Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 623156, 15 pages
http://dx.doi.org/10.1155/2014/623156

Research Article

Comparison of Three Different Curves Used in Path Planning Problems Based on Particle Swarm Optimizer

J. J. Liang,[1] H. Song,[1] B. Y. Qu,[2] and Z. F. Liu[2]

[1] *School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China*
[2] *School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China*

Correspondence should be addressed to B. Y. Qu; qby1984@hotmail.com

Received 23 December 2013; Accepted 11 March 2014; Published 17 April 2014

Academic Editor: Fang Zong

In path planning problems, the most important task is to find a suitable collision-free path which satisfies some certain criteria (the shortest path length, security, feasibility, smoothness, and so on), so defining a suitable curve to describe path is essential. Three different commonly used curves are compared and discussed based on their performance on solving a set of path planning problems. Dynamic multiswarm particle swarm optimizer is employed to optimize the necessary parameters for these curves. The results show that Bezier curve is the most suitable curve for producing path for the certain path planning problems discussed in this paper. Safety criterion is considered as a constrained condition. A new constraint handling method is proposed and compared with other two constraint handling methods. The results show that the new method has a better characteristic to improve the performance of algorithm.

1. Introduction

The mobile robot path planning is an important research field of robotics. One of the most important tasks to realize navigation and control of the robots is path planning. In an environment with obstacles, the aim of path planning is to find a suitable collision-free path, which satisfies some certain optimal criteria (such as the shortest path length, security, and feasibility), for a mobile robot to move from a start position to a target position. Most researches have focused on finding the shortest path, the minimum-time path, or the safest path, but the generated paths may be discontinued. Smoothness of the path is essential for the navigation of mobile robots, because nonsmooth motions have effect on slip [1–5]. So finding a suitable curve to describe the path is a very important task in path planning problems.

In [6], η^3 curve with parallel variable-length genetic algorithm has been used to realize path planning problems. Ferguson is another curve which is used commonly with particle swarm optimizer in [7, 8] and particle filter in [9]. Bezier curve is one of the most common curves which is combined with de Casteljau algorithm in [10], genetic algorithm in [11], and particle swarm optimizer in [12] in recent years. These curves which are used to generated curve have their own specific characteristics and requirements, but which one is the best has not been discussed.

The approaches of traditional path planning are artificial potential field [13], neural network [14], D^* algorithm [15], and so on. With the appearance and development of evolutionary computation algorithms, many nature inspired optimization computing methods have been proposed to solve path planning problems, including genetic algorithms [16, 17] and differential evolution [18, 19].

Particle swarm optimization (PSO) which was proposed by Eberhart and Kennedy in 1995 [20, 21] is based on swarm intelligence. It has been applied to many areas successfully such as artificial neural network training [22], path planning problems [23, 24], multiworking modes product-color planning [25], and robust control of 3RPS parallel manipulators [26], for its easiness to use, robustness, and strong ability of global optimization. An improved particle swarm optimizer is applied to solve the path planning problems in this paper.

The robotic path planning problem is to find a suitable path for a mobile robot to move from the start location to the target location, which satisfies some optimum criteria in an environment full of obstacles. In this paper, we define the security and the shortest path as the optimum criteria. The security means no collision between the robot and all the obstacles and the shortest path describes the distance the robot moves from the initial point to the end point [5, 27]. The three times Bezier curve, Ferguson curve, and $\eta 3$ curve are used to generate the path and their performances are compared. With these curve generating methods the path planning can be transformed into optimizing a few limited anchor points which are used to form the path. Then dynamic multiswarm particle swarm optimizer (DMS-PSO) is employed to optimize the locations of these anchor points.

In the previous work, two different constraint handling methods, dynamic threshold ε and dynamic balance function, have been tested [28]. Based on the analysis on the weakness of these two constraint handling methods, a novel constraint handling method "dynamic compared Δ" is proposed to be incorporated intothe algorithm to improve the search efficiency. The experimental result shows that this new method has a better performance on most path planning problems discussed in this paper.

The rest of this paper is organized as follows. The characteristics of the three curves are introduced in detail in Section 2. Section 3 gives a brief introduction on the dynamic multiswarm particle swarm optimizer and the constraint handling mechanisms employed in this work. The experimental setup and the results are presented in Section 4. Conclusions and future work are given in Section 5.

2. Description of Curves

2.1. The Definition and Properties of Bezier Curve. Bezier curve was proposed by the French engineer Pierre Bezier, who used Bezier curve to design for the body of the car in 1962 [29]. In recent years, Bezier curve was applied to various occasions for its advantages on describing both straight line and curve.

A Bezier curve of degree n is a parametric curve composed of Bernstein basis polynomials of degree n [22]:

$$P(t) = \sum_{i}^{N} P_i B_{i,N}(t), \quad t \in [0,1]. \tag{1}$$

In this equation, basis function $\{B_{i,T}(t)\}$ is a famous n times Bernstein polynomial [23], which is defined as

$$B_{i,N}(t) = C_N{}^i t^k (1-t)^{N-i} \quad (i = 0, 1, \ldots, n). \tag{2}$$

The parameter equation of every point for three times Bezier curve could be generated by formulas (1) and (2) as follows:

$$P(t) = P_0(1-t)^3 + 3P_1 t(1-t)^2 + 3P_2 t^2(1-t) + P_3 t^3, \tag{3}$$

where t is in the range of $[0,1]$. Bezier curve starts at $t = 0$ and ends at $t = 1$.

The properties of Bezier curves [22] can be described as follows.

(1) Bezier curves start at the start point and stop at the end point.

(2) First derivatives of the start point and the end point are only related to the two near control points and in the same direction of the line of the two points.

The calculation formula

$$P'(0) = 3 \times (P_1 - P_0), \qquad P'(n) = 3 \times (P_n - P_{n-1}). \tag{4}$$

A complex first-order continuous Bezier curve can be formed by connecting several segments of low-order Bezier curves. Each segment has four control points. Assuming we have two segments, $P_1(P_{10}, P_{11}, P_{12}, P_{13})$ and $P_2(P_{20}, P_{21}, P_{22}, P_{23})$, in order to ensure the continuousness of the curve after connection, the following equation should be satisfied:

$$P_{13} - P_{12} = P_{21} - P_{20}, \quad P_{13} = P_{20}. \tag{5}$$

Therefore, in order to meet the property of first-order continuous when using n segments of Bezier curves to describe a path, $2n$ points ($4n$ parameters) are needed. The path can be generated using the following:

$$P(t) = \begin{cases} P_0(1-t)^3 + 3P_1^i t(1-t)^2 \\ \quad + 3P_2^i t^2(1-t) + P_3^i t^3, & i = 1 \\ P_3^{i-1}(1-t)^3 + 3\left(2P_3^{i-1} - P_2^{i-1}\right) t(1-t)^2 \\ \quad + 3P_2^i t^2(1-t) + P_3^i t^3, & 1 < i < n \\ P_3^{i-1}(1-t)^3 + 3\left(2P_3^{i-1} - P_2^{i-1}\right) t(1-t)^2 \\ \quad + 3P_2^i t^2(1-t) + P_1 t^3, & i = n, \end{cases}$$

$$P(t) = [x(t), y(t)]^T, \tag{6}$$

where P_0 represents the start point while P_1 stands for the end point. When t changes in the interval $(0, 1)$, we can get a cubic Bezier curve of segment i. These n segments of cubic Bezier curve constitute the entire path of the curve.

2.2. The Properties of Ferguson Curve. Ferguson curve is also a famous curve which has many excellent properties and plays an important role in the shape description.

Since Ferguson curve is smooth and easy to implement, it is also often used to describe the path in path planning problems. One segment of Ferguson curve can be defined as follows:

$$C(t) = [x(t), y(t)]^T,$$

$$k: C(t) = P_0 F_1(t) + P_1 F_2(t) + P_0{}' F_3(t) + P_1{}' F_4(t), \quad t \in [0,1]. \tag{7}$$

P_0 and P_1 are the start point and the end point of the curve, respectively, and P_i and $P_i{}'$are control points which control the shape of the curve. F_i represents Ferguson polynomial and is defined as follows:

$$\begin{aligned} F_1(t) &= 2t^3 - 3t^2 + 1, \\ F_2(t) &= -2t^3 + 3t^2, \\ F_3(t) &= t^3 - 2t^2 + t, \\ F_4(t) &= t^3 - t^2. \end{aligned} \quad (8)$$

Assuming that a curve consists of n segments of Ferguson curves, these Ferguson curves should satisfy certain requirements. Two-segment Ferguson curves are taken as an example to illustrate the requirements they should satisfy to ensure smooth connection between these two segments. Suppose that the other Ferguson curve is described as follows:

$$\overline{k} : C(t) = \overline{P_0} F_1(t) + \overline{P_1} F_2(t) + \overline{P_0}' F_3(t) + \overline{P_1}' F_4(t). \quad (9)$$

In order to make the path smooth, the curve which is used to describe the path must be first-order continuous, and then k and $\overline{k}$ must satisfy

$$P_1 = \overline{P_0}, \qquad P_1' = \overline{P_0}'. \quad (10)$$

The same as Bezier curve, if n segments Ferguson curves are used to generate the path, there will be $2n$ control points which means that $4n$ variables are to be optimized.

2.3. The Properties of η3 Curve. The same as the Bezier curve and Ferguson curve, η3 curve which is used in path planning problems for its good properties in describing lines, arcs, and clothoid is also a widely used curve. As shown in [8], first we set two arbitrary combinations as follows:

$$\begin{aligned} \Omega_A &= \left[x_A, y_A, \theta_A, k_A, \dot{k}_A\right]^T, \\ \Omega_B &= \left[x_B, y_B, \theta_B, k_B, \dot{k}_B\right]^T. \end{aligned} \quad (11)$$

Here x, y, and θ represent coordinates and direction, respectively, while k and $\dot{k}$ denote curvature and curvature derivative of the path at one point. A 7th-order polynomial of η3 curve can be formed by the following formulas:

$$\begin{aligned} P(t) &= [x(t), y(t)]^T, \\ P_x(t) &= \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 \\ &\quad + \alpha_4 t^4 + \alpha_5 t^5 + \alpha_6 t^6 + \alpha_7 t^7; \quad t \in [0,1], \\ P_y(t) &= \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \\ &\quad + \beta_4 t^4 + \beta_5 t^5 + \beta_6 t^6 + \beta_7 t^7. \end{aligned} \quad (12)$$

In order to ensure the smoothness of the curve after connecting, the following formula should be satisfied:

$$\begin{aligned} \Omega_A &= \left[x_A, y_A, \theta_A, k_A = 0, \dot{k}_A = 0\right]^T, \\ \Omega_B &= \left[x_B, y_B, \theta_B, k_B = 0, \dot{k}_B = 0\right]^T. \end{aligned} \quad (13)$$

In addition, the polynomial has extra six degrees of freedom. In order to reduce the calculation of degrees of freedom, we use Euclidean distance of two terminal configurations to represent some variables of vectors while the other variables of vectors are set to 0. Therefore, x coordinate coefficients used to generate the curve can be obtained according to the above formulas as follows:

$$\begin{aligned} \alpha_0 &= x_{A;} \\ \alpha_1 &= \left\|(x_A - x_B, y_A - y_B)\right\| \cos\theta_A; \\ \gamma_1 &= \left\|(x_A - x_B, y_A - y_B)\right\| \cos\theta_B; \\ \alpha_2 &= 0; \qquad \gamma_2 = 0; \\ \alpha_3 &= 0; \qquad \gamma_3 = 0; \\ \alpha_4 &= 35(x_B - x_A) - 20\alpha_1 - 10\alpha_2 \\ &\quad -4\alpha_3 - 15\gamma_1 + 5\gamma_2 - \gamma_3; \\ \alpha_5 &= -84(x_B - x_A) + 45\alpha_1 + 20\alpha_2 \\ &\quad +6\alpha_3 + 39\gamma_1 - 14\gamma_2 + 3\gamma_3; \\ \alpha_6 &= 70(x_B - x_A) - 36\alpha_1 - 15\alpha_2 \\ &\quad -4\alpha_3 - 34\gamma_1 + 13\gamma_2 - 3\gamma_3; \\ \alpha_7 &= -20(x_B - x_A) + 10\alpha_1 + 4\alpha_2 \\ &\quad +\alpha_3 + 10\gamma_1 - 4\gamma_2 + \gamma_3. \end{aligned} \quad (14)$$

y coordinate coefficients can be obtained by changing $\cos\theta$ into $\sin\theta$. When m segments of η3 curve are used to describe the path, $m + 1$ control points are needed. However, the start point and the end point are known in the path planning problems discussed in this paper; thus the number of the control points which are needed to be optimized is $m - 1$. In other words, there are $2(m - 1)$ variables to be optimized. Except the location of the control points, the tangent directions of each control point for the path are also controllable, so there are other $m + 1$ points to be optimized for m segments. Therefore, there are $3m - 1$ variables for an η3 curve with m segments.

So from the above information, we could know that $x(t)$ and $y(t)$ are the coordinates of every point which should be optimized. What is more, these three curves are smooth and suitable for path planning for robots. If the same number of segments is needed to generate the path, there are $4n$ parameters to be optimized for the first two curves and $3n - 1$ variables for η3 curve.

3. Brief Introduction about Algorithm and Constraint Handling Mechanisms

Particle swarm optimizer is an intelligent evolutionary algorithm which is constructed by mimicking the birds' behavior of preying food [21]. The basic idea of particle swarm optimization algorithm is to find the optimal solution through collaboration among groups and information sharing among individuals.

The idea of dynamic multiswarm based on periodically changed neighborhood structure was firstly proposed by Liang and Suganthan in 2005 [27]. The good information obtained by each subswarm is exchanged among the subswarms and the diversity of the population is increased simultaneously by using the dynamic changing topology. Considering its good performance on complex optimization problems, the dynamic multiswarm particle swarm optimizer (DMS-PSO) is employed to solve the path planning problems in this paper.

The position updating equations of DMS-PSO with crossover can be described as follows [28]:

$$\begin{aligned}
&\text{If rand} < 0.5\\
&\quad V_i^d \longleftarrow \omega * V_i^d\\
&\qquad\quad + c_1 * \text{rand}1_i^d * \left(p\text{best}_i^d - X_i^d\right)\\
&\qquad\quad + c_2 * \text{rand}2_i^d * \left(l\text{best}_k^d - X_i^d\right),\\
&\quad V_i^d \longleftarrow \min\left(V_{\max}^d, \max\left(-V_{\max}^d, V_i^d\right)\right),\\
&\quad X_i^d \longleftarrow X_i^d + V_i^d.\\
&\text{Otherwise}\\
&\quad X_i^d \longleftarrow p\text{best}_i^d,
\end{aligned} \tag{15}$$

where X_i^d represents the position of the ith particle in dimension d. V_i^d represents the velocity of the ith particle in dimension d. $p\text{best}_i^d$ is the best position in history of the ith particle in dimension d. $V_{\max}^d$ is the predefined maximum value in dimension d.

DMS-PSO was firstly used in path planning problems in [5], where the path planning problem has been solved by the following means.

(1) Security and the shortest path criteria are combined into a punitive function with a constant to balance them.

(2) Path length is regarded as the objective function, while the security criterion is regarded as a constraint for the shortest path.

In these path planning problems, a series of circles are used to represent obstacles, and the safe distance between path and obstacles is set as D_{safe} (which is radius of the circle). The minimum distance between path and obstacles is $d_{\min}$. If and only if $d_{\min}$ is larger than D_{safe}, the path could be defined as secure. Otherwise, penalty will be imposed. f_{safe} is treated as security penalty function as follows:

$$\begin{aligned}
f_{\text{safe}} &= \begin{cases} 0, & d_{\min} > D_{\text{safe}} \\ d_{\min}, & 0 \le d_{\min} \le D_{\text{safe}}, \end{cases}\\
f_{\text{len}} &= \min_{o\in C_{\text{obs}}} \min_{t\in[0,1]} \sqrt{\left(x(t) - o_x\right)^2 + \left(y(t) - o_y\right)^2},
\end{aligned} \tag{16}$$

where o_x and o_y are the centre of the obstacles and C_{obs} is a collection of all the obstacles in the space. The total cost is calculated as

$$f(x) = f_{\text{len}}(x) + \alpha f_{\text{safe}}(x), \quad \alpha = 1000. \tag{17}$$

α is a constant value, which is used to balance the proportion of f_{safe} and f_{len}. A large α will lead to local optimum easily while a small α will make a collision with obstacles, so choosing a suitable value is difficult.

In order to overcome this drawback, two constraint handling methods have been used to improve the above static constrain in [27]; FEs is the current fitness evaluation times and MaxFEs is the predefined max fitness evaluation times.

Constraint Handling Method 1 (Dynamic Threshold ε). One has

$$\begin{aligned}
\varepsilon = \max\Biggl(\min\left(D_{\text{safe}}\left(1 - 2 * \frac{\text{FEs}}{\text{MaxFEs}}\right),\right.\\
\left.\text{mean}\left(f_{\text{safe}}\right)\right), 0\Biggr).
\end{aligned} \tag{18}$$

Figure 1 describes the dynamic changing process of ε. x is considered to be better than y if

$$\begin{aligned}
&(1)\ f_{\text{safe}}(x) <= \varepsilon \text{ and } f_{\text{safe}}(y) <= \varepsilon \text{ and } f_{\text{len}}(x) < f_{\text{len}}(y)\\
&(2)\ f_{\text{safe}}(x) <= \varepsilon \text{ and } f_{\text{safe}}(y) > \varepsilon\\
&(3)\ \ \varepsilon < f_{\text{safe}}(x) < f_{\text{safe}}(y).
\end{aligned} \tag{19}$$

We observed that the potential good solutions which locate near to the global optimum but do not satisfy the current constraint will be replaced. In this way, the useful information obtained along the search process may be lost.

Constraint Handling Method 2 (Dynamic Balance Function). This method is similar to the previous static penalty function (17) except that the balance factor α is gradually increasing. The dynamic α is defined as follows:

$$\alpha = 10000 * \max\left(\max\left(\text{mean}\left(f_{\text{safe}}\right), 0.2\right), \frac{\text{FEs}}{\text{MaxFEs}}\right)^2. \tag{20}$$

x is considered to be better than y if

$$F(x) < F(y). \tag{21}$$

The dynamic changing process of α with the FEs is presented in Figure 2. α is changed with the FEs and its rising

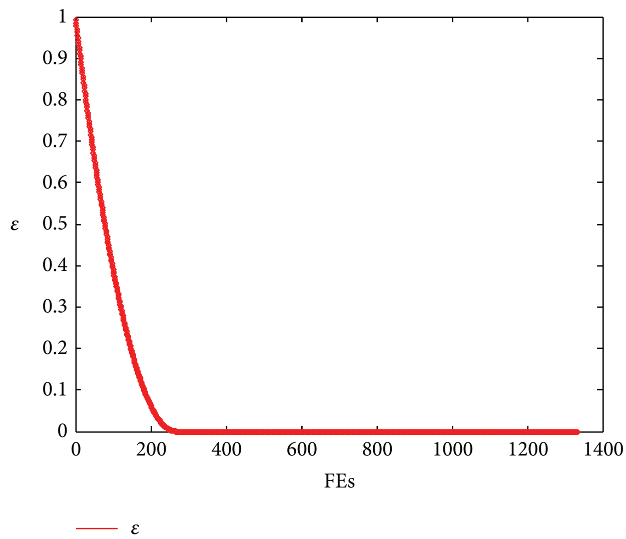


Figure 1: Dynamic change of ε.

trend is gentle and continuous. It is better than the static penalty function, but it is still difficult to control and select a suitable value to avoid losing some potential solutions.

Constraint Handling Method 3 (Dynamic Compared Function, Described with Δ). Constraint handling methods 1 and 2 improved the feature of algorithm which has been discussed in previous work, but they still have some drawbacks: the first method may lose some potential solutions which have been abandoned for dissatisfying constraint condition in current generation while the second method may not find the best solution for its gentle change. On the other hand, the first one has large space while the second has small space to be improved. A new constraint handling method which has a larger constraint range is introduced to overcome the shortage of the first constraint handling method. It is expected to have a better ability of global search. In this new method, two different ε, ε_1 and ε_2, are employed to judge if a solution satisfies the constraint. And the mean value of f_{safe} values of current particles is used to control the value of ε_1:

$$\begin{aligned}\varepsilon_1 &= \min\left(\max\left(D_{\text{safe}} * \left(\max\left(1 - \frac{\text{FEs}}{(0.2 * \text{MaxFEs})}, 0\right)\right)^2, \text{mean}\left(f_{\text{safe}}\right)\right),\right. \\ &\quad \left. D_{\text{safe}} * \left(\max\left(1 - \frac{\text{FEs}}{(0.9 * \text{MaxFEs})}, 0\right)\right)^{0.5}\right); \\ \varepsilon_2 &= D_{\text{safe}} * \left(\max\left(1 - \frac{FEs}{(0.5 * \text{MaxFEs})}, 0\right)\right)^2.\end{aligned} \tag{22}$$

Figure 3 provides the possible range of ε_1 and ε_2. ε is the effective range of ε_1 and ε_2. The new constraint method is generated by the comparison of ε_1 and ε_2. For any two solutions x and y to be compared, the following comparison criterion is used:

$$tp = \begin{cases}(1)\ f_{\text{safe}}(x) < \varepsilon_2, f_{\text{safe}}(y) <= \varepsilon_2, f_{\text{len}}(x) < f_{\text{len}}(y) \\ (2)\ f_{\text{safe}}(x) < \varepsilon_2, f_{\text{safe}}(y) > \varepsilon_2 \\ (3)\ \varepsilon_2 <= f_{\text{safe}}(x) < f_{\text{safe}}(y), f_{\text{safe}}(x) <= \varepsilon_1 \\ (4)\ f_{\text{safe}}(x) > \varepsilon_2, f_{\text{safe}}(y) > \varepsilon_2, f_{\text{len}}(x) < f_{\text{len}}(y).\end{cases} \tag{23}$$

If tp is equal to 1, x is considered to be better than y. This constraint handling method overcomes the shortage of the dynamic threshold ε which may be trapped into local optimum and improves the exploration property of the algorithm.

4. Experimental Setup and Results

From the previous test in the path planning problems, some conclusions have been made that DMS-PSO with crossover outperforms DMS-PSO and PSO with crossover performs better than PSO. So in this task, DMS-PSO with crossover and PSO with crossover are combined with the above three constraint handling methods which are designed to test the characteristics of the curves in path planning problems.

(1) The following six algorithms are used to test characteristics of each curve in path planning problems:

- (i) PSO-ε: basic particle swarm optimizer with dynamic ε and crossover operator;
- (ii) PSO-DP: basic particle swarm optimizer with dynamic balance and crossover operator;
- (iii) PSO-Δ: basic particle swarm optimizer with dynamic compared Δ and crossover operator;
- (iv) DMS-PSO-ε: dynamic multiswarm particle swarm optimizer with dynamic ε and crossover operator;
- (v) DMS-PSO-DP: dynamic multiswarm particle swarm optimizer with dynamic balance and crossover operator;
- (vi) DMS-PSO-Δ: dynamic multiswarm particle swarm optimizer with dynamic compared Δ and crossover operator.

(2) Some parameters settings during the experiment are as follows:

MaxFEs (the max fitness evaluation): 40000;

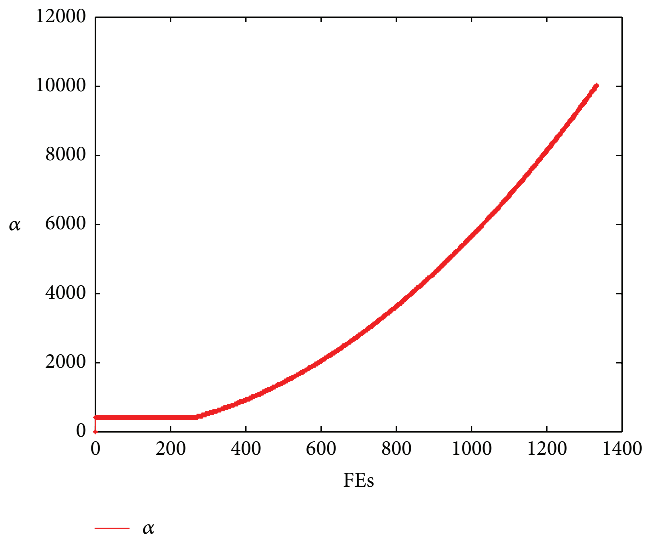


Figure 2: Dynamic change of α.

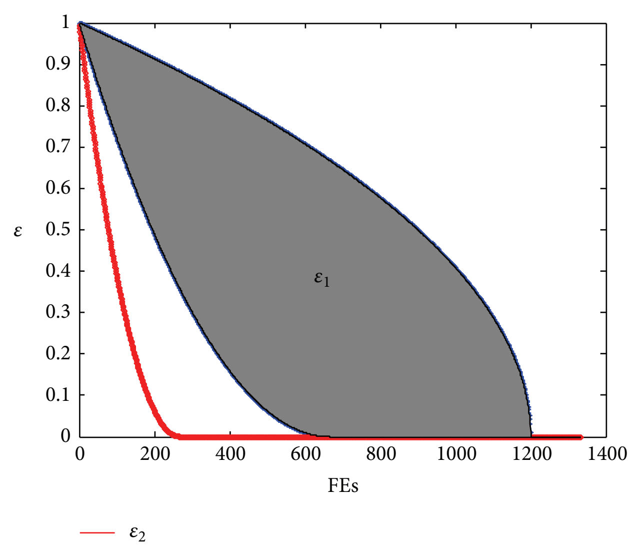


Figure 3: Dynamic change of ε_1 and ε_2 (plot in condition of $R = 1$ which means the radius of obstacles is 1 and the maximum ε is 1).

independent runs for every algorithm: 25;
population size of PSO: 30;
number of subswarms in DMS-PSO: 10;
particles in each subswarm in DMS-PSO: 3.

(3) The settings of every curve and the parameters need to be optimized.

In this task, three different curves are used to generate path. The segment of each curve and parameters needed to be optimized are set in detail as follows. Generally speaking, more points make the path more smooth and complex, while fewer points take less time in optimization.

(a) The first case is that the segments of every curve are uniform ($n = 2$), so there are 8 parameters to be optimized in Bezier curve and Ferguson curve while 5 parameters should be optimized in $\eta 3$ curve.

(b) The second case is that the optimized parameters of every curve are equal (8 parameters for all curves), so three segments of $\eta 3$ curve are used to describe the path.

4.1. Comparison of Best Satisfied Paths for Each Problem. Eight artificial designed path planning problems which have different properties are used to test characteristics of Bezier curve, Ferguson curve, and $\eta 3$ curve. In order to show how the robot moves in an environment full of obstacles, the following landscapes with the best path of these tested problems are plotted in Figure 4. The yellow circles describe the dangerous distance around the obstacles. What is more, A represents the start point while B stands for the end point.

These eight path planning problems can be classified into two classes. F1, F2, F3, and F4 are simple problems which have less local optima and are easier to find the shortest path that satisfies the safety criterion. F5, F6, F7, and F8 can be classified into complex problems which have more local optima and make the algorithms be easily trapped into the local optima.

4.2. Comparison Results of the Different Curves. Nonparametric statistical method t-test is used to evaluate the difference between two algorithms. For each problem, the results of the best algorithm which obtains the best average value in the 25 independent runs are compared with those of other algorithms by t-test method. $h = 1$ indicates a rejection of the null hypothesis at the 5% significance level. $h = 0$ indicates a failure to reject the null hypothesis at the 5% significance level.

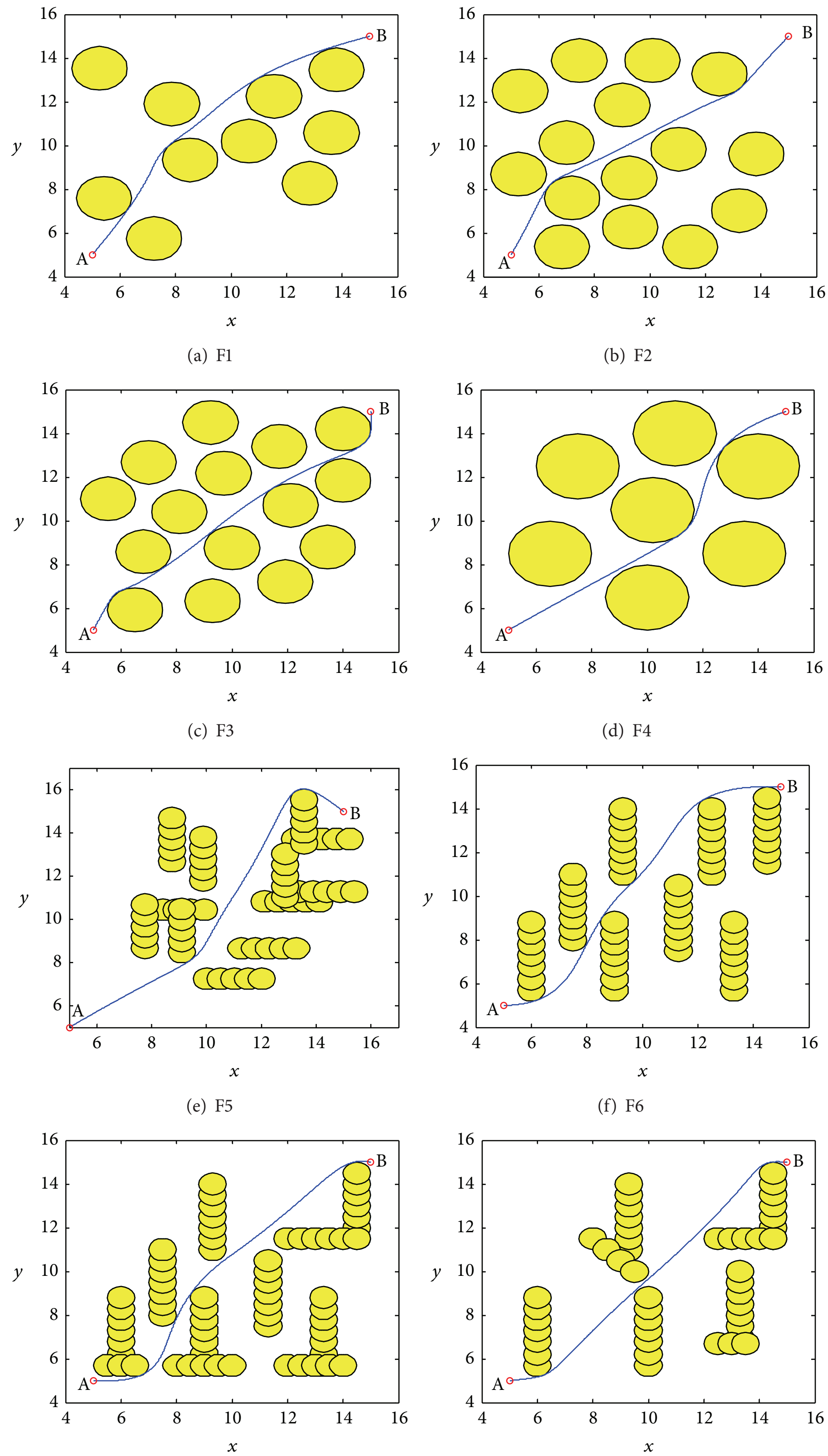
Case 1. Two segments for all curves are used to describe the path, so there are eight points for Bezier curve and Ferguson curve to be optimized while five parameters are needed for $\eta 3$ curve. The experiment results are listed in Tables 1 to 3.

Some conclusions could be drawn from Table 1 as follows.

(1) DMS-PSO outperforms PSO in all constraint handling methods correspondingly, which shows that DMS-PSO has better global search ability.

(2) The result of t-test 2 shows that there is no obvious difference between these two algorithms, so this phenomenon is regarded as these two algorithms have the similar performance on these problems. But DMS-PSO-Δ performs better on problems F1, F2, F4, F5, and F6 while DMS-PSO-ε outperforms on F4, F6, and F7 on average.

(3) Compared with the best solutions obtained by DMS-PSO-ε and DMS-PSO-Δ, the distribution of optimal solutions of DMS-PSO-DP is significantly different on problems F2, F3, F4, F5, F6, and F7 which could be seen from the results of t-test 2.



(a) F1 (b) F2 (c) F3 (d) F4 (e) F5 (f) F6 (g) F7 (h) F8

Figure 4: Landscapes of the test problems.

Table 1: Result of Bezier curve.

Problems		PSO-ε	PSO-DP	PSO-Δ	DMS-PSO-ε	DMS-PSO-DP	DMS-PSO-Δ
F1	Mean	14.9058	15.1830	14.8969	14.9294	14.7792	**14.7559**
	Std.	0.2910	0.4454	0.0282	0.4494	0.0500	0.0087
	Min	14.6726	14.6638	14.6969	14.6547	14.6566	14.6626
	Max	16.9975	16.9843	15.2326	17.1550	15.7071	15.0132
	h	**0**	**1**	**1**	**0**	**0**	—
F2	Mean	14.7260	15.6209	15.1451	14.6492	14.7087	**14.6486**
	Std.	0.0034	1.8989	0.1415	0.00039	0.0034	0.0002
	Min	14.6533	14.6889	14.8150	14.6251	14.6274	14.6242
	Max	14.8920	19.4449	16.2972	14.7009	14.8882	16.3361
	h	**0**	**1**	**1**	**0**	**1**	—
F3	Mean	15.9481	16.7858	16.5723	**14.9475**	16.3286	14.9701
	Std.	1.1006	1.1209	0.9196	0.2976	0.4483	0.1306
	Min	14.8502	15.7943	15.1810	14.7240	15.6394	14.7290
	Max	18.8319	18.7404	19.0984	17.2080	17.2206	15.7963
	h	**1**	**1**	**1**	—	**1**	**0**
F4	Mean	14.8243	15.9185	15.4251	14.7366	15.0635	**14.7363**
	Std.	0.0407	1.5323	1.2592	$8.586E-5$	0.2979	$1.451E-5$
	Min	14.7430	14.7225	14.7851	14.7150	14.7151	14.7313
	Max	15.7812	18.1838	19.8821	14.7713	17.3141	14.7455
	h	**1**	**1**	**1**	**0**	**1**	—
F5	Mean	16.5000	16.6254	16.4928	16.3466	16.3800	**16.3361**
	Std.	0.0544	0.1889	0.0048	0.0007	0.0029	0.0012
	Min	16.3327	16.3102	16.3666	16.2868	16.3246	16.2827
	Max	17.3413	17.9407	16.6432	16.4001	16.5518	16.4123
	h	**0**	**1**	**1**	**0**	**1**	—
F6	Mean	16.2485	16.4073	16.2432	15.3892	16.1031	**15.2860**
	Std.	0.3219	0.2167	0.2125	0.2282	0.3366	0.0103
	Min	15.3576	15.2832	15.5306	15.2319	15.2461	15.2400
	Max	16.6793	16.6649	16.8466	17.2616	16.6399	15.7535
	h	**1**	**1**	**1**	**0**	**1**	—
F7	Mean	16.2040	16.4125	16.4490	**15.1967**	15.9038	15.2458
	Std.	0.4474	0.3349	0.2576	0.1794	0.5311	0.2027
	Min	15.1081	15.1079	15.1656	15.0355	15.0516	15.0504
	Max	16.6948	16.8264	16.9785	16.6090	16.6292	16.9341
	h	**1**	**1**	**1**	—	**1**	**0**
F8	Mean	15.0239	15.2377	14.8827	**14.6644**	14.6787	14.6699
	Std.	0.5322	0.6693	0.0905	0.0003	0.0011	0.0003
	Min	14.6303	14.6327	14.6697	14.6291	14.6262	14.6296
	Max	16.6699	16.6459	16.2430	14.7046	14.7831	14.7093
	h	**1**	**1**	**1**	—	**0**	**0**

These three points show that although DMS-PSO-Δ and DMS-PSO-ε have the same characteristic on t-test 2, DMS-PSO-Δ overcomes the drawback of DMS-PSO-ε which is easy to be trapped into local optimum on average. Compared with other algorithms, the feature of DMS-PSO-Δ stands out on path planning problems where Bezier curve is used to generate path. The result also tells that Bezier curve is suitable on path planning problems for its stable feature when we employ evolutionary algorithm to optimize its parameters.

Table 2 gives us the following information.

(1) DMS-PSO has a better global search ability compared with PSO on the whole.

(2) All best solutions about Ferguson curve spread in DMS-PSO-ε and DMS-PSO-Δ, while the best results of problems F1, F6, F7, and F8 accept DMS-PSO-DP in the distribution of optimal solutions for 25 independent runs. The result of problem F6 has no difference with DMS-PSO-DP, DMS-PSO-ε, and DMS-PSO-Δ on the distribution of 25 independent

Table 2: Result of Ferguson curve.

Problems		PSO-ε	PSO-DP	PSO-Δ	DMS-PSO-ε	DMS-PSO-DP	DMS-PSO-Δ
F1	Mean	15.4745	14.9230	15.8539	**14.6720**	14.7393	15.0440
	Std.	1.0221	0.3217	0.8098	$3.065E-5$	0.1175	0.1913
	Min	14.6617	14.6565	14.7062	14.6626	14.6532	14.6601
	Max	17.3684	16.3953	17.5340	14.6830	16.3743	16.3967
	h	**1**	**1**	**1**	—	**0**	**1**
F2	Mean	14.6541	14.8765	14.7016	14.6399	14.6550	**14.6395**
	Std.	$1.507E-4$	0.9416	0.0013	$2.085E-5$	$2.167E-4$	$6.640E-5$
	Min	14.6335	14.6344	14.6415	14.6320	14.6309	14.6245
	Max	14.6787	19.5190	14.8158	14.6481	14.6860	14.6531
	h	**1**	**1**	**1**	**0**	**1**	—
F3	Mean	17.6613	16.8770	15.1573	**14.8069**	16.3085	15.7254
	Std.	1.0504	1.1620	0.9308	0.0133	0.4618	0.0223
	Min	15.5203	15.5515	14.6219	14.6170	15.9185	15.3465
	Max	19.6494	19.4469	19.7183	15.2238	17.8514	15.9672
	h	**1**	**1**	**1**	—	**1**	**1**
F4	Mean	14.7474	15.4290	14.7654	**14.7361**	14.7525	14.7364
	Std.	$1.037E-4$	1.0917	$7.884E-4$	$2.494E-5$	$1.735E-4$	$2.205E-5$
	Min	14.7356	14.7337	14.7378	14.7291	14.7373	14.7288
	Max	14.7714	18.3901	14.8401	14.7497	14.7928	14.7488
	h	**0**	**1**	**1**	—	**1**	**0**
F5	Mean	18.9011	17.8758	17.8816	17.7712	17.5169	**16.8374**
	Std.	19.3752	0.9286	4.7299	4.3717	0.3750	0.7757
	Min	16.4034	16.4773	16.6205	16.4129	16.9094	16.3204
	Max	28.7950	19.3831	26.4327	25.2178	18.6831	17.8547
	h	**1**	**1**	**1**	**1**	**1**	—
F6	Mean	20.4977	17.2938	21.4213	16.7939	17.0746	**16.5777**
	Std.	21.7799	7.5871	23.4940	5.1557	1.6796	0.3785
	Min	17.4722	15.7391	15.4976	15.2908	15.7525	15.6083
	Max	29.3754	27.9776	29.9679	22.5225	21.5453	17.9068
	h	**1**	**1**	**1**	**0**	**0**	—
F7	Mean	23.5527	17.7175	19.9267	**17.3071**	18.5039	19.3598
	Std.	27.7538	0.5265	35.7375	15.6542	1.9334	32.9936
	Min	17.8796	16.8212	15.1296	15.0326	16.8277	15.6002
	Max	30.0131	20.5929	29.5523	28.4035	21.4855	30.3578
	h	**1**	**1**	**1**	—	**0**	**0**
F8	Mean	18.9859	15.9555	17.1540	**15.4485**	15.6366	18.54158
	Std.	8.6686	0.6078	11.4477	0.0218	0.6935	3.5024
	Min	17.4710	15.3587	15.4066	15.1707	15.4429	16.2014
	Max	28.6708	17.9973	28.8277	15.7448	19.6240	26.7133
	h	**1**	**1**	**1**	—	**0**	**1**

runs though there is much difference between these four algorithms on the mean value.

(3) The constraint handling method ε is little better than dynamic balance function and dynamic compared Δ when path is produced by Ferguson curve.

(4) Except for problems F1, DMS-PSO-ε outperforms obviously than other algorithms on the whole no matter on the result of mean value or t-test 2.

In a word, ε and Δ are suitable and DMS-PSO-ε and DMS-PSO-Δ are smart choices when Ferguson curve is applied in path planning problems especially on complex problems.

From Table 3, we could observe the following.

(1) Dynamic multiswarm has improved the search ability of traditional particle swarm optimizer which means that DMS-PSO with crossover performs better than PSO under all constraint handling methods.

Table 3: Result of $\eta 3$ curve ($n = 2$).

Problems		PSO-ε	PSO-DP	PSO-Δ	DMS-PSO-ε	DMS-PSO-DP	DMS-PSO-Δ
F1	Mean	15.1590	15.3283	15.2681	14.9161	15.1670	**14.8622**
	Std.	0.3263	0.3658	0.4875	0.2284	0.1987	0.2007
	Min	14.6853	14.6728	14.6932	14.6608	14.6801	14.6614
	Max	16.98428	16.72636	17.6536	16.4115	16.2402	16.9015
	h	**1**	**1**	**1**	**0**	**1**	—
F2	Mean	15.0589	15.0396	15.7143	**14.7317**	14.9698	14.9601
	Std.	0.4267	0.2972	1.4186	0.0902	0.1457	0.4676
	Min	14.6589	14.6626	14.7605	14.65536	14.6640	14.6555
	Max	16.2536	16.3979	19.6726	16.1707	16.2063	17.2056
	h	**1**	**1**	**1**	—	**1**	**0**
F3	Mean	16.9309	16.7421	16.6452	16.2764	16.7070	**15.8886**
	Std.	0.3723	0.5148	0.8288	0.8953	0.3677	0.2801
	Min	15.6267	15.6372	15.6718	15.6049	15.7619	15.6068
	Max	17.4024	17.3621	19.6036	18.6914	17.3590	17.3939
	h	**1**	**1**	**1**	**0**	**1**	—
F4	Mean	15.1204	15.6049	15.0711	**15.8886**	15.3697	14.9027
	Std.	0.2261	0.7872	0.0990	0.2801	0.4204	0.4031
	Min	14.7725	14.7932	14.7791	15.6068	14.7799	14.7403
	Max	15.8243	18.1527	15.9261	17.3939	16.7863	17.9495
	h	**1**	**1**	**1**	—	**1**	**0**
F5	Mean	16.5948	16.4176	16.61561	16.4206	**16.4134**	16.4329
	Std.	0.1251	$1.910E-4$	0.1206	0.0097	$1.865E-4$	0.0102
	Min	16.4121	16.4029	16.4001	16.3722	16.3787	16.3842
	Max	17.6328	16.4500	17.5537	16.8877	16.4457	16.9001
	h	**1**	**0**	**1**	**0**	—	**0**
F6	Mean	16.6868	16.6765	16.7663	16.5985	16.5424	**16.0209**
	Std.	0.0188	0.0731	0.0481	0.1177	0.1248	0.3553
	Min	16.1306	15.4302	15.8551	15.4915	15.4417	15.3031
	Max	16.9352	16.9246	16.9671	16.9530	16.7489	16.8915
	h	**1**	**1**	**1**	**1**	**1**	—
F7	Mean	16.5802	16.3344	16.4030	16.2061	15.9442	**15.6069**
	Std.	0.2244	0.45243	0.3946	0.4480	0.4248	0.3488
	Min	15.2971	15.2954	15.3358	15.2914	15.2947	15.2917
	Max	16.9642	16.8365	16.9896	16.8971	16.8196	16.8491
	h	**1**	**1**	**1**	**1**	**0**	—
F8	Mean	15.4772	15.0808	15.5568	15.1574	15.0251	**15.0064**
	Std.	0.5238	0.0665	0.4020	0.2922	$9.807E-4$	$4.680E-4$
	Min	15.0152	14.9885	15.0526	14.9821	14.9887	14.9863
	Max	16.9498	16.3030	16.9771	16.9514	15.0988	15.0621
	h	**1**	**0**	**1**	**0**	**1**	—

(2) DMS-PSO-ε has a better mean value than other algorithms on the whole.

(3) DMS-PSO-Δ is better than DMS-PSO-ε+ and DMS-PSO-DP except for F2, F4, and F5 while DMS-PSO-Δ is similar to DMS-PSO-ε on problem F2 and DMS-PSO-DP+ on problem F5 on average.

(4) Except for problems F6 and F7, DMS-PSO-Δ and DMS-PSO-ε have similar performance.

(5) Although DMS-PSO-DP is better than other algorithms on F3, DMS-PSO-Δ, DMS-PSO-ε+, and DMS-PSO-DP have the same acceptance which means they have no difference under 25 independent runs at the 5% significance level.

Generally speaking, DMS-PSO-Δ possesses a good feature so that it could be applied in path planning problems when $\eta 3$ curve is used to describe path.

Table 4: Result of $\eta 3$ curve ($n = 3$).

Problems		PSO-ε	PSO-DP	PSO-Δ	DMS-PSO-ε	DMS-PSO-DP	DMS-PSO-Δ
F1	Mean	15.1269	15.2927	15.8549	**14.8915**	15.2233	14.8945
	Std.	0.2000	0.3962	0.6258	0.1053	0.2890	0.1533
	Min	14.6743	14.6922	14.8753	14.6599	14.6810	14.6746
	Max	16.4027	16.9510	17.1939	15.7512	16.5228	16.4695
	h	**1**	**1**	**1**	—	**1**	**0**
F2	Mean	15.1828	15.8735	17.3363	14.8915	15.5943	**14.7678**
	Std.	0.4999	0.5877	2.4413	0.1053	0.4150	0.0323
	Min	14.6786	14.6978	15.0084	14.6599	14.6649	14.6682
	Max	17.1109	17.5539	20.7949	15.7512	17.0959	15.5873
	h	**1**	**1**	**1**	**0**	**1**	—
F3	Mean	16.9903	17.3458	17.8328	16.2678	17.13297	**15.3092**
	Std.	0.3941	0.34185	1.5596	1.7428	0.1416	0.7615
	Min	15.1838	15.7395	15.8694	14.7341	15.9525	14.7554
	Max	17.6611	18.7940	20.4837	18.8251	17.4754	17.1826
	h	**1**	**1**	**1**	**1**	**1**	—
F4	Mean	15.1503	16.2299	16.5532	**14.7421**	16.1149	15.0292
	Std.	0.4168	1.0923	1.9814	$6.941E-4$	1.2537	1.8730
	Min	14.7527	14.8084	14.9267	14.6988	14.7411	14.7161
	Max	17.4512	18.0449	19.4632	14.8068	18.0394	21.5966
	h	**1**	**1**	**1**	—	**1**	**0**
F5	Mean	16.8508	16.6488	16.9656	16.4054	16.5114	**16.3798**
	Std.	0.2432	0.1425	0.1367	0.0569	0.0180	0.0037
	Min	16.4201	16.3486	16.5099	16.3191	16.3300	16.3254
	Max	17.8626	17.4785	17.9822	17.5422	16.9302	16.6216
	h	**1**	**1**	**1**	**0**	**1**	—
F6	Mean	16.6472	16.5453	16.9163	16.2758	16.2548	**15.7087**
	Std.	0.0727	0.1016	0.0788	0.3668	0.2496	0.32256
	Min	15.4516	15.5077	16.6653	15.2211	15.2660	15.2387
	Max	16.9744	16.9084	18.13603	17.0442	16.73439	16.7263
	h	**1**	**1**	**1**	**1**	**1**	—
F7	Mean	16.6997	16.4166	16.8786	15.8356	16.1770	**15.6007**
	Std.	0.0129	0.3524	0.0291	0.5831	0.2656	0.3273
	Min	16.5565	15.1092	16.5886	15.0152	15.0720	15.0145
	Max	16.8951	16.9036	17.3592	16.7376	16.7343	16.7801
	h	**1**	**1**	**1**	**0**	**1**	—
F8	Mean	15.1227	14.9834	15.5447	**14.6521**	14.7546	14.6624
	Std.	0.3083	0.2759	0.5752	$7.786E-4$	0.0216	0.0016
	Min	14.6780	14.6695	14.7240	14.6327	14.6510	14.6180
	Max	16.6986	16.6473	18.1863	14.7619	15.3003	14.7838
	h	**1**	**1**	**1**	—	**1**	**0**

Case 2. Eight parameters are satisfied to generate path; Table 4 is the result (the result concludes $\eta 3$ curve only).

The following information is given from Table 4.

(1) DMS-PSO is absolutely better than PSO in three-segment $\eta 3$ curve no matter the result of mean value or the null hypothesis at the 5% significance level.

(2) When DMS-PSO-ε is better than DMS-PSO-Δ on the mean value, the previous accepts the latter on all problems which means they have no difference.

(3) When DMS-PSO-Δ is better than DMS-PSO-ε+ on the mean value, the previous rejects the latter on F3 and F6.

Table 5: Best result of every curve.

Problems		Bezier curve	Ferguson curve	$\eta 3$ curve $n = 2$	$\eta 3$ curve $n = 3$
F1	Mean	14.7559	**14.6720**	14.8622	14.8915
	Std.	0.0087	$3.065E-5$	0.2007	0.1053
	Min	14.6626	14.6626	14.6614	14.6599
	Max	15.0132	14.6830	16.9015	15.7512
F2	Mean	14.6486	**14.6395**	14.7317	14.7678
	Std.	0.0002	$6.640E-5$	0.0902	0.0323
	Min	14.6242	14.6245	14.65536	14.6682
	Max	16.3361	14.6531	16.1707	15.5873
F3	Mean	14.9475	**14.8069**	15.8886	15.3092
	Std.	0.2976	0.0133	0.2801	0.7615
	Min	14.7240	14.6170	15.6068	14.7554
	Max	17.2080	15.2238	17.3939	17.1826
F4	Mean	14.7363	**14.7361**	15.8886	14.7421
	Std.	$1.451E-5$	$2.494E-5$	0.2801	$6.941E-4$
	Min	14.7313	14.7291	15.6068	14.6988
	Max	14.7455	14.7497	17.3939	14.8068
F5	Mean	**16.3361**	16.8374	16.4134	16.3798
	Std.	0.0012	0.7757	$1.865E-4$	0.0037
	Min	16.2827	16.3204	16.3787	16.3254
	Max	16.4123	17.8547	16.4457	16.6216
F6	Mean	**15.2860**	16.5777	16.0209	15.7087
	Std.	0.0103	0.3785	0.3553	0.32256
	Min	15.2400	15.6083	15.3031	15.2387
	Max	15.7535	17.9068	16.8915	16.7263
F7	Mean	**15.1967**	17.3071	15.6069	15.6007
	Std.	0.1794	15.6542	0.3488	0.3273
	Min	15.0355	15.0326	15.2917	15.0145
	Max	16.6090	28.4035	16.8491	16.7801
F8	Mean	14.6644	15.4485	15.0064	**14.6521**
	Std.	0.0003	0.0218	$4.680E-4$	$7.786E-4$
	Min	14.6291	15.1707	14.9863	14.6327
	Max	14.7046	15.7448	15.0621	14.7619

(4) DMS-PSO-Δ and DMS-PSO-ε are better than DMS-PSO-DP on all problems.

So, when three-segment $\eta 3$ curve is used to generate path, it is clear to ensure DMS-PSO performs better than PSO. The constraint handling method dynamic threshold ε and dynamic compared Δ are better than dynamic balance function and dynamic compared Δ overcomes the drawback of dynamic threshold ε and outperforms it.

Case 3. The comparison of best result under each curve is as follows.

Having compared the best result of every curve, when two segments of Bezier curve, Ferguson curve, and $\eta 3$ curve are used to describe path, we could observe the following.

(1) Ferguson curve performs better than Bezier curve and $\eta 3$ curve on problems F1 to F4 which are easy to find global optimum and has a worse feature on problems F5 to F8 which are described as complex problems.

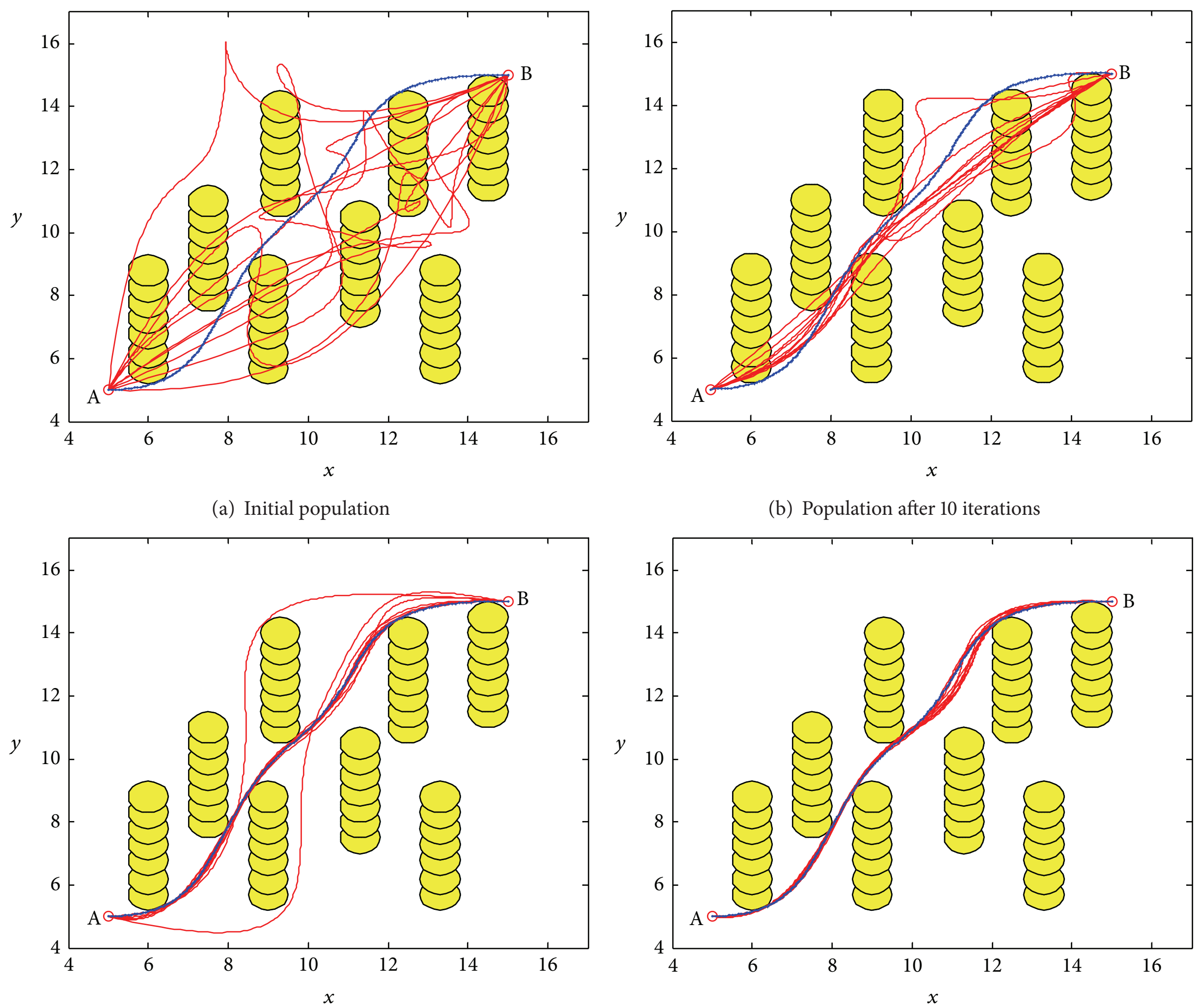
(2) Although $\eta 3$ curve is not so good as Ferguson curve and Bezier curve on simple problem, it outperforms Ferguson curve on complex problem which means $\eta 3$ curve possesses a better search ability.

(3) Bezier curve is better in producing path compared with Ferguson curve and $\eta 3$ curve on the whole.

When three-segment $\eta 3$ curve is applied in these problems, conclusions could be made as follows.

(1) Compared with two segments, three-segment $\eta 3$ has a small range of the distribution between the best and worst solutions on the whole.

(2) There is no difference between problems F1, F2, F5, and F7 no matter it is two-segment or three-segment $\eta 3$ curve.



(a) Initial population

(b) Population after 10 iterations

(c) Population after 400 iterations

(d) Population after 700 iterations

Figure 5: Processes of iteration.

(3) Three-segment $\eta 3$ curve outperforms on problems F3, F4, F6, and F8 in generating path obviously.

(4) On complex problems which are easily trapped into local optimum, three-segment $\eta 3$ curve is much better than two-segment curve which means that the previous has obvious difference compared to the latter such as F6, F8, F3, and F4.

(5) Three-segment $\eta 3$ curve performs better than Ferguson on complex problems but obviously expresses an inferior characteristic in generating path compared with Bezier curve.

From so many points of discussion from Tables 1 to 5, conclusions could be made that DMS-PSO overcomes the drawback of PSO which is easy to fall into local optimal and premature. Dynamic compared Δ overcomes the drawback and inherits the advantage of dynamic threshold ε and shows better constraint characteristics than dynamic balance function, which makes it show good binding properties in path planning problems.

When all curves are composed by the same number of segments, $\eta 3$ curve outperforms Ferguson curve on complex problems specifically but is worse than Bezier curve for all problems. Fewer points would be optimized when $\eta 3$ curve is used to describe path, so less time is needed.

Three-segment $\eta 3$ curve is better than two-segment one for generating path because it has more anchor points to control and can generate a more flexible path, especially on complex problems.

Bezier curve expresses better performance on path planning problems compared with Ferguson curve and $\eta 3$ curve. The most possible reason may be that Bezier curve is easier to change the shape of the path via adjustment of a fixed number of anchor points than the other two curves. So Bezier curve is the most suitable curve to produce path in this paper and DMS-PSO with crossover combined with dynamic compared Δ is the best choice to optimize path in path planning problems.

In order to show the property of Bezier curve and DMS-PSO with crossover combined with dynamic compared Δ, complex problem F6 is an example to show how particles learn from their neighborhood and avoid being trapped into local optimum. The yellow circles describe the dangerous distance around the obstacles, red paths mean the current local paths, and blue path is the best path satisfying some certain criteria. The processes of iteration are in Figure 5.

Figure 5 shows the search process which could be seen that although the robot always runs into obstacles in the first 400 iterations, it is far away from obstacles step by step. After 700 iterations, solutions are converged into the best path gradually which shows that DMS-PSO with crossover combined with dynamic compared Δ has a good ability of global search in early stage and global convergence in latter stage.

5. Conclusion

In order to solve path planning problems in static environment, suitable curves and algorithms with constraint mechanisms are designed in this paper. Three curves are compared under six algorithms, and the results have proved that DMS-PSO has a better ability of global search than PSO again. At the same time, the analyses of three constraint handling methods and curves are carried on. Firstly, dynamic constraint methods are designed well for path planning problems compared with the previous work where static constraint is used. Then the dynamic compared Δ possesses a better feature than dynamic threshold ε and dynamic balance function which has overcome the drawback of dynamic threshold ε which might lose the previous good solutions found in the search process, which are near to the global optimum but do not satisfy the current constraint. So dynamic compared Δ is more suitable to be applied in path planning problems. From the results, we could observe that $\eta 3$ curve outperforms Ferguson curve when the same segment is used to describe path, especially on the distribution of solutions for complex problems. What is more, compared with two segments, three segments of $\eta 3$ curve are more suitable to generate path for the reason that more points may make the path more flexible and easier to change the direction of path. The most important is that Bezier curve outperforms $\eta 3$ curve and Ferguson curve no matter on simple or complex problems and it improves the solutions further, which is more likely depending on the property of its flexible shape changed by adjusting a fixed number of anchor points. So when PSO and its improved versions are used to solve path planning problems, Bezier curve possesses a higher status. For the limitation of experimental conditions, only the path length and security criteria are compared in this paper. Bezier curve will be used to evaluate more criteria in path planning problems as well as in the condition of dynamic environment where the obstacles are changed with time in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported partially by National Natural Science Foundation of China (61305080, U1304602), Postdoctoral Science Foundation of China (Grant 20100480859), Specialized Research Fund for the Doctoral Program of Higher Education (20114101110005), Scientific and Technological Project of Henan Province (132102210521, 122300410264), and Key Foundation of Henan Educational Committee (14A410001).

References

[1] T. Zhao and J. Fang, "A transit path planning model based on the heterogeneous road network," in *Proceedings of the 17th International Conference on Geoinformatics*, Fairfax, Va, USA, August 2009.

[2] G. Chesi, "Designing image trajectories in the presence of uncertain data for robust visual servoing path-planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 1492–1497, Kobe, Japan, May 2009.

[3] L. Guangshun and S. Hongbo, "Study of technology on path planning for mobile robots," in *Proceedings of the Chinese Control and Decision Conference (CCDC '08)*, pp. 3295–3300, Yantai, China, July 2008.

[4] Y.-H. Qu, Q. Pan, and J.-G. Yan, "Flight path planning of UAV based on heuristically search and genetic algorithms," in *Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society*, pp. 6–10, November 2005.

[5] J.-J. Liang, H. Song, B.-Y. Qu, and X.-B. Mao, "Path planning based on dynamic multi-swarm particle swarm optimizer with crossover," in *Intelligent Computing Theories and Applications*, vol. 7390 of *Lecture Notes in Computer Science*, pp. 159–166, Springer, Berlin, Germany, 2012.

[6] J.-H. Wei and J.-S. Liu, "Collision-free composite η^3-splines generation for nonholonomic mobile robots by parallel variable-length genetic algorithm," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA '08)*, pp. 545–550, Vienna, Austria, December 2008.

[7] W. Xianxiang, M. Yan, and W. Juan, "An improved path planning approach based on particle swarm optimization," in *Proceedings of the 11th International Conference on Hybrid Intelligent Systems (HIS '11)*, pp. 157–161, December 2011.

[8] M. Saska, M. Macaš, L. Přeučil, and L. Lhotská, "Robot path planning using particle swarm optimization of ferguson splines," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA '06)*, pp. 833–839, September 2006.

[9] Y. Gao, S.-D. Sun, and D.-F. He, "Global path planning of mobile robot base on particle filter," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering (CSIE '09)*, pp. 205–209, Los Angeles, Calif, USA, April 2009.

[10] J.-W. Choi, R. Curry, and G. Elkaim, "Path planning based on Bézier curve for autonomous ground vehicles," in *Proceedings of the Advances in Electrical and Electronics Engineering—IAENG Special Edition of the World Congress on Engineering and Computer Science*, pp. 158–166, San Francisco, Calif, USA, October 2008.

[11] L. Yang, Z. Luo, Z. Tang, and W. Lv, "Path planning algorithm for mobile robot obstacle avoidance adopting Bezier curve based on genetic algorithm," in *Proceedings of the Chinese Control and*

Decision Conference (CCDC '08), pp. 3286–3289, Yantai, China, July 2008.

[12] D. W. Zhu and X. B. Mao, "Path planning algorithm based on improved particle swarm optimization of Bezier curves," *Application Research of Computers*, vol. 5, pp. 1–6, 2012.

[13] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[14] N. G. Bourbakis, D. Goldman, R. Fematt, I. Vlachavas, and L. H. Tsoukalas, "Path planning in a 2-D known space using neural networks and skeletonization," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2001–2005, Orlando, Fla, USA, October 1997.

[15] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 968–975, May 2002.

[16] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '97)*, pp. 138–143, Monterey, Calif, USA, July 1997.

[17] I. Al-Taharwa, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.

[18] H. Mo and Z. Li, "Bio-geography based differential evolution for robot path planning," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '12)*, pp. 1–6, Shenyang, China, June 2012.

[19] J. Chakraborty, A. Konar, U. K. Chakraborty, and L. C. Jain, "Distributed cooperative multi-robot path planning using differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 718–725, June 2008.

[20] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, December 1995.

[22] Z. A. Bashir and M. E. El-Hawary, "Short-term load forecasting using artificial neural network based on particle swarm optimization algorithm," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECD '07)*, pp. 272–275, Vancouver, Canada, April 2007.

[23] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 2, pp. 511–526, 2012.

[24] H. E. Espitia and J. I. Sofrony, "Path planning of mobile robots using potential fields and swarms of Brownian particles," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 123–129, New Orleans, La, USA, June 2011.

[25] M. Ding, W. Sun, and H. Chen, "Multi-working modes product-color planning based on evolutionary algorithms and swarm intelligence," *Mathematical Problems in Engineering*, vol. 2010, Article ID 871301, 15 pages, 2010.

[26] T.-S. Zhan and C.-C. Kao, "Modified PSO method for robust control of 3RPS parallel manipulators," *Mathematical Problems in Engineering*, vol. 2010, Article ID 302430, 25 pages, 2010.

[27] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 9–16, Vancouver, Canada, July 2006.

[28] J. J. Liang, H. Song, and B. Y. Qu, "Performance evaluation of dynamic multi-swarm particle swarm optimizer with different constraint handling methods on path planning problems," in *Proceedings of the IEEE Workshop on Memetic Computing*, pp. 65–71, April 2013.

[29] R. Wagner, O. Birbach, and U. Frese, "Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, pp. 3305–3312, San Francisco, Calif, USA, September 2011.