-u)^2 + 3P_2u^2(1-u) + P_3u^3, & i=1; \\ P_3^{i-1}(1-u)^3 + 3(2P_3^{i-1} - P_2^{i-1})u(1-u)^2 + 3P_2u^2(1-u) + P_3u^3, & 1 < i < n; \\ P_3^{i-1}(1-u)^3 + 3(2P_3^{i-1} - P_2^{i-1})u(1-u)^2 + 3P_2u^2(1-u) + P_1u^3, & i=n. \end{cases} \quad (5)$$

## 2 算法描述

### 2.1 标准粒子群优化算法

粒子群算法基本思想起源于对鸟群行为的模拟. 设想一群鸟在一个特定的空间内寻找食物, 并且在这个空间里有且仅有一个食物. 假如每只鸟都不知道食物被放在什么位置, 但是知道自己现在的位置与食物之间的距离. 为了能够快速简单地找到食物, 最好的办法是在离食物最近的鸟的周围进行搜索. 粒子群优化算法就是从这种设想中受到启发而被提出, 用于解决各种优化问题<sup>[11]</sup>.

在粒子群优化算法中, 每一个潜在的解代表的是搜索范围中的一只鸟, 我们把它叫做“粒子”, 每一个粒子有一个适应度值, 这个适应度值是由优化函数决定的, 每个粒子都有一个决定他们下一步的飞行方向和距离的速度. 具体优化的算法模型如下

$$V_i^d = \omega \cdot V_i^d + c_1 \cdot \text{rand}1_i^d \cdot (pbest_i^d - X_i^d) + c_2 \cdot \text{rand}2_i^d \cdot (gbest^d - x_i^d). \quad (6)$$

$$X_i^d = X_i^d + V_i^d. \quad (7)$$

式中:  $c_1, c_2$  是学习因子;  $r_1, r_2$  是介于  $[0,1]$  之间的随机数;  $\omega$  是惯性因子.

但是标准粒子群优化算法易出现局部最优和早熟现象.

### 2.2 动态多组群粒子群优化算法

动态多组群粒子群优化算法 (Dynamic Multi-Swarm Particle Swarm Optimizer, 简称 DMS-PSO) 是利用分群的思想把整个种群均分成若干个小种群, 这些小种群在自己的范围内寻找其内部的最优解 ( $lbest$ ), 经过  $L$  (粒子的分群周期, 每隔  $L$  代粒子进行重组) 代之后, 再把所有的

$$P_3 - P_2 = P_5 - P_4, P_3 = P_4. \quad (4)$$

由以上关系可以得知: 假如一条曲线由  $n$  段 Bezier 组成, 规定起点和终点已知, 那么这段曲线将由  $2n$  个未知点, 即是  $4n$  个变量, 每一段的变量形式为  $P_{1x}^1, P_{1y}^1, P_{2x}^1, P_{2y}^1, P_{3x}^1, P_{3y}^1, P_{2x}^2, P_{2y}^2, P_{3x}^2, P_{3y}^2, \dots, P_{2x}^n, P_{2y}^n$ .

$u \in [0,1]$ , 由式子(3)和(4)可以得出每段曲线各点的参数方程如下.

小种群重新随机组合, 继续寻找自己范围内的  $lbest$ . DMS-PSO 克服了标准粒子群优化算法的缺点, 扩大了粒子寻优的离散度, 不容易陷入局部最优的情况<sup>[12]</sup>, 交叉策略增大了粒子寻找最优解的可能性<sup>[9]</sup>.

基于交叉策略的动态多组群粒子群优化算法速度和位置更新公式为

$$\begin{aligned} & \text{If } \text{rand} < 0.5 \\ & V_i^d \leftarrow \omega \cdot V_i^d + c_1 \cdot \text{rand}1_i^d \cdot (pbest_i^d - X_i^d) + c_2 \cdot \text{rand}2_i^d \cdot (lbest_k^d - X_i^d) \\ & V_i^d = \min(V_{\max}^d, \max(-V_{\max}^d, V_i^d)) \\ & X_i^d \leftarrow X_i^d + V_i^d \\ & \text{Otherwise} \\ & X_i^d \leftarrow pbest_i^d \end{aligned} \quad (8)$$

### 2.3 评价指标

机器人路径优化解决的是在机器人和障碍物之间满足的技术指标问题, 在本研究中, 考虑的惩罚包括安全性惩罚和最短路径惩罚<sup>[9-10]</sup>.

安全性惩罚描述的是机器人和障碍物之间的无碰撞问题. 假设障碍物是圆,  $D_{\text{safe}}$  表示圆的半径, 当所有障碍物与圆心之间的最短距离  $d_{\min}$  小于  $D_{\text{safe}}$  时, 就会产生惩罚, 用  $f_{\text{safe}}$  来表示安全性惩罚, 其函数定义如下:

$$d_{\min} = \min_{o \in C_{\text{obs}}} \min_{u \in [0,1]} \sqrt{(x(u) - o_x)^2 + (y(u) - o_y)^2}. \quad (9)$$

$$f_{\text{safe}} = \begin{cases} 0, & d_{\min} > D_{\text{safe}}; \\ d_{\min}, & 0 \leq d_{\min} \leq D_{\text{safe}}. \end{cases} \quad (10)$$

式中:  $x(u)$  和  $y(u)$  是每个点的坐标;  $o_x$  和  $o_y$  是障

障碍物的圆心（假设障碍物是圆）； $C_{\text{obs}}$  是障碍物的集合。

而最短路径描述的是机器人通过障碍物从起点到终点的最短距离。用  $f_{\text{len}}$  表示路径最短惩罚，其定义如下：

$$f_{\text{len}} = l = \int_0^1 \sqrt{(x'(t))^2 + (y'(t))^2} dt. \tag{11}$$

2.4 动态约束策略

在文献[9]中，采用了静态的惩罚系数来平衡  $f_{\text{safe}}$  和  $f_{\text{len}}$  的比例，但是很难选择一个合适的系数，如果该值选择过大，粒子很容易陷入局部最优，选择过小，又会使机器人和障碍物相撞。为了克服这种缺点，笔者采用新的动态约束策略。

把安全性指标  $f_{\text{safe}}$  作为一个约束，和动态临界值  $\varepsilon$  相比较，比较的结果看作总的惩罚约束， $\varepsilon$  的定义如下：

$$\varepsilon = \max(\min(D_{\text{safe}} \cdot (1 - 2 \cdot FEs / \max FEs), f_{\text{safe}}(pbest)), 0). \tag{12}$$

其中， $FEs$  是当前适应度评价次数， $\max FEs$  预先定义的最大适应度评价次数，当  $pbest$  和  $gbest$  更新时，如果：

$$\begin{cases} f_{\text{safe}}(x_i) < \varepsilon \& f_{\text{safe}}(x_j) < \varepsilon \& f_{\text{len}}(x_i) < f_{\text{len}}(x_j); \\ f_{\text{safe}}(x_i) < \varepsilon \& f_{\text{safe}}(x_j) > \varepsilon; \\ f_{\text{safe}}(x_i) > \varepsilon \& f_{\text{safe}}(x_j) > \varepsilon \& f_{\text{safe}}(x_i) < f_{\text{safe}}(x_j). \end{cases} \tag{13}$$

那么， $x_i$  就比  $x_j$  好。

3 实验设置和结果

6 种路径问题用来测试以上约束机制下的 6 种算法和 Bezier 曲线在描述路径时的特性。图 1 描述的是被测问题的最好的路径，圆代表障碍物。在实验时，用两段的 Bezier 曲线来描述路径，由于  $n$  段 Bezier 曲线需要优化点的个数是  $4n$ ，那么，本实验所需要优化的参数就是 8 个。实验中用到的 6 种算法描述如下。

- (1) 标准粒子群优化算法 (PSO)；
- (2) 标准粒子群优化算法 + 动态约束 (PSO- $\varepsilon$ )；
- (3) 标准粒子群优化算法 + 动态约束 + 交叉策略 (PSO- $\varepsilon$ +)；
- (4) 动态多群体粒子群优化算法 (DMS-PSO)；
- (5) 动态多组群粒子群优化算法 + 动态约束 (DMS-PSO- $\varepsilon$ )；
- (6) 动态多组群粒子群优化算法 + 动态约束 + 交叉策略 (DMS-PSO- $\varepsilon$ +)。

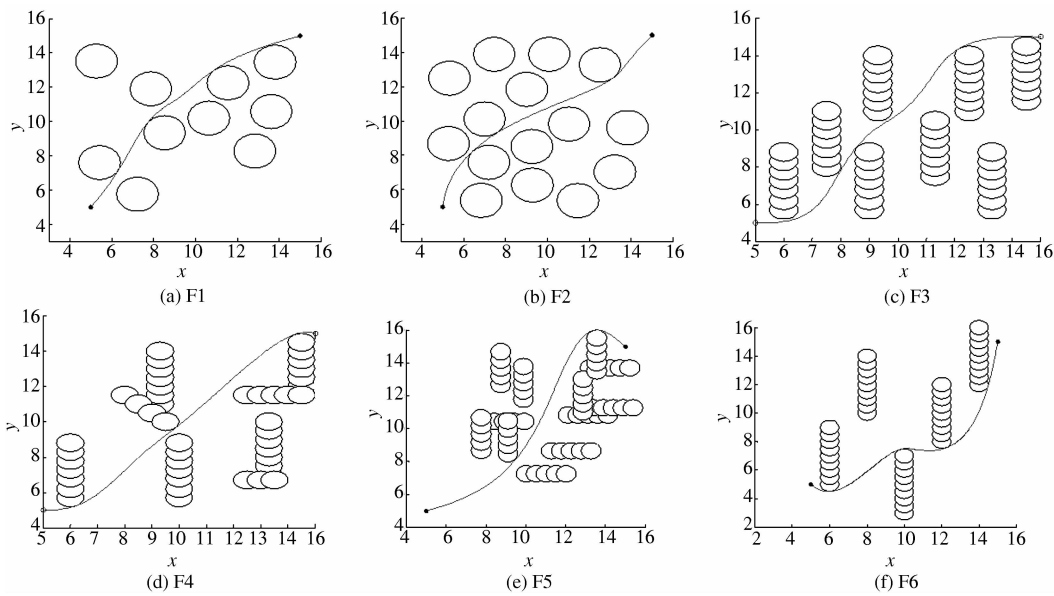


图 1 被测试问题  
Fig.1 Landscapes of the test problems

在本实验中，种群的规模设为 30 个粒子，在动态多群体粒子群优化算法中，把粒子随机平分

为 3 个小种群，那么小种群的规模是 10。对于所有的算法最大适应度评价次数为 20 000，即  $\max FEs$

$= 20\ 000$ ， $c_1 = c_2 = 1.494\ 45$ ， $\omega = 0.729$ ， $L = 5$ 。所有的问题都是基于 25 次以上的随机独立实验下完成的。表 1 是该实验的结果。

表 1 实验结果  
Tab. 1 Experiment result

问题		PSO	PSO-ε	PSO-ε +	DMS-PSO	DMS-PSO-ε	DMS-PSO-ε +
F1	平均值	15.764 0	15.141 9	14.899 5	15.038 9	15.007 3	14.846 8
	方差	1.116 7	0.266 2	0.274 9	0.028 1	0.593 2	0.254 5
	Min	14.911 8	14.767 6	14.673 1	14.910 6	14.651 1	14.651 0
	Max	18.386 4	17.150 4	16.975 9	15.479 8	16.928 4	16.915 0
F2	平均值	16.505 4	15.231 6	14.777 9	15.632 1	14.717 9	14.682 2
	方差	3.853 7	0.244 0	0.008 1	1.661 2	0.014 3	0.001 4
	Min	14.680 7	14.755 8	14.677 8	14.6676	14.625 1	14.632 2
	Max	19.407 5	16.746 2	15.091 5	18.770 9	15.073 8	14.745 0
F3	平均值	16.726 9	16.826 9	16.326 4	16.473 5	16.376 8	16.244 8
	方差	0.180 5	0.350 7	0.344 8	0.123 8	0.247 7	0.243 6
	Min	16.491 6	15.638 0	15.362 2	15.389 1	15.222 6	15.239 7
	Max	18.590 9	18.886 7	17.139 1	16.687 8	16.661 8	16.688 5
F4	平均值	16.186 3	15.465 7	15.084 5	14.859 4	14.977 5	14.746 0
	方差	1.072 0	0.563 2	0.503 9	0.184 3	0.531 1	0.150 1
	Min	14.617 8	14.902 1	14.672 0	14.616 7	14.612 8	14.613 9
	Max	17.932 0	17.593 7	16.695 7	16.602 8	16.682 0	16.599 1
F5	平均值	17.556 5	16.982 4	16.775 3	16.875 2	16.909 7	16.470 0
	方差	0.142 1	0.270 9	0.311 6	0.325 2	0.480 2	0.082 8
	Min	16.897 8	16.481 6	16.385 6	16.357 2	16.292 6	16.275 8
	Max	18.097 2	18.076 0	17.910 8	17.828 7	18.098 7	17.368 1
F6	平均值	18.165 7	17.996 5	17.483 8	17.354 9	17.614 6	17.329 2
	方差	0.736 8	0.180 0	0.484 3	0.743 0	0.909 0	0.991 4
	Min	16.741 7	17.085 7	16.826 3	16.705 8	16.705 5	16.706 9
	Max	19.372 1	18.929 8	20.390 1	19.440 7	20.014 5	20.698 4

实验结果表明:1)DMS-PSO 比 PSO 的整体结果要好;2)PSO-ε + 的性能优于 PSO 和 PSO + , DMS-PSO-ε + 的性能优于 DMS-PSO 和 DMS-PSO + ;3)DMS-PSO-ε + 在这 6 种算法中性能最好.

从以上 3 个方面的比较可以得到:DMS-PSO 具有较好的全局搜索能力,克服了标准粒子群算法易于陷入局部最优的缺点.交叉操作在路径优化过程中是非常有用的,扩大了搜索范围的离散度.动态约束中由于参数的变化性,增大了粒子找到与障碍物最安全的距离并且使整个路径最短的可能性.

4 结论

在本设计中,用 Bezier 曲线和动态多组群粒子群优化算法结合交叉策略和动态约束来实现路径优化.通过应用分段的 Bezier 曲线来描述路径,把路径问题转化为求最优问题.考虑到路径优化的本身特性,利用优化算法、交叉操作和动态约束来提高它的性能.实验结果表明利用动态约束下的交叉操作的 DMS-PSO 能够避免陷入局部最优和找到和障碍物保持安全距离的最短路线.

参考文献:

[1] 刘华军,杨静宇,陆建峰,等. 移动机器人运动规划研究综述[J]. 中国工程科学, 2006, 8(1): 85 - 94.

[2] QU Yao-hong, PAN Q, YAN Jian-guo. Flight path planning of UAV based on heuristically search and genetic algorithms[C]// 31st Annual Conference of IEEE Industrial Electronics Society. Xi'an:Industrial Electronics Society, 2005: 45 - 49.

[3] JU Ming-yi, CHENG Ching-wen. Smooth path planning using genetic algorithms[C]// the 8th World Congress on Intelligent Control and Automation. Tainan: Intelligent Control and Automation, 2011: 1103 - 1107.

[4] MO Hong-wei, Li Zhen-zhen. Bio-geography based differential evolution for robot path planning[C]// IEEE Internation. Shengyang: Information and Automation, 2012: 1 - 6.

[5] CHAKRABORTY J, KONAR A, CHAKRABORTY U K. Distributed cooperative multi-robot path planning using differential evolution[C]// IEEE Congress on Evolutionary Computation. Hongkong: Springer Verlag, 2008: 718 - 725.

[6] 纪震, 廖惠连, 吴青华. 粒子群算法与应用[M]. 北京: 科学出版社, 2009.

- [7] HUANG X Y, WEN Z. Vehicle routing optimization based on improved PSO[J]. Natural Science Journal of Xiangtan University, 2009, 38(4): 166 – 171.
- [8] 杭后俊, 李汪根. 有理三次 Bezier 曲线表示圆弧的一种实用方法[J]. 计算机应用与工程, 2011, 48(6): 185 – 189.
- [9] LIANG Jing, SONG Hui, QU Bo-yang. Path planning based on dynamic multi-swarm particle swarm optimizer with crossover[C]// 8th International Conference on Intelligent Computing Theories and Applications, ICIC 2012. Huangshan: Springer Verlag, 2012: 159 – 166.
- [10] 朱东伟, 毛晓波, 陈铁军. 基于改进粒子群三次 Bezier 曲线优化的路径规划[J]. 计算机应用研究, 2012, 29(5), 1710 – 1712.
- [11] 黄日朋. 有理 q-Bernstein-Bezier 曲线的构造及其应用[J]. 计算机应用, 2010, 30(5): 1359 – 1362.
- [12] 刘会刚, 秦国亮. 一种基于 Bezier 曲线的军事箭标实现[J]. 四川兵工学报, 2009, 30(2): 67 – 68.

## Research on Path Planning Problems Based on Improved Particle Swarm Optimizer

LIANG Jing<sup>1</sup>, SONG Hui<sup>1</sup>, QU Bo-yang<sup>2</sup>, MAO Xiao-bo<sup>1</sup>

(1. School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China; 2. School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China)

**Abstract:** This paper describes that dynamic constraint mechanism and improved Particle Swarm Optimizer (Dynamic Multi-Swarm Particle Swarm Optimizer with Crossover) are combined to solve path planning problems combining with Bezier curves. Different algorithms are used to test characteristics of the curves in the path planning process and Bezier curve is used to describe the path. The results show that the DMS-PSO overcomes the phenomenon of precocious and easy convergence compared with standard PSO, and constraint mechanism improves performance of the optimization algorithm which is combined with crossover, can deal with the constraints flexibly and overcomes the shortcomings of static penalty function.

**Key words:** constraint mechanism; dynamic multi-swarm particle swarm optimizer; bezier curves

(上接第 33 页)

## Delay Compensation of Network Control Systems Based on JGPC

LIU Yan-hong, LUO Yong-ping, LIU Yan-xing

(School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** The uncertain factors such as the network delays usually deteriorate the control performance and stability of networked control systems (NCS), even result in instability. In this paper, we propose a novel delay compensation method to enhance the stability of NCS. First, an auto regressive model for the network time delays is established and the parameter self-tuning LMS(Least mean square) algorithm is used to predict the network delay online. Then, a improved JGPC algorithm is proposed for the compensation of the networked time delay. This algorithm can get a better system performance than GPC. Simulation results show that the proposed method can effectively compensate the network delay and improve the dynamic performance of the NCS.

**Key words:** Network control systems; Auto regressive model; JGPC algorithm; Random delay prediction; time delay compensation