A new approach to optimal smooth path planning of mobile robots with continuous-curvature constraint

Lin Xu, Baoye Song and Maoyong Cao

College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, People’s Republic of China

ABSTRACT
Smooth path planning is very important to mobile robots with continuous-curvature constraint, but there are still some limitations and drawbacks on traditional planning approach. To deal with this problem, a new approach combined with parametric cubic Bezier curve (PCBC) and particle swarm optimization with adaptive delayed velocity (PSO-ADV), is developed to plan the smooth path of mobile robots. Unlike the traditional smooth path consisting of several linear and curve segments with discontinuous curvature at the joints, the smooth path composed of PCBC segments has equivalent curvature at the segment joints, thereby it is able to attain continuous curvature along the whole smooth path. In terms of the mathematical formulation of PCBC, the smooth path planning is essentially an optimization problem to seek the optimal control points and parameters of PCBC segments. To handle this intractable problem and some frequently encountered troubles (e.g. premature convergence and local trapping), a new PSO-ADV algorithm is developed by blending the term of adaptive delayed velocity, and its superiority can be confirmed by several simulation experiments. The new approach is finally applied to produce the smooth path with continuous-curvature constraint, and can achieve superior performance in comparison with traditional method.

1. Introduction

With the rapid development of robotics in recent years, autonomous mobile robots have been widely used for various service tasks (Liu et al., 2018), and path planning (PP) is one of the most critical technics for the autonomy of mobile robots (Lin et al., 2017). The main task of path planning is to produce an optimal and feasible path for a mobile robot from the start to the destination, and till now, a large amount of algorithms have been developed to tackle the issues of mobile robot path planning, see e.g. Gonzalez et al. (2016), Fong et al. (2015), Elbanshawi and Simic (2014), Atyabi and Powers (2013), and the references therein.

Among these algorithms, the heuristic intelligent optimization algorithms have been extensively investigated to deal with the path planning problem, which is subject to several performance criteria and constraints (e.g. the shortest path and collision avoidance) (Atyabi & Powers, 2013; Fong et al., 2015; Qu et al., 2013; Zeng, Zhang, et al., 2016). For example, a hybrid algorithm has been proposed for the path planning of a mobile robot in Contreras-Cruz et al. (2015), where the artificial bee colony algorithm is firstly utilized to plan a set of feasible paths in the local searching procedure, and then the evolutionary programming algorithm is applied to refine the local results in the searching space. The firefly algorithm has been applied to navigate the mobile robot in an uncertain environment, in which the uncertainty is introduced by the variation of the environmental conditions from static to dynamic (Patle et al., 2018). In Zeng, Zhang, et al. (2016), a novel PSO (particle swarm optimization) algorithm combined with the DE (differential evolution) algorithm and a non-homogeneous Markov chain has been presented for the path planning of intelligent robot, and the effectiveness and feasibility of the algorithm are verified by several simulation experiments. A hybrid algorithm combining the two-dimensional cellular automata with the ant colony has been put forward to plan a feasible path for the mobile robots in the environment with both convex and concave obstacles (Akbarimajd & Hassanzadeh, 2012), where the working environment of the mobile robots is decomposed into a great deal of rectangular grids, and the four states of the automata updated by the evolutionary rules are used...
to guide the mobile robots towards their destination. In Nazarahari et al. (2019), a hybrid approach combining the artificial potential field algorithm with an enhanced genetic algorithm has been developed to deal with a multi-objective path planning problem in the environment with several mobile robots, and some comprehensive evaluations on several benchmark path planning problems are carried out to demonstrate the superiority of the proposed novel approach in comparison with some well-known algorithms. In B. Tang et al. (2016), the PSO algorithm is hybridized with the DE algorithm to tackle the issues such as stagnation and premature in the global path planning of mobile robots, and the superior performances of the proposed algorithm are verified by comparing with several state-of-the-art intelligent optimization algorithms.

However, there are still several limitations in the aforementioned works, and one of them is the multi-objective path planning problem (Nazarahari et al., 2019), which usually includes not only the minimization of the path length, but also several other objectives such as smoothness and safety of the path (Song et al., 2016), etc. Among others, the smoothness of a planning path, which has proven to be an imperative criterion that is closely related to other criteria (Castillo et al., 2007), has attracted increasing research attentions in recent years. For example, an A* algorithm has been employed to seek for a feasible path from the start to the destination on the directed graph (On & Yazici, 2011, July), and the generated optimal path is smoothed by using an arc-line approach. A parallel PSO algorithm has been proposed to plan the feasible linear path in Huang (2014), and then the path is smoothed by using a cubic B-spline curve. In Zhou et al. (2011), the Dijkstra algorithm has been combined with the Voronoi digram to devise a piecewise linear path, whose endpoints are regarded as the control points of the smooth Bezier curve path. A linear path has been generated by using a hybrid approach combining the Dijkstra algorithm with the MAKLINK Graph, then the $\eta^3$-splines are employed to smoothly interpolate the endpoints of the path (S. Zhang et al., 2014, December). A genetic algorithm has been implemented to achieve an optimal obstacle-avoidance path connecting the initial and final positions of the mobile robots, and then the piecewise cubic Hermite interpolating polynomials are exploited to produce a smooth optimal path (Bakdi et al., 2017).

Nevertheless, it is worth noting that the smooth path planning methodologies in the aforementioned works often include two steps, i.e. a linear path is usually firstly generated by using an intelligent search algorithm (e.g. PSO) upon some criteria and constraints such as the length and feasibility of the path, and then the linear path is smoothed via a parametric curve (e.g. Bezier curve). The two-step methodology would undoubtedly bring some performance degradations and other undesirable results, because the path smoothing is not directly linked to the path optimization criteria (Song et al., 2016). To overcome the drawbacks identified above, a few approaches have been put forward to directly produce an optimal smooth path for the mobile robots. In Chang and Liu (2009, February), an IPGA algorithm (island parallel genetic algorithm) has been combined with the $\eta^3$-splines to plan a continuous collision-avoidance smooth path with differentiable curvature for autonomous mobile robots, but only the minimization of path length is considered for the optimization criteria. In Song et al. (2016), a new approach combining Bezier curve with genetic algorithm has been developed to design a smooth path for mobile robots, however, the control points of the Bezier curve are limited to the centre of the predefined grids rather than arbitrary positions of the whole workspace. To combat this obstacle issue, an improved PSO combined with Bezier curve segments has been investigated to devise a smooth path for mobile robots (Song et al., 2017), but it is subject to the trouble of discontinuous curvature on the connecting points of the curve segments. Thus, a new scheme has been proposed to plan the smooth path of mobile robots using a continuous high-degree Bezier curve in combination with an improved PSO algorithm in Xu et al. (2017, October), yet, the computational burden is huge because of the complexity of the path optimization problem, and some constraints such as maximum curvature and maximum curvature-derivative are not simultaneously considered. Additionally, a new method to optimal curvature smoothing has been presented for the trajectory generation of flying robots in Dong et al. (2019), while it is to deal with the issue of smooth transition between straight linear segments rather than smooth path planning of mobile robots.

To handle the above-mentioned problems in the smooth path planning of mobile robots, a new approach combined with the parametric cubic Bezier curve (PCBC) and a novel PSO with adaptive delayed velocity (PSO-ADV) is developed to plan an optimal smooth path for the mobile robots with continuous-curvature constraint. The main contributions of the current paper can be outlined from the following aspects. (1) A new approach based on parametric cubic Bezier curve is presented to overcome the drawbacks of the traditional smooth paths composed of several curve segments, thereby the continuous curvature can be achieved at the joints of the curve segments. (2) A modified PSO with adaptive delayed velocity is developed to deal with some obstacles encountered frequently in the smooth robot path optimization, e.g. premature convergence and local trapping, and the
performance of the modified PSO is evaluated via some simulation experiments on several famous benchmark functions. (3) The problem of smooth path planning with continuous curvature constraint is formulated mathematically as an optimization problem, which is resolved by the new approach combined with PCBC and PSO-ADV algorithms. (4) The superiority of the new approach on the smooth path planning is confirmed on the basis of several comparisons with traditional studies.

The remaining part of the present paper is organized as follows. Section 2 firstly presents the preliminary of the parametric cubic Bezier curve in detail, and then delineates the problem statement of the smooth path planning of mobile robots. Section 3 elaborates the newly developed PSO algorithm with adaptive velocity, and its superiority is also validated by the comparisons with some famous PSOs on several benchmark functions. Section 4 reports the results of the simulation experiments on smooth path planning based on the new approach combined with PCBC and PSO-ADV algorithms. In the last section, the paper is concluded and some future works are pointed out.

2. Preliminary and problem statement

2.1. Preliminary on parametric cubic Bezier curve

The parametric cubic Bezier curve (PCBC) is an analytical continuous-curvature path-smoothing algorithm proposed by Yang and Sukkarieh (2010). By regulating the parameter of PCBC, the straight linear segments can be smoothly connected and the constraint of continuous-curvature can also be satisfied.

The cubic Bezier curve can be expressed as follows:

\[ B(u) = \sum_{i=0}^{3} \mathbf{B}_i \binom{3}{i} u^i (1-u)^{3-i}, \]  

where \( u \) is subject to \( 0 \leq u \leq 1 \); \( \mathbf{B}(u) \) indicates the cubic Bezier curve; \( \mathbf{B}_i \) denotes the \( i \)th control point. With this definition, the Bezier transition curves illustrated in Figure 1 can be built by using two symmetric cubic Bezier curves that are expressed as follows:

\[ B_1(u) = \sum_{i=0}^{3} \mathbf{B}_{1,i} \binom{3}{i} u^i (1-u)^{3-i}, \]  

\[ B_2(u) = \sum_{i=0}^{3} \mathbf{B}_{2,(3-i)} \binom{3}{i} u^i (1-u)^{3-i}, \]  

where \( \mathbf{B}_{1,0} (j = 1, 2) \) denotes the connecting point between the Bezier curve and the straight line, and \( \mathbf{B}_{3,0} (j = 1, 2) \) denotes the connecting point of the two Bezier curves.

\[ B_{10} = P_2 - T_1 d, \]

\[ B_{11} = P_2 - T_1 (1-c_1 c_3) d, \]

\[ B_{12} = P_2 - T_1 (1-c_1 c_3 - c_3) d, \]

\[ B_{13} = B_{12} + \eta d u d, \]

\[ B_{20} = P_2 + T_2 d, \]

\[ B_{21} = P_2 + T_2 (1-c_1 c_3) d, \]

\[ B_{22} = P_2 + T_2 (1-c_1 c_3 - c_3) d, \]

\[ B_{23} = B_{22} - \eta d u d, \]

where \( T_1 = \frac{P_1P_2}{||P_1P_2||}, \ T_2 = \frac{P_2P_3}{||P_2P_3||}, \ c_1 = 2(\sqrt{6} - 1)/5, \ c_2 = (c_1 + 4)(c_1 + 1), \ c_3 = (c_1 + 4)/(c_2 + 6), \ \eta = 6c_3 \cos \beta/(c_1 + 4), \) and \( u_d = \frac{B_{12}B_{22}}{||B_{12}B_{22}||}. \)

It should be highlighted that the quality of the Bezier transition curves are fully depended on the transition length \( d \), a control parameter of PCBC that can be linked to the optimization criteria of smooth path planning. It has proven that the continuous-curvature can be achieved at the optimization point \( B_{3,0} \) of the two Bezier transition curves \( \mathbf{B}_1(u) \) and \( \mathbf{B}_2(u) \) via the PCBC algorithm (Yang & Sukkarieh, 2010). Besides, the connecting point \( B_{3,0} \) is the point that has the maximum curvature, which can be computed as

\[ k_{\text{max}} = \frac{2c_3 \sin \beta}{3d\eta^2}. \]

In view of Equation (12), the maximum curvature \( k_{\text{max}} \) has an inverse relation with the control parameter \( d \). So that, if
the following constraint: 

\[ d \geq \frac{2c_3 \sin \beta}{3\kappa_{\max}h^2}. \]  

(13)

Therefore, the control parameter \( d \) of PCBC is a critical variable to be optimized for each Bezier transition curve in the following smooth path planning of mobile robots.

### 2.2. Problem statement of mobile robot smooth path planning

To compare the performance of the smooth path planning algorithms conveniently, the workspace of the mobile robot is supposed to be a two-dimensional plane like (Song et al., 2016, 2017). As shown in Figure 2, the workspace is a square plane cut into \( 2^n \times 2^n \) grids, each of which is a square with 10 x 10 units. Each grid of the workspace is assigned a grid number, where the white grids represent the feasible space for the mobile robots, while the infeasible area with obstacles is marked with black grids. It is obvious that the workspace can be described more detailedly using more grids (i.e. a larger parameter \( n \)), which also means more computational cost. In this paper, the parameter \( n \) is set as 4 in accordance with the performance requirement of the smooth path planning.

It should be noted that the obstacles in the workspace have been extended according to the size of the mobile robot, which can be regarded as a unit point of the workspace regardless of their true sizes. Thus, a feasible unit point can be determined by the following defined criterion.

### Definition 2.1: A unit point is feasible if and only if it is included in the feasible white grids, which should not contain any part of the extended obstacles, i.e. \( N_{B(u)}(u) \in N_w \), where \( N_{B(u)} \) represents the number of the grid containing the unit point on the Bezier path \( B(u) \), and \( N_w \) represents the set of the numbered feasible white grid, and the conversion from the coordinates of a unit point to the corresponding numbered grid can be formulated as:

\[ N_{B(u)} = \lfloor x(u)/10 \rfloor + \lfloor y(u)/10 \rfloor \times 16, \quad (14) \]

where \( x(u) \) and \( y(u) \) stand for the X- and Y-coordinate components of the unit point; \( \lfloor \cdot \rfloor \) indicates the rounding down operation.

Now, we are in the position to describe the problem to be solved in this paper. On account of the aforementioned workspace of the mobile robot, the objective of this paper is to plan a feasible and optimal smooth path between the given start and destination of the mobile robot, where the obtained smooth path should satisfy the optimization criteria and constraints as follows: (1) the smooth path devised in this paper should be an obstacle-avoidance route for the movement of the mobile robot; (2) the planned smooth path should satisfy the requirement of continuous-curvature along the path, i.e. a G\(^2\)-path (see Piazzi et al., 2007 for more details); (3) the maximum curvature along the smooth path should be minimized for the mobile robot with limited turning radius; and (4) the full length of the path should be minimized except for the above-mentioned criteria and constraints.

To accomplish this smooth path planning task, the segmented parametric cubic Bezier curves are employed to produce a satisfying smooth path by optimizing their control points and control parameters. Thus, the planning of smooth path in this paper is essentially an optimization problem to seek a series of control points \( P_i \) (\( i = 1, 2, \ldots, m + 1 \)) and control parameters \( d_i \) (\( i = 1, 2, \ldots, m \)), which can lead to a feasible and optimal smooth path for the mobile robot. Consequently, considering some constraints of the parametric cubic Bezier curves, the objective function of the optimization problem can be formulated as follows:

\[
\arg \min_{P_i, d_i} \sum_{i=1}^{m} \left( \sum_{j=1}^{2} \int_{0}^{1} \| \dot{B}_j(u; d_i) \| \, du \right),
\]

\[
\text{s.t. } N_{B_j(u; d_i)}(u) \in N_w, \quad 0 \leq u \leq 1,
\]

\[
|k_{ij}(u; d_i)| \leq \kappa_{\max},
\]

where \( k_{ij}(u; d_i) \) denotes the curvature of the ith Bezier transition curve at the point \( u \) on the \( j \)th segment. The above optimization problem can be solved using the incremental gradient descent method to find the optimal control parameters and control points to achieve the desired smooth path.
where $B_i(u; d_i)$ is one of $2m$ segmented parametric cubic Bezier curves, which are used for $m$ transitions of straight linear segments; $G^2$ denotes the $G^2$-path that has continuous curvature; $\kappa_{\text{max}}$ is the given constrained curvature for the smooth movement of the mobile robot; $\kappa_{\text{max},i}$ denotes the maximum curvature on the $i$th transition curve; $l_i = \|P_{i+1} - P_i\|$ is the length of the $i$th straight linear segment; and $d_i$ is the $i$th control parameter of the transition. It is apparent that the issue of smooth path planning is a rather complicated optimization problem that is apt to encounter some obstacle issues, e.g. local trapping and premature convergence. To handle these obstacle issues, a novel PSO algorithm is developed and will be discussed later.

3. A novel PSO algorithm with adaptive delayed velocity

3.1. Classical PSO and its variants

The PSO, which was developed by Kennedy and Eberhart (1995, November) to simulate the behaviours of animal swarms (e.g. the flocks of birds and fishes), is a meta-heuristic intelligent optimization algorithm that has been widely and successfully utilized to search for the optimum solutions of various optimization problems.

In PSO, the particle of the swarm acts as a candidate optimal solution of a specific optimization problem, which determines the dimension of the particle’s velocity and position vectors. By flying around to explore the solution space of the optimization problem, an optimal solution can be achieved and the updating functions of the velocity and position vectors at the $k$th iteration can be expressed as follows:

$$V_i(k + 1) = wV_i(k) + c_1r_1(P_{ib}(k) - X_i(k)) + c_2r_2(G_b(k) - X_i(k)), \quad (16)$$

$$X_i(k + 1) = X_i(k) + V_i(k + 1), \quad (17)$$

where $V_i(k)$ and $X_i(k)$ represent the $i$th particle’s velocity and position vectors, respectively; $P_{ib}(k)$ and $G_b(k)$, respectively, indicate the best position that the $i$th particle has ever experienced and the best position of the global swarm till the $k$th iteration; $w$ denotes the inertia weight of the velocity; $c_1$ and $c_2$ are defined as the cognitive and social parameters which can also be called the acceleration coefficients; $r_1$ and $r_2$ are the random numbers uniformly distributed on $[0, 1]$.

By now, a large amount of improved PSO algorithms have been presented to advance the performance of the above-mentioned classical PSO, e.g. PSO-LDIW (Shi & Eberhart, 1998, May), PSO-CK (Clerc & Kennedy, 2002), PSO-TVAC (Ratnaweera et al., 2004), FIPS (Mendes et al., 2004), CLPSO (Liang et al., 2006), PS-ACO (Shuang et al., 2011), and PSO-SA (Behnamian & Fatemi Ghomi, 2010), to name but a few. These improved PSOs are usually developed based on diverse methodologies, e.g. the parameter regulation, local swarm division, and hybridization with other intelligent algorithms, etc. However, to the best of our knowledge, the idea about adaptively regulating the updating functions of the particle swarm is still scarcely considered, except for a few scattered works. For instance, the velocity updating function has been adaptively switched among several evolutionary states depending on the evolutionary factor of the particle swarm (Z. Zhan et al., 2009). An SPSO (switching PSO) has been developed in Y. Tang et al. (2011) to switch the updating function in accordance with the predicted evolutionary states determined by the calculated evolutionary factor and a Markov chain. Afterwards, an SDPSO (switching PSO with random delay) has been put forward to update the velocity function based on the evolutionary state of the particle swarm (Zeng, Wang, et al., 2016).

Besides, an MDPSO (PSO with multimodel delayed velocities) has been developed in Song et al. (2017) to overcome some obstacle issues, such as the local trapping and the premature convergence. Nevertheless, the aforementioned improved PSOs are usually subject to several rather severe requirements, e.g. the powerful computational capacity and the large memory storage, etc.

3.2. PSO-ADV algorithm and its performance evaluation

In this paper, a novel PSO with adaptive delayed velocity (denoted by PSO-ADV) is proposed to overcome the obstacle issues identified above and promote the searching performance of the above-mentioned improved PSO algorithms, and hence effectively tackle the smooth path planning problem of mobile robots. The updating functions of the proposed novel PSO algorithm can be formulated as follows:

$$V_i(k + 1) = wV_i(k) + w_1V_i(k - 1) + c_1r_1(P_{ib}(k) - X_i(k)) + c_2r_2(G_b(k) - X_i(k)),$$  \hspace{1cm} (18)

$$X_i(k + 1) = X_i(k) + V_i(k + 1), \quad (19)$$
where \( c_1 \) and \( c_2 \) are computed by

\[
c_1 = (c_{1i} - c_{1f}) \times \frac{k_{\text{max}} - k}{k_{\text{max}}} + c_{1f}, \quad (20)
\]

\[
c_2 = (c_{2i} - c_{2f}) \times \frac{k_{\text{max}} - k}{k_{\text{max}}} + c_{2f}, \quad (21)
\]

where \( k_{\text{max}} \) denotes the maximum iteration; \( c_{1i} \) (\( c_{2i} \)) and \( c_{1f} \) (\( c_{2f} \)) are, respectively, the initial and final values of the acceleration coefficient \( c_1 \) (\( c_2 \)); \( w \), which is inspired by the FOPSO algorithm (Pires et al., 2010), stands for the inertia weight of the additional delayed velocity \( V_i(k-1) \), and can be computed depending on the inertia weight \( w \) of the current velocity \( V_i(k) \) by

\[
w_1 = \frac{1}{2} w(1 - w), \quad (22)
\]

where \( w \) is linearly and adaptively regulated depending on the evolutionary state of the swarm in this paper, and can be computed by

\[
w = 0.9 - \frac{1}{1 + e^{E_f(k)}} \times \frac{k}{k_{\text{max}}}, \quad (23)
\]

where \( E_f(k) \) indicates the evolutionary factor to reflect the evolutionary state of the swarm, and can be defined by

\[
E_f(k) = \frac{D_{gb}(k) - D_{\text{min}}(k)}{D_{\text{max}}(k) - D_{\text{min}}(k)}, \quad (24)
\]

where \( D_{gb} \) indicates the average Euclidean distance from the global best particle to the other particles of the swarm; \( D_{\text{max}}(k) \) and \( D_{\text{min}}(k) \) indicate the maximum and minimum average Euclidean distances between one particle and the others in the swarm; the average Euclidean distance of the \( i \)th particle at the \( k \)th iteration (denoted by \( D_{i}(k) \)) can be computed by

\[
D_{i}(k) = \frac{1}{S-1} \sum_{j=1;j\neq i}^{S} \left( \sum_{k=1}^{D} (x_{i}(k) - x_{j}(k))^2 \right), \quad (25)
\]

where \( D \) denotes the dimension of the particle; and \( S \) denotes the size of the particle swarm.

Remark 3.1: In the PSO-ADV algorithm, the velocity of the next iteration is updated according to not only the velocity of the current iteration but also the velocity of the previous iteration, i.e. the delayed velocity. This scheme can bring some ‘useful’ disturbances to the convergence process of the algorithm, and hence make the particle more likely to fly out of the local optima of the searching space. Besides, the inertia weights of \( V_i(k) \) and \( V_i(k-1) \) are adaptively regulated depending on the evolutionary state of the swarm, which makes it possible for the particle to exploit and explore the whole searching space more thoroughly so as to obtain an optimum of the global space.

In order to appraise the performance of the PSO-ADV algorithm, several simulation experiments have been implemented in this paper upon a few standard benchmark functions, which can be expressed by Equations (26)–(31), and all of them are frequently used evaluation functions that are fairly difficult to attain the optimum solutions. TABLE 1 shows the configuration information of the benchmark functions, including the name of each function, dimension of each particle swarm, searching range of each dimension, theoretical optimal solutions. TABLE 1 shows the configuration information of the benchmark functions, including the name of each function, dimension of each particle swarm, searching range of each dimension, theoretical optimal solution of each benchmark function, and threshold of the successful optimum solution.

\[
f_1(x) = \sum_{i=1}^{D} x_i^4, \quad (26)
\]

\[
f_2(x) = \sum_{i=1}^{D-1} \left( (x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2 \right), \quad (27)
\]

\[
f_3(x) = \frac{\pi}{D} \left( 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right) + \sum_{i=1}^{D} u(x_i), \quad (28)
\]

\[
u(x_i) = \begin{cases} 100(-x_i - 10)^4, & x_i < -10, \\ 0, & |x_i| \leq 10, \\ 100(x_i - 10)^4, & x_i > 10, \\ \end{cases}
\]

\[
y_i = 1 + 1/4(x_i + 1),
\]

\[
f_4(x) = \sum_{i=1}^{D} u(x_i) + 0.1 \left( \sin^2(3\pi x_1) + (x_D - 1)^2 \right.
\]

\[
\left. + \sum_{i=1}^{D-1} (x_D - 1)^2 [1 + \sin^2(\pi x_{i+1})] \right), \quad (29)
\]

\[
u(x_i) = \begin{cases} 100(-x_i - 5)^4, & x_i < -5, \\ 0, & |x_i| \leq 5, \\ 100(x_i - 5)^4, & x_i > 5, \\ \end{cases}
\]

\[
f_5(x) = 418.9829D - \sum_{i=1}^{D} x_i \sin(\sqrt{|x_i|}), \quad (30)
\]

\[
f_6(x) = 1 + \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}). \quad (31)
\]

To verify the superiority of the novel PSO-ADV algorithm, its performance is compared with some well-known
Table 1. Configuration information of the benchmark functions.

| Function | Name      | Dimension | Searching space | Optimum | Threshold |
|----------|-----------|-----------|-----------------|---------|-----------|
| $f_1(x)$ | Sphere    | 20        | $[-100, 100]^D$ | 0       | 0.01      |
| $f_2(x)$ | Rosenbrock | 20        | $[-30, 30]^D$   | 0       | 100       |
| $f_3(x)$ | Rastrigin  | 20        | $[-5.12, 5.12]^D$ | 0       | 50        |
| $f_4(x)$ | Schwefel 2.22 | 20 | $[-10, 10]^D$ | 0       | 0.01      |
| $f_5(x)$ | Schwefel 1.2 | 20 | $[-100, 100]^D$ | 0       | 0.01      |
| $f_6(x)$ | Schwefel 2.21 | 20 | $[-100, 100]^D$ | 0       | 0.01      |

standard or modified PSOs, including PSO-LDIW (Shi & Eberhart, 1998, may), PSO-TVAC (Ratnaweera et al., 2004), SDPSO (Zeng, Wang, et al., 2016), FOPSO (Pires et al., 2010), and MDPSO (Song et al., 2017). For all simulation experiments, their parameters are taken as follows: $R = 20$ for the repetition of each experiment; $N = 20000$ for the maximum iteration of each algorithm; $D = 20$ for the dimension of the particle, and $S = 40$ for the size of the particle swarm.
As shown in Figures 3–8, the logarithmic average fitness values of each iteration are clearly demonstrated with diverse line and mark styles for all benchmark functions. Besides, the statistical results of the optimal fitness values, including the minimum (MIN), average value (AVG), standard deviation (SD), and success rate (SR) of the results, have also been illustrated in TABLE 2. Obviously, it can be concluded from the performance comparisons that PSO-ADV is superior to other standard or modified PSOs. For the \textit{Sphere} function, which is a typical unimodal function to test the convergence ability of an intelligent optimization algorithm, PSO-ADV can outperform all other PSOs on the obtained optimum. For the \textit{Rosenbrock} function, which is often regarded as a multimodal benchmark function fairly difficult to obtain the optimal solution because of its famous banana-valley, PSO-ADV is superior to others on two aspects of the convergence ability, i.e. obtainable optimum and convergence rate. For the \textit{Rastrigin} function, which is a typical complex multimodal benchmark function hard to converge, the performance of PSO-ADV can further verify its superiority in comparison with other algorithms. Moreover, the superior performance of PSO-ADV can also be confirmed by the optimization results of the \textit{Schwefel 2.22}, \textit{Schwefel 1.2} and \textit{Schwefel 2.21} functions, all of which are severe benchmark functions very hard to achieve the optimum. Moreover, it is worth noting that the PSO-ADV can attain 100% convergence success rate and better optimization result than others for all benchmark functions. This is benefited by the introduction of the adaptive delayed velocity, which can more likely make the particles jump out of the local trapping and robustly converge to the optimum, rather than some other PSOs that can trap into several local minima or diverge to the boundary of the searching space sometimes.

4. Simulation experiments

In this section, several simulation experiments are carried out to affirm the effectiveness and superiority of the presented smooth path planning approach, which is combined with the PCBC and the PSO-ADV algorithms. For the sake of contrast, the path planning algorithm is implemented in the workspace sketched in Figure 2, and the configuration of the experiments are as follows: $N = 7$ for the amount of control points; $S = 50$ for the size of particle swarm; and $T = 100$ for the maximum iteration of each simulation experiment.

Figures 9 and 10 demonstrate two of the representative simulation results, whose starts and destinations are respectively set as Grid 15 (respectively, Grid 0) to Grid 240 (respectively, Grid 255), and their X-Y-coordinate values are specified as (155, 5) for Grid 15, (5, 5) for Grid 0, (5, 155) for Grid 240, and (155, 155) for Grid 240. In these figures, the control points of the Bezier curves are indicated by the blue hollow circles; the corresponding convex hulls are depicted by the blue solid lines; and the red solid curves represent the obtained optimal smooth path.

To compare the smooth paths produced via different approaches, Figures 11 and 12 illustrate the counterpart results generated by using square Bezier curve (SBC) segments and MDPSO algorithm. Essentially, the smooth paths in Figure 9 (respectively, Figure 11) and Figure 10 (respectively, Figure 12) have different properties, though they look like each other. For example, both of the smooth paths in Figures 9 and 10 are composed of many cubic Bezier transition curve segments, which have
equal curvatures at the segment joints. Thus, the $G^2$-path can be acquired along the whole Bezier curve paths. In contrast, the smooth paths in Figures 11 and 12 consist of square Bezier curve segments, which can not satisfy the curvature continuity at the joints of the curve segments. Undoubtedly, this will cause discontinuities of the velocity and acceleration of the mobile robot, and can result in over actuation and slippage in the case of fast moving. The aforementioned properties can also be confirmed by the curvature curves of these smooth paths described in Figures 13 and 14, where the curvatures of the smooth paths are computed for every segments and the curvature of each segment is sampled on 100 uniformly distributed path points. It is apparent that there are several ‘steps’ on the curvature curves of the smooth paths devised by using SBC and MDPSO algorithms (see e.g. the points 200, 400, 600, and 800), which can not satisfy the requirement of continuous movement of the mobile robot and will lead to inevitable frequent switches on the motion states of the mobile robot. While, the continuous curvature of the smooth paths (which have the same curvature ‘0’ at the segment joints) presented in this paper can ensure the movement continuity of the mobile

Table 2. Statistical results of the optimal fitness values.

|       | PSO-LDIW | PSO-TVAC | SDPSO | FOPSO | MDPSO | PSO-ADV |
|-------|----------|----------|-------|-------|-------|---------|
| $f_1(x)$ MIN | 9.07e-243 | 3.45e-323 | 8.75e-18 | 2.60e-304 | 7.05e-200 | 9.88e-324 |
| AVG | 2.59e-230 | 1.22e-215 | 3.48e-11 | 3.20e-215 | 2.77e-151 | 0.00 |
| SD | 0.00 | 0.00 | 1.18e-10 | 0.00 | 1.23e-150 | 0.00 |
| SR | 100% | 100% | 100% | 100% | 100% | 100% |
| $f_2(y)$ MIN | 5.02e-2 | 3.76e-7 | 3.05 | 7.99e-6 | 7.89e-4 | 1.46e-14 |
| AVG | 4.53e+3 | 7.23 | 1.99e+1 | 3.67 | 7.00 | 2.81 |
| SD | 2.01e+4 | 5.59 | 2.83e+1 | 4.45 | 3.46 | 3.69 |
| SR | 80% | 100% | 90% | 100% | 100% | 100% |
| $f_3(z)$ MIN | 9.94e-1 | 1.98 | 3.98 | 1.98 | 2.98 | 2.98 |
| AVG | 7.86 | 6.96 | 1.51e+1 | 6.51 | 7.01 | 6.06 |
| SD | 9.06 | 2.25 | 1.02e+1 | 2.98 | 2.93 | 1.79 |
| SR | 100% | 100% | 100% | 100% | 100% | 100% |
| $f_4(w)$ MIN | 1.60e-151 | 1.70e-83 | 9.60e-10 | 3.51e-95 | 9.36e-75 | 5.36e-157 |
| AVG | 7.00 | 4.01e-49 | 2.00 | 7.25e-56 | 2.16e-56 | 2.78e-128 |
| SD | 8.01 | 1.79e-48 | 5.23 | 3.24e-55 | 6.55e-56 | 1.23e-127 |
| SR | 50% | 100% | 100% | 100% | 100% | 100% |
| $f_5(x)$ MIN | 1.69e-32 | 1.42e-73 | 2.65e-2 | 8.75e-67 | 1.08e-61 | 2.44e-156 |
| AVG | 1.25e-3 | 1.36e-51 | 9.23e-1 | 2.88e-42 | 9.31e-47 | 4.08e-137 |
| SD | 2.75e-3 | 6.01e-51 | 1.18 | 8.89e-42 | 3.77e-46 | 1.82e-136 |
| SR | 80% | 100% | 0% | 100% | 100% | 100% |
| $f_6(y)$ MIN | 9.23e-24 | 5.90e-32 | 5.01e-4 | 3.11e-39 | 2.12e-28 | 1.36e-83 |
| AVG | 7.37e-21 | 4.15e-26 | 8.57e-3 | 6.69e-34 | 1.85e-23 | 9.13e-78 |
| SD | 8.98e-21 | 1.43e-25 | 1.02e-2 | 2.74e-33 | 4.29e-23 | 3.03e-77 |
| SR | 100% | 100% | 80% | 100% | 100% | 100% |

Figure 9. Smooth path I (PCBC+PSO-ADV).

Figure 10. Smooth path I (PCBC+PSO-ADV).
robot, and therefore make the trajectory tracking of the mobile robot more easier.

Remark 4.1: It should be highlighted that it is really a challenging issue to plan the smooth path in the cases of this paper. The difficulty and complexity come from at least the following two aspects. Firstly, there is only a narrow valley surrounded by obstacles for the optimizer to find a feasible smooth path for the mobile robot, which is a global optimization problem that it prone to trap into the local optima, i.e. infeasible smooth paths. Secondly, several criteria and constraints are linked to the smooth path optimization, which is actually a multi-objective optimization problem that is difficult to obtain the global optimum. It is fortunate that the challenging task can be fully accomplished via the approach presented in this paper, which can acquire superior performance comparing with the results of our previous papers.

5. Conclusions

In this paper, a new approach combined with PCBC and PSO-ADV algorithms is developed to plan the smooth path of mobile robots. The smooth path composed of PCBC segments can achieve equivalent curvature at the joints of the segments, and thus it is able to attain continuous curvature along the whole smooth path. Based on the mathematical formulation of the smooth path planning problem, the optimal smooth path can be obtained by regulating the control points and control parameters.
of PCBC, which is essentially an intractable optimization problem. To tackle this optimization problem and some frequently encountered troubles (e.g. premature convergence and local trapping), a novel PSO-ADV algorithm is developed by blending an adaptive delayed velocity term that can give the particles more ‘power’ to jump out of the local minima, and the validity of PSO-ADV can be affirmed on account of some simulation experiments on several famous benchmark functions. The new approach is finally applied to the smooth path planning of mobile robots, and its effectiveness and superiority can be verified through several simulation experiments.

In the future work, we will pay more attentions to some interesting topics including: (1) the convergence analysis of the PSO-ADV algorithm; (2) the new schemes that can be employed to boost the performance of PSO; (3) the application of the new smooth path planning approach to some more complicated cases, e.g. 3-D path planning (Sun et al., 2018), multi-robot path planning (Nazarahari et al., 2019), and path planning in severe environment (Berglund et al., 2010), etc.

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**Funding**

This work was supported in part by the National Natural Science Foundation of China under Grant 61703242, and the Natural Science Foundation of Shandong Province China under Grant ZR2020MF071.

**References**

Akbarimajd, A., & Hassanzadeh, A. (2012). Autonomously implemented versatile path planning for mobile robots based on cellular automata and ant colony. *International Journal of Computational Intelligence Systems, 5*(1), 39–52. https://doi.org/10.18756/j.ijciss.2012.670520

Atyabi, A., & Powers, D. M. W. (2013). Review of classical and heuristic-based navigation and path planning approaches. *International Journal of Advancements in Computing Technology, 5*, 1–14.

Bakid, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., & Bouzouia, B. (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robotics and Autonomous Systems, 89*, 95–109. https://doi.org/10.1016/j.robot.2016.12.008

Behnamian, J., & Fatemi Ghomi, S. (2010). Development of a PSO-SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. *Expert Systems with Applications, 37*(2), 974–984. https://doi.org/10.1016/j.eswa.2009.05.079

Berglund, T., Brodnik, A., Jonsson, H., Staffanson, M., & Soderquist, I. (2010). Planning smooth and obstacle-avoidance B-splines paths for autonomous mining vehicles. *IEEE Transactions on Automation Science and Engineering, 7*(1), 167–172. https://doi.org/10.1109/TASE.2009.2015886

Castillo, O., Trujillo, L., & Melin, P. (2007). Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing, 11*(3), 269–279. https://doi.org/10.1007/s00500-006-0068-4

Chang, H., & Liu, J. (2009, February). High-quality path planning for autonomous mobile robots with n3-splines and parallel genetic algorithms. In *Proceedings of IEEE international conference on robotics and biomimetics* (pp. 1671–1677). IEEE Press.

Clerc, M., & Kennedy, J. (2002). The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73. https://doi.org/10.1109/4235.985692

Contreras-Cruz, M. A., Ayala-Ramirez, V., & Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing, 30*, 319–328. https://doi.org/10.1016/j.asoc.2015.01.067

Dong, W., Ding, Y., Huang, J., Yang, L., & Zhu, X. (2019). An optimal curvature smoothing method and the associated real-time interpolation for the trajectory generation of flying robots. *Robotics and Autonomous Systems, 115*, 73–82. https://doi.org/10.1016/j.robot.2019.02.004

Elbanhawi, M., & Simic, M. (2014). Sampling-based robot motion planning: A review. *IEEE Access, 2*(1), 56–77. https://doi.org/10.1109/ACCESS.2014.2302442

Fong, S., Deb, S., & Chaudhary, A. (2015). A review of metaheuristics in robotics. *Computers & Electrical Engineering, 43*, 278–291. https://doi.org/10.1016/j.compeleceng.2015.01.009

Gonzalez, D., Perez, J., Milanes, V., & Nashashibi, F. (2016). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems, 17*(4), 1135–1145. https://doi.org/10.1109/TITS.2015.2498841

Huang, H.-C. (2014). FPGA-based parallel metaheuristic PSO algorithm and its application to global path planning for autonomous robot navigation. *Journal of Intelligent & Robotic Systems, 76*, 475–488. https://doi.org/10.1007/s10866-013-9884-9

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of IEEE international conference on neural network* (pp. 1942–1948). IEEE Press.

Liang, J., Qin, A., Suganthan, P., & Baskar, S. (2006). Comprehensive learning particle optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation, 10*(3), 281–295. https://doi.org/10.1109/TEVC.2005.857610

Lin, D., Shen, B., Liu, Y., Alsaadi, F. E., & Alsaedi, A. (2017). Genetic algorithm-based compliant robot path planning: An improved Bi-RRT-based initialization method. *Assembly Automation, 37*(3), 261–270. https://doi.org/10.1108/AA-12-2016-173

Liu, Y., Xu, W., Dobaie, A. M., & Zhuang, Y. (2018). Autonomous road detection and modeling for UGVs using vision-laser data fusion. *Neurocomputing, 275*, 2752–2761. https://doi.org/10.1016/j.neucom.2017.11.042
Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation, 8*(3), 204–210. https://doi.org/10.1109/TEVC.2004.826074

Nazarhari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications, 115*, 106–120. https://doi.org/10.1016/j.esswa.2018.08.008

On, S., & Yazici, A. (2011, July). A comparative study of smooth path planning for a mobile robot considering kinematic constraints. In *Proceedings of IEEE international symposium on innovations in intelligent systems and applications* (pp. 565–569). IEEE Press.

Patle, B. K., Pandey, A., Jagadeesh, A., & Parhi, D. R. (2018). Path planning in uncertain environment by using firefly algorithm. *Defence Technology, 14*(6), 691–701. https://doi.org/10.1016/j.dt.2018.06.004

Piazzi, A., Bianco, C. G. L., & Romano, M. (2007). $\eta^3$-splines for the smooth path generation of wheeled mobile robots. *IEEE Transactions on Robotics, 23*(5), 1089–1095. https://doi.org/10.1109/TRO.2007.903816

Pires, E. J. S., Machado, J. A. T., Oliveira, P. B. M., Cumha, J. B., & Mendes, L. (2010). Particle swarm optimization with fractional-order velocity. *Nonlinear Dynamics, 61*, 295–301. https://doi.org/10.1007/s11071-009-9649-y

Qu, H., Xing, K., & Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing, 120*, 509–517. https://doi.org/10.1016/j.neucom.2013.04.020

Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*(3), 240–255. https://doi.org/10.1109/TEVC.2004.826071

Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In *Proceedings of IEEE international conference on evolutionary computation* (pp. 69–73). IEEE Press.

Shuang, B., Chen, J., & Li, Z. (2011). Study on hybrid PS-ACO algorithm. *Applied Intelligence, 34*(1), 64–73. https://doi.org/10.1007/s10489-009-0179-6

Song, B., Wang, Z., & Sheng, L. (2016). A new genetic algorithm approach to smooth path planning for mobile robots. *Assembly Automation, 36*(2), 138–145. https://doi.org/10.1108/AA-11-2015-094

Song, B., Wang, Z., & Zou, L. (2017). On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. *Cognitive Computation, 9*(1), 5–17. https://doi.org/10.1007/s12559-016-9442-4

Sun, B., Zhu, D., & Yang, S. X. (2018). An optimized fuzzy control algorithm for three-dimensional auv path planning. *International Journal of Fuzzy Systems, 20*(2), 597–610. https://doi.org/10.1007/s40815-017-0403-1

Tang, Y., Wang, Z., & Fang, J. (2011). Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm. *Expert Systems with Applications, 38*, 2523–2535. https://doi.org/10.1016/j.eswa.2010.08.041

Tang, B., Zhu, Z., & Luo, J. (2016). Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning. *International Journal of Advanced Robotic Systems, 13*, 1–17. https://doi.org/10.5772/62058

Xu, L., Wang, D., Song, B., & Cao, M. (2017, October). Global smooth path planning for mobile robots based on continuous Bezier curve. In *Proceedings of 2017 Chinese automation congress* (pp. 2081–2085). IEEE Press.

Yang, K., & Sukkarieh, S. (2010). An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics, 26*(3), 561–568. https://doi.org/10.1109/TRO.2010.2042990

Zeng, N., Wang, Z., Zhang, H., & Alsaaedi, F. E. (2016). A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay. *Cognitive Computation, 8*, 143–152. https://doi.org/10.1007/s12559-016-9396-6

Zeng, N., Zhang, H., Chen, Y., Chen, B., & Liu, Y. (2016). Path planning for intelligent robot based on switching local evolutionary PSO algorithm. *Assembly Automation, 36*(2), 120–126. https://doi.org/10.1108/AA-10-2015-079

Zhan, Z., Zhang, J., Li, Y., & Chung, H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39*(6), 1362–1381. https://doi.org/10.1109/TSMCB.2009.2015956

Zhang, S., Sun, L., Chen, Z., Lu, X., & Liu, J. (2014, December). Smooth path planning for a home service robot using $\eta^3$-splines. In *Proceedings of IEEE international conference on robotics and biomedicine* (pp. 1910–1915). IEEE Press.

Zhou, F., Song, B., & Tian, G. (2011). Bezier curve based smooth path planning for mobile robot. *Journal of Information and Computational Science, 8*(1), 2441–2450.