

ORIGINAL ARTICLE

Open Access



Multi-objective optimal trajectory planning for manipulators based on CMOSPBO

Tingting Bao¹, Zhijun Wu^{1,2} and Jianliang Chen^{1*}

Abstract

Feasible, smooth, and time-jerk optimal trajectory is essential for manipulators utilized in manufacturing process. A novel technique to generate trajectories in the joint space for robotic manipulators based on quintic B-spline and constrained multi-objective student psychology based optimization (CMOSPBO) is proposed in this paper. In order to obtain the optimal trajectories, two objective functions including the total travelling time and the integral of the squared jerk along the whole trajectories are considered. The whole trajectories are interpolated by quintic B-spline and then optimized by CMOSPBO, while taking into account kinematic constraints of velocity, acceleration, and jerk. CMOSPBO mainly includes improved student psychology based optimization, archive management, and an adaptive ε -constraint handling method. Lévy flights and differential mutation are adopted to enhance the global exploration capacity of the improved SPBO. The ε value is varied with iterations and feasible solutions to prevent the premature convergence of CMOSPBO. Solution density estimation corresponding to the solution distribution in decision space and objective space is proposed to increase the diversity of solutions. The experimental results show that CMOSPBO outperforms than SQP, and NSGA-II in terms of the motion efficiency and jerk. The comparison results demonstrate the effectiveness of the proposed method to generate time-jerk optimal and jerk-continuous trajectories for manipulators.

Keywords: Trajectory planning, Manipulator, Multi-objective student psychology based optimization, Adaptive ε constrained method, Quintic B-spline

1 Introduction

Robotic manipulators play an important role in advance manufacturing process such as welding, casting, and milling, since they can replace manual labor to achieve highly efficient and accurate work throughout the day [1–4]. They are highly nonlinear dynamic mechanism subjected to complex nonlinear constraints [5]. Trajectory planning problem is the fundamental issue of manipulator joint motion, determining the efficiency and stability of manipulators. Generally, the trajectory planning is carried out in the joint space and the operational space [6, 7]. Although the task is performed in the operational space, the motion of

the manipulator is usually planned and controlled in the joint space. In addition, trajectory planning techniques in the joint space guarantee the smoothness of joint movement to decrease vibrations on the actuators, extending the lifespan of robotic manipulators [8].

In the joint space, numerous methods for trajectory planning have been developed in the last decades by researchers from all over the world. In terms of trajectory models, various interpolation functions such as polynomials, spline, Bézier, NURBS, radial basis functions, and a combination of them have been employed to construct smooth trajectories [9–19]. Among them, B-splines are distinguished by its good local support characteristics. Various types of B-spline have been successfully applied for generating acceleration-continuous and jerk-continuous trajectories for robotic manipulators. Rout et al. employed cubic spline to achieve acceleration-continu-

*Correspondence: cjl555@zjvit.edu.cn

¹Automobile School, Zhejiang Institute of Communications, Hangzhou, 311112, China

Full list of author information is available at the end of the article

ous trajectory planning for a 6-DOF manipulator [16]. Zeeshan and Xu adopted fifth-order B-splines to interpolate jerk-bounded trajectory [19]. Cao et al. utilized quintic B-spline to obtain smooth and continuous trajectories to achieve fruit picking operation for a manipulator with a specific end-effector [20].

In order to achieve the motion efficiency and low noise of robotic manipulators to enhance the manufacturing productivity, trajectories with minimum execution time, minimum jerk, or minimum energy are desired. Trajectory optimization is a non-convex and highly nonlinear problem due to the characteristics of manipulators. In terms of the number of optimization objectives, the trajectory optimization can be divided into two categories: single objective optimizations and multiple objective optimizations. Genetic algorithm [5], cuckoo search algorithm [12], grey wolf optimization [15], particle swarm optimization [21], harmony search algorithm [22], and their varieties [23] are adopted to search the optimal solution for single objective trajectory optimization problem. In terms of multi-objective optimization algorithm for trajectory planning, nondominated sorting genetic algorithm II (NSGA-II) [14], multi-objective particle swarm optimization algorithm [20], sequential quadratic programming (SQP) [24], and their varied combinations [25] are utilized to obtain Pareto solutions. However, the performance of these algorithms highly depends on their certain parameter configurations. The certain parameters need to be tuned properly to achieve the performance of the algorithm. Otherwise, it may not result in the optimal solutions.

Student psychology based optimization (SPBO) algorithm is a population based and parameter-free optimization algorithm [26]. SPBO has no tunable parameters. Population size that is general for all metaheuristic optimization algorithms is the only parameter that needs to be selected for SPBO. The performance of SPBO has been validated its superiority over several optimization algorithms in terms of the achievement of fast convergence mobility and global optimum solution. Balu and Mukherjee utilized SPBO to obtain the optimum site and size of distributed generation in radial distribution system [27]. Wang et al. employed SPBO to optimize the output power of TEG modules [28]. Sunflower earthworm algorithm, Garra-Rufa fish optimization, and whale optimization algorithm have been integrated with SPBO for various applications [29–31]. These integrated algorithms may increase the exploitation capacity of SPBO, with low improvement in the exploration capacity. Furthermore, although SPBO and its varieties have been employed for multi-objective optimization problems, these multi-objective problems were converted into single objective ones using weight method. The adjustment of weight values is needed when the trade-off of multiple objectives varies, weakening the superiority

of SPBO. To the best of our knowledge, the existing literature has not provided any solution integrated with SPBO and multi-objective handling method. Hence, it is worth exploring for multi-objective optimization problems.

Optimization algorithms need to deal with the constraints of optimization problems properly for feasible and optimal solutions. In the literatures, various constraint handling techniques have been proposed to tackle the constraints of equality and inequality. These techniques are mainly divided into three categories: penalty functions, constrained dominance principle, and ε -constrained method [32–35]. The preference to the feasible solutions tends to lead the population to fall into local feasible regions. The parameter tuning of penalty factor and ε is a labor-intensive task [36]. Ming et al. proposed an adaptive ε -constrained method that ε was updated according to feasible ratio and generation, while two extra control parameters were added [37]. Feng et al. utilized infeasible individuals to update ε , while an extra parameter is added to determine the rate of decline of ε [38]. Therefore, a parameter-free constraint handling technique that improves the exploration capacity of algorithms for global region and feasible solutions is still a challenge.

Motivated by above researches, this study reports a multi-objective method to generate time-jerk optimal and jerk-continuous trajectories with respect to the kinematic constraints for manipulators. This method is based on quintic B-spline and the proposed CMOSPBO. CMOSPBO integrates an improved SPBO with a multi-objective handling method and an adaptive ε -constraint handling method. The essential property of no tunable parameters of SPBO is preserved for CMOSPBO. Population size is a general parameter for all metaheuristic optimization algorithms, which is the only parameter that needs to be selected for CMOSPBO. Lévy flights and differential mutation are adopted in the improved SPBO to enhance its global exploration capacity. Fast nondominated sorting approach is adopted for nondominated Pareto solutions. An external repository with management rules and solution density estimation are proposed to store a fixed amount of nondominated Pareto solutions. Solution density estimation corresponding to the solution distribution in decision space and objective space is taken into account to increase the diversity of solutions. The proposed adaptive ε -constraint handling method is a parameter-free technique. The ε value is varied with iteration and the feasible solutions in population to prevent the premature convergence and improve the exploration capacity of CMOSPBO for global region and feasible solutions.

The rest of this paper is arranged as follows. In Sect. 2, the problem of multi-objective trajectory optimization is defined. Section 3 briefly introduces the trajectory model of quintic B-spline and the details of CMOSPBO. Performance measures are delineated in Sect. 5. Section 6 is de-

voted to the simulation results and relevant discussions. Finally, the main conclusions are presented in Sect. 6.

2 Formulation of the multi-objective optimization problem

Since trajectories in the operation space should be converted into that in the joint space to achieve a specific task for manipulator, trajectory planning in most cases is carried out in the joint space that this paper focuses on.

We expect that the trajectory should be smooth enough to avoid unnecessary vibration in joints, as well as the less execution time of a task for a manipulator to hence the productivity, with respect to the kinematic constraints of velocity, acceleration, and jerk at the same time. The kinematic limitation of velocity, acceleration, and jerk provides an in directed way to achieve the joint torque bound, while it is unnecessary to consider the dynamic constraints in the trajectory planning method. In this paper, the definition of the multi-objective optimization problem is similar to Ref. [24]. The optimization objective functions are given as

$$F_1 = \sum_{i=1}^{n_p-1} t_i, \quad (1)$$

$$F_2 = \sum_{j=1}^{n_j} \int_0^{T_t} (\text{jerk}_j(t))^2 dt, \quad (2)$$

where t_i denotes the time interval between expected adjacent positions, n_p is the number of via-positions, T_t denotes the execution time of an entire trajectory, $\text{jerk}_j(t)$ is the j th joint jerk, and n_j is the number of manipulator joints. F_1 is an execution time optimization objective, which function is related with the motion efficiency of a manipulator. F_2 is the jerk optimization objective, with the goal of reducing jerk and enhancing the smooth operation performance. These two functions are contradictory to each other due to their opposite effects. Short execution time will result in larger jerk value, while smaller jerk objective will lead to longer execution time and lower efficiency. Trade-off between these two objectives must be made in the applications utilized manipulators. This can be achieved by providing a set of feasible and time-jerk optimal solutions for user to choose.

In order to ensure the normal operation of a manipulator, the joint trajectories must be within the kinematic constraints, given as follows.

$$|\text{vel}_j(t)| \leq \text{vel}_j^{\text{limit}}, j = 1, \dots, n_j, \quad (3)$$

$$|\text{acc}_j(t)| \leq \text{acc}_j^{\text{limit}}, j = 1, \dots, n_j, \quad (4)$$

$$|\text{jerk}_j(t)| \leq \text{jerk}_j^{\text{limit}}, j = 1, \dots, n_j, \quad (5)$$

where $\text{vel}_j(t)$ and $\text{acc}_j(t)$ are the velocity and acceleration of the j th joint, respectively, $\text{vel}_j^{\text{limit}}$, $\text{acc}_j^{\text{limit}}$, and $\text{jerk}_j^{\text{limit}}$ are

the limit of velocity, acceleration, and jerk of the j th joint, respectively. We assume that the absolute value of the velocity limit is the same regardless of whether the joint is in forward or reverse rotation, so as that of acceleration and jerk.

3 Proposed method

3.1 Trajectory interpolation by means of quintic B-spline

The motion trajectory interpolated by B-spline has stronger stability for manipulator, due to the characteristic of local support of B-spline curve. In this paper, in order to achieve the jerk continuity, quintic B-spline function is utilized to construct the joint trajectories. A quintic B-spline curve can be expressed as follows.

$$p(u) = \sum_{i=0}^{n_{cp}} q_i N_{i,5}(u), \quad (6)$$

where $p(u)$ is the joint position, q_i is the i th control points of the quintic B-spline, n_{cp} is the total number of the control points, $N_{i,5}(u)$ is the i th basis function of the quintic B-spline, and u is the normalized knot vector variable. $N_{i,5}(u)$ can be obtained from the Deboor-Cox recursion formula.

The trajectory of a manipulator joint is discretized to some desired positions on the time scale, due to the needs of joint control. Two extra virtual positions are adopted at the second and the second-last positions of the desired position sequence to obtain an integrated position sequence, because of the consideration of zero jerk value at the starting and end motions. Normalized knot vector \mathbf{u} is generated from the sequence of \mathbf{t} corresponding to the integrated position sequence, given as follow.

$$\mathbf{u} = [u_0, u_1, \dots, u_{n_p+10}], \quad (7)$$

where

$$\begin{cases} u_0 = u_1 = \dots = u_5 = 0, \\ u_i = u_{i-1} + |h_{i-6}| / \sum_{j=0}^{n_p-1} |h_j|, \\ u_{n_p+5} = u_{n_p+6} = \dots = u_{n_p+10} = 1. \end{cases} \quad (8)$$

The l th order derivatives can be described as

$$p^l(u) = \sum_{j=i-5+r}^i q_j^r N_{i,5-r}(u), \quad (9)$$

where

$$q_j^l = \begin{cases} q_j, l = 0, \\ (6-l) \frac{q_j^{l-1} - q_{j-1}^{l-1}}{u_{j+6-l} - u_j}, l = 1, 2, \dots, r; j = i-5+l, \dots, i. \end{cases} \quad (10)$$

The boundary conditions on the starting and end jerks, accelerations, and velocities can be expressed as

$$\text{jerk}(0) = \sum_{j=3}^5 q_j^3 N_{i,2}(u) = q_3^3, \quad (11)$$

$$\text{jerk}(1) = \sum_{j=n_{cp}+2}^{n_{cp}+4} q_j^3 N_{i,2}(u) = q_{n_{cp}+4}^3, \quad (12)$$

$$\text{acc}(0) = \sum_{j=2}^5 q_j^2 N_{i,3}(u) = q_2^2, \quad (13)$$

$$\text{acc}(1) = \sum_{j=n_{cp}+1}^{n_{cp}+4} q_j^2 N_{i,3}(u) = q_{n_{cp}+4}^2, \quad (14)$$

$$\text{vel}(0) = \sum_{j=i-4}^i q_j^1 N_{i,4}(u) = q_1^1, \quad (15)$$

$$\text{vel}(1) = \sum_{j=n_{cp}}^{n_{cp}+4} q_j^1 N_{i,4}(u) = q_{n_{cp}+4}^1, \quad (16)$$

where $\text{jerk}(0)$, $\text{jerk}(1)$, $\text{acc}(0)$, $\text{acc}(1)$, $\text{vel}(0)$, and $\text{vel}(1)$ denote the angular jerk, acceleration, and velocity at the starting and end motions, respectively. Let \mathbf{k} be the coefficient matrix determined by the time intervals between expected adjacent positions. Then, the control knot vector can be calculated as follow.

$$p = kq, \quad (17)$$

where

$$q = [q_0 q_1 \dots q_{np} q_{np+1} q_{np+2} q_{np+3} q_{np+4}]^T, \quad (18)$$

$$p = [p_1 \dots$$

$$p_{np-1} \text{vel}(0) \text{vel}(1) \text{acc}(0) \text{acc}(1) \text{jerk}(0) \text{jerk}(1)]^T. \quad (19)$$

For a specific task of a manipulator, the jerk-continuous trajectories can be generated once the time intervals t_i ($i = 1, \dots, n_{p+1}$) is obtained.

3.2 Proposed CMOSPBO

CMOSPBO mainly includes improved SPBO, archive management, and adaptive ε -constraint handling method. SPBO integrated with Pareto ranking scheme could be a direct way to extend the algorithm to handle multi-objective optimization problems. The utilization of global attraction mechanisms integrated with an archive of historically found nondominated vectors will lead convergence to globally nondominated solutions.

3.2.1 Improved SPBO

SPBO is inspired by the psychology of the students trying to improve their examination up in a class [26]. The students in the class are divided into four categories: best student, good student, average student, and students who try to improve randomly. The selection of student category is a random process due to the different psychologies of different students, except the best student. Several improvements are adopted to enhance the global exploration capacity of SPBO.

- Best student

The student obtained the highest overall grade in an examination is the best student in the class. Improvement of the best student is expressed as follow.

$$x_{\text{best}}^{it+1} = x_{\text{best}}^{it} + (-1)^{r_0} \times \text{rand} \times (x_{\text{best}}^{it} - x_{\text{rand}}^{it}), \quad (20)$$

where x_{best}^{it} and x_{rand}^{it} respectively denote the best student and randomly selected student in it th iteration, rand is a random value in between 1 and 0, r_0 is a parameter that is randomly selected as 1 or 2. Note that the student with maximum density is selected as the best student for multi-objective optimization problems. The detail of density estimation is shown in Sect. 3.2.2.

- Good student

Some of the good students tend to offer similar or better kind of effort from the best student. The improvement of some good student is represented in Equation (18). In addition, some good students are affected by the best student and the mean performance of the class, represented in Equation (19).

$$x_i^{it+1} = x_{\text{best}}^{it} + \text{rand} \times (x_{\text{best}}^{it} - x_i^{it}), \quad (21)$$

$$x_i^{it+1} = x_i^{it} + \text{rand}_1 \times (x_{\text{best}}^{it} - x_i^{it}) \\ + \text{rand}_2 \times (x_i^{it} - x_{\text{mean}}^{it}), \quad (22)$$

where x_i^{it} denotes the i th student, x_{mean}^{it} is the mean performance of the class, and rand_1 and rand_2 are different random values. Note that x_{mean}^{it} is obtained from feasible solutions and relaxation solutions for constrained problems.

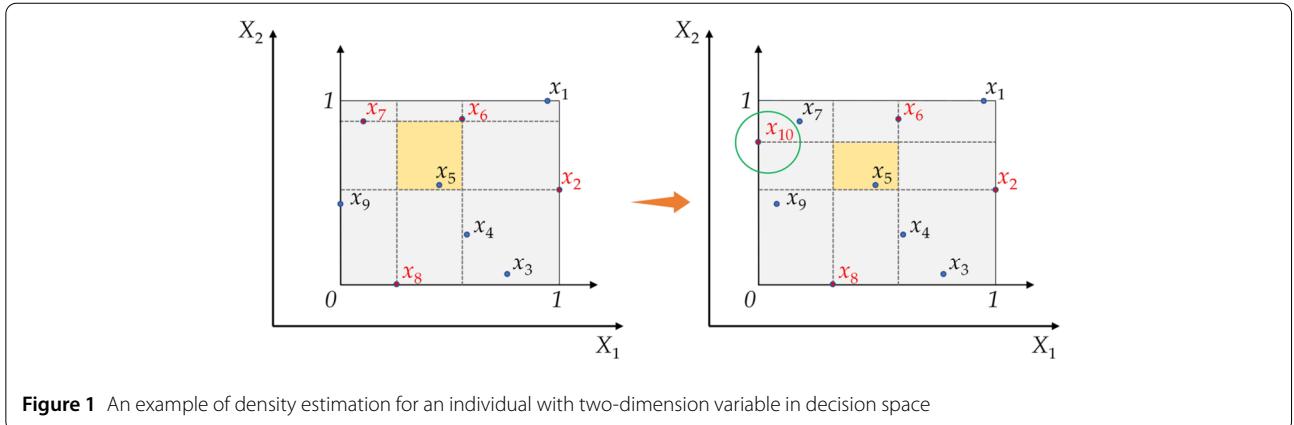
- Average student

Average student can be affected by any student in the class, which is the first difference between the traditional SPBO and the proposed algorithm. Improvement of the average student is expressed as follow.

$$x_i^{it+1} = x_i^{it} + \text{rand} \times (x_{\text{rand}}^{it} - x_i^{it}). \quad (23)$$

- Students who try to improve randomly

Except the above three categories of students, other students tend to improve their performance by themselves.



Lévy flights and differential mutation are adopted to enhance its global exploration capacity, which is the second difference between the traditional SPBO and the proposed algorithm. The selection of the two methods is a random process. Improvement of this type of student is expressed as follow.

$$x_i^{it+1} = x_i^{it} + L(s, \lambda), \quad (24)$$

$$x_i^{it+1} = rand_1 \times (x_{best}^{it} - x_i^{it}) + rand_2 \times (x_{rand}^{it} - x_i^{it}), \quad (25)$$

where $L(s, \lambda)$ is Lévy flight function. Lévy step can be expressed as:

$$L(s, \lambda) = \frac{\mu}{(\gamma)^{\frac{3}{2}}}, \quad (26)$$

where μ and γ are normal distribution, respectively.

3.2.2 Density estimation

In order to maintain a sustainable diversity in a population, the density estimation method is proposed to eliminate the individuals with small density in the decision space and the objective space. The normalization is considered to eliminate the effect of the order of magnitude. The density of an individual in archive in the objective space can be calculated by using two adjacent individuals along a nondominated front, represented as follow.

$$D_f(i) = \begin{cases} 1, & \text{if } i = 1 \text{ or } i = Pop_Size, \\ \prod_{j=1}^{n_f} \frac{f_{j,i+1} - f_{j,i-1}}{f_j^{\max} - f_j^{\min}}, & \text{otherwise,} \end{cases} \quad (27)$$

where D_f denoted the density of an individual in the objective space, n_f denotes the dimension of objective function, Pop_Size is the size of population in archive.

In the decision space, the density of an individual in archive can be calculated by several adjacent individuals

along the variable axis, represented as follow.

$$D_x(i) = \begin{cases} 1, & \text{if } i = 1 \text{ or } i = Pop_Size, \\ \prod_{j=1}^{n_x} \frac{x_{j,i+1} - x_{j,i-1}}{x_j^{\max} - x_j^{\min}}, & \text{otherwise,} \end{cases} \quad (28)$$

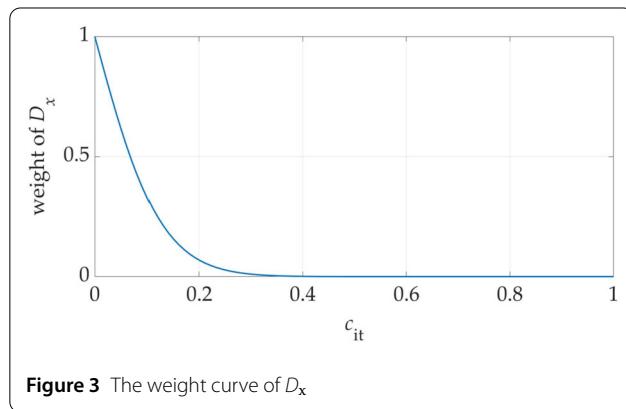
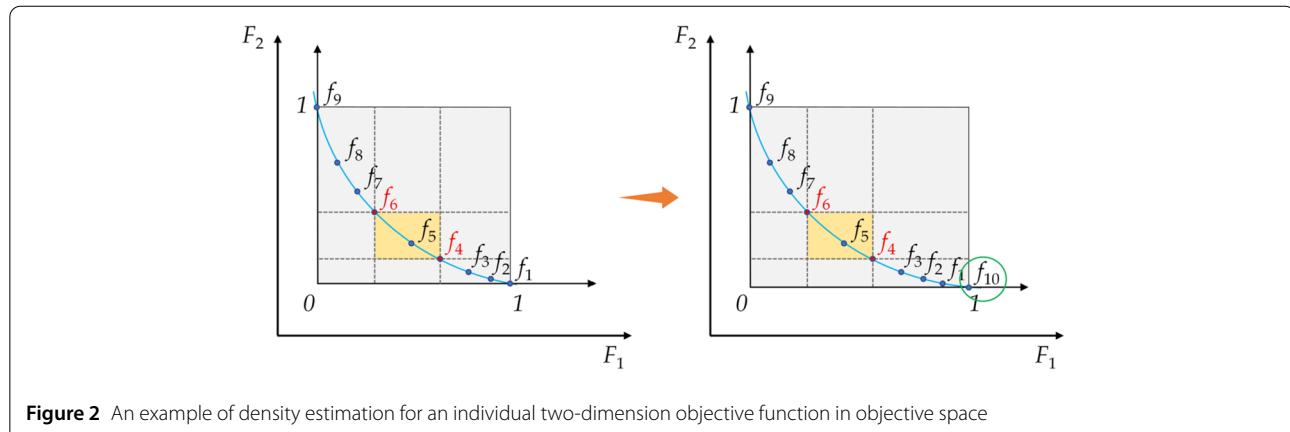
where D_x denotes the density of an individual in the decision space, n_x denotes the dimension of variable.

For instance, let's consider a problem with two variables and two objectives. Figure 1 shows density estimation for an individual with two-dimension variable in decision space. In an iteration, the density estimation for x_5 is calculated be using four solutions, x_2 , x_6 , x_7 , and x_8 . The density of x_5 becomes small after x_{10} is added to archive, since x_{10} expands the gray area and is closer to x_5 compared with x_7 along X_2 . The density estimation for f_5 in Fig. 2 is similar to x_5 in Fig. 1.

The overall density of an individual is calculated as the sum of individual area in the objective function space and the decision space with weights.

$$D(i) = \frac{2}{1 + e^{8\left(\frac{1}{1-c_{it}} - \frac{1}{c_{it}+1}\right)}} \times D_x(i) + \left(1 - \frac{2}{1 + e^{8\left(\frac{1}{1-c_{it}} - \frac{1}{c_{it}+1}\right)}}\right) \times D_f(i), \quad (29)$$

where c_{it} denotes the ratio of cumulative iterations to total iterations. The weight of D_x becomes small as the increase of iteration, as shown in Fig. 3. The diversity of solutions in the decision space in the early iteration process is considered for the improvement of the exploration capacity of the proposed algorithm. The density of solutions in the objective function space has decisive influence on the diversity of solutions, due to the consideration of diversity of solutions in Pareto front.



3.2.3 Archive management

Archive is an external repository utilized to store non-dominated Pareto solutions found so far. The function of archive management is to decide whether a solution should be added to the archive or not. Archive is empty until a nondominated Pareto solution is found in an iteration. Its size is generally specific and fixed. If the number of nondominated Pareto solutions found so far is beyond the fixed size, the archive will be updated to keep the size. Its update rules are given as follows.

- If a new found solution in the current iteration is dominated by at least one solution in the archive, then the new solution is not allowed to be added to the archive;
- If at least one solution in the archive is dominated by a new found solution in the current iteration, then the new found solution is allowed to be added to the archive and the dominated solutions will be removed from the archive;
- If no solution in the archive is dominated by a new found solution in the current iteration and the new found solution is not dominated by any solution in the archive, then the new solution is allowed to be added to the archive;

- If the size of archive after the solution replacement is beyond the fixed value, density estimation is adopted to remove the solutions with small density until the size of archive is equal to the fixed value.

3.2.4 Adaptive ϵ -constraint handling method

Constraint handling techniques are utilized for constrained optimization problems to tackle the constraints to find the feasible solutions. Adaptive ϵ -constraint handling method is proposed to enable the population to search the feasible solution region and the relaxation boundary of infeasible solution region. Infeasible individuals within ϵ violation are converted into relaxation feasible solutions, enhancing the exploration capacity of algorithm for feasible and optimal solution. The formulation of updating ϵ is expressed as follow.

$$\epsilon(c_{it}, c_f) = \begin{cases} \frac{1}{1 + e^{\left(\frac{1}{1 - \min(c_{it}, c_f)} - \frac{1}{\min(c_{it}, c_f)}\right)}}, & it \leq 0.5it_{max}, \\ \frac{1}{1 + e^{\left(\frac{1}{1 - \max(c_{it}, c_f)} - \frac{1}{\max(c_{it}, c_f)}\right)}}, & it > 0.5it_{max}, \end{cases} \quad (30)$$

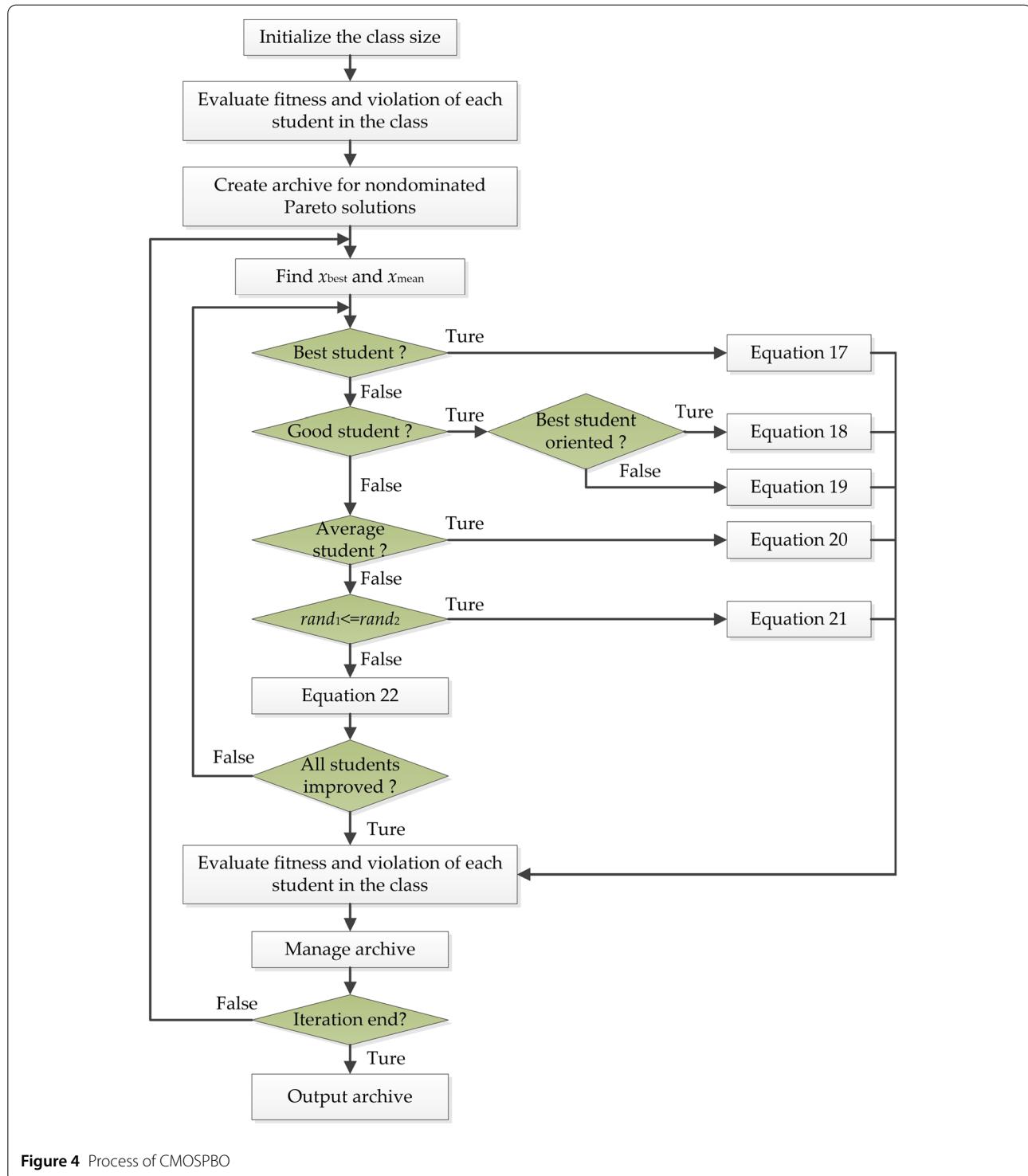
where c_f is the proportion of feasible solutions to all solutions, it_{max} is the maximum iteration.

ϵ is updated until the iteration reaches the termination condition. At the stage of $it \leq it_{max}$, the smaller value between c_f and c_{it} is selected for the calculation of ϵ , allowing the population to explore the entire region. This enables algorithm to search for optimal solutions and prevents population from trapping to local optima. At the stage of $it > it_{max}$, the parameter with larger value determines ϵ , achieving the diversity of feasible solutions in archive.

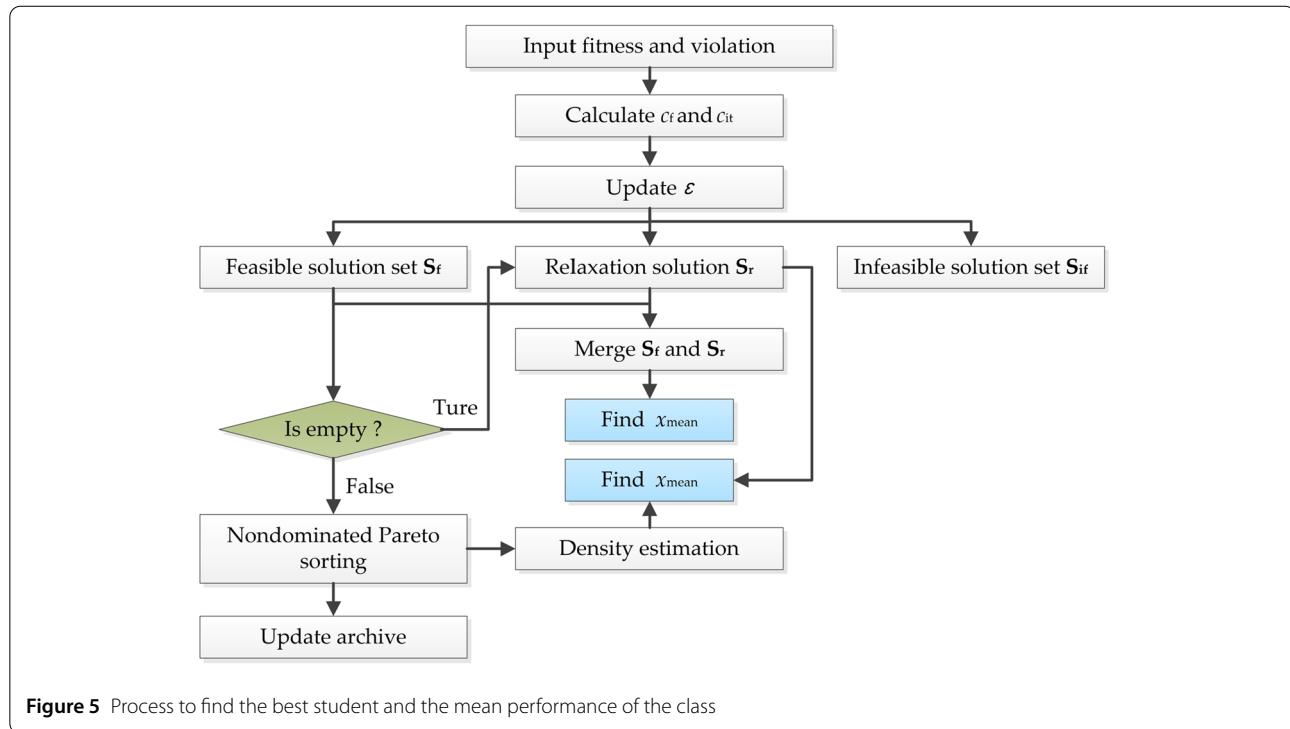
In summary, the flowchart of the proposed CMOSPBO is shown in Fig. 4.

The algorithm of CMOSPBO is the following.

- Initialize the class (population);
- Evaluate each student in the class;



- Create archive for nondominated Pareto solutions;
 - Find x_{best} and x_{mean} , as shown in Fig. 5;
- 1) Input fitness and violation of each student from Step 2;
 - 2) Calculate c_f and c_{it} ;
 - 3) Update ε (more detail in Sect. 3.3);
 - 4) Divide the class into three sets (feasible solution set S_f , relaxation solution S_r , and infeasible solution set S_{if}) according to updated ε ;
 - 5) Calculate the mean performance of merged solutions and assign the result to x_{mean} ;

**Table 1** Desired via-positions of the continuous task

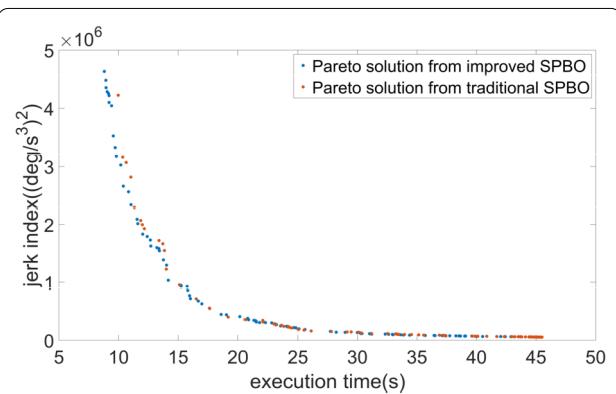
Via-Positions	Joint No. (°)					
	1	2	3	4	5	6
1	-10	20	15	150	30	120
2 (Virtual point)	-	-	-	-	-	-
3	60	50	100	100	110	60
4	20	120	-10	40	90	100
5 (Virtual point)	-	-	-	-	-	-
6	55	35	30	10	70	25

Table 2 Kinematic constraints of the continuous task

Kinematic Constraints	Joint No.					
	1	2	3	4	5	6
Velocity (°/s)	100	95	100	150	130	110
Acceleration (°/s ²)	60	60	75	70	90	80
Jerk (°/s ³)	60	66	85	70	75	70

6) Check S_f , if S_f is empty, then select the student with the smallest violation in S_r as x_{best} ; otherwise, sort S_f by the nondominated Pareto sorting method, estimate the density of Pareto solutions, and select the student with the biggest density as x_{best} :

- If the student is the best student, improve the student using Equation (17);
- If the student is the good student affected by the best student, improve the student using Equation (18); otherwise, improve the student using Equation (19);

**Figure 6** Pareto solution sets from CMOSPBO and the traditional SPBO with the same multi-objective handling method and constraint handling method

- If the student is the average student, improve the student using Equation (20);
- Randomly select the student who try to improve randomly and improve using Equation (21) or (22);
- If all students have been improved, then evaluate the new class and manage archive; otherwise, go to Step 5;
- If termination condition is satisfied, then output archive; otherwise, go to Step 4.

4 Performance measures

For solutions in archive from the proposed algorithm, it is necessary to evaluate the fitness factor for each solution

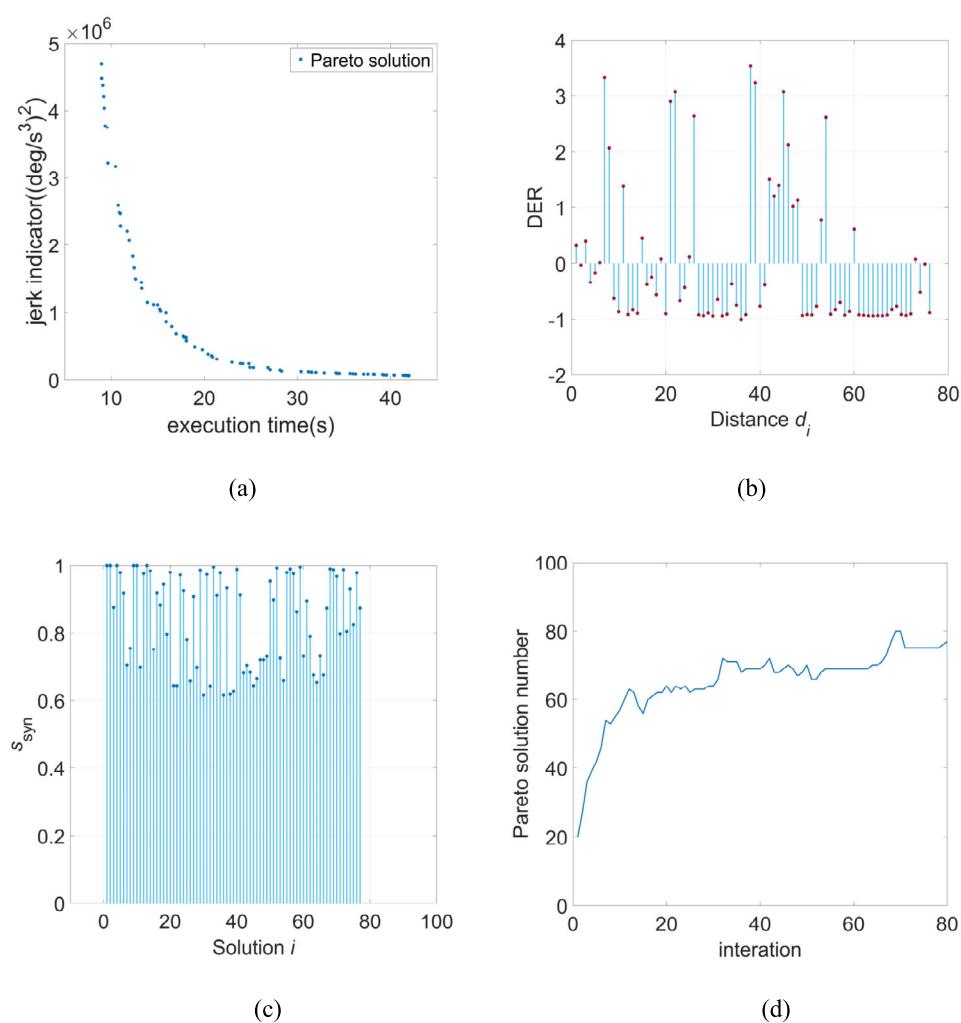


Figure 7 A Pareto solution set includes a solution with execution time of 9.1 s. (a) Pareto solutions for time-jerk optimal trajectory planning; (b) Values of distance error ratio for the Pareto front; (c) Synthetical membership value for the Pareto front; (d) Number variation of Pareto solution in archive

and the distribution along the Pareto Front. A distance error ratio is defined to present the Euclidean change in the objective function space, expressed as follow.

$$DER(i) = (d_i - d_{\text{mean}})/d_{\text{mean}}, \quad (31)$$

where d_i denotes the distance between consecutive solutions, d_{mean} denotes the mean value of all d_i .

In order to evaluate the fitness of a solution in archiving in terms of execution time and jerk, a fuzzy membership function is adopted as a decision-maker. Fuzzy comprehensive evaluation is an effective decision-making technique for multi-objective problems. The equations of the fuzzy membership and synthetical membership value are

expressed as follow.

$$s_j(i) = (F_{j\max} - F_j(i))/(F_{j\max} - F_{j\min}), \quad (32)$$

$$s_{\text{syn}}(i) = (s_1(i) + s_2(i))/\max\{s_1 + s_2\}, \quad (33)$$

where F_{\max} and F_{\min} are the maximum and minimum objective function values, respectively.

5 Results and discussion

The numerical simulations were conducted by solving two test tasks to validate the effectiveness of the proposed method for time-jerk optimal trajectories for multi-joint manipulators. All the experiments were run on a desktop. The specifications of the desktop are 13th Gen Intel(R) Core(TM) i7-13700KF, RAM of 32.0 GB, and Windows 10 64-bit operating system.

Table 3 Comparison results from NSGA-II [6] and proposed CMOSPBO

Work	Execution time range (s)	Jerk index range
NSGA-II [6]	[9.058, 13.96]	[55.55 deg/s ³ , 188.98 deg/s ³]
CMOSPBO	[8.954, 41.97]	[6.2326E4(deg/s ³) ² , 4.6911E6(deg/s ³) ²]

Table 4 Maximum kinematic values of all joints from different methods with the same execution time

Work	Kinematic	Joint No.					
		1	2	3	4	5	6
Gasparetto-Zanotto [8]	Velocity	37.57	41.96	61.49	28.90	41.53	38.14
	Acceleration	39.07	43.65	65.60	15.87	33.88	39.94
	Jerk	54.94	65.90	78.73	23.02	52.73	65.15
Zeng-Huang [6]	Velocity	39.27	47.05	62.31	29.15	43.35	41.98
	Acceleration	42.67	53.04	70.52	17.60	35.75	48.19
	Jerk	59.00	62.28	82.92	26.23	55.4	60.20
Piazz-Visioli [13]	Velocity	34.89	45.79	56.47	25.24	37.89	40.59
	Acceleration	42.36	55.92	71.62	15.89	34.61	50.86
	Jerk	35.74	51.3	50.69	16.56	36.05	45.73
Simon-Isik [19]	Velocity	39.84	47.67	57.64	28.67	44.29	43.00
	Acceleration	39.03	46.40	62.05	19.89	35.61	43.73
	Jerk	54.82	59.28	80.84	25.99	49.04	61.47
Proposed	Velocity	36.61	50.58	58.34	26.35	40.07	44.80
	Acceleration	37.72	53.27	62.00	15.27	30.30	47.60
	Jerk	52.08	65.77	73.26	23.01	46.92	62.48

Table 5 Mean kinematic values of all joints from different methods with the same execution time

Work	Kinematic	Joint No.					
		1	2	3	4	5	6
Gasparetto-Zanotto [8]	Velocity	16.94	70.7	27.45	15.38	15.92	19.47
	Acceleration	19.08	18.15	30.60	6.50	14.92	21.84
	Jerk	26.26	20.69	41.26	6.39	18.18	30.35
Zeng-Huang [6]	Velocity	16.33	21.05	26.33	15.39	15.59	19.78
	Acceleration	19.36	20.74	30.93	6.41	14.61	23.51
	Jerk	27.86	25.33	43.61	7.00	17.93	33.95
Piazz-Visioli [13]	Velocity	13.14	16.75	20.9	11.94	11.65	16.38
	Acceleration	16.82	17.55	26.67	5.89	11.56	20.96
	Jerk	29.84	27.98	46.07	7.76	17.16	37.73
Simon-Isik [19]	Velocity	16.87	22.11	26.41	15.38	16.76	20.24
	Acceleration	19.48	19.99	30.07	6.53	15.02	23.51
	Jerk	27.51	23.97	41.27	9.10	18.07	33.77
Proposed	Velocity	11.82	17.67	18.27	11.01	10.88	15.99
	Acceleration	15.42	18.23	23.94	5.38	10.57	20.31
	Jerk	27.15	25.87	40.53	9.31	16.54	35.06

A typical continuous trajectory task for a manipulator with six joints was employed [6]. The desired via-positions of the continuous task are listed in Tables 1. Table 2 shows the kinematic constraints of joints, including velocity, acceleration, and jerk for the continuous task. The general parameters of CMOSPBO are iteration number = 80, population size = 100.

Figure 6 illustrates the Pareto solution sets by CMOSPBO and the traditional SPBO with the same multi-objective handling method and constraint handling method. The results from the traditional SPBO show that solutions are mainly concentrated in the region with smaller jerk values, indicating that the traditional SPBO is prone to fall into local exploration. Furthermore, the Pareto so-

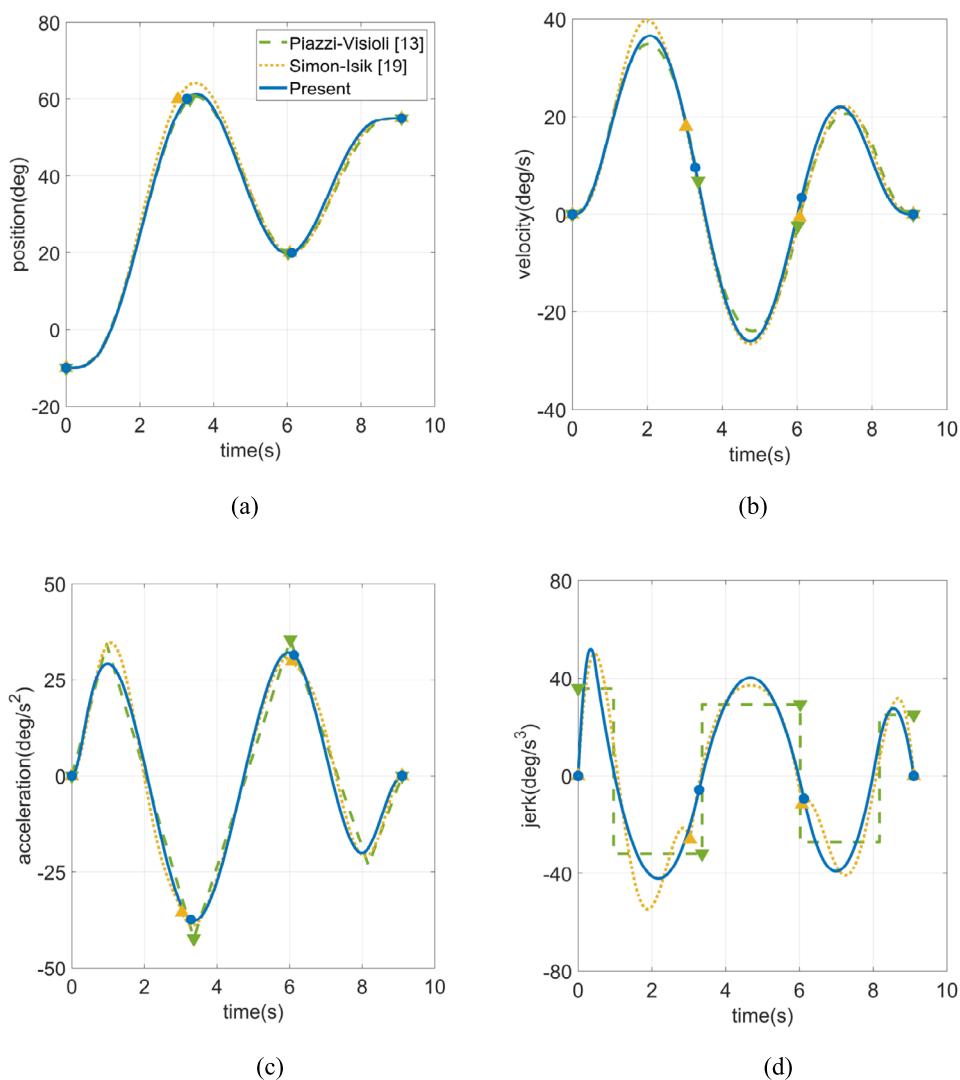


Figure 8 Comparison curves of Joint 1 with execution time of 9.1 s. (a) Joint position curves; (b) Joint velocity curves; (c) Joint acceleration curves; (d) Joint jerk curves

lution set by improved SPBO is closer to the Pareto Front than that by the traditional SPBO. The results verify that the improved SPBO enhance the explore capacity of the traditional SPBO.

The trajectories were also interpolated by quintic B-spline and separately optimized by SQP [8] and NSGA-II [6]. Furthermore, Cubic splines [13] and trigonometric splines [19] were utilized for trajectories. The execution time was around 9.1 s in [6, 8]. The proposed method has the capacity to provide a solution with the similar execution time. A Pareto solution set includes a solution with execution time of 9.1 s generated by CMOSPBO is illustrated in Fig. 7. The time intervals of 9.1 s from the proposed method are 0.4549 s, 2.8247 s, 2.8412 s, 1.9595 s, and 1.0189 s, respectively.

Figure 7 shows that the execution time in the Pareto solutions ranges from 8.9539 s to 41.9666 s while the jerk indicator from 6.2326E4(deg/s³)² to 4.6911E6(deg/s³)², which has the larger range than that by NSGA-II [6], shown in Table 3. This indicates that CMOSPBO outperforms NSGA-II in terms of exploration capacity. As shown in Fig. 7(a), the longest execution time corresponds to the smallest jerk index. The Pareto optimal solutions provide a number of selections for decision-makers. Figure 7(b) shows the values of distance error ratio for the Pareto front. Most DER values between two consecutive points of the generated Pareto front are near the mean distance, while there are a few distances beyond the mean distance. Figure 7(c) presents the synthetical membership value for the generated Pareto solutions. The value for the 1st, 2nd, 4th,

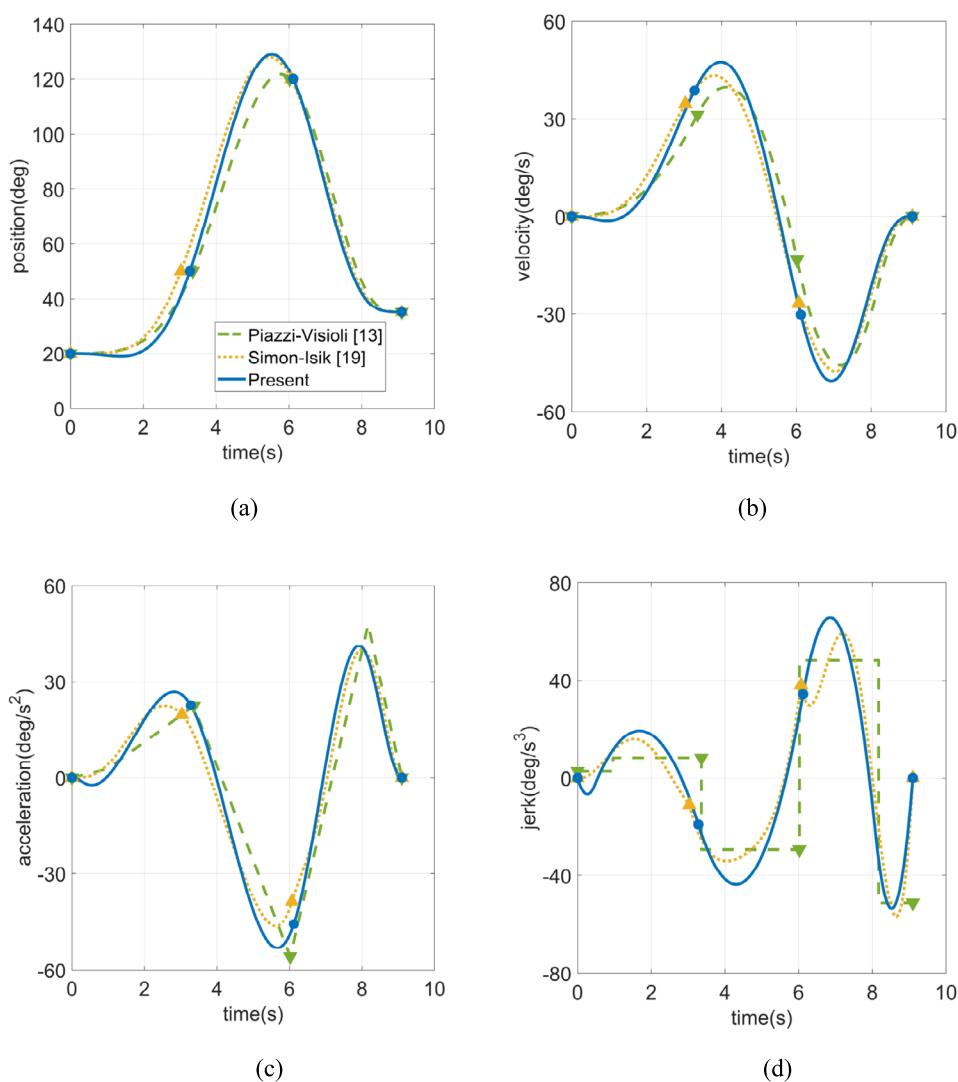


Figure 9 Comparison curves of Joint 2 with execution time of 9.1 s. (a) Joint position curves; (b) Joint velocity curves; (c) Joint acceleration curves; (d) Joint jerk curves

9th, 10th, 13rd solutions is 1, indicating that these solutions provide the best trade-off between execution time and jerk. Figure 7(d) presents the number variation of Pareto solution in archive. In general, the Pareto solution number increases as the iteration number increases. The minor number fluctuations indicates that CMOSPBO has the capacity to avoid premature convergence.

Tables 4 presents the maximum kinematic values for the solutions, compared with the works [6, 8, 13, 19]. The results in Table 4 indicates that the whole trajectory of all joint satisfy the kinematic constraints. For the comparison with the same trajectory model, the results of 1st, 3rd, 4th, and 5th joints from our work are the lowest, compared with that yielded by benchmark methods [6, 8]. The maximum jerks of 1st, 3rd, 4th, and 5th joints

yielded by CMOSPBO are lower than those from SQP [8] with 5.21%, 6.95%, 0.04%, and 11.02%, respectively, and the results optimized by NSGA-II [6] with 13.29%, 13.19%, 13.99%, and 18.07%, respectively. The maximum accelerations of 1st, 3rd, 4th, and 5th joints yielded by CMOSPBO are lower than those from SQP [8] with 3.46%, 5.49%, 3.78%, and 10.57%, respectively, and the results optimized by NSGA-II [6] with 13.12%, 13.74%, 15.26%, and 17.99%, respectively. The maximum velocities of 1st, 3rd, 4th, and 5th joints yielded by CMOSPBO are lower than those from SQP [8] with 2.56%, 5.12%, 8.82%, and 3.52%, respectively, and the results optimized by NSGA-II [6] with 7.27%, 6.80%, 10.63%, and 8.19%, respectively. CMOSPBO performed better than SQP [8] and NSGA-II [6] in terms of jerk optimization. For the comparison with the different

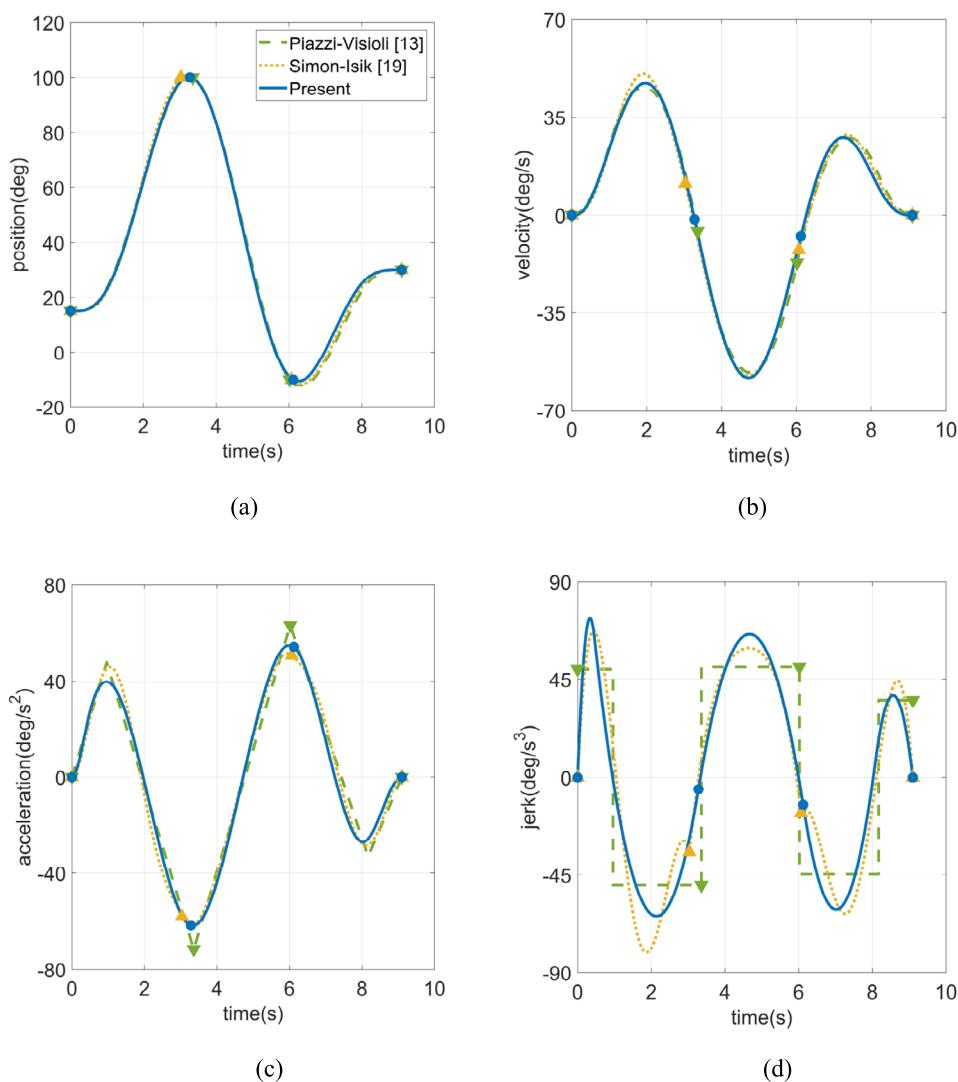


Figure 10 Comparison curves of Joint 3 with execution time of 9.1 s. (a) Joint position curves; (b) Joint velocity curves; (c) Joint acceleration curves; (d) Joint jerk curves

trajectory models and optimization algorithms, the proposed method outperforms that by Simon-Isik [19], but is slightly inferior to that by Piazzoli-Visioli [13] in terms of maximum velocity and jerk, since the cubic trajectory model [13] generate no-zero jerk values at initial and final movements.

Tables 5 presents the mean kinematic values for the solutions, compared with the works [6, 8, 13, 19]. Regardless of trajectory models, the mean velocities of all joints from our work are the lowest, compared with that yielded by benchmark methods. The mean velocities from 1st to 6th joints yielded by CMOSPBO are lower than those from SQP [8] with 30.22%, 75.01%, 33.44%, 28.41%, 31.66%, and 17.87%, respectively, and the results optimized by NSGA-II [6] with 38.16%, 19.13%, 44.12%, 39.78%, 43.29%, and 23.70%,

respectively. The mean accelerations of 1st, 3rd, 4th, 5th, and 6th joints yielded by CMOSPBO are lower than those from SQP [8] with 19.18, 21.76%, 17.23%, 29.16%, and 7.01%, respectively, and the results optimized by NSGA-II [6] with 13.12%, 13.74%, 15.26%, and 17.99%, respectively. The mean jerks of 3rd and 5th joints yielded by CMOSPBO are lower than those from SQP [8] with 1.77% and 9.02%, respectively, and the results optimized by NSGA-II [6] with 7.60% and 17.99%, respectively. The results validate that the proposed CMOSPBO is competitive with those algorithms proposed by Huang et al. [6] and, Gasparetto and Zanotto [8]. For the comparison with the different trajectory models and optimization algorithms [13, 19], the proposed method has huge superiority in generation of mean kinematic values including velocity, acceleration, and jerk.

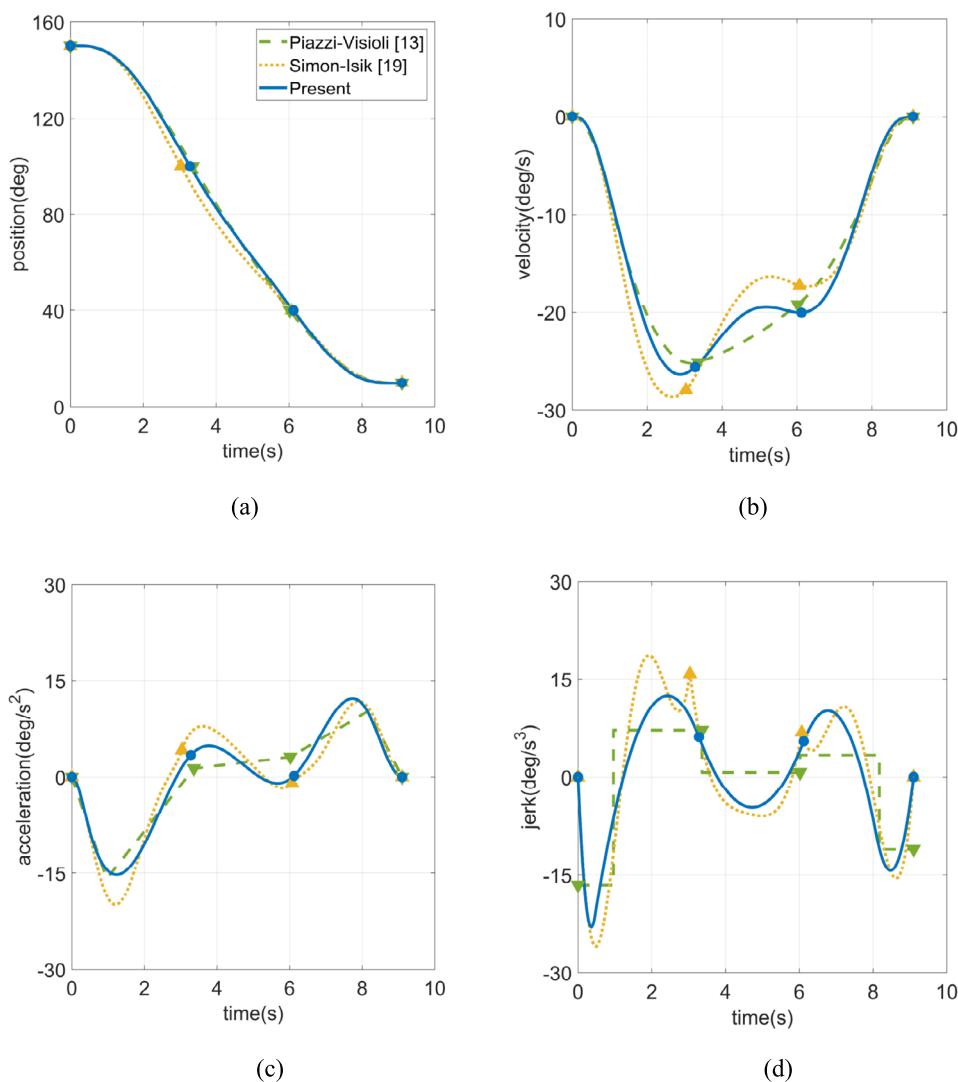


Figure 11 Comparison curves of Joint 4 with execution time of 9.1 s. (a) Joint position curves; (b) Joint velocity curves; (c) Joint acceleration curves; (d) Joint jerk curves

In order to show the resulting trajectories interpolated by different splines, the motion curves of the joint position, velocity, acceleration, and jerk resulted from the proposed method and benchmark methods [13, 19] are illustrated in Figs. 8 to 13. The curves by the proposed method and Simon-Isik [19] are jerk-continuous and the trajectories by Piazzoli-Visioli [13] are acceleration-continuous, achieving the smoothness of the optimized trajectories. In addition, the values of velocity, acceleration, and jerk of the resulting trajectories by the proposed method and Simon-Isik [19] are all zero at the initial and final motions. The no-zero jerk values at initial and final movements by Piazzoli-Visioli [13] may increase the vibration of the joint actuators and exacerbate the wear between moving parts in the joints of ma-

nipulators. The execution time need to be configured before optimization, while the proposed method outputs the execution time as a result. The curves demonstrate that the trajectories obtain by the proposed method are smooth, which may decrease vibrations on the actuators and extend the lifespan of robotic manipulators. This is significant for manipulators to improve their reliability and maybe helpful for users to enhance production cost, especially in the automation field requiring long-time operations using manipulators. Therefore, the proposed multi-objective optimal trajectory planning technique is a competitive solution for manipulators to generate time-jerk optimal joint trajectories.

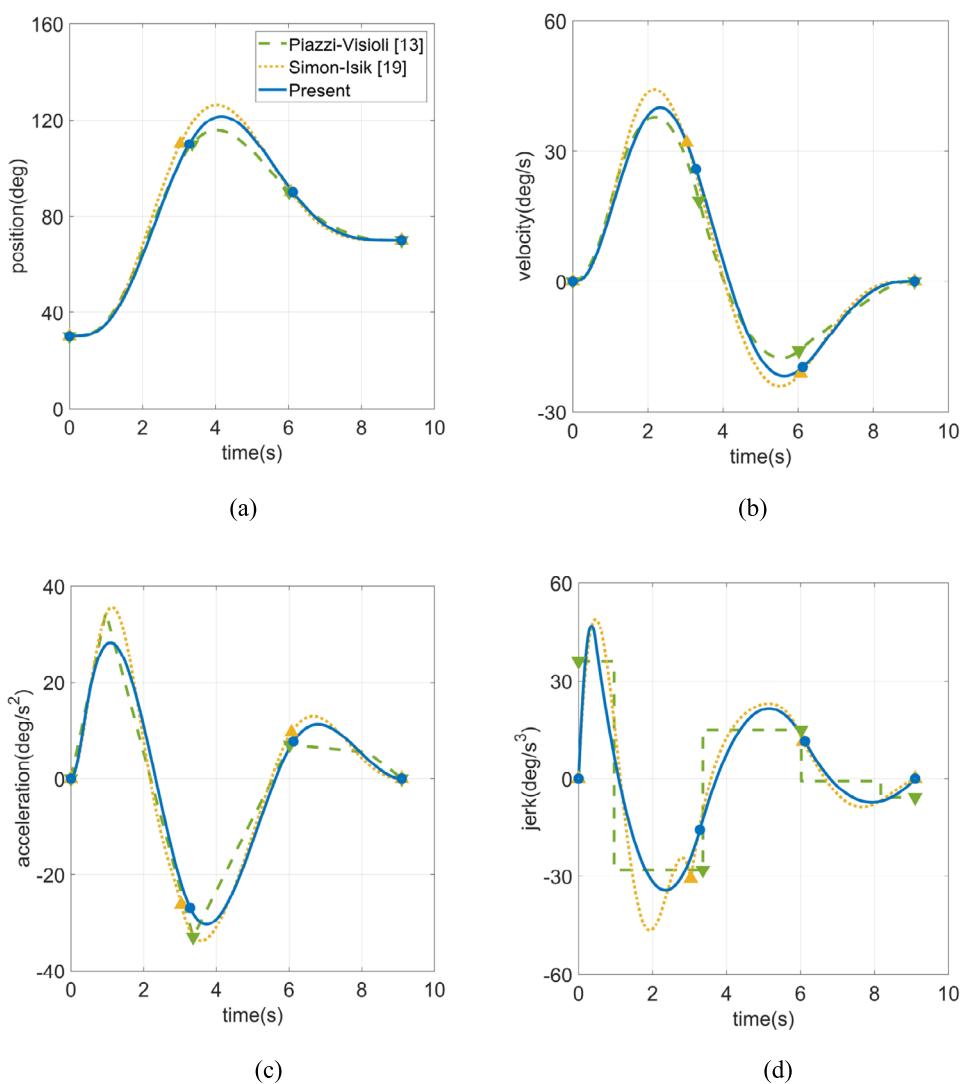


Figure 12 Comparison curves of Joint 5 with execution time of 9.1 s. (a) Joint position curves; (b) Joint velocity curves; (c) Joint acceleration curves; (d) Joint jerk curves

6 Conclusions

The problem of generation of time-jerk optimal trajectories with jerk continuity in the joint space is considered for manipulators in this paper, with respect to the kinematic constraints. A novel technique based on CMOSPBO is proposed for this problem. Quintic B-spline is employed to interpolate entire trajectories. The proposed CMOSPBO algorithm is constructed to optimize the execution time and jerk of the trajectories. Density estimation and archive management are taken into account for the diversity and convergence of CMOSPBO. An adaptive ε -constraint handling method enables CMOSPBO to search for the optimal and feasible solutions and prevents population from trapping to local optima. A typical experiment is conducted

to validate the effectiveness of the proposed technique for manipulators. The results can be summarized as follows.

- 1) The comparison between the improved SPBO in CMOSPBO and the traditional SPBO verifies that improved SPBO is capable of enhancing the explore capacity of the traditional SPBO.
- 2) The ranges of execution time and jerk indicator from CMOSPBO are larger than that from NSGA-II, verifying the global exploration capacity of CMOSPBO.
- 3) The maximum kinematic values yielded by CMOSPBO is decreased by up to 11.02% compared with SQP, and by up to 18.07% compared with NSGA-II. The mean kinematic values yielded by CMOSPBO is decreased by up to 75.01% compared

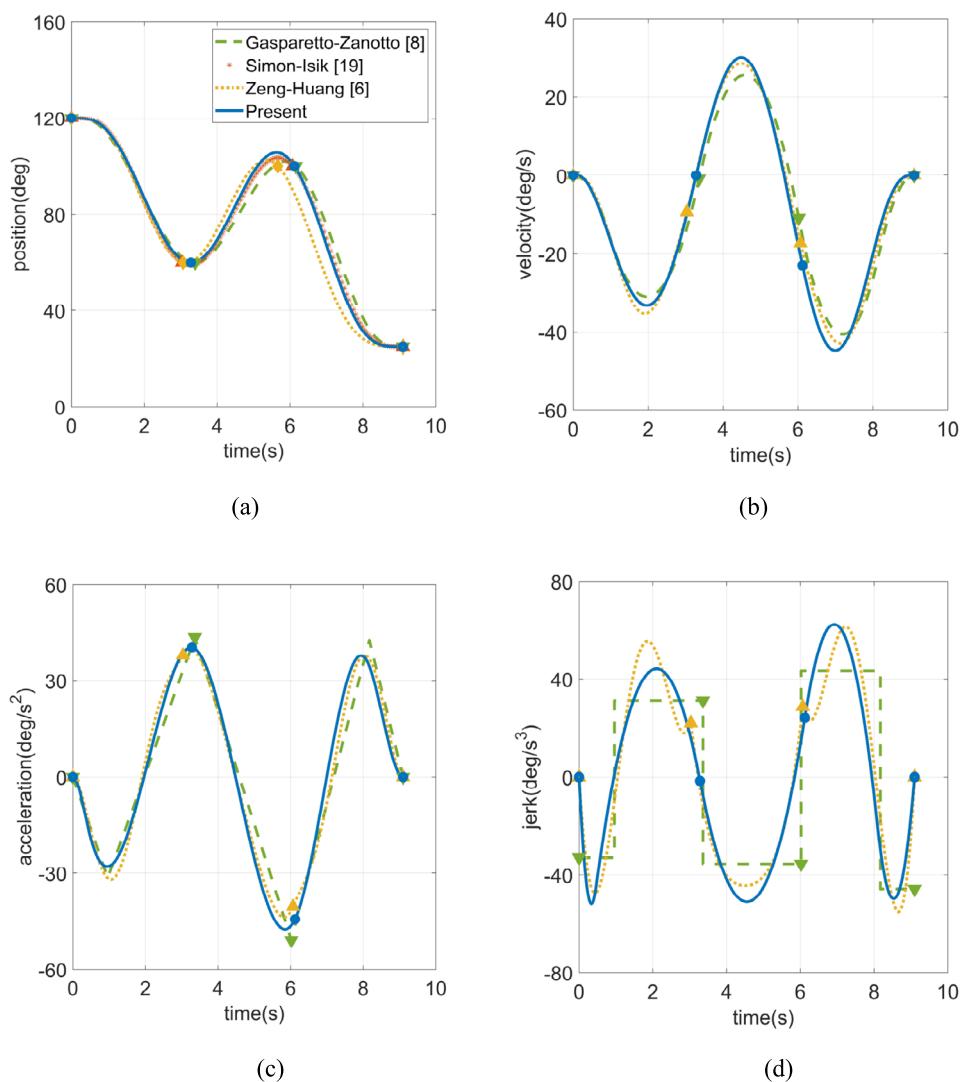


Figure 13 Comparison curves of Joint 6 with execution time of 9.1 s. (a) Joint position curves; (b) Joint velocity curves; (c) Joint acceleration curves; (d) Joint jerk curves

with SQP, and by up to 44.12% compared with NSGA-II. The experimental results show that the CMOSPBO outperform than SQP and NSGA-II in terms of the motion efficiency and jerk.

- 4) All joints start at the initial positions and reach the via-positions synchronously, with zero values of velocity, acceleration, and jerk at the initial and final movements. This might decrease vibrations on the actuators, which is significant for manipulators to improve their reliability.
- 5) The joint movement curves demonstrate the effectiveness of the proposed method to generate time-jerk optimal and jerk-continuous trajectories for manipulators.

In future work, the proposed method can be further validated through the experiments using a 6-axis UR manipulator. Various trajectory models such as Bézier and NURBS, could be extended to replace the quintic B-spline to verify the generality of MOSPBO algorithm. In addition, the proposed method could be implemented for some practical applications like assembly and welding.

Acknowledgements

The authors acknowledge the support and inspiration of Zhejiang provincial soft science project.

Author contributions

Conceptualization, TB, JC and ZW. Methodology, TB and ZW. Validation, ZW. Writing—original draft preparation, ZW. Writing—review and editing, TB, JC

and ZW. Supervision, JC. Project administration, ZW. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by Zhejiang Provincial Soft Science Project of China under Grant Number 2023C35088.

Data availability

Not applicable.

Declarations

Competing interests

The authors declare no competing interests.

Author details

¹Automobile School, Zhejiang Institute of Communications, Hangzhou, 311112, China. ²Key Laboratory of E&M, Ministry of Education, Zhejiang University of Technology, Hangzhou, 310012, China.

Received: 12 May 2024 Revised: 7 August 2024

Accepted: 5 September 2024 Published online: 01 November 2024

References

1. F. Mo, H.U. Rehman, F.M. Monetti, J.C. Chaplin, D. Sanderson, A. Popov, A. Maffei, S. Ratchev, A framework for manufacturing system reconfiguration and optimisation utilising digital twins and modular artificial intelligence. *Robot. Comput.-Integr. Manuf.* **82**, 102524 (2023). <https://doi.org/10.1016/j.jrcim.2022.102524>
2. S. Sampathkumar, F. Augustin, M.K. Kaabar, X-G. Yue, An integrated intuitionistic dense fuzzy Entropy-COPRAS-WASPAS approach for manufacturing robot selection. *Adv Mech. Eng.* **15**(3) (2023). <https://doi.org/10.1177/16878132231160265>
3. Y. Zhang, H. Liu, W. Cheng, L. Hua, D. Zhu, A novel trajectory planning method for robotic deburring of automotive castings considering adaptive weights. *Robot. Comput.-Integr. Manuf.* **86**, 102677 (2024). <https://doi.org/10.1016/j.jrcim.2023.102677>
4. S. Ma, K. Deng, Y. Lu, X. Xu, Robot error compensation based on incremental extreme learning machines and an improved sparrow search algorithm. *Int. J. Adv. Manuf. Technol.* **125**, 5431–5443 (2023). <https://doi.org/10.1007/s00170-023-10957-6>
5. F.J. Abu-Dakka, I.F. Assad, R.M. Alkhodour, M. Abderahim, Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *Int. J. Adv. Manuf. Technol.* **89**, 389–406 (2017). <https://doi.org/10.1007/s00170-016-9050-1>
6. J.S. Huang, P.F. Hu, K.Y. Wu, M. Zeng, Optimal time-jerk trajectory planning for industrial robots. *Mech. Mach. Theory* **121**, 530–544 (2018). <https://doi.org/10.1016/j.mechmachtheory.2017.11.006>
7. P. Marcinko, J. Semjón, R. Jánoš, J. Svetlík, M. Sukop, Š. Ondočko, Analysis of the methodology for experimental measuring of the performance criteria of the laser-using collaborative robot's path accuracy. *Appl. Sci.* **14**, 1414 (2024). <https://doi.org/10.3390/app14041414>
8. A. Gasparetto, V. Zanotto, A new method for smooth trajectory planning of robot manipulators. *Mech. Mach. Theory* **42**, 455–471 (2007). <https://doi.org/10.1016/j.mechmachtheory.2006.04.002>
9. X. Li, H. Lv, D. Zeng, Q. Zhang, An improved multi-objective trajectory planning algorithm for kiwifruit harvesting manipulator. *IEEE Access* **11**, 65689–65699 (2023). <https://doi.org/10.1109/ACCESS.2023.3289207>
10. B. Nadir, O. Mohammed, N. Minh-Tuan, et al., Optimal trajectory generation method to find a smooth robot joint trajectory based on multiquadric radial basis functions. *Int. J. Adv. Manuf. Technol.* **120**, 297–312 (2022). <https://doi.org/10.1007/s00170-022-08696-1>
11. F. Wang, Z. Wu, T. Bao, Time-jerk optimal trajectory planning of industrial robots based on a hybrid WOA-GA algorithm. *Processes* **10**, 1014 (2022). <https://doi.org/10.3390/pr10051014>
12. W. Wang, Q. Tao, Y. Cao, X. Wang, X. Zhang, Robot time-optimal trajectory planning based on improved cuckoo search algorithm. *IEEE Access* **8**, 86923–86933 (2020). <https://doi.org/10.1109/ACCESS.2020.2992640>
13. A. Piazza, A. Visioli, Global minimum-jerk trajectory planning of robot manipulators. *IEEE Trans. Ind. Electron.* **47**(1), 140–149 (2000). <https://doi.org/10.1109/41.824136>
14. Z. Wang, Y. Shi, X. Wang, NURBS function closed-loop mapping trajectory planning of serial robotic plasma cladding for complex surface coatings. *Int. J. Adv. Manuf. Technol.* **121**, 8285–8298 (2022). <https://doi.org/10.1007/s00170-022-09709-9>
15. C. Choubey, O.J. Optimal, Trajectory generation for a 6-DOF parallel manipulator using grey wolf optimization algorithm. *Robotica* **39**(3), 411–427 (2021). <https://doi.org/10.1017/S0263574720000442>
16. A. Rout, G.B. Mahanta, D. Bbvl, B.B. Biswal, Kinematic and dynamic optimal trajectory planning of industrial robot using improved multi-objective ant lion optimizer. *J. Inst. Eng. (India), Ser. C* **101**, 559–569 (2020). <https://doi.org/10.1007/s40032-020-00557-8>
17. Z. Wu, J. Chen, D. Zhang, J. Wang, L. Zhang, F. Xu, A novel multi-point trajectory generator for robotic manipulators based on piecewise motion profile and series-parallel analytical strategy. *Mech. Mach. Theory* **181**, 105201 (2023). <https://doi.org/10.1016/j.mechmachtheory.2022.105201>
18. F.J. Abu-Dakka, I.F. Assad, R.M. Alkhodour, et al., Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *Int. J. Adv. Manuf. Technol.* **89**, 389–406 (2017). <https://doi.org/10.1007/s00170-016-9050-1>
19. D. Simon, C. Isik, A trigonometric trajectory generator for robotic arms. *Int. J. Control.* **57**, 505–517 (1993). <https://doi.org/10.1080/00207179308934404>
20. X. Cao, H. Yan, Z. Huang, S. Ai, Y. Xu, R. Fu, X. Zou, A multi-objective particle swarm optimization for trajectory planning of fruit picking manipulator, *Agronomy* **11**, 2286 (2021). <https://doi.org/10.3390/agronomy1112286>
21. S. Kucuk, Optimal trajectory generation algorithm for serial and parallel manipulators. *Robot. Comput.-Integr. Manuf.* **48**, 219–232 (2017). <https://doi.org/10.1016/j.rcim.2017.04.006>
22. P. Tangpattanakul, P. Artrit, Minimum-time trajectory of robot manipulator using harmony search algorithm, in *Proceedings of the 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Chonburi, Thailand, 06–09 May 2009. (2009). <https://doi.org/10.1109/ECTICON.2009.5137025>
23. Y. Yang, H. Xu, S. Li, L. Zhang, X. Yao, Time-optimal trajectory optimization of serial robotic manipulator with kinematic and dynamic limits based on improved particle swarm optimization. *Int. J. Adv. Manuf. Technol.* **120**, 1253–1264 (2022). <https://doi.org/10.1007/s00170-022-08796-y>
24. A. Gasparetto, V. Zanotto, A technique for time-jerk optimal planning of robot trajectories. *Robot. Comput.-Integr. Manuf.* **24**, 415–426 (2008). <https://doi.org/10.1016/j.rcim.2007.04.001>
25. J. Zhao, X. Zhu, T. Song, Serial manipulator time-jerk optimal trajectory planning based on hybrid IWOA-PSO algorithm. *IEEE Access* **10**, 6592–6604 (2022). <https://doi.org/10.1109/ACCESS.2022.3141448>
26. B. Das, V. Mukherjee, D. Das, Student psychology based optimization algorithm: a new population based optimization algorithm for solving optimization problems. *Adv. Eng. Softw.* **146**, 102804 (2020). <https://doi.org/10.1016/j.advengsoft.2020.102804>
27. K. Balu, V. Mukherjee, Optimal siting and sizing of distributed generation in radial distribution system using a novel student psychology-based optimization algorithm. *Neural Comput. Appl.* **33**, 15639–15667 (2021). <https://doi.org/10.1007/s00521-021-06185-2>
28. X. Wang, P. Henshaw, D. Ting, Applying student psychology-based optimization algorithm to optimize the performance of a thermoelectric generator. *Int. J. Green Energy* **21**(1), 1–12 (2024). <https://doi.org/10.1080/15435075.2023.2194395>
29. V. Shanmugam, T.V. Madhusudhana Rao, H.J. Rao, B. Maram, Internet of things based smart application for rice leaf disease classification using optimization integrated deep maxout network. *Concurr. Comput., Pract. Exp.* **35**, e7545 (2023). <https://doi.org/10.1002/cpe.7545>
30. S.A.T. Subramanian, M.S. Kumar, Hybrid optimization technique-based maximum power point tracking for single-stage grid-connected PV systems. *Clean Technol. Environ. Policy* **25**, 2999–3025 (2023). <https://doi.org/10.1007/s10098-023-02542-y>
31. J. Peta, S. Koppu, An IoT-based framework and ensemble optimized deep maxout network model for breast cancer classification. *Electronics* **11**, 4137 (2022). <https://doi.org/10.3390/electronics11244137>
32. F. Ming, W. Gong, L. Wang, L. Gao, A constraint-handling technique for decomposition-based constrained many-objective evolutionary algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* **53**, 7783–7793 (2023). <https://doi.org/10.1109/TSMC.2023.3299570>
33. H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems. *Inf. Sci.* **478**, 499–523 (2019). <https://doi.org/10.1016/j.ins.2018.11.041>

34. G. Yavuz, B. Durmuş, D. Aydin, Artificial bee colony algorithm with distant savants for constrained optimization. *Appl. Soft Comput.* **116**, 108343 (2022). <https://doi.org/10.1016/j.asoc.2021.108343>
35. R. Fitas, G. Carneiro, C.C. António, Swarm intelligence hybridized with genetic search in multi-objective design optimization under constrained-Pareto dominance. *Compos. Struct.* **319**, 117155 (2023). <https://doi.org/10.1016/j.compstruct.2023.117155>
36. F. Ming, W. Gong, L. Wang, L. Gao, A constrained many-objective optimization evolutionary algorithm with enhanced mating and environmental selections. *IEEE Trans. Cybern.* **53**, 4934–4946 (2023). <https://doi.org/10.1109/TCYB.2022.3151793>
37. F. Ming, W. Gong, L. Wang, C. Lu, A tri-population based co-evolutionary framework for constrained multi-objective optimization problems. *Swarm Evol. Comput.* **70**, 101055 (2022). <https://doi.org/10.1016/j.swevo.2022.101055>
38. J. Feng, S. Liu, S. Yang, J. Zheng, J. Liu, An adaptive tradeoff evolutionary algorithm with composite differential evolution for constrained multi-objective optimization. *Swarm Evol. Comput.* **83**, 101386 (2023). <https://doi.org/10.1016/j.swevo.2023.101386>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com