On Global Smooth Path Planning for Mobile Robots Using A Novel Multimodal Delayed PSO Algorithm

Baoye Song, Zidong Wang* and Lei Zou

Abstract

Background: The planning problem for smooth paths for mobile robots has attracted particular research attention, but the strategy combining the heuristic intelligent optimization algorithm (e.g. particle swarm optimization) with smooth parameter curve (e.g. Bezier curve) for global yet smooth path planning for mobile robots has not been thoroughly discussed because of several difficulties such as the local trapping phenomenon in the searching process.

Methods: In this paper, a novel multimodal delayed particle swarm optimization (MDPSO) algorithm is developed for the global smooth path planning for mobile robots. By evaluating the evolutionary factor in each iteration, the evolutionary state is classified by equal interval division for the swarm of the particles. Then, the velocity updating model would switch from one mode to another according to the evolutionary state. Furthermore, in order to reduce the occurrence of local trapping phenomenon and expand the search space in the searching process, the so-called multimodal delayed information (which is composed of the local and global delayed best particles selected randomly from the corresponding values in previous iterations) is added into the velocity updating model.

Results: A series of simulation experiments are implemented on a standard collection of benchmark functions. The experiment results verify that the comprehensive performance of the developed MDPSO algorithm is superior to other well-known PSO algorithms. Finally, the presented MDPSO algorithm is utilized in the global smooth path planning problem for mobile robots, which further confirms the advantages of the MDPSO algorithm over the traditional genetic algorithm (GA) investigated in previous studies.

Conclusions: The multimodal delayed information in the MDPSO reduces the occurrence of local trapping phenomenon and the convergence rate is satisfied at the same time. Based on the testing results on a selection of benchmark functions, the MDPSO's performance has been shown to be superior to other five well-known PSO algorithms. Successful application of the MDPSO for planning the global smooth path for mobile robots further confirms its excellent performance compared with the some typical existing algorithms.

Index Terms

Particle swarm optimization; Mobile robot; Multimodal delayed information; Smooth path planning; Bezier curve.

This work was supported in part by the Research Fund for the Taishan Scholar Project of Shandong Province of China and the Higher Educational Science and Technology Program of Shandong Province of China under Grant J14LN34.

B. Song and L. Zou are with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China.

Z. Wang is with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China. He is also with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. (Email: Zidong.Wang@brunel.ac.uk)

∗ Corresponding author.

I. INTRODUCTION

The past few decades have seen the accelerated development in robotics due mainly to the wide applications of mobile robots in a wide range of areas including agricultural production robotics, routine material transport, indoor and outdoor security patrols, harsh material handling, harsh site cleanup and underwater applications [19], [20], [48]. Among others, the path planning is a critical task in mobile robotics whose objective is to look for a feasible yet optimal path from the start point to the target point. Such an issue could be regarded as an optimization issue on certain indices (e.g. shortest distance and minimum energy) with certain constraints (e.g. collision-free route), see, e.g. [4], [28], [29], [38]. Up to now, researchers have developed many heuristic algorithms to solve this problem (see, e.g. [10], [12], [32], [35], [37], and the references cited therein), and particle swarm optimization (PSO) algorithm is arguably the most studied algorithms which have been widely employed in the path planning for mobile robots [49].

In the past few years, a large number of PSO-based approaches have been proposed for the path planning for mobile robots. For example, a study on the comparison of dynamic path planning has been presented for mobile robots in [11] where the PSO algorithm has shown the better convergence performance compared with the genetic algorithm (GA). In [31], A new method called biogeography PSO (BPSO) has been developed by combining the biogeography-based optimization (BBO) and PSO algorithm to tackle the path planning problem in static environments, where the BPSO algorithm is employed to optimize the network of the paths through approximate voronoi boundary network (AVBN) modeling. An accelerated PSO has been developed in [33] towards the global path planning for mobile robots, where only the global best particle of the PSO is utilized and the local best particle is discarded in the updating function. Furthermore, it has been shown by simulation that such a simplified PSO gives the same order of convergence as the conventional PSO. In [14], a hybrid heuristic GA-PSO scheme has been put forward to plan the paths for mobile robots in a grid environment, where crossover and mutation operators of GA are applied to the evolution of the particles of PSO, thereby avoiding the premature convergence and time complexity in conventional GA and PSO algorithms. A parallel metaheuristic PSO algorithm has been developed in [15] for the global path planning for mobile robots, in which three parallel PSO algorithms combined with a communication operator are utilized to generate the feasible line path, and the generated feasible line path is then smoothed by a cubic B-spline smoother. However, the path planning algorithms proposed in the aforementioned papers have mainly been concerned with certain simple optimal criterion (e.g. the minimum length of the path), while other important performances of the path (e.g. the smoothness of the path) are seldom considered in the path planning [44]. Usually, a traditional path planning algorithm could generate a path composed of several polygonal lines which involve inevitable sharp turns sometimes. Moving along such a path would cause frequent switches for a mobile robot between different modes (e.g. stop, rotate and restart) which leads to unnecessary waste on time and energy. Furthermore, the jerk resulting in the state switching is not permissible when the smoothness of the movement is required to ensure service quality [45]. Therefore, in addition to the path length, the path smoothness has been considered as another important criterion for its close relation to other optimization criteria [5].

So far, the planning problem for smooth paths for mobile robots has attracted particular research attention. For example, Bezier curve has been applied to plan paths of a robot soccer system modeled as multi-agents considering velocity and acceleration limits [16]. A smooth path planning approach has been proposed in [13] to generate a feasible path composed of piecewise Bezier curve with curvature constraint. A so-called A* algorithm has been introduced in [34] to find the minimum-cost path from the starting node to a given target node over a directed graph, and the arc-line approach is utilized to smooth the generated path. In [55], the Voronoi diagram and Dijkstra algorithm have been employed to plan a piecewise line path, where the endpoints of the lines in the path have been exploited as the control points of the smooth Bezier curve path. A stochastic PSO has been proposed in [7] to optimize certain cubic spline described by polynomials whose coefficients are set as variables/free parameters to form the swarm of particles. An approach based on the radial basis function (RBF) neural networks has been presented in [3] to deal with the smooth path planning of mobile robots where a Bezier curve has been trained to realize the local path planning processes. Nevertheless, the strategy combining the heuristic intelligent optimization algorithm (e.g. PSO) with smooth parameter curve (e.g. Bezier curve) for global yet smooth path planning for mobile robots has not been adequately discussed in the literature because of several difficulties such as the local trapping phenomenon in the searching process.

In this paper, we aim to develop a novel multimodal delayed PSO (MDPSO) algorithm combining with the Bezier curve to handle the global smooth planning for mobile robot paths. The multimodal delayed information adopted in the proposed MDPSO would decrease the trapping possibility to the local minimum and hence help explore the whole search space thoroughly. The main contributions of this paper are outlined as threefold. *(1) A novel multimodal delayed PSO (MDPSO) with adaptive multimodal delayed information is proposed to overcome the local trapping phenomenon frequently appearing in the global smooth planning for mobile robot paths. (2) The performance of the MDPSO is shown, via comprehensive simulation experiments, to outperform the other well-known PSO algorithms on a standard collection of benchmark functions. (3) The developed MDPSO is successfully applied to globally smoothly planning paths for mobile robots and the derived smooth path performs better than the one generated by GA in the previous studies*.

The remainder of this paper is organized as follows. In Section II, the particle swarm optimization and its developments are briefly introduced. In Section III, the novel multimodal delayed PSO is proposed and discussed in great detail. In Section IV, we conduct the simulation experiments in order to compare and contrast the performances of the MDPSO with several existing PSO algorithms. In Section V, the strategy of MDPSO combining with Bezier curve is exploited to the global smooth planning for mobile robot paths and the performance is then discussed. Finally, concluding remarks are given and future work is pointed out in the last section.

II. Particle Swarm Optimization and Its Developments

A. Traditional PSO Algorithm

As a heuristic intelligent optimization algorithm, PSO was developed by Kennedy and Eberhart to simulate the swarm behaviors of birds flocking or fish schooling, where each particle of the swarm acts as a potential solution of certain optimization problem [17].

For PSO, the swarm consisting of particles moves around at certain velocity in the search space of D-dimension. At the kth iteration in the searching process, the position of the ith particle (denoted by a vector $x_i(k) = (x_{i1}(k), x_{i2}(k), \cdots, x_{iD}(k))$ will be updated to reach the global optimum based on the corresponding velocity vector (denoted by $v_i(k) = (v_{i1}(k), v_{i2}(k), \cdots, v_{iD}(k))$). Moreover, the velocity vector of the ith particle at the kth iteration will be updated according to 1) the historical best position of the ith particle, which is also called local best particle (*pbest*) as denoted by $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$; and 2) the historical best position of the entire swarm, which is also named as global best particle (*gbest*) as represented by $p_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. The details of the updating models for velocity and position of the ith particle at the next iteration is given as follows:

$$\begin{aligned} &v_i(k+1) = wv_i(k) + c_1 r_1(p_i(k) - x_i(k)) + c_2 r_2(p_g(k) - x_i(k)), \\ &x_i(k+1) = x_i(k) + v_i(k+1), \end{aligned} \tag{1}$$

where k is the number of the current iteration; w is the inertia weight; the acceleration coefficients c_1 and c_2 are called, respectively, cognitive and social parameters; and r_1 and r_2 are two uniformly distributed random numbers on $[0, 1]$.

B. Developments of Traditional PSO

The traditional PSO scheme described above has widely been used in various optimization problems for its simple concept and efficient implementation, and a variety of approaches have been proposed to improve the capability of the traditional PSO [46].

PSO with linearly decreased inertia weight w on iteration generations (PSO-LDIW) has been introduced by Shi and Eberhart [40]–[42], the inertia weight of the current iteration w is calculated as follows:

$$w = (w_1 - w_2) \times \frac{iter_{\max} - iter}{iter_{\max}} + w_2, \tag{2}$$

where w_1 and w_2 are, respectively, the initial value and the final value of the inertia weight; $iter$ denotes the number of current iteration and $iter_{\max}$ is the number of maximum iteration. Generally speaking, a larger inertia weight would make the PSO tend to the global exploration and, on the other hand, a smaller one could achieve the local exploitation. Therefore, the initial and final values w_1 and w_2 are customarily set as 0.9 and 0.4, respectively. Furthermore, PSO with time-vary acceleration coefficients (PSO-TVAC) has been proposed in [39] as computed by the following equations:

$$c_1 = (c_{1i} - c_{1f}) \times \frac{iter_{\max} - iter}{iter_{\max}} + c_{1f}, \tag{3}$$

$$c_2 = (c_{2i} - c_{2f}) \times \frac{iter_{\max} - iter}{iter_{\max}} + c_{2f}, \tag{4}$$

where c_{1i} (c_{2i}) is the initial value, and c_{1f} (c_{2f}) is the final value of the acceleration coefficient c_1 (c_2). Usually, we set $c_{1i} = 2.5$ ($c_{2i} = 0.5$) and $c_{1f} = 0.5$ ($c_{2f} = 2.5$) in this strategy. Additionally, PSO with constriction factor (PSO-CK) has been proposed by Clerc and Kennedy to enhance the searching performance of PSO, where $w = 0.729$ and $c_1 = c_2 = 1.49$ are recommended [9]. The improvement strategies of the traditional PSO mentioned above are mainly concerning with the parameter studies, while other strategies including combination with auxiliary operations and topological structures have also been constructed to enhance the capability of PSO [51]. One of the most remarkable research trends is the hybrid PSO, which combines some auxiliary operations with the traditional PSO, e.g., selection [2], crossover [8], mutation [1], local search [22] and differential evolution [54], etc. A fully informed particle swarm optimization (FIPSO) scheme has been developed to guide the particles of the swarm using the information of entire neighborhood [30]. A comprehensive-learning PSO (CLPSO) introduced in [23] employs local best particle from different neighbors to update the swarm flying at different dimensions so as to improve the performance in the case of multimodal optimizations. The analyses and experiments have shown that the performance of the traditional PSO has been promoted greatly by using these strategies.

Recently, an adaptive PSO (APSO) has been put forward by Zhan et al., which introduces an evolutionary factor to quantify the mean distance (between the global best particle and other particles). By a series of fuzzy membership functions according to the evolutionary factor, four states have been defined, which are the exploration state, the exploitation state, the convergence state and the jumping out state. These four states have been used to adaptively control the inertia weight and the acceleration coefficients in each iteration [51]. A switching PSO (SPSO) has been proposed by Tang et al. to overcome the shortcomings of Zhan's algorithm. A Markov chain is used to predict the next state according to the current state as decided by the evolutionary factor, and the velocity updating rule is switched from one mode to another depending on the evolutionary state [46]. Moreover, a switching delayed PSO (SDPSO) has been proposed more recently, which could switch the velocity updating model according to the evolutionary state predicted by a Markov chain. In the SDPSO, in addition to the inertia weight and acceleration coefficients that are adaptively adopted based on the evolutionary factor and the evolutionary state, the local/global best particles are randomly chosen based on the corresponding values from previous iterations [50].

III. A Novel Multimodal Delayed PSO Algorithm

The main purpose of this section is to develop a novel multimodal delayed PSO (MDPSO) so as to improve the searching performance further. The main novelty of such a new PSO algorithm is to add two delayed terms in the traditional velocity updating model of the PSO algorithm. The new terms, composed of both the local and global delayed best particles selected from the corresponding values in the previous iterations stochastically, are added into the velocity updating model according to the evaluated evolutionary state. This improvement strategy aims to reduce the convergence speed of the traditional PSO and thereby decreasing the likelihood of converging to local minimum. As such, the entire search space could be explored more thoroughly.

A. Framework of MDPSO

The updating equations for the velocity and the position of the novel MDPSO algorithm are presented by:

$$\begin{aligned} v_i(k+1) &= wv_i(k) + c_1 r_1(p_i(k) - x_i(k)) + c_2 r_2(p_g(k) - x_i(k)) \\ &\quad + s_i(k) c_3 r_3(p_i(k - \tau_i(k)) - x_i(k)) \\ &\quad + s_g(k) c_4 r_4(p_g(k - \tau_g(k)) - x_i(k)), \\ x_i(k+1) &= x_i(k) + v_i(k+1), \end{aligned} \tag{5}$$

where w is the inertia weight determined by equation (2); c_1 and c_2 are the coefficients for acceleration updated by equations (3) and (4), and c_3 and c_4 are equal to c_1 and c_2 without loss of generality, i.e. $c_1 = c_3$ and $c_2 = c_4$, respectively; $r_i (i = 1, 2, 3, 4)$ are the random uniformly distributed numbers in $[0, 1]$; $\tau_i(k)$ and $\tau_g(k)$ are the random delays uniformly distributed in $[0, k]$ for the local and the global delayed best particle, respectively; $s_i(k)$ and $s_g(k)$ are the intensity factor of the newly added terms in the velocity updating model depending on the evolutionary factor.

In the novel MDPSO algorithm, the newly added terms in the velocity updating model is closely related to the evolutionary factor (EF) defined in [51] to describe the swarm distribution properties. According to the EF in the searching process, the four states (i.e. the exploration state, the exploitation state, the convergence state and the jumping out state) are denoted by $\xi(k) = 1$, $\xi(k) = 2$, $\xi(k) = 3$ and $\xi(k) = 4$, respectively. The mean distance between the ith particle and the other particles in the swarm denoted as d_i could be calculated by

$$d_i = \frac{1}{S-1} \sum_{j=1, j\neq i}^{S} \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2}, \tag{6}$$

where S and D denote the swarm size and the particle dimension, respectively. Accordingly, the evolutionary factor denoted as E_f could be calculated by

$$E_f = \frac{d_g - d_{min}}{d_{\max} - d_{min}}, \tag{7}$$

where d_g is the mean distance between the global best particle and the other particles in the swarm. $d_{\max}$ and d_{min} are the maximum and minimum of d_i in the swarm, respectively. The evolutionary state are classified according to the evolutionary factor by a series of fuzzy functions in [51], equal division strategy is used for the evolutionary state classification in [46], [50] for the state prediction depending on a Markov chain. In this paper, the formulation of evolutionary state classification in [46], [50] has been adopted and expressed as follows:

$$\xi(k) = \begin{cases} 1, & 0 \leq E_f < 0.25, \\ 2, & 0.25 \leq E_f < 0.5, \\ 3, & 0.5 \leq E_f < 0.75, \\ 4, & 0.75 \leq E_f \leq 1. \end{cases} \tag{8}$$

B. Strategies for Multimodal Delayed Information

PSO-TVAC is one of the most successful improvements of the traditional PSO, in which the linear varied acceleration coefficients are used in the searching process. Four evolutionary states are classified by a series of fuzzy functions according to the evolutionary factor, and different acceleration coefficients are recommended for each state in APSO. In SPSO, the same strategy for selecting the varied acceleration coefficients is adopted according to the switching evolutionary state based on a Markov chain. Furthermore, the switching evolutionary state is employed to switch the velocity updating model with delayed information from one mode to another in SDPSO. The main idea in the above strategies is to adjust the velocity updating model in an adaptive mode according to the evaluated evolutionary state. In this paper, a novel strategy with multimodal delayed information is developed to adaptively adjust the velocity updating model and introduced as follows:

- In the convergence state ($\xi(k) = 1$), the particles in the swarm are expected to fly into the region around the global optimum as soon as possible. Hence, only the normal terms in the velocity updating model are remained and the delayed information is ignored, i.e. both of $s_i(k)$ and $s_g(k)$ are set to zero.
- In the exploitation state ($\xi(k) = 2$), the particles in the swarm are willing to exploit the region around the local best particles. So that, the local delayed information is added into the velocity updating

model, i.e. only the local best particles in the previous iterations are randomly selected for the velocity updating with the intensity factor $s_i(k) = E_f(k)$.

- In the exploration state ($\xi(k) = 3$), it is important to search the optima as many as possible. Therefore, the global delayed information is added to explore the whole search space in a more thorough way, that is, the global best particles in the previous iterations are randomly selected for the velocity updating with the intensity factor $s_g(k) = E_f(k)$.
- In the state of jumping out ($\xi(k) = 4$), the local best particles are eager to jump out from the region around the local optima. Thus, it is necessary to provide more power for these particles to escape from this region, so that both of the local and global delayed information are used for this purpose with the intensity factor $s_i(k) = E_f(k)$ and $s_g(k) = E_f(k)$, respectively.

The above discussed strategies for multimodal delayed information can be summarized in Table I, where $s_i(k)$ and $s_g(k)$ are the intensity factor determined by the evolutionary state and evolutionary factor $E_f(k)$ in each iteration; rand is the function to generate a randomly uniformly distributed number in $[0, 1]$; the delay $\tau_i(k)$ and $\tau_g(k)$ are randomly selected integer uniformly distributed in $[0, k]$, where k denotes the number of the current iteration and $\lfloor \cdot \rfloor$ means the round down function.

Remark 1: Note that the delayed information stems from the randomly selected previous local and global particles, both of which would bring some kind of "turbulence" to the convergence process in the traditional PSO. Such a novel strategy would, without any doubt, reduce the convergence rate slightly, yet the entire search space could be explored more thoroughly and the global optima could be more likely to be obtained. This superiority is crucial for the global planning of smooth paths for mobile robots where traditional planning algorithms suffer typically from the local trapping problems leading to the infeasibility of the planning problems.

The flowchart of our novel multimodal delayed PSO (MDPSO) algorithm is depicted as in Fig. 1.

TABLE I
STRATEGIES FOR MULTIMODAL DELAYED INFORMATION

State	Mode	$s_i(k)$	$s_g(k)$	$\tau_i(k)$	$\tau_g(k)$
Convergence	$\xi(k) = 1$	0	0	–	–
Exploitation	$\xi(k) = 2$	$E_f(k)$	0	$\lfloor k \cdot \mathrm{rand}_{\tau_i} \rfloor$	–
Exploration	$\xi(k) = 3$	0	$E_f(k)$	–	$\lfloor k \cdot \mathrm{rand}_{\tau_g} \rfloor$
Jumping-out	$\xi(k) = 4$	$E_f(k)$	$E_f(k)$	$\lfloor k \cdot \mathrm{rand}_{\tau_i} \rfloor$	$\lfloor k \cdot \mathrm{rand}_{\tau_g} \rfloor$

IV. SIMULATION EXPERIMENTS

A. Configuration of Benchmark Functions

In the following simulation examples, some frequently used benchmark functions are employed to evaluate the performance of the novel MDPSO scheme. The benchmark functions are given by equation (9) to (14), where the Sphere function $f_1(x)$ is a typical unimodal optimization problem usually utilized to examine the convergence rate of the optimization algorithm; the Rosenbrock function $f_2(x)$ can be regarded as a multimodal function because it is hard to obtain the optimum in the narrow banana-like valley; $f_3(x)$ to $f_6(x)$ are other typical unimodal and multimodal functions undoubtedly very hard to obtain the optimum. To this end, the configuration of the benchmark functions are shown in Table II,

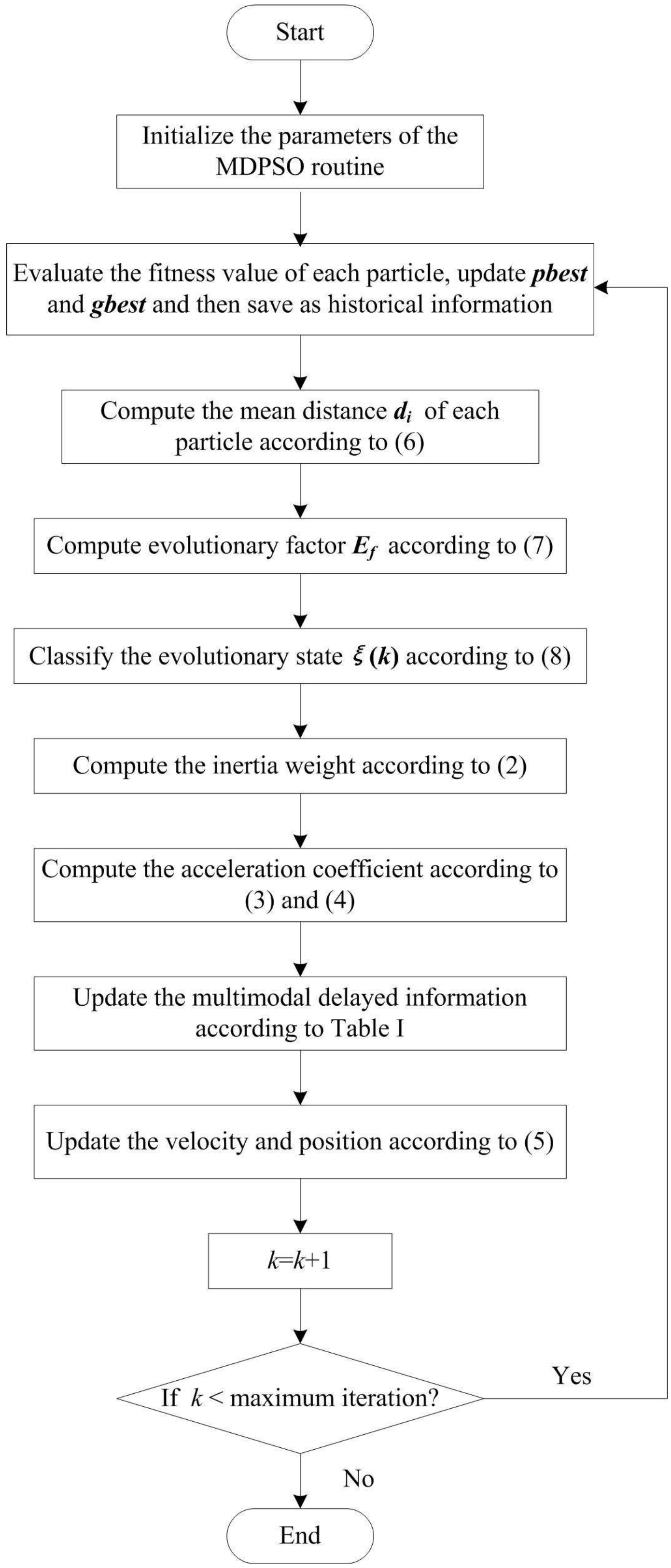

Fig. 1. Flowchart of MDPSO algorithm

where the fourth column indicates the search space of each dimension; the fifth column is the threshold to determine whether a searching process is successful or not; and the optima of all the benchmark functions are zero as given in the sixth column of the table [46].

$$\text{Sphere}: \ f_1(x) = \sum_{i=1}^{D} x_i^2. \tag{9}$$

$$\text{Rosenbrock}: \ f_2(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2). \tag{10}$$

$$\text{Ackley}: \ f_3(x) = -20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i} + 20 + e. \tag{11}$$

$$\text{Rastrigin}: \ f_4(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos 2\pi x_i + 10). \tag{12}$$

$$\text{Schwefel 2.22}: \ f_5(x) = \sum_{i=1}^{D} \mid x_i \mid + \prod_{i=1}^{D} \mid x_i \mid . \tag{13}$$

$$\text{Schwefel 1.2}: \ f_6(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2. \tag{14}$$

TABLE II
CONFIGURATION OF BENCHMARK FUNCTIONS

Functions	Name	Dimension	Search space	Threshold	Minimum
$f_1(x)$	Sphere	20	$[-100, 100]^D$	0.01	0
$f_2(x)$	Rosenbrock	20	$[-30, 30]^D$	100	0
$f_3(x)$	Ackley	20	$[-32, 32]^D$	0.01	0
$f_4(x)$	Rastrigin	20	$[-5.12, 5.12]^D$	50	0
$f_5(x)$	Schwefel 2.22	20	$[-10, 10]^D$	0.01	0
$f_6(x)$	Schwefel 1.2	20	$[-100, 100]^D$	0.01	0

B. Simulation and Discussion

A series of simulations are implemented to 1) test the performance of our novel MDPSO scheme and 2) compare with some other standard PSO algorithms in order to demonstrate the advantages of the MDPSO. The parameter settings are given as follows: the population of the swarm $S = 20$, the dimension of the particle $D = 20$, the maximum iteration number $N = 20000$, and each experiment is repeated 50 times independently in one routine for the subsequent statistical analysis. The proposed MDPSO algorithm is compared with five other rather standard PSO algorithms, including the PSO-LDIW [40], PSO-TVAC [39], PSO-CK [9], SPSO [46] and SDPSO [50].

The mean fitness values of the above PSO algorithms for each benchmark function are shown in Fig. 2 to Fig. 7, where the horizontal coordinate indicates the iteration number and the values of the vertical

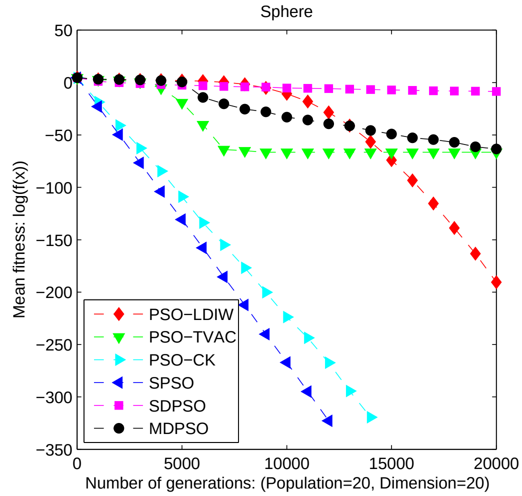

Fig. 2. Performance test for Sphere function $f_1(x)$

coordinate are represented in logarithmic formation. The statistical comparisons of the optimization results listed in Table III are also presented to demonstrate the minimum, the mean value and the standard deviation of the fitness values as well as the success ratio of each PSO algorithm on the benchmark functions. Note that some of the success ratios are very low as shown in the table, i.e. not all the optimal solutions of the algorithms could converge to the fitness value below the threshold with the increasing iteration generation, so that some of the mean values are extremely large in comparison with the proposed MDPSO algorithm as shown in the figures. It is illustrated that the random initialization could not ensure the successful convergence for all the algorithms and the premature convergence could not be avoided. However, the MDPSO algorithm could be able to tend to the optimum robustly for all the benchmark functions. Besides, the convergence performances of the algorithms are quite different from each other. Obviously, the convergence rate of the proposed MDPSO is slightly slower than some of the other PSO algorithms, e.g. PSO-CK and SPSO on the Sphere function, yet better solutions could be achieved for all the other benchmark functions. It is worthwhile to note that the multimodal delayed information in the MDPSO make it not easy to converge to the local minima, and thus the whole search space could be explored more thoroughly as a result of the super capability of escaping from the local optima. Hence, the proposed MDPSO algorithm outperforms the other PSO algorithms for both unimodel and multimodel benchmark functions in a series of criteria such as success ratio and mean fitness value.

V. Smooth path planning for mobile Robots

A. Preliminary of Bezier curve

As a parametric curve, the Bezier curve has been successfully used in practice such as computer graphics. Different from the traditional interpolation-based curves such as polynomials and cubic splines, the Bezier curve consists of a number of control points, which are not in the curve except the start and end points. Let a set of control point vectors $\mathbf{P}_0, \mathbf{P}_1, \cdots, \mathbf{P}_n$ be given. In this case, the Bezier curve denoted as $\mathbf{P}(t)$ is defined as

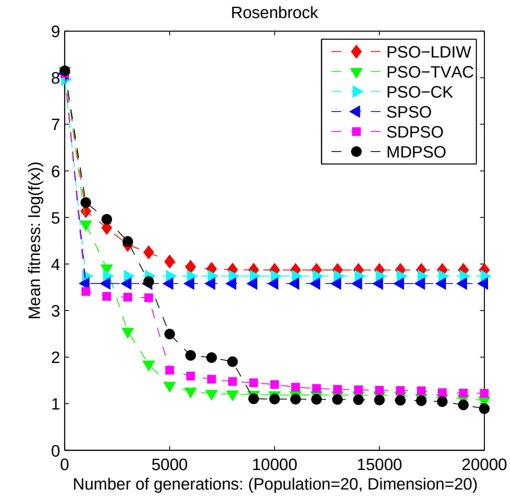

Fig. 3. Performance test for Rosenbrock function $f_2(x)$

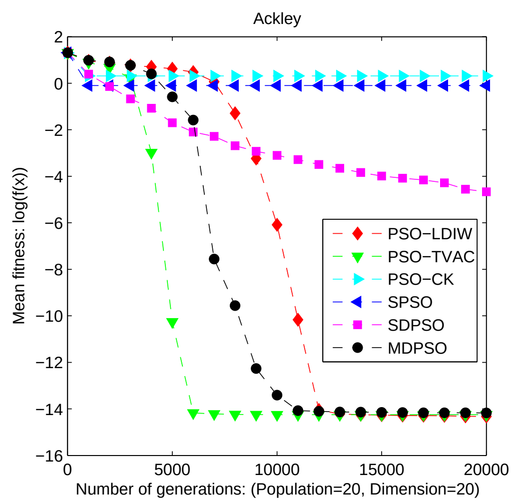

Fig. 4. Performance test for Ackley function $f_3(x)$

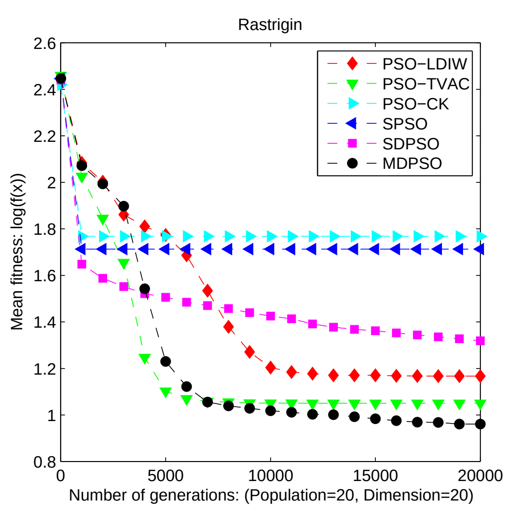

Fig. 5. Performance test for Rastrigin function $f_4(x)$

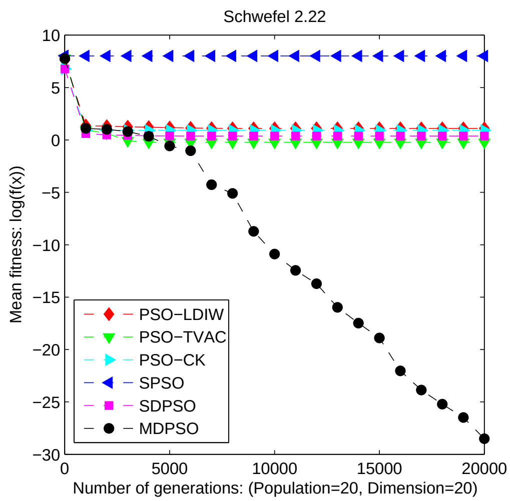

Fig. 6. Performance test for Schwefel 2.22 function $f_5(x)$

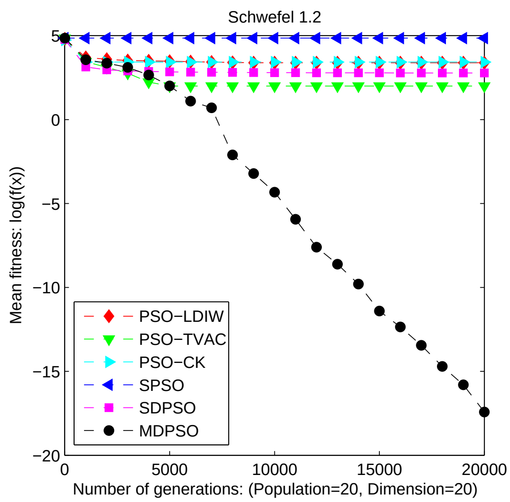

Fig. 7. Performance test for Schwefel 1.2 function $f_6(x)$

$$\mathbf{P}(t) = \sum_{i=0}^{n} B_i^n(t)\mathbf{P}_i,\ \ 0 \le t \le 1, \tag{15}$$

where t is the normalized time variable; $\mathbf{P}_i = (x_i, y_i)^T$ stands for the coordinate vector of the ith control point with x_i and y_i as X and Y coordinate components, respectively; $B_i^n(t)$ is the Bernstein polynomial expressed as:

$$B_i^n(t) = \begin{pmatrix} n \\ i \end{pmatrix} t^i(1-t)^{n-i},\ \ i = 0, 1, \ldots, n. \tag{16}$$

The smoothness of a Bezier curve based path is closely related to the curvature function of the path. In the two-dimensional plane, the Bezier curve's curvature can be expressed as:

$$\kappa(t) = \frac{1}{R(t)} = \frac{\dot{\mathbf{P}}_x(t)\ddot{\mathbf{P}}_y(t) - \dot{\mathbf{P}}_y(t)\ddot{\mathbf{P}}_x(t)}{(\dot{\mathbf{P}}_x^2(t) + \dot{\mathbf{P}}_y^2(t))^{3/2}}, \tag{17}$$

TABLE III
STATISTICAL COMPARISONS OF THE OPTIMIZATION RESULTS

		PSO-TVAC	PSO-LDIW	PSO-CK	SPSO	SDPSO	MDPSO
$f_1(x)$	Minimum	2.53×10^{-145}	2.49×10^{-201}	0.0000	0.0000	1.23×10^{-12}	2.72×10^{-94}
	Mean	3.03×10^{-67}	3.07×10^{-191}	0.0000	0.0000	3.09×10^{-9}	4.50×10^{-64}
	Std. Dev.	2.14×10^{-66}	0.0000	0.0000	0.0000	1.00×10^{-8}	3.10×10^{-63}
	Ratio	100%	100%	100%	100%	100%	100%
$f_2(x)$	Minimum	2.21×10^{-4}	1.85×10^{-2}	4.45×10^{-12}	2.23×10^{-3}	3.5339	2.70×10^{-5}
	Mean	1.17×10^{1}	7.41×10^{3}	5.47×10^{3}	3.79×10^{3}	1.67×10^{1}	7.7743
	Std. Dev.	1.48×10^{1}	2.46×10^{4}	2.15×10^{4}	1.77×10^{4}	1.52×10^{1}	5.1952
	Ratio	100%	80%	90%	88%	100%	100%
$f_3(x)$	Minimum	2.66×10^{-15}	2.66×10^{-15}	6.21×10^{-15}	2.66×10^{-15}	1.99×10^{-7}	2.66×10^{-15}
	Mean	5.64×10^{-15}	4.72×10^{-15}	2.0694	7.91×10^{-1}	2.14×10^{-5}	6.64×10^{-15}
	Std. Dev.	1.81×10^{-15}	1.77×10^{-15}	2.2971	2.3036	5.03×10^{-5}	2.22×10^{-15}
	Ratio	100%	100%	18%	72%	100%	100%
$f_4(x)$	Minimum	5.9697	3.9798	2.78×10^{1}	2.18×10^{1}	2.9852	1.9899
	Mean	1.12×10^{1}	1.46×10^{1}	5.86×10^{1}	5.16×10^{1}	2.08×10^{1}	9.1337
	Std. Dev.	3.4876	1.26×10^{1}	1.95×10^{1}	1.90×10^{1}	1.26×10^{1}	2.8119
	Ratio	100%	100%	32%	48%	98%	100%
$f_5(x)$	Minimum	3.86×10^{-43}	7.84×10^{-120}	5.68×10^{-31}	1.63×10^{-162}	1.04×10^{-7}	6.50×10^{-40}
	Mean	6.00×10^{-1}	1.26×10^{1}	8.40	1.05×10^{8}	2.4000	3.10×10^{-29}
	Std. Dev.	2.3989	1.21×10^{1}	9.33	1.67×10^{8}	4.7638	1.14×10^{-28}
	Ratio	94%	32%	46%	2%	78%	100%
$f_6(x)$	Minimum	1.31×10^{-29}	2.63×10^{-28}	1.05×10^{-106}	2.83×10^{4}	2.64×10^{-1}	1.41×10^{-30}
	Mean	1.00×10^{2}	2.43×10^{3}	2.70×10^{3}	7.26×10^{4}	6.04×10^{2}	3.77×10^{-18}
	Std. Dev.	7.07×10^{2}	4.01×10^{3}	3.57×10^{3}	2.46×10^{4}	2.39×10^{3}	1.67×10^{-17}
	Ratio	98%	66%	58%	0%	0%	100%

where $R(t)$ is the curvature's radius; $\dot{\mathbf{P}}_x(t)$, $\dot{\mathbf{P}}_y(t)$, $\ddot{\mathbf{P}}_x(t)$ and $\ddot{\mathbf{P}}_y(t)$ are the coordinate components on X and Y of the first and second derivatives of the Bezier curve $\mathbf{P}(t)$, which are expressed as the following equations:

$$\dot{\mathbf{P}}(t) = \frac{d\mathbf{P}(t)}{dt} = n \sum_{i=0}^{n-1} B_i^{n-1}(t)(\mathbf{P}_{i+1} - \mathbf{P}_i), \tag{18}$$

$$\ddot{\mathbf{P}}(t) = n(n-1) \sum_{i=0}^{n-2} B_i^{n-2}(t)(\dot{\mathbf{P}}_{i+2} - 2\dot{\mathbf{P}}_{i+1} + \dot{\mathbf{P}}_i). \tag{19}$$

B. Optimization model of smooth path planning

A two-dimensional workspace with several obstacles is supposed to be the working environment of a mobile robot, and the purpose of planning the smooth path is to seek a feasible and optimal Bezier curve path from the start to the target points. The whole workspace is divided into many square grids according to the requirement of the smooth path planning, e.g. Fig. 8 shows the workspace of a mobile robot with numbered grids. For each grid, it is defined to be either empty (denoted as the white square

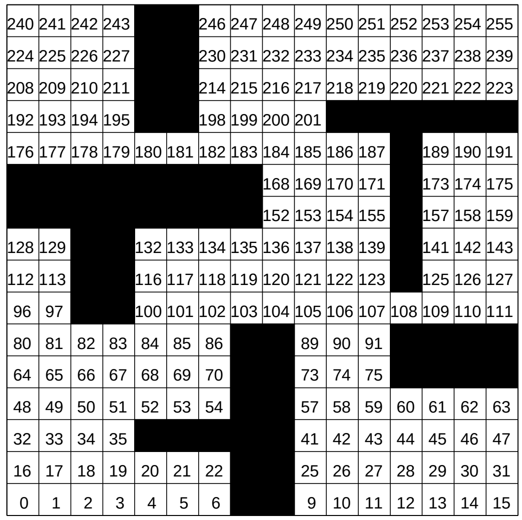

Fig. 8. Workspace of a mobile robot with numbered grids

grid) or occupied (denoted as the black square grid), which is determined by whether the boundary of the obstacles is within the square grid. A mobile robot in the workspace could be treated as a point. In terms of the size of a mobile robot, the boundary of the obstacles has been constructed from 1) the actual boundary; and 2) the minimum safety distance.

In this paper, the proposed MDPSO is combined with the Bezier curve for planning the smooth path of a mobile robot by considering feasibility, smoothness and distance of the path at the same time. Generally, a Bezier curve path comprises several segments denoted as a sequence of control points. Hence, the smooth path planning is to find a sequence of control points that define a feasible and shortest Bezier curve path by using the MDPSO algorithm. The objetive function is given as follows:

$$\min\ J = \sum_{i=1}^{n} \|\mathbf{P}_i(t)\| + N_o \times P_r, \tag{20}$$

where $\|\mathbf{P}_i(t)\|$ denotes the length of the ith segment of a Bezier curve path; N_o and P_r denote the amount of the occupied grids in the path and the penalty ratio for each occupied grid, respectively.

C. Results and analyses

In this section, the algorithm of planning smooth path based on MDPSO and Bezier curve has been applied to the workspace in Fig. 8 in order to show the usefulness of the proposed approach. The parameters of the experiments are given as follows: the population of the swarm is set as 100, the maximum of the iteration generation is set as 100, the dimension of the particle that denotes the amount of control points of the Bezier curve is taken as 7, and the experimental penalty ratio is taken as 50 for each occupied obstacle.

Two results of numerous simulation experiments have been illustrated in Fig. 9, where the start position and target position are $(5, 5)$ to $(155, 155)$ and $(155, 5)$ to $(5, 155)$, respectively; blue circles indicate the control points of the Bezier curve path, blue solid lines compose the convex hull and red solid curve depicts the optimum smooth path. Obviously, it is really a challenging issue for the path planner to find a satisfied path in these two cases for at least two reasons. Firstly, the acceptable path must pass through a

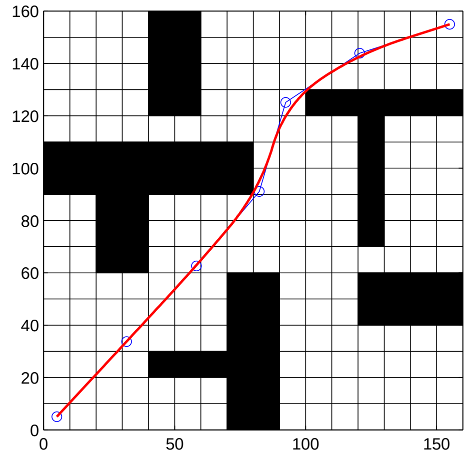

Fig. 9. Smooth path planning from (5, 5) to (155, 155)

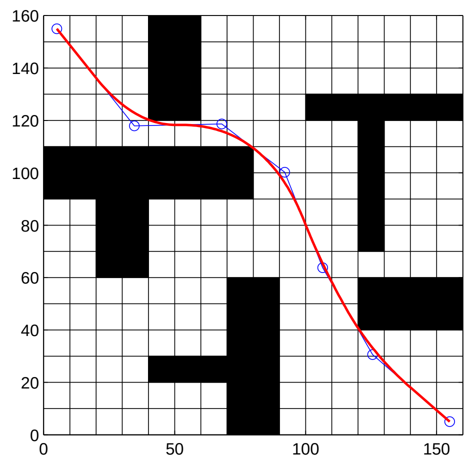

Fig. 10. Smooth path planning from (155, 5) to (5, 155)

few narrow gaps formed by the nearby obstacles. Secondly, the optimal solution is very easy to be attracted into the surrounding traps, i.e. several infeasible paths with suboptimal objective function value. However, the MDPSO algorithm could accomplish the difficult task of planning smooth path successfully, which profits from its ability of escaping from the local optima by using the multimodal delayed information. In comparison with the approach proposed in [44] where genetic algorithm (GA) is combined with Bezier curve for planning the smooth path for a mobile robot, the smooth path produced in this paper has a smaller curvature as shown in Fig. 11, which means a smoother path for the movement of a mobile robot. The best objective function value of the optimum particle in each iteration is shown in Fig. 13, which demonstrates the fast convergence rate of the MDPSO algorithm for this problem.

VI. Conclusions

A novel multimodal delayed particle swarm optimization (MDPSO) has been developed in this paper for planning the global smooth path for mobile robots. In the proposed MDPSO algorithm, the delayed

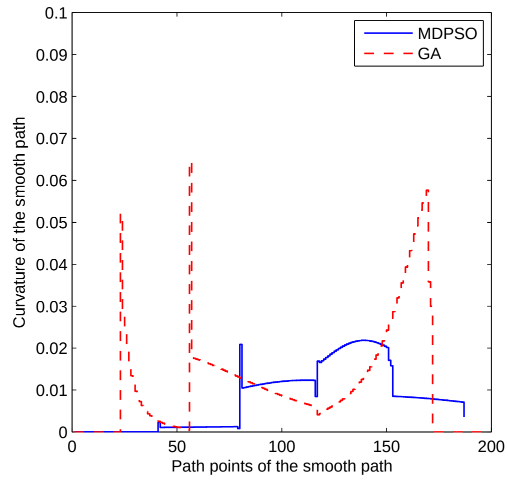

Fig. 11. Path curvature in the case of Fig. 9

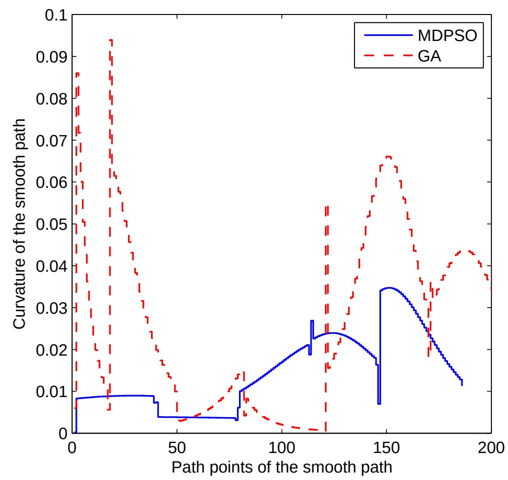

Fig. 12. Path curvature in the case of Fig. 10

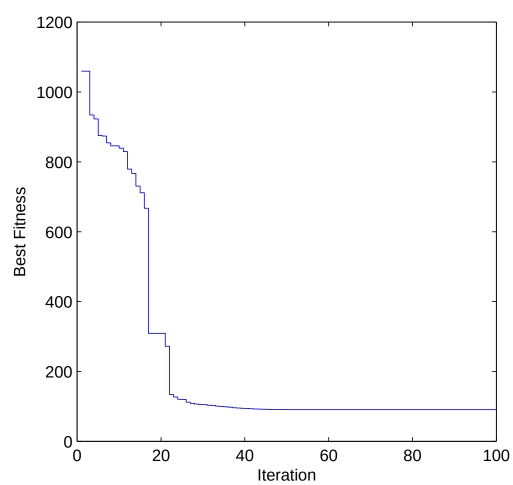

Fig. 13. Optimum objective function value in the case of Fig. 9

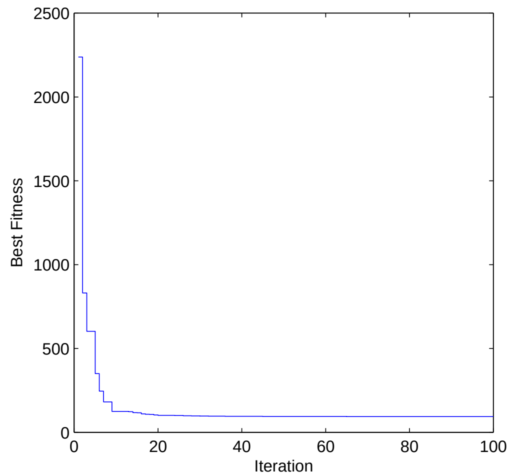

Fig. 14. Optimum objective function value in the case of Fig. 10

information composed of local and global delayed best particle, which are chosen randomly from the previous iterations, have been added into the velocity updating model according to the evolutionary state and evolutionary factor. The multimodal delayed information in the MDPSO reduces the occurrence of local trapping phenomenon and the convergence rate is satisfied at the same time. Based on the testing results on a selection of benchmark functions, the MDPSO's performance has been shown to be superior to other five well-known PSO algorithms. Finally, the successful application of the MDPSO for planning the global smooth path for mobile robots further confirms its excellent performance compared with the one generated by GA in the previous studies.

In the future work, the focus will be on 1) how to develop new strategy to further enhance/improve the performance of the developed PSO (e.g. the approach to adaptively select the delayed information) and 2) how to apply the proposed algorithms to more complicated systems (see e.g. [6], [18], [21], [24]–[27], [43], [47], [52], [53]).

VII. Compliance with Ethical Standards

Conflict of Interest B. Song, Z. Wang and L. Zou declare that they have no conflict of interest.

Informed Consent All procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1975, as revised in 2008 (5). Additional informed consent was obtained from all patients for which identifying information is included in this article.

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by the any of the authors.

References

[1] Andrews P. S., "An investigation into mutation operators for particle swarm optimization," In *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1044-1045, 2006.

[2] Angeline P. J., "Using selection to improve particle swarm optimization," In *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 84-89, 1998.

[3] Arana-Daniel N., Gallegos A. A., Lopez-Franco C., and Alanis A. Y., “Smooth global and local path planning for mobile robot using particle swarm optimization, radial basis functions, splines and Bezier curves,” In *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 175-182, 2014.

[4] Atyabi A., and Powers D. M. W., “Review of clasical and heuristic-based navigation and path planning approaches,” *International Journal of Advancements in Computing Technology*, Vol. 5, pp. 1-14, 2013.

[5] Castillo O., Ttrujillo L., and Melin P., “Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots,” *Soft Computing*, Vol. 11, pp. 269-279, 2007.

[6] H. Chen, J. Liang, and Z. Wang, Pinning controllability of autonomous Boolean control networks, *Science China Information Sciences*, vol. 59, no. 7, Art. No. 070107, DOI: 10.1007/s11432-016-5579-8, 2016.

[7] Chen X., and Li Y., “Smooth path planning of a mobile robot using stochastic particle swarm optimization,” In *Proceedings of IEEE International Conference on Mechatronics and Automation*, pp. 1722-1727, 2006.

[8] Chen Y. P., Peng W. C., and Jian M. C., “Particle swarm optimization with recombination and dynamic linkage discovery,”, *IEEE Transactions on Systems, Man and Cybernetics–Part B: Cybernetics*, Vol. 37, No. 6, pp. 1460-1470, 2007.

[9] Clerc M., and Kennedy J., “The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space,” *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp. 58-73, 2002.

[10] Contreras-Cruz M. A., Ayala-Ramirez V., and Hernandez-Belnonte U. H., “Mobile robot path planning using artificial bee colony and evolutonary programming,” *Applied Soft Computing Journal*, Vol. 30, pp. 319-328, 2015.

[11] Fetanat M., Haghzad S., and Shouraki S. B., “Optimization of dynamic mobile robot path planning based on evolutionary methods,” In *Proceedings of AI & Robotics (IRANOPEN 2015)*, pp. 1-7, 2015.

[12] Fong S., Deb S., and Chaudhary A., “A review of metaheuristics in robotics,” *Computers and Electrical Engineering*, Vol. 43, pp. 278-291, 2015.

[13] Ho Y. J., and Liu J. S., “Collision-free curvatue-bounded smooth path planning using composite Bezier curve based on Voronoi diagram,” In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 463-468, 2009.

[14] Huang H. C., and Tsai C. C., “Global path planning for autonomous robot navigation using hybrid metaheuristi GA-PSO algorithm,” In *Proceedings of SICE Annual Conference*, pp. 1338-1348, 2011.

[15] Huang H. C., “FPGA-based parallel metaheuristic PSO algorithm and its application to global path planning for autonomous robot navigation,” *Journal of Intelligent & Robotic Systems*, Vol. 76, pp. 475-488, 2014.

[16] Jolly K. G., Kumar R. S., and Vijayakumar R., “A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits,” *Robotics and Autonomous Systems*, Vol. 57, No. 1, pp. 23-33, 2009.

[17] Kennedy J., and Eberhart R., “Particel swarm optimization,” In *Proceedings of IEEE International Conference on Neural Network*, pp. 1942-1948, 1995.

[18] Q. Li, B. Shen, Y. Liu, and F. E. Alsaadi, Event-triggered H_∞ state estimation for discrete-time stochastic genetic regulatory networks with Markovian jumping parameters and time-varying delays, *Neurocomputing*, vol. 174, pp. 912–920, 2016.

[19] Z. Li, C. Yang, C.-Y. Su, and W. Ye, Adaptive fuzzy-based motion generation and control of mobile under-actuated manipulators, *Engineering Applications of Artificial Intelligence*, vol. 30, pp. 86–95, 2014.

[20] Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, Vision-based model predictive control for steering of a nonholonomic mobile robot, *IEEE Transactions on Industrial Electronics*, vol. 24, no. 2, pp. 553–564, 2016.

[21] W. Li, G. Wei, F. Han, and Y. Liu, Weighted average consensus-based unscented Kalman filtering, *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 558–567, 2016.

[22] Liang J. J., and Suganthan P. N., “Dynamic multi-swarm particle swarm optimizer with local search,” In *Proceedings of IEEE Congress on Evolutionary Computation.*, pp. 522-528, 2005.

[23] Liang J. J., Qin A. K., Suganthan P. N., and Baskar S., “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” *IEEE Trans. Evol. Comput.*, Vol. 10, No. 3, pp. 281-295, 2006.

[24] D. Liu, Y. Liu, and F. E. Alsaadi, A new framework for output feedback controller design for a class of discrete-time stochastic nonlinear system with quantization and missing measurement, *International Journal of General Systems*, vol. 45, no. 5, pp. 517–531, 2016.

[25] S. Liu, G. Wei, Y. Song, and Y. Liu, Extended Kalman filtering for stochastic nonlinear systems with randomly occurring cyber attacks, *Neurocomputing*, vol. 207, pp. 708–716, 2016.

[26] S. Liu, G. Wei, Y. Song, and Y. Liu, Error-constrained reliable tracking control for discrete time-varying systems subject to quantization effects, *Neurocomputing*, vol. 174, pp. 897–905, 2016.

[27] Y. Liu, W. Liu, M. A. Obaid, and I. A. Abbas, Exponential stability of Markovian jumping Cohen-Grossberg neural networks with mixed mode-dependent time-delays, *Neurocomputing*, vol. 177, pp. 409–415, 2016.

[28] Manikas T. W., Ashenayi K., and Wainwright R. L., “Genetic algorithms for autonomous robot navigation,” *IEEE Instrumentation and Measurement Magazine*, Vol. 12, No. 1, pp. 26-31, 2007.

[29] Masehian E., and Sedighizadeh D., "Classic and heuristic approaches in robot motion planning-a chronological review," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol. 1, No. 5, pp. 228-233, 2007.

[30] Mendes R., Kennedy J., and Neves J., "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation.*, Vol. 8, No. 3, pp. 204-210, 2004.

[31] Mo H., and Xu L., "Research of biogeography particle swarm optimization for robot path planning," *Neurocomputing*, Vol. 148, pp. 91-99, 2015.

[32] Mohajer B., Kiani K., Samiei E., and Sharifi M., "A new online random particles optimization algorithm for mobile robot path planning in dynamic environments," *Mathematical Problems in Engineering*, Vol. 2013, pp. 1-9, 2013.

[33] Mohamed A. Z., Lee S. H., Hsu H. Y., and Nath N., "A faster path planner using accelerated particle swarm," *Artificial Life & Robotics*, Vol. 17, pp. 233-240, 2012.

[34] On S., and Yazici A., "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, 2011.

[35] Pol R. S., and Murgugan M., "A review on indoor human aware autonomous mobile robot navigation through a dynamic environment," In *Proceedings of IEEE International Conference on Industrial Instrumentation and Control*, pp. 1339-1344, Pune, May, 2015.

[36] Purcaru C., Precup R., Iercan D., Fedorovici L. O., and David R. C., "Hybrid PSO-GSA robot path planning algorithm in static environments with danger zones," In *Proceedings of International Conference on System Theory, Control and Computing*, pp. 434-439, Sinaia, October, 2013.

[37] Qu H., Xing K., and Alexander T., "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, Vol. 120, pp. 509-517, 2013.

[38] Raja P., and Pugazhenthi S., "Optimal path planning of mobile robots: a review," *International Journal of Physical Sciences*, Vol. 7, No. 9, pp. 1314-1320, 2012.

[39] Ratnaweera A., Halgamure S. K., and Watson H. C., "Self-organizing hierarchical particle swarm ooptimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, Vol. 8, pp. 240-255, 2004.

[40] Shi Y., and Eberhart R., "A modified particle swarm optimizer," In *Proceedings of IEEE Internatonal Conference on Evolutionary Computation*, pp. 69-73, 1998.

[41] Shi Y., and Eberhart R., "Parameter selection in particle swarm optimization," In *Proceedings of the 7th International Conference on Evolutionary Programming*, pp. 591-600, 1998.

[42] Shi Y., and Eberhart R., "Empirical study of particle swarm optimization," In *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1945-1959, 1999.

[43] H. Shu, S. Zhang, B. Shen, and Y. Liu, Unknown input and state estimation for linear discrete-time systems with missing measurements and correlated noises, *International Journal of General Systems*, vol. 45, no. 5, pp. 648–661, 2016.

[44] Song B., Wang Z., and Sheng L., "A new genetic algorithm approach to smooth path planning for mobile robots," *Assembly Automation*, Vol. 36, No. 2, pp. 138-145, 2016.

[45] Song B., Tain G., and Zhou F., "A comparison study on path smoothing algorithms for laser robot navigatioed mobile robot path planning in intelligent space," *Journal of Information and Computational Science*, Vol. 7, No. 1, pp. 2943-2950, 2010.

[46] Tang Y., Wang Z., and Fang J., "Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm," *Expert Systems with Applications*, Vol. 38, pp. 2523-2535, 2011.

[47] C. Wen, Y. Cai, Y. Liu, and C. Wen, A reduced-order approach to filtering for systems with linear equality constraints, *Neurocomputing*, vol. 193, pp. 219–226, 2016.

[48] H. Xiao, Z. Li, C. Yang, L. Zhang, P. Yuan, L. Ding, and T. Wang, Robust stabilization of a wheeled mobile robot using model predictive control based on neuro-dynamics optimization, *IEEE Transactions on Industrial Electronics*, In press, DOI: 10.1109/TIE.2016.2606358.

[49] Zeng N., Zhang H., Chen Y., Chen B., and Liu Y., "Path planning for intelligent robot based on switching local evolutonary PSO," *Assembly Automation*, Vol. 36, No. 2, pp. 120-126 , 2016.

[50] Zeng N., Wang Z., Zhang H., and Alsaadi F. E., "A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay," *Cognitive Computation*, Vol. 8, pp. 143-152, 2016.

[51] Zhan Z., Zhang J., Li Y., and Chung H., "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, Vol. 39, No. 6, pp. 1362-1381, 2009.

[52] J. Zhang, L. Ma, and Y. Liu, Passivity analysis for discrete-time neural networks with mixed time-delays and randomly occurring quantization effects, *Neurocomputing*, vol. 216, pp. 657–665, 2016.

[53] W. Zhang, Z. Wang, Y. Liu, D. Ding, and F. E. Alsaadi, Event-based state estimation for a class of complex networks with time-varying delays: a comparison principle approach, *Physics Letters A*, vol. 381, no. 1, pp. 10–18, 2017.

[54] Zhang W. J., and Xie X. F., "DEPSO: Hybrid particle swarm with differential evolution operator," In *Proceedings of IEEE International Conference on Systtems, Man, and Cybernetics*, pp. 997-1006, 2004.

[55] Zhou F., Song B., and Tian G., "Bezier curve based smooth path planning for mobile robot," *Journal of Information and Compuational Science*, Vol. 8, No. 1, pp. 2441-2450, 2011.