Research Article

Path Planning for Mobile Objects in Four-Dimension Based on Particle Swarm Optimization Method with Penalty Function

Yong Ma,1 M. Zamirian,2 Yadong Yang,1 Yanmin Xu,1 and Jing Zhang3

1 Navigation College, Wuhan University of Technology, Wuhan, Hubei 430063, China
2 Department of Mathematics, Islamic Azad University, Bojnourd Branch, Bojnourd 94186-54145, Iran
3 Chutian College, Huazhong Agricultural University, Wuhan, Hubei 430205, China

Correspondence should be addressed to Yanmin Xu; whyanmin@163.com

Received 4 December 2012; Accepted 1 January 2013

Academic Editor: Yang Tang

Copyright © 2013 Yong Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present one algorithm based on particle swarm optimization (PSO) with penalty function to determine the conflict-free path for mobile objects in four-dimension (three spatial and one-time dimensions) with obstacles. The shortest path of the mobile object is set as goal function, which is constrained by conflict-free criterion, path smoothness, and velocity and acceleration requirements. This problem is formulated as a calculus of variation problem (CVP). With parametrization method, the CVP is converted to a time-varying nonlinear programming problem (TNLPP). Constraints of TNLPP are transformed to general TNLPP without any constraints through penalty functions. Then, by using a little calculations and applying the algorithm PSO, the solution of the CVP is consequently obtained. Approach efficiency is confirmed by numerical examples.

1. Introduction

Mobile objects such as autonomous unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs) [1–6] have been applied for specific utilities in the engineering application field, especially in offshore drilling, spacewalk, and so forth. To guarantee the successful operation of above kinds of mobile objects, the problem of path planning for mobile objects in four-dimension (three spatial and one-time dimensions) becomes indispensable and ever-increasingly important. Therefore, we focus our investigations on the path planning for mobile objects in the presence of obstacles in four-dimension.

Path planning in a time-varying environment with static or moving obstacles is inherently hard [7, 8]. Even for a simple case in two dimensions, the problem is NP-hard and is not solvable in polynomial time [8–10]. Our addressed problem is characterized by objects dynamically moving in large outdoor space denoted as four-dimension \((x_1, x_2, x_3, t)\) with obstacles, where \((x_1, x_2, x_3)\) denotes the three-dimensional space and \(t\) denotes the temporal dimension. Therefore, in this problem, we should consider the constraints of safety, velocity and acceleration brought by the influence among obstacles and objects, and the curse of dimensionality [2] brought by the time-varying motion in three spatial dimensions. This fact makes path planning for mobile object in four-dimension with obstacles a challenging research problem [9].

Generous excellent methods have been applied to path planning problems. Many recent researches have focused on the gird-based techniques [3, 5, 8]. However, lots of broken lines and excessive interrupted turnings have existed in the planned path with grid-based approaches [8]. Then, gird-based approaches violate the constraint of path smoothness [11]. In [12], Bzier-curve-based approach has been adopted to achieve smooth path, but index of their curve is restricted to one certain number. In [7, 13], parametrization-based method is proposed to fulfill the ideal path and their solution is obtained through software such as Lingo, Matlab with the embedded function. Due to the error of the embedded function of software itself, the resultant path in [7, 13] can be remarkably improved with some techniques.

In our problem, the goal is to minimize path length such that the path is smooth and safe. This goal is formulated as the calculus of a variation problem (CVP) whose variable is
the path \( x(\cdot) \) of the mobile object. Then using parametrization method [7, 13] and some calculations, the CVP is converted to the sequence of TNLPPs whose variable is a polynomial function with unknown constant coefficients. After some calculation, the TNLPP is equivalent to a conventional nonlinear programming problem (NLPP).

In path planning problem, classic methods proved to be inefficient for high-dimension space, requiring considerably long time and huge storage memory. Consequently, heuristic methods are developed to cope with the curse of dimensionality in path planning problem. Many efficient metaheuristic approaches such as simulated annealing (SA), genetic algorithm (GA), ant colony optimization (ACO), taboo search (TS), and particle swarm optimization (PSO) are applied to path planning problems. PSO is one prevalent algorithm founded by Kennedy and Eberhart [14]. For the last decades, PSO has been extensively used in the field of path planning in two dimensions [15, 16]. PSO is a swarm intelligence method inspired by the collective behavior of birds flock. PSO algorithm is famous for its concise mechanism, fine convergence, and little computational time [17–19]. Up to the present time, little attention has been paid to path planning in four-dimension with PSO. To achieve the much better solution than [4, 13], we extend PSO to solve the above TNLPP.

As fitness function in two-dimension cannot be directly used in four-dimension, the foremost issue of PSO is how to determine the proper fitness function for our problem. The TNLPP is subject to path smoothness, safety criterion, and velocity and acceleration constraints, which can be classified as multiobject optimization problem. To devise the rational fitness function, penalty function is adopted to transform the constrained TNLPP to general unconstrained TNLPP. The constraints are placed into the fitness function via the penalty parameter in such a way as to penalize any violation of the constraints. Then, solution of the TNLPP can be attained with PSO. The advantage of our approach is that the paths for mobile objects in four-dimension can be achieved promptly and the quality of the paths is high in the perspective of path length and smooth. Within the algorithm based on PSO, the population can converge to the feasible region and reach the ideal fitness value effectively.

The rest of this paper is organized as follows. Section 2 describes the problem formulation. Section 3 introduces penalty function and PSO for path planning of mobile objects. Section 4 presents illustrative path planning examples for mobile objects in four-dimension using the proposed approach. Finally, Section 5 presents conclusions.

### 2. Problem Formulation

Compared with former researches in [1, 6, 20, 21] that treated the objects as points, the actual shape of mobile objects and obstacles are taken into account, which would be more suitable for practicality. Consider a single rigid and free moving object \( A \) with the center \( x(t) = (x_1(t), x_2(t), x_3(t)) \), \( t \in [0, t_f] \), moving from \( x_0 \in \mathbb{R}^3 \) at time 0 to \( x_f \in \mathbb{R}^3 \) at time \( t_f \) in the presence of obstacles. Where \( x(\cdot) \) is a continuously unknown differentiable real vector-valued function, which is the path of \( A \). \( t_f \) is a given real number as final moving time of \( A \), and \( x_0 = x(0), x_f = x(t_f) \). We suppose obstacle \( obs_k \) is a rigid object with the center \( \alpha_k(t) = (a_{k1}(t), a_{k2}(t), a_{k3}(t)), k = 1, 2, \ldots, q, t \in [0, t_k] \), where \( \alpha_k(\cdot), k = 1, 2, \ldots, q \) are known continuous real valued functions, which are the paths of motion obstacles, and \( t_k \) is a given real number as final moving time of \( obs_k \). Mobile objects and all obstacles are simplified as spheres. The radius of \( A \) and obstacles can be denoted as \( r_A \) and \( r_k, k = 1, 2, \ldots, q \), respectively. In Figure 1, the two smaller spheres are the start and final position of \( A \), respectively; the other two larger spheres stand for obstacles.

![Figure 1: Mobile object A and obstacles obs_k, k = 1, 2, …, q.](image)

Planned path \( x(\cdot) \) of \( A \) should comply with the following constraints: the dynamic constraints of \( A \), the reachable of \( x(\cdot) \), and the safety distance between \( A \) and obstacles. Satisfied with all above constraints, the goal is to minimize the length of \( x(\cdot) \).

To satisfy dynamic constraints of the mobile object \( A \), the velocity and acceleration of \( A \) are restricted to certain regions. Thus, we suppose \( x(\cdot) \in X = \{x(t)|x(t) \in C^1(0, t_f)\), \( a(t) \leq x(t) \leq b(t), c(t) \leq \dot{x}(t) \leq d(t), e(t) \leq \ddot{x}(t) \leq f(t), x(0) = x_0, x(t_f) = x_f, t \in [0, t_f]\} \), where \( a(t), b(t), c(t), d(t), e(t), f(t) \), and \( f(t) \) all belong to \( \mathbb{R}^3 \), are known as continuous real vector-valued functions as the boundaries of \( x(t), \dot{x}(t) \), and \( \ddot{x}(t) \) for all \( t \in [0, t_f] \), respectively.

The reachable of \( x(\cdot) \) means mobile object \( A \) should follow the \( x(\cdot) \) from \( x_0 \) to \( x_f \). Then, criteria of dynamic and reachable are correspondingly given as below:

\[
x(\cdot) \in X.
\]
The smoothness of \( x(\cdot) \) is satisfied inherently, that is, the planned path is introduced by a polynomial function belonging to \( C^\infty[0,t_f] \) (the set of highly smooth functions).

For the safety distance criterion, set
\[
\varphi_k(x(t)) = \left\| x(t) - \alpha_k(t) \right\|_2 - (r_A + r_k),
\]
where \( \varphi_k(x(t)) \), \( k = 1, 2, \ldots, q \), is the distance between object \( A \) (or the planned path \( x(\cdot) \)) and obstacle \( k \) at the moment \( t \). Set \( d_k \) is a given safety distance between \( A \) and \( \text{obs}_k \), which guarantees \( A \) and \( \text{obs}_k \) are free of collision in the motion of \( A \). Then, the distances between \( A \) and obstacles \( \text{obs}_k \) are denoted as \( (d_1, \ldots, d_k, \ldots, d_q) \), where \( k = 1, 2, \ldots, q \). Then, the safety criterion is listed as below:
\[
\varphi_k(x(t)) \geq d_k.
\]

Length of \( x(\cdot) \) can be expressed as follows:
\[
L_0(x(t_f)) = \int_0^{t_f} \sqrt{\dot{x}_1^2(t) + \dot{x}_2^2(t) + \dot{x}_3^2(t)} dt = \int_0^{t_f} \left\| \dot{x}(t) \right\|_2 dt.
\]

Then, in the environment \( X \) to achieve the minimum path length of \( A \) named \( L_0 \), which is constrained by above criteria, for \( k = 1, \ldots, q, t \in [0, t_f] \), the following CVP is defined:
\[
\min L_0(x(t_f)) = \int_0^{t_f} \left\| \dot{x}(t) \right\|_2 dt
\]
\[
\text{s.t.} \quad \begin{cases} x(\cdot) \in X \\ \varphi_k(x(t)) \geq d_k. \end{cases}
\]

Similar to the former work in [7, 13], for all \( t \in [0, t_f] \), we set \( p_n(t) = (p_{n1}(t), p_{n2}(t), p_{n3}(t)) \), where \( p_{nj}(\cdot), j = 1, 2, 3 \) are polynomials of degree at most \( n \) with unknown constant coefficients. Then, with parametrization method, by substituting \( p_n(\cdot) \) for \( x(\cdot) \) in the problems (5) and (6), the sequence of the TNLPP is obtained as follows:
\[
\min L_0(p_n(t_f)) = \int_0^{t_f} \left\| \dot{p}_n(t) \right\|_2 dt
\]
\[
\text{s.t.} \quad \begin{cases} p_n(\cdot) \in X \\ \varphi_k(p_n(t)) \geq d_k \quad k = 1, \ldots, q, t \in [0, t_f] \end{cases}
\]

Suppose \( Q \) is the set of \( x(\cdot) \) such that the CVP (5) and (6) is feasible and \( Q(n) \) is the set of \( p_n(\cdot) \) such that the TNLPP (7) and (8) is feasible. Also, we suppose \( Q \) and \( Q(n) \) are not empty. Then, by the following theorem is proven that the sequence of the solutions of the problems (7) and (8) converges to the solution of the problems (5) and (6).

**Theorem 1.** If \( \eta = \inf_Q I_0(x(t_f)) \) and \( \eta(n) = \inf_Q I_0(p_n(t_f)) \), Then \( \eta = \lim_{n \to \infty} \eta(n) \).

*Proof. See [13].*

When all constraints of \( A \) are considered, such as bounded velocity and acceleration constraints, reachable of \( x(\cdot) \), and safety criterion, TNLPP (7) and (8) is transformed as follows:
\[
\min \int_0^{t_f} \left\| \dot{p}_n(t) \right\|_2 dt
\]
\[
\begin{cases}
p_n(t) \geq a(t) \\ p_n(t) \leq b(t) \\ \dot{p}_n(t) \geq c(t) \\ \dot{p}_n(t) \leq d(t) \\ p_n(t) \geq e(t) \\ \dot{p}_n(t) \leq f(t) \\ p_n(0) = x_0, p_n(t_f) = x_f \\ \varphi_k(p_n(t)) \geq d_k \end{cases}
\quad k = 1, \ldots, q.
\]

Now, we partition the interval \( [0, t_f] \) into \( M \) equal parts as \( h = t_f/M \), set \( E(t) = \left\| \dot{p}_n(t) \right\|_2 \). Thus, by using a numerical integration method such as trapezoidal rule, for \( k = 1, \ldots, q \), problems (9) and (10) are converted to the following problem:
\[
\min h \left( 2 \sum_{m=0}^{M} E(mh) - E(0) - E(t_f) \right)
\]
\[
\begin{cases}
p_n(mh) \geq a(mh) \\ p_n(mh) \leq b(mh) \\ \dot{p}_n(mh) \geq c(mh) \\ \dot{p}_n(mh) \leq d(mh) \\ p_n(mh) \geq e(mh) \\ \dot{p}_n(mh) \leq f(mh) \\ p_n(0) = x_0, p_n(t_f) = x_f \\ \varphi_k(p_n(mh)) \geq d_k \end{cases}
\quad k = 1, \ldots, q.
\]

**Theorem 2.** Solutions of the problems (11), (12), (9), and (10) are the same, if in problems (11) and (12) \( M \) tends to infinity.

*Proof. See [22].*

With parametrization method, our path planning problem in four-dimension passages from the CVP to TNLPP. When taking into account all constraints, the TNLPP is converted into NLPP. Thus, how to achieve the ideal solution of this NLPP becomes the first and foremost problem. In the following section, we propose the PSO with penalty function to resolve this NLPP.

### 3. Determine the Optimal Path for Objects in Four-Dimension Based on PSO with Penalty Function

Generally, many softwares such as lingo, Matlab can be used to solve above NLPP with their embedded functions. In [7, 13], fine results have been reported with the embedded functions of above softwares. However, due to the error of the functions themselves, the obtained resolution which costs much computational time and spatial memory could not be the optimal one. Even in some complicated cases, no solution
can be generated owing to insufficiency of the functions. Thus, to resolve the path planning problem in four-dimension taking on character of optimization function with multiple constraints, the PSO with penalty function is proposed.

3.1. Multiconstrained NLPP Converts to Unconstrained NLPP with Penalty Function. In constrained optimization (CO) problems such as constrained TNLPP and NLPP, the search space consists of two kinds of points: feasible and unfeasible. Feasible points satisfy all the constraints, while unfeasible points violate at least one of them. Penalty function technique solves CO problems through a sequence of unconstrained optimization problems (UOPs) [23–25]. With proper penalty parameters, which could evaluate the degree of penalty, the constraints are deleted and added into the goal function. Thus, CO problems are changed into general UOP with penalty function.

Based on penalty function, the CO problems for path planning of mobile objects in four-dimension are converted to the following UOP. Then, for \( t' = m h \), expression of the UOP can be adopted as the fitness function of PSO as follows:

\[
\min F_{\text{pena}} = \frac{1}{2} \left[ \sum_{m=1}^{M} \left( E(T') - E(0) - E(t_f) \right) \right] + c_1 \sum_{m=0}^{M} \max(0, a(t') - p_n(t')) \\
+ c_2 \sum_{m=0}^{M} \max(0, p_n(t') - b(t')) \\
+ c_3 \sum_{m=0}^{M} \max(0, c(t') - \hat{p}_n(t')) \\
+ c_4 \sum_{m=0}^{M} \max(0, \hat{p}_n(t') - d(t')) \\
+ c_5 \sum_{m=0}^{M} \max(0, e(t') - \tilde{p}_n(t')) \\
+ c_6 \sum_{m=0}^{M} \max(0, \tilde{p}_n(t') - f(t')) \\
+ c_7 \max(0, \| p_n(0) - x_0 \|) \\
+ c_8 \max(0, \| p_n(t_f) - x_f \|) \\
+ c_9 \sum_{k=1}^{M} \max(0, d_k - q_k (p_n(t'))) 
\]

In (13), \( c_1, c_2, \ldots, c_9 \) and \( c_9 \) stand for large scalars to penalize the violation of the constraints. \( c_1 \) and \( c_2 \) are the lower and higher penalty coefficients of region boundary of the mobile object, respectively; \( c_3 \) and \( c_4 \) are the penalty coefficients of velocities boundary, respectively; \( c_5 \) and \( c_6 \) are the penalty coefficients of acceleration boundary, respectively; \( c_7 \) and \( c_8 \) are the penalty coefficients of reachable constraints, respectively; and \( c_9 \) is the penalty coefficients of safety constraints. All the constraints should be satisfied well. Thus, on the one hand, the penalty function parameters would reflect the equal importance of each constraint, and on the other hand, the defined parameters can lead the population within the feasible region. The ideal value of penalty parameters should be specified moderately which is not too large or too small. Then, when any constraint violation occurred, the penalty function parameters are defined as \( c_i = 100 \) \( (i = 1, 2, \ldots, 9) \). Then, the focus of the above minimization problem becomes how to determine the polynomials of \( P_3(t) \).

After ascertainment of the fitness function for PSO, the following section presents the solution for the above problem based on PSO.

3.2. Optimal Path for Mobile Objects in Four-Dimension with Obstacles Using PSO. In PSO, with particles size \( s \), particle \( i \) \((1 \leq i \leq s)\) stands for one potential solution of the given problem in \( d \)-dimensional space. Each particle \( i \) has three vectors at the \( l \)th iteration, current position vector \( u_i(l) = (u_i^1(l), u_i^2(l), \ldots, u_i^d(l)) \), velocity vector \( v_i(l) = (v_i^1(l), v_i^2(l), \ldots, v_i^d(l)) \), and its own best position vector \( p_i = (p_i^1, p_i^2, \ldots, p_i^d) \) which represents the best objective function value or fitness value. One global best position vector \( p_g = (p_g^1, p_g^2, \ldots, p_g^d) \) is defined as the position of the best particle among the \( s \) particles in the population. Figure 2 shows the iterative process of position vector \( u_i(l) \) in PSO.

When all the terms are stated, the standard PSO formula can be

\[
u_i^j(l+1) = w \nu_i^j(l) + c_1 r_1 (p_i^j - u_i^j(l)) + c_2 r_2 (p_g^j - u_i^j(l)),
\]

\[
u_i^j(l+1) = u_i^j(l) + v_i^j(l+1) \quad j = 1, 2, \ldots, d.
\]
In (14), the upper equation factor stands for the movement update of particle $i$ in dimension $j$ at the iteration number $l+1$, where $w$ is a control factor controlling the magnitude of $v_i^l$, $c_p$ and $c_g$ are positive acceleration coefficients, and $r_1$ and $r_2$ are uniform random numbers in $[0,1]$; the lower equation gives the corresponding position $u_i(l+1)$ update of particle $i$. Customarily, as in [26], we set the parameters to $c_p = c_g = 1$, and $w = 0.8$.

When it comes to our problem, the solution of $x_i$ corresponds to $p_k^i$, where $p_k^i = (p_k^i_{x_1}(t), p_k^i_{x_2}(t), p_k^i_{x_3}(t))$. We set the coefficients of $u_i$ correspond to $(p_{in}^i, p_{in}^i, p_{in}^i)$. Thus, the dimension of $u_i$, which denoted as $d$ equals to all three indexes summation of $p_{in}^i$, $p_{in}^i$, and $p_{in}^i$. For example, the highest sequential indexes of polynomials for $p_{in}^i$, $p_{in}^i$, and $p_{in}^i$ are 3, 4, and 5, respectively, then the value of $d$ is 12. With (11), after choosing proper parameters, the velocity and position update of particles can be implemented until the predefined conditions such as the generation number are satisfied. Thus, the achieved $u_i$ can be treated as the polynomial coefficients of $p_{in}^i$, and the corresponding value of fitness function should be the optimal path length $l_0$.

Basic flow of PSO algorithm named PSObasic is given as below, from which we can obtain the optimal solution of our path planning problem for mobile objects in four-dimension promptly. In PSObasic, fitness and $N$ represent fitness function $F_{pena}$ and iteration number, respectively. Means of $c_p$ and $c_p$, $s$, $d$, and $w$ are the same as depicted above.

**Algorithm**. PSObasic($f_{fitness}, s, c_p, c_p, w, N, d$)

**Input**: fitness function $f_{fitness}$, particle size $s$, acceleration coefficients $c_p$, $c_p$, inertia factor $w$, iteration number $N$, and spatial dimension $d$.

**Output**: the latest position $u_i$ and the optimal value of $f_{fitness}$ which represent the coefficients of $p_{in}^i$ and $l_0$, respectively.

1. Initialize $u_i$ and $v_i$ of all particles in the swarm.
2. $f_{fitness}(u_i^l); u_i^l; v_i^l \rightarrow p_i$.
3. if the particle $i$ with minimum $f_{fitness}$, then $v_i, u_i \rightarrow p_g$.
4. end if
5. $l = 1$.
6. for each particle $i$ in swarm while $l \leq N$ do
7. Update the $v_i^l, x_i^l$ with (14).
8. if $f_{fitness}(u_i(t)) \leq p_i$, then $p_i = f_{fitness}(u_i(t))$.
9. end if
10. if $p_i \leq p_g$, then $p_g = p_i$.
11. end if
12. $l = l + 1$;
13. end for each
14. return $u_i, f_{fitness}(u_i)$.

With algorithm PSObasic, the optimal planned path expressions $p_k^i$ and length $l_0$ of mobile object $A$ can be achieved which corresponding to the value of $u_i$ and $f_{fitness}(u_i)$, respectively.

### 4. Experimental Results

Simulations are provided to validate the effectiveness of the proposed PSO with penalty function, and path planning for mobile objects in four-dimension under different circumstances is discussed. We set $v(t) = \dot{x}(t)$, where $v(t)$ is interpreted as control function which shows the speed of object $A$ in the direction of $\dot{x}(t)$ at the moment $t \in [0, t_f]$. In the following figures, the scales are the same and all quantities conform to a given unit system, for instance, meters, per second, and so forth. In this paper, all the computations were run on a PC with CPU Intel Core 2 Duo and 2 GB of RAM, and all the codes are written in Matlab 7.0 software.

#### 4.1. Case Study with Stationary Obstacles in Four-Dimension

To prove the validity of the proposed approach, we test it using the case data in [13] and give the experiment results. Consider the mobile object $A$ in a space $\mathbb{R}^3$, in the presence of five stationary obstacles with the following information: $r = 0.01, x_0 = (0, 0, 0), x_f = (1, 1, 1), (0, 0, 0) \leq (x_1(t), x_2(t), x_3(t)) \leq (1, 1, 1), (0, 0, 2, 0.3) \leq (v_1(t), v_2(t), v_3(t)) \leq (7.5, 4, 3.5)$, where $t \in [0, 1]$. Table 1 gives the raw information data list of the obstacles.

With our proposed approach, the path length is 1.819 while in [13] the value is 1.844, and their approach was based on the embedded functions of softwares such as Matlab, lingo, and so forth. As shown in Figure 3, the planned path is smooth and conflict-free with the five obstacles. The control function denoted by velocity changes on time is illustrated in Figure 4, wherein the curves of velocity change meet the requirements of velocity constraints. The distances between $A$ and obstacles, $k = 1, \ldots, 5$ are shown in Figure 5, and all the curves reveal that the distances satisfy the predefined safety distances.

Results of this example demonstrate that our approach fits for path planning problems of mobile objects in four dimensions with static obstacles.

#### 4.2. Case Study with Stationary and Moving Obstacles in Four-Dimension

Circumstance settings of three dimensions 100*100*100 with static and moving obstacles are given as follows.

1. Parameters of mobile object $A$: radius $r = 1.5$, initial points of center position $x_0 = (70, 80, 20)$, final points of center position $x_f = (20, 35, 85)$.

### Table 1: Information of obstacles $O_{s_k}$.

<table>
<thead>
<tr>
<th>$O_{s_k}$</th>
<th>$(\alpha_{s_1}(t), \alpha_{s_2}(t), \alpha_{s_3}(t))$</th>
<th>$r_s$</th>
<th>$d_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs1</td>
<td>(0.5, 0.5, 0.5)</td>
<td>1/6</td>
<td>0.1</td>
</tr>
<tr>
<td>Obs2</td>
<td>(0.3, 0.3, 0.1)</td>
<td>1/5</td>
<td>0.15</td>
</tr>
<tr>
<td>Obs3</td>
<td>(0.8, 0.2, 0.6)</td>
<td>1/5</td>
<td>0.11</td>
</tr>
<tr>
<td>Obs4</td>
<td>(0.2, 0.2, 0.8)</td>
<td>1/5</td>
<td>0.12</td>
</tr>
<tr>
<td>Obs5</td>
<td>(0.2, 0.8, 0.4)</td>
<td>1/5</td>
<td>0.2</td>
</tr>
</tbody>
</table>
(2) Boundary conditions: velocity boundary 
\((-7, -5.5, 0) \leq (v_1, v_2, v_3) \leq (0, 0, 8)\). Time boundary 
\(T_{\text{max}} = 10\). Safety distance \(d_1 = 3, d_2 = 5, d_3 = 4\).

(3) Parameters of obstacles: radius \(r_1 = 5, r_2 = 3, r_3 = 4\).
Center position \(\alpha_1 = (30, 50, 40), \alpha_2 = (55, 70, 26), \alpha_3 = (42, 20, 51)\). Path of moving obstacle \(\text{Obs}_3\)

denoted by \(\alpha_3(t) = (\alpha_{13}(t), \alpha_{23}(t), \alpha_{33}(t)), t \in [0, 10]\)
is listed as follows:
\[
\begin{align*}
\alpha_{13}(t) & = 42 + 0.058t - 0.4011t^2 + 0.0175t^3, \\
\alpha_{23}(t) & = 20 + 5.0098t - 0.0236t^2 - 0.011t^3 - 0.00158t^4, \\
\alpha_{33}(t) & = 51 - 3.98t + 0.677t^2 - 0.130t^3 + 0.0070t^4.
\end{align*}
\]

(15)

After all constraints related to the motion of static and moving obstacles are taken into account, with transformation of parametrization method, the problems are resolved by algorithm \(\text{PSO}_{\text{basic}}\). For \(t \in [0, 10]\), paths of mobile object \(A\) are given as below:
\[
\begin{align*}
x_1(t) & = 70 - 5.26325t + 0.0263t^2, \\
x_2(t) & = 80 - 3.726t - 0.2798t^2 + 0.0202t^3, \\
x_3(t) & = 20 + 8.015t - 0.3326t^2 + 0.0557t^3 - 0.0038t^4.
\end{align*}
\]

(16)

From two different angles of view, the condensed and discontinuous planned paths of \(A\) sequentially showed in Figures 6 and 7 are conflict-free with all obstacles moving and stationary in environment of time-varying three dimensions. In our approach, the navigation path length of \(A\) is 93.84 which corresponding to the best fitness value of PSO. Compared with [13], with which method the planned path length is 96.56.

The PSO curve of fitness is shown in Figure 8, with the proposed approach, the set of solving space converges to the best value within the predefined iteration number 1000 in 5 s.
Planned paths in Figures 6 and 7, and fitness curve in Figure 8 verify that the method of PSO with penalty function can successfully deal with the mobile objects path planning in four-dimension with moving obstacles.

Velocities of A and the distances between A and obs<sub>k</sub>, \( k = 1, 2, 3 \) are shown in Figures 9 and 10, respectively. As all velocities and the distances are limited to their corresponding boundaries, consequently all the velocity constraints and safety distance criterion are satisfied. Results of this example indicate that the proposed approach is valid for mobile objects path planning with static or moving obstacles in four dimensions.

### 5. Conclusion

Path planning problems for mobile objects in four-dimension with static and moving obstacles, which have been paid little attention, are introduced. We have formulated the problem as one TNLPP. Particle swarm optimization method with penalty function is proposed for the problems. Results of several numerical examples have verified the effectiveness of our approach. With the presented approach, the resultant path is collision-avoidance with all other obstacles, smooth and much shorter than other methods. Based on the work of
this paper, in future works, we will focus on the multiobject path planning problems in four-dimension with stationary and moving obstacles.

Acknowledgments

The authors wish to sincerely thank the Editor-in-Chief and anonymous reviewers for their constructive and valuable comments which lead to the better presentation of this paper. The authors (Y. Ma and Y. Xu) are partially supported by National Science Foundation of China (no. 51109173).

References


Submit your manuscripts at http://www.hindawi.com