IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 2, 2021, accepted January 5, 2022, date of publication January 7, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141448

Serial Manipulator Time-Jerk Optimal Trajectory Planning Based on Hybrid IWOA-PSO Algorithm

JING ZHAO, XIJING ZHU, AND TIANJIAO SONG
School of Mechanical Engineering, North University of China, Taiyuan 030051, China

Corresponding author: Xijing Zhu (zxj161501@nuc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Program 51975540, and in part by the Innovation Project of Graduate Education of Shanxi Province under Grant 2019BY102.

ABSTRACT Loading and unloading operations are widely employed in industrial robots and CNC machine tools. In the present study, an effective algorithm is established for optimum time-jerk path planning and reducing the vibration of the serial manipulator, and enhancing the robot efficiency. To this end, the manipulator's trajectory is constructed using quintic B-spline interpolation in the joint space and the optimal objective function is constructed in terms of time and mean jerk. A hybrid improved whale optimization and particle swarm optimization (IWOA-PSO) method is proposed to optimize the objective function. First, WOA is improved by employing adaptive weight and threshold to balance exploration and exploitation of the WOA to prevent falling into the local optimum solution. Then, PSO method and IWOA techniques are combined to enhance the convergence speed of the IWOA. Applying different algorithms to 23 benchmark functions demonstrate superiority of the proposed algorithm to state-of-the-art algorithms. A 6-DOF serial industrial manipulator integrated in CNC machine was taken as a case study and the joint positions are considered the IWOA-PSO inputs. All calculations are performed in the MATLAB 2018b environment. The obtained results demonstrate that the proposed IWOA-PSO can effectively reduce the jerk of the robot while improving its work efficiency.

INDEX TERMS Optimum time-jerk path planning, IWOA-PSO algorithm, quintic B-spline, serial manipulator.

I. INTRODUCTION

Path planning is the fundamental field of study in robot kinematics, determining the manipulator's working efficiency, motion stability, and energy consumption [1]. The main objective in optimum path planning is to plan the optimal trajectory through these task points according to the given path points of the execution task while meeting the motion constraints [2], [3]. Currently, industrial manipulators are widely used as auxiliary equipment in CNC machine tools [4]. It is worth noting that all types of manipulators integrated into a CNC machine need a set of high-performance path planning algorithms to manage the feeding and unloading motions smoothly and efficiently. In this regard, various studies have been accomplished in the past few years on the optimal path planning of manipulators.

As the earliest and most popular research in this area, Lin *et al.* [5] proposed the minimum time path planning to find the optimal path of the robot with the shortest time as the performance index under given constraints and improved its effectiveness. Studies show that the joint jerk plays an essential role in robot movement. More specifically, too much jerk causes vibration, making it difficult for the robot to perform the task properly. Accordingly, minimum jerk is an important performance criterion to decrease vibration and improve trajectory tracking accuracy based on minimum-jerk trajectory planning [6]–[8]. Recently, the time-jerk optimal path planning issue has attracted many scholars to decrease vibration of manipulators and improve efficiency. The main objective of the present study is to obtain a smooth and time-jerk optimum path for a serial manipulator. Moreover, reducing the computational expense is another goal to improve the efficiency of path planning [9].

Time-jerk optimal path planning problem could be represented with a mathematical objective function [10], [11]. Then optimal techniques can be applied to solve the established mathematical objective function. Currently, the interpolated functions used in the joint space mainly

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

include polynomial interpolation and B-spline interpolation. Moreover, optimal techniques, including monarch butterfly optimization [12], moth swarm algorithm [13] cuckoo search algorithm [14], ameliorated harris hawk optimization [15] and so on, are applied to solve the equations. Huang *et al.* [16] proposed an optimization approach for the time-jerk synthetic optimal trajectory planning of robots. In this regard, both time and jerk were considered as objective functions subjected to the velocity, acceleration, and jerk limitations. Then the path was interpolated in the joint space using a five-order B-spline, and the elitist non-dominated sorting genetic algorithm (NSGA-II) was utilized to optimize the two objectives. Lu *et al.* [17] adopted a combination of constrained particle swarm optimization with the augmented Lagrange multiplier approach for time-jerk optimal path planning. Liu and Liu [18] proposed an enhanced adaptive genetic algorithm and attained an optimal model for electrolytic copper transport robots. Amruta *et al.* [19] established a teaching learning-based optimization method to derive a minimum time-jerk path and attain a smooth movement according to the kinematic constraints of the manipulator.

Recently, Mirjalili and Lewis [20] proposed the whale optimization algorithm based on heuristic algorithms. Guo *et al.* [21] combined an improved whale optimization algorithm with social learning and wavelet mutation strategies to estimate the water resources demand. Wang *et al.* [22] proposed an improved whale optimization algorithm (IWOA) and evaluated its effectiveness to realize the optimal path planning of the grinding robot. In this regard, Liu *et al.* [23] proposed a hybrid WOA combined with differential evolution and Levy flight (WOA-LFDE) to solve job shop scheduling problems.

In the present study, it is intended to propose an evolutionary algorithm, called the hybrid IWOA-PSO algorithm, to improve the efficiency and reduce the jerk of serial manipulators. The proposed algorithm is expected to inherit the advantages of the IWOA and PSO algorithms and provide remarkable benefits such as minimum setting parameters and higher searchability compared with the conventional algorithms.

This article is mainly arranged as follows: The minimum time-jerk optimal path planning issue is introduced in Section 2. Then details of the IWOA-PSO algorithm are discussed in Section 3. Section 4 is devoted to the simulation results and the corresponding discussions. Finally, the main achievements and conclusions are summarized in Section 5. Finally, section 6 puts forward the future research.

II. TRAJECTORY PLANNING PROBLEM STATEMENT

A. THE TIME-JERK OPTIMAL OBJECTIVE FUNCTION

During the feeding process of the manipulator, the path is expected to be smooth enough to prevent excessive mechanical vibration. Although the manipulator's stability can be improved by alleviating the jerk, this scheme significantly increases the whole traveling time, thereby affecting the robot's efficiency. In order to resolve this problem, the weight coefficients K_T and K_J are introduced to minimize the whole traveling time and the jerk function. In this regard, the end feeding trajectory should be initially described in the Cartesian coordinate system. The inverse kinematics of the 6-DOF serial manipulator is then solved to obtain the angle of each joint. In the present study, speed, acceleration, and jerk constraints of the joint are considered. Accordingly, the following time-jerk optimal objective function can be defined to express the problem mathematically.

$$\begin{aligned} \min f &= K_T N \sum_{i=1}^{n-1} t_i + \partial K_J \sum_{j=1}^{N} \sqrt{\int_0^T ((\dddot{q}_j(t))^2 dt / T} \\ &\quad + a_1 \sum V_c + a_2 \sum A_c + a_3 \sum J_c \\ &\textit{subject to} \begin{cases} |\dot{q}_j(t)| \le V_{jm}, & j = 1, 2, \cdots, 6 \\ |\ddot{q}_j(t)| \le A_{jm}, & j = 1, 2, \cdots, 6 \\ |\dddot{q}_j(t)| \le J_{jm}, & j = 1, 2, \cdots, 6 \end{cases} \end{aligned} \tag{1}$$

where, $K_T + K_J = 1$. Since the running time and the square of jerk of the manipulator may have different order magnitudes, the elastic coefficient is employed to balance the order of magnitude of the motion time and jerk. Table 1 illustrates the definition of the symbols mentioned above. The first term in equation (1) stands for the running time of the robot, while the second term is an expression about the jerk. Meanwhile, the last three terms denote penalty functions, giving a limit area for optimization. In an ideal event, penalty elements approach zero. Eq. (1) should be solved to obtain the optimal time intervals subjected to joint speed, acceleration, and jerk constraints. Then, the time-jerk optimal path subjected to the above mentioned constraints can be attained through the quintic B-spline.

B. CONSTRUCTION OF QUINTIC B-SPLINE INTERPOLATION FUNCTION

B-spline function was evolved from the Bezier curve. Accordingly, it has the characteristics of local controllability in addition to the inherited characteristics from the Bezier curve, including geometric invariance and affine invariance. When a control point changes, it can only affect part of the B-spline curve, while the rest of the trajectory curve shape remains intact. This property can be well applied to robot trajectory planning [24]. In the present study, quintic B-spline was applied to construct the path and realize the jerk continuity to guarantee that its beginning and ending acceleration are set zero. A B-spline curve can be expressed as follows:

$$p(u) = \sum_{i=0}^{n} V_i N_{i,k}(u) \tag{2}$$

where V_i is the control point of the B-spline, $N_{i,k}(u)$ denotes the base function of the $k+1$th order B-spline, which can be

TABLE 1. Meaning of symbols in the objective function.

symbol	Meaning
N	Number of robot joints
n	Number of via-points on the trajectory path
K_T	Time weighting parameter
t_i	Time interval between two via-points
K_J	Jerk weighting coefficient
T	Total traveling time
$\dot{q}_j(t)$	Velocity of the jth joint
$\ddot{q}_j(t)$	Acceleration of the jth joint
$\dddot{q}_j(t)$	Jerk of the jth joint
V_{jm}	Upper limit for the velocity of the jth joint
A_{jm}	Upper limit for the acceleration of the jth joint
J_{jm}	Upper limit for the jerk of the jth joint
V_c	The exceeded value of the limit velocity
A_c	The exceeded value of the limit acceleration
J_c	The exceeded value of the limit jerk

obtained via the Cox-de Boor expression [25].

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & if\ u_i \le u \le u_{i+1} \\ 0, & \text{others} \end{cases} \\ N_{i,k}(u) = \dfrac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) \\ + \dfrac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ and \dfrac{0}{0} = 0 \end{cases} \tag{3}$$

Since the normalized node vector u varies with the range (0, 1), all points of the line segment can be calculated by substituting u into equation (2). Yet the control vertices V_i are unknown, the control vertices in quintic B-spline can be computed through the following expression:

$$\begin{bmatrix} 1 & 2 & -6 & 2 & 1 & & & & & \\ 1 & 10 & 0 & -10 & 1 & & & & & \\ 1 & 26 & 66 & 26 & 1 & & & & & \\ & 1 & 26 & 66 & 26 & 1 & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & 1 & 26 & 66 & 26 & 1 & \\ & & & & -1 & 10 & 0 & 10 & 1 & \\ & & & & 1 & 2 & -6 & 2 & 1 & \end{bmatrix} \begin{bmatrix} V_{i-2} \\ V_{i-1} \\ V_i \\ V_{i+1} \\ \cdots \\ V_{i+n} \\ V_{i+n+1} \\ V_{i+n+2} \end{bmatrix}$$
$$= 120 \begin{bmatrix} 0 \\ 0 \\ P_0 \\ P_1 \\ \cdots \\ P_n \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

Eq. (4) provides n + 1 equations. Obviously, it is insufficient to solve n+5 control points with n+1 equations. In order to solve all control vertex vectors, the first two and last two laws of the matrix equation are obtained through velocity and acceleration conditions of the starting and the ending via-points. Combining the quantic B-spline curve base function, the control vertices generated by Eq. (4) can be replaced with Eq. (5) to obtain the trajectory points.

$$p(u) = \frac{1}{120!}[1 \quad u \quad u^2 \quad u^3 \quad u^4 \quad u^5] \begin{bmatrix} 1 & 26 & 66 & 26 & 1 & 0 \\ -5 & 50 & 0 & 50 & 5 & 0 \\ 10 & 20 & -60 & 20 & 10 & 0 \\ -10 & 20 & 0 & -20 & 10 & 0 \\ 5 & -20 & 30 & -20 & 5 & 0 \\ -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix} \begin{bmatrix} V_{i-2} \\ V_{i-1} \\ V_i \\ V_{i+1} \\ V_{i+2} \\ V_{i+3} \end{bmatrix} \tag{5}$$

III. HYBRID IMPROVED WHALE OPTIMIZATION AND PARTICLE SWARM OPTIMIZATION (IWOA-PSO)

A. PARTICLE SWARM OPTIMIZATION

The PSO is a meta-heuristic optimization technique based on population. In the PSO algorithm, each particle dynamically adjusts the search trajectory according to its optimal solution and the optimal solution of the population to approach the global optimal solution. Consequently, the whole population can achieve good positive feedbacks during the iterative solution, thereby improving the convergence rate and accuracy of the algorithm. The standard PSO algorithm can be described in the form below [26]:

$$v_{ij}^{t+1} = w v_{ij}^t + c_1 r_1 \left(p_{ij}^t - x_{ij}^{t+1} \right) + c_2 r_2 \left(p_{gj}^t - x_{ij}^t \right) \tag{6}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \tag{7}$$

where $x_i = (x_{i1}, x_{i1}, \ldots, x_{ij})$ and $\boldsymbol{v}_i = (v_{i1}, v_{i1}, \ldots, v_{ij})$ denote the position and velocity vector of the ith particle, respectively. Moreover, p_{ij} is the best position of particle i, and p_{gj} is the optimal location of the population. c_1 and c_2 are learning factors called acceleration constants. r_1 and r_2 are random numbers in the range [0, 1], and w is the inertial weight that can be described as the following:

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})t}{t_{\max}} \tag{8}$$

where $w_{\max}$ denotes the maximum value of w, which is set to 0.9. Furthermore, $w_{\min}$ denotes the minimum value of w, which is normally set to 0.4. t and $t_{\max}$ stand for the current number and the maximum number of iterations, respectively.

B. WHALE OPTIMIZATION ALGORITHM

As a new intelligent optimization algorithm, the whale optimization algorithm (WOA) was proposed in 2016 [20]. Studies show that WOA has superior characteristics such as simple principle, few parameters, and easy implementation [27], [28]. Accordingly, WOA has been successfully applied in diverse engineering optimization areas. It is worth noting that the WOA mainly consists of three parts, including encircling the prey, the bubble-net foraging, and searching for prey. These parts are described as follows:

TABLE 2. Test functions.

Function	Dim	Range	Min
$f_1(x)=\sum_{i=1}^{d}x_i^2$	30	[-100,100]	0
$f_2(x)=\sum_{i=1}^{d}\lvert x_i\rvert+\prod_{i=1}^{d}\lvert x_i\rvert$	30	[-10,10]	0
$f_3(x)=\sum_{i-1}^{d}(\sum_{j-1}^{i}x_j)^2$	30	[-100,100]	0
$f_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le d\}$	30	[-100,100]	0
$f_5(x)=\sum_{i=1}^{d}[100(x_{i+1}-x_i^2)^2-(x_i-1)^2$	30	[-10,10]	0
$f_6(x)=\sum_{i=1}^{d}([x_i+0.5])^2$	30	[-30,30]	0
$f_7(x)=\sum_{i=1}^{d}ix_i^4+random[0,1)$	30	[-1.28,1.28]	0
$f_8(x)=\sum_{i=1}^{d}-x_i\sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.93E+5
$f_9(x)=\sum_{i=1}^{d}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12,5.12]	0
$f_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)+20+e$	30	[-32,32]	0
$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]	0
$f_{12}(x)=\frac{\pi}{d}\left\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})+(y_d-1)^2]\right\}+\sum_{i-1}^{d}u(x_i10,100,4)$ $y_i=1+\frac{x_{i+1}}{4},\quad u(x_ia,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	30	[-50,50]	0
$f_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{d}(x_i-1)^2[1+\sin^2(3\pi x_1+1)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{d}u(x_i,5,100,4)$	30	[0,π]	0
$f_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-b_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$f_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
$f_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
$f_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\cos x_1+10\right)$	2	[-5,5]	0.398
$f_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3
$f_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$	3	[1,3]	-3.86
$f_{20}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right)$	6	[0,1]	-3.32
$f_{21}(x)=-\sum_{i=1}^{4}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
$f_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
$f_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.5363

1) ENCIRCLING THE PREY

When $p < 0.5$ and $\left|\overrightarrow{A}\right| < 1$, humpback whales can detect the position of prey and surround them. However, its location cannot be known in advance. In the WOA, it is assumed that the optimum position of the current population is its corresponding global optimal solution. Then, the other whales update their location using the current global optimal location. This phenomenon can be mathematically modeled in the form below:

$$\overrightarrow{D}=\left|\overrightarrow{C}\cdot\overrightarrow{X^*}-\overrightarrow{X}(t)\right| \tag{9}$$

$$\vec{X}(t+1)=\overrightarrow{X^*}(t)-\overrightarrow{A}\cdot\overrightarrow{D} \tag{10}$$

where, p stands for random number in the range [0, 1], $\overrightarrow{X}(t)$ is the position vector, and $\overrightarrow{X^*}$ denotes the position vector of the best solution. Moreover, vectors $\vec{A}$ and $\vec{C}$ can be calculated from the following expressions:

$$\vec{A}=2*\vec{a}*\vec{r}-\vec{a} \tag{11}$$

$$\vec{C}=2*\vec{r} \tag{12}$$

where, $\vec{r}$ is a random vector in the range [0, 1], while $\vec{a}$ is a variable that linearly reduces from 2 to 0 through the following expression:

$$\vec{a}=2-\frac{2t}{MaxIter} \tag{13}$$

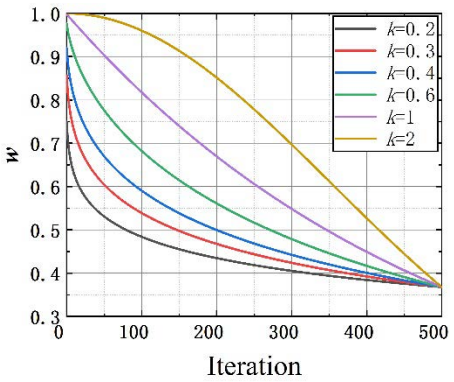

FIGURE 1. Variations of the inertia weight for different adjustment coefficient.

where t and $MaxIter$ denote the current number and the maximum number of iterations, respectively.

2) EXPLOITATION PHASE

When $p \geq 0.5$, the humpback preys its prey with a spiral bubble net. The spiral predation of humpback whales can be mathematically modelled as follows:

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) \tag{14}$$

where i is the distance of the ith whale to the prey, b is a parameter that describes the logarithmic spiral shape, and l denotes a random number in the range [0, 1]. It is assumed that there exists a 50% probability to select among the shrinking encircling mechanism and the spiral model to adjust the whale's position. This assumption can be mathematically modeled as follows:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D}, & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t), & if\ p \geq 0.5 \end{cases} \tag{15}$$

3) SEARCHING FOR PREY

When $\left|\overrightarrow{A}\right| \geq 1$, humpback whales update the whales' positions using a random searching concept. This process was described in Eqs. (16) and (17).

$$\overrightarrow{D} = \left|\overrightarrow{C} \cdot \overrightarrow{X_{rand}} - \overrightarrow{X}\right| \tag{16}$$

$$\overrightarrow{X(t+1)} = \overrightarrow{X_{rand}} - \overrightarrow{A} \cdot \overrightarrow{D} \tag{17}$$

where $\overrightarrow{X_{rand}}$ describes the arbitrary location of a random whale.

C. IMPROVED WHALE OPTIMIZATION ALGORITHM

Reviewing the literature indicates that the whale optimization algorithm has been applied and studied in many areas. Despite remarkable achievements, this algorithm has some short-comings such as defects in premature convergence and falling into the local optimization. In order to improve the search capabilities of the WOA, it is intended to propose an improved whale optimization algorithm (IWOA) in this section.

TABLE 3. The parameter settings of the comparative algorithms.

Algorithm	Parameter settings
WOA [20]	N=30, a=2-t×(2/ t_{max})
IWOA	N=30, t_{max}:500, D:30; k=0.4
IWOA-PSO	N=30, t_{max}:500, D:30; k=0.4, c_1=c_2=2.0; P_c=0.3
GWO [29]	N=30, a=2-t×(2/ t_{max})
PSO [26]	N=30, w_{max}=0.9, w_{min}=0.4, c_1=c_2=2.0
MPA[30]	N=30, P=0.5,P_f=0.2
HHO[31]	N=30, t_{max}:500,

It is worth noting that the probability threshold of the standard whale optimization algorithm is 0.5. Therefore, a logarithmic form of adaptive probability threshold is proposed to balance the ability of the global search and local development with the following expression:

$$p' = 1 - \log_{10}(1 + \frac{9t}{\max_i ter}) \tag{18}$$

From Eq. (18), it is clear that the range of p' is linearly reduces from 1 to 0. The adaptive probability threshold is constantly updated to keep the whale population close to the optimal solution, so as to improve the convergence accuracy of WOA.

When the standard WOA is locally developed in the later stage, there is no corresponding weight to adjust the update rule. In order to resolve this problem, another improvement of WOA algorithm is the introduction of inertia weight in particle swarm optimization algorithm. The expression for the w is

$$w(t) = e^{-(\frac{t}{\max_i ter})^k} \tag{19}$$

where, t and max_iter denote the number of the current and maximum iterations, respectively. k is the adjustment coefficient. Figure 1 presents variations of the weight change for different adjustment coefficients.

According to PSO, when $0.4 \leq w \leq 0.9$, the performance of the algorithm improves significantly. In this regard, Fig.1 indicates that when $k = 0.4$, the weight range is in the range [0.4, 0.9]. Accordingly, the parameter k is set to 0.4 in all calculations to balance the global and local search ability of the WOA. When the weight coefficient of the algorithm at the initial stage is relatively large, it has strong global searchability. As the number of iterations increases, the weight coefficient decreases.

Then the whale position is updated through the following expression:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - w\vec{A} \cdot \vec{D}, & if\ p < p' \\ w\overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t), & if\ p \geq p' \end{cases} \tag{20}$$

D. THE PROCEDURES OF THE HYBRID IWOA-PSO ALGORITHM

The main core of IWOA-PSO algorithm is to combine the improved WOA and the PSO algorithm. The unity of both

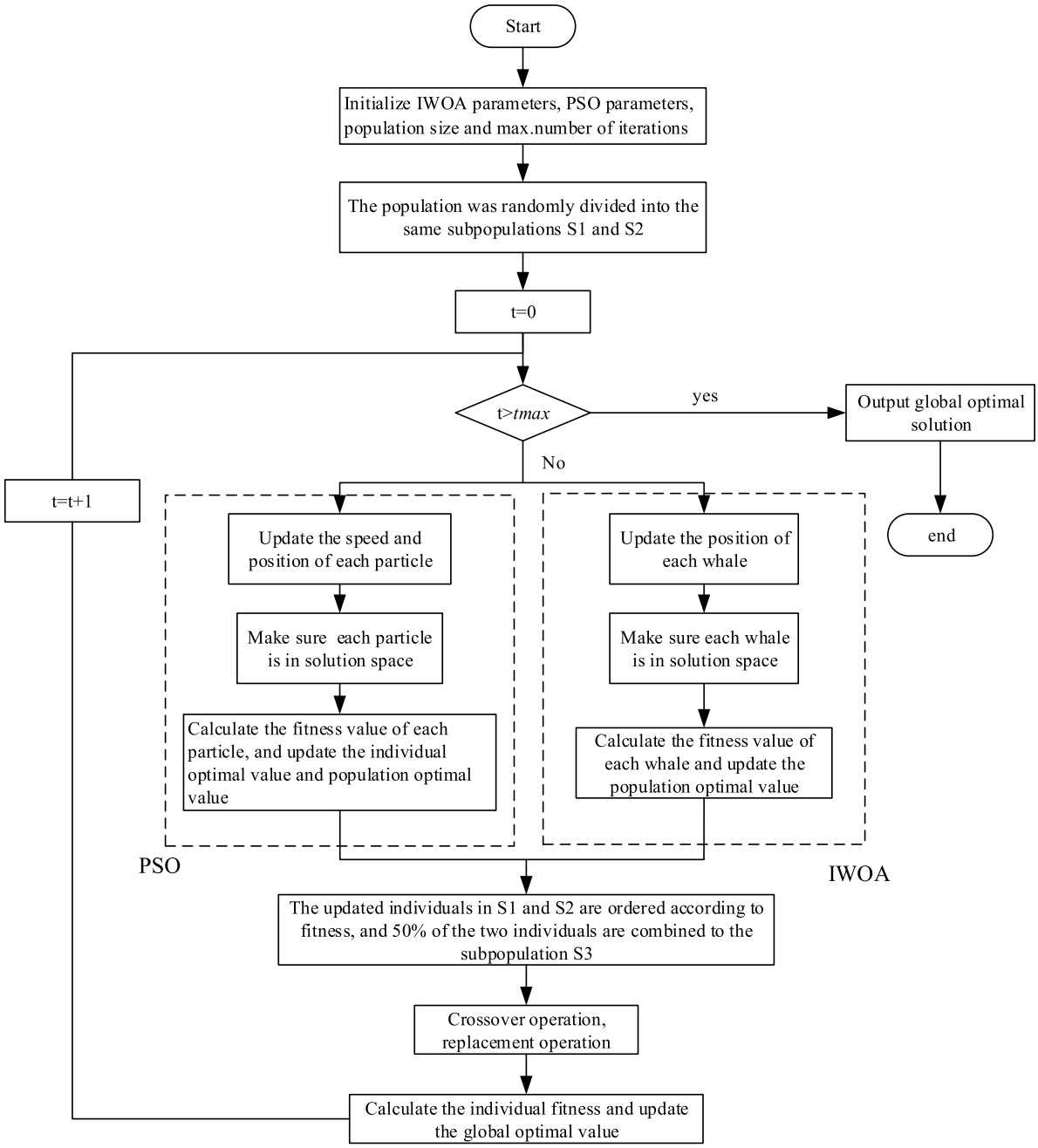

FIGURE 2. Flowchart of hybrid IWOA-PSO algorithm.

IWOA and PSO ensures the optimal solution of the optimization problem can be found faster and more efficiently. The procedures of the hybrid IWOA-PSO algorithm as follows:

Initialize all parameters of IWOA and PSO.

The population was randomly divided into subpopulations S1 and S2 of the same size. Individuals in S1 and S2 subpopulations perform IWOA and PSO algorithms, respectively.

The updated individuals in S1 and S2 subpopulations are ordered according to fitness, and 50% of the two individuals are combined to the subpopulation S3.

The crossover strategy is used to exchange information among individuals in S3, then the individuals in S3 are recombined and randomly returned to the subpopulations S1 and S2 according to the previous selection proportion to replace the position of the corresponding individuals in the original population and keep the size of S1 and S2 unchanged.

The individual fitness value is calculated and the global optimal solution is updated. The iterative process continues until the termination conditions are met.

Finally, the global optimal solution is obtained and the calculation ends. Otherwise, return to iteration stage and repeat the optimization process.

The crossover operation is as follows:

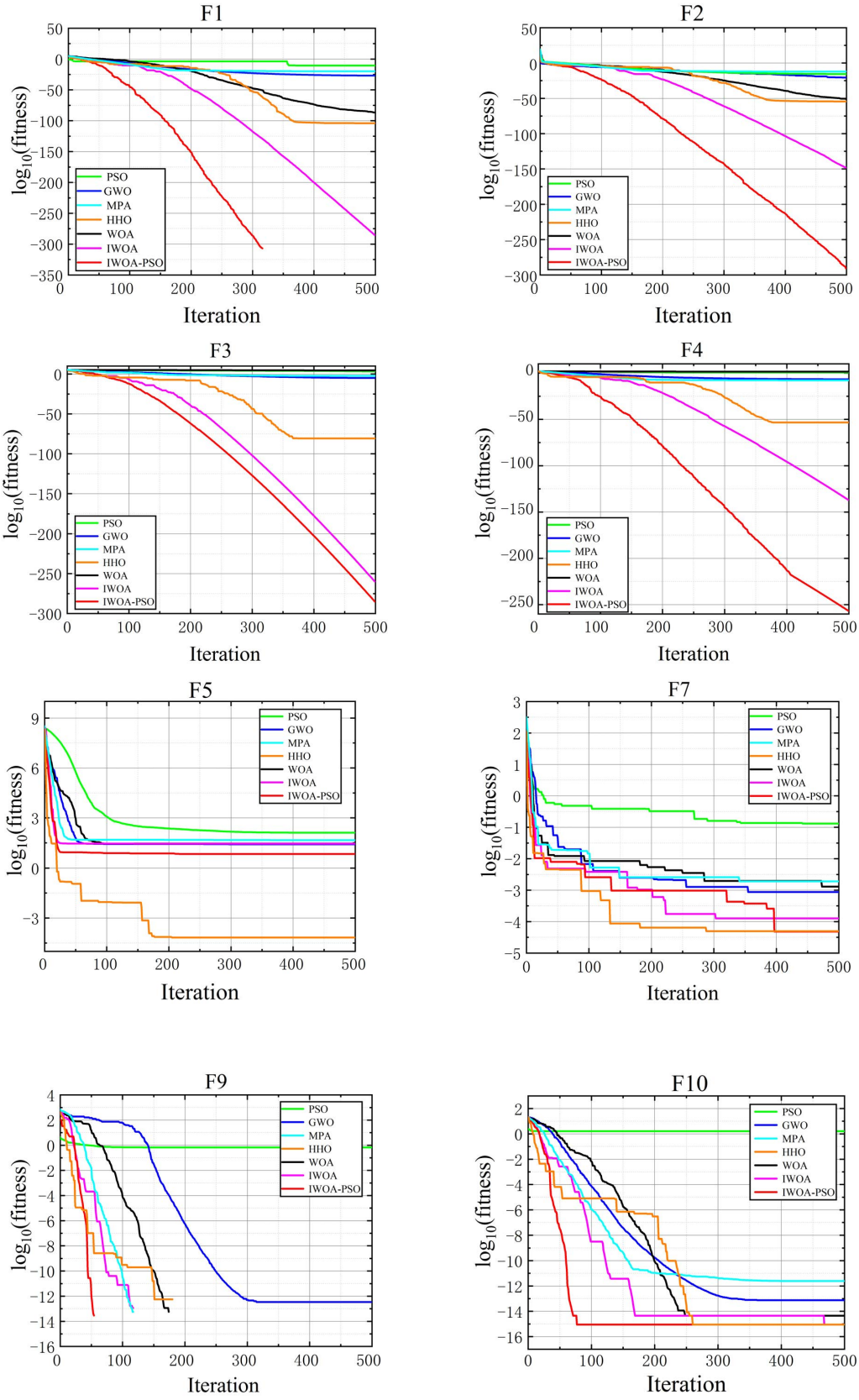

FIGURE 3. The convergence plots of some typical test functions.

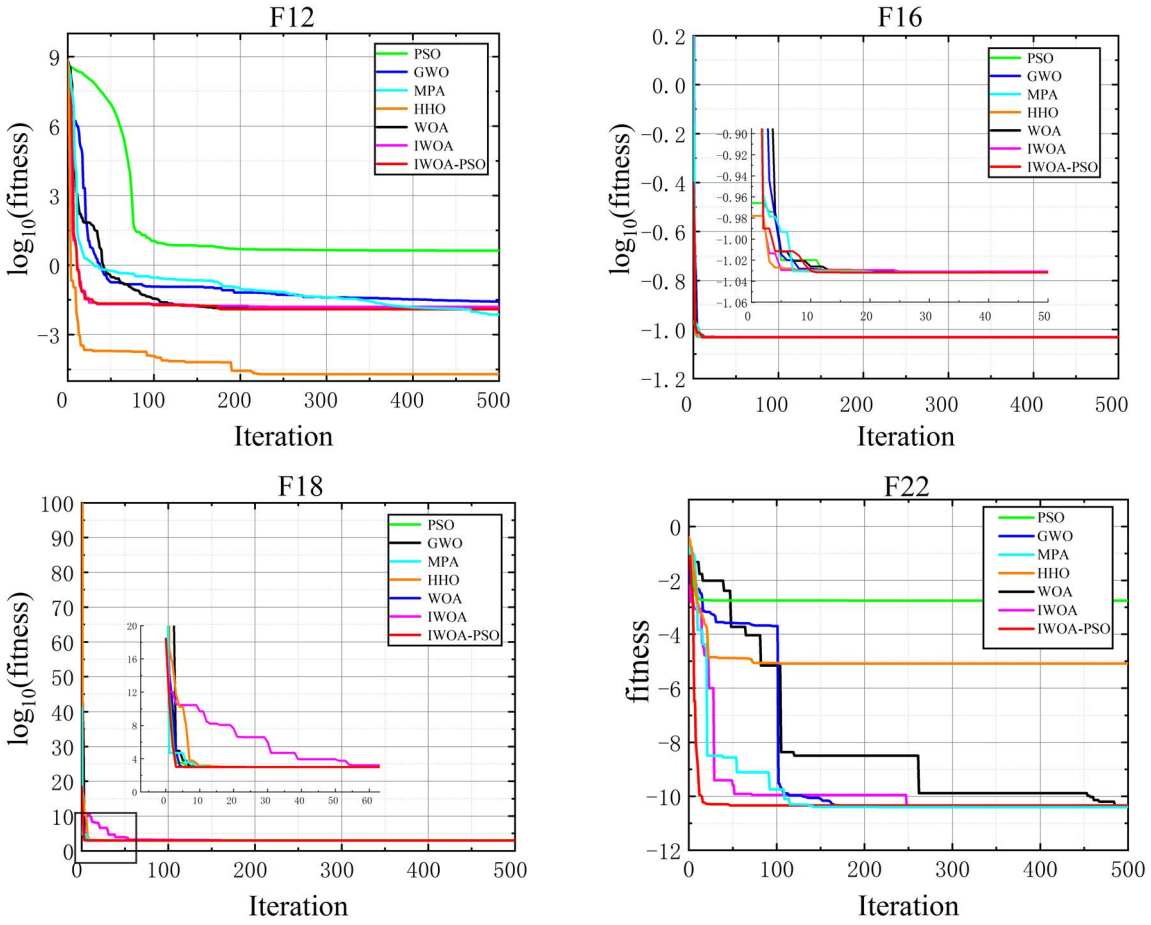

FIGURE 3. *(Continued.)* **The convergence plots of some typical test functions.**

The crossover strategy is similar to the crossover operation of GA (Genetic Algorithm).In order to let search agents perform a crossover operation, half of the individuals in subpopulations S1 and S2 with high fitness values are combined to a new subpopulation S3. Then, take the individuals in S3 as the parent and perform the crossover operation. Suppose the crossover probability P_c. Then generate a random number in the range [0, 1]. If the generated random number is greater than P_c, two parents directly enter the next generation. Otherwise, some attributes of the two-parent individuals are crossly recombined, that is, the genes of a pair of chromosomes cross each other.

The corresponding relationship between the new attribute and the original parent attribute can be expressed through the following expression:

$$\begin{cases} x_i(k+1) = ry_i(k) + (1-r)x_i(k) \\ y_i(k+1) = rx_i(k) + (1-r)y_i(k) \end{cases} \tag{21}$$

where r is the generated random number, and $x_i(k)$ and $y_i(k)$ are the ith attribute of two individuals in the kth generation. Figure 2 shows the flowchart of the hybrid IWOA-PSO algorithm.

IV. EXPERIMENT AND RESULTS ANALYSIS

A. EXPERIMENT 1: TEST FUNCTION

In the first experiment, 23 different benchmark functions are selected to evaluate the performance of the proposed algorithm. Table 2 gives the data of those benchmark functions. These test functions include unimodal ($f1$-$f7$), multimodal ($f8$-$f13$), and fixed dimensional multimodal functions ($f14$-$f23$). It should be indicated that each unimodal function has only one individual optimal solution, which is utilized to evaluate the exploitation and convergence rate of the algorithm. Furthermore, the multimodal function includes multiple optimum solutions, where only one of them is the global optimal solution, while the remaining ones are local optimal solutions employed to evaluate the global searchability of the algorithm.

1) PARAMETER SETTINGS

In this section, a wide variety of experiments are performed to verify the efficiency of the proposed IWOA-PSO. Moreover, IWOA, WOA [20], PSO [26], GWO [29], MPA [30] and HHO [31] algorithms are also used for comparative analysis. During the experiment, the size of the test population and

TABLE 4. Obtained results from different algorithms on unimodal functions.

Funtion		IWOA-PSO	IWOA	WOA	PSO	GWO	MPA	HHO
F1	Mean	**0**	4.01E-282	1.41E-30	1.36 E-4	6.59E-28	6.68E-22	-2.59E-44
	Std	0	0	4.91E-30	2.02 E-4	6.34E-27	3.85E-21	1.21E-43
F2	Mean	**0**	2.12E-145	1.06E-21	4.21 E-2	7.18E-17	1.79E-21	4.15E-53
	Std	0	3.12E-146	2.39E-21	4.54 E-2	2.90 E-17	1.96E-21	1.55E-49
F3	Mean	**0**	1.04E--253	5.39E-7	7.01 E+1	3.29E-6	-2.02E-14	-2.32E-16
	Std	0	0	2.93E-6	2.21 E+1	7.91 E+1	9.67E-02	2.29E-17
F4	Mean	**0**	2.23E-132	7.26 E-2	1.08 E-0	5.61E-7	1.99E-01	3.22E-40
	Std	0	0	3.97 E-1	0.32 E-0	1.32 E-0	4.16E-09	2.74E-40
F5	Mean	7.05E+0	6.39E+0	2.79 E+1	9.67 E+1	2.68 E+1	3.68E+01	**9.94E-01**
	Std	2.70E-1	2.64 E-1	7.64 E-1	6.01 E+1	6.99 E+1	3.14E-01	8.20E-03
F6	Mean	5.10E-1	1.54 E-1	3.11 E-0	**1.02 E-4**	8.16 E-1	-5.00E-01	-4.99E-01
	Std	2.18 E-1	4.76 E-2	5.32 E-1	8.28E-5	1.26 E-4	1.24E-01	2.76E-02
F7	Mean	**6.1E-5**	1.02E-4	1.43E-3	1.23 E-1	2.13 E-1	-1.32E-02	-8.31E-02
	Std	4.4E-5	1.05E-4	1.15 E-3	4.49 E-2	1.00E-1	1.01E-01	6.48E-02

TABLE 5. Obtained results from different algorithms on multimodal functions.

Funtion		IWOA-PSO	IWOA	WOA	PSO	GWO	MPA	HHO
F8	Mean	**-1.26E+4**	-1.24E+4	-5.08E+03	-4.84E+-3	-5.97E+03	1.57E+03	1.44E+02
	Std	123	2.12E+02	6.95E+02	1.15E+03	7.10E+02	8.11E+02	3.06E+02
F9	Mean	**0**	**0**	**0**	4.67E+01	3.10E-01	**0**	**0**
	Std	0	0	0	1.16E+01	4.73E+01	0	0
F10	Mean	**8.88E-16**	**8.88E-16**	7.40E+00	2.76E-01	1.06E-13	1.18E-12	**8.88E-16**
	Std	0	0	9.89E+00	5.09E-01	7.78E-02	2.64E-12	4.01E-33
F11	Mean	**0**	0	2.89E-03	9.21E-03	4.48E-03	0	0
	Std	0	0	1.58E-03	7.72E-03	6.66E-03	0	0
F12	Mean	**8.00E-5**	7.09E-3	3.39E-01	6.91E-03	5.34E-02	-1.00E-02	1.90E+01
	Std	3.00E-5	3.92E-04	2.14E-01	2.63E-02	2.07E-02	3.48E-02	3.31E+00
F13	Mean	9.31E-2	1.09E-01	1.88E+00	**6.67E-03**	6.54E-02	1.03E+00	1.89E+01
	Std	4.07E-2	4.00E-3	2.66E-01	8.91E-03	4.47E-03	4.50E-02	1.56E+00

the maximum number of iterations are set to 30 and 500, respectively. Moreover, each algorithm is performed 20 times in the benchmark function. Settings of various algorithms are presented in Table 3.

2) SIMULATION RESULTS AND DISCUSSION

All simulations are performed in the MATLAB R2018b environment. Configurations of the PC are Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz with 4 GB RAM. In the experiment, mean value (Mean) and standard deviation (Std) are used as evaluation indices. The obtained results are shown in Tables 4-6. The BOLD data reflect the optimal results obtained by comparing the five algorithms after optimizing the same function. The smaller the standard deviation, the more stable and robust the proposed algorithm.

Table 4 indicates that the IWOA-PSO algorithm can precisely determine the global optimal solution. More especially, functions $f1$-$f4$ can reach theoretical optimal value 0, while other algorithms cannot do so, demonstrating its powerful searchability. Furthermore, IWOA-PSO obtains a good standard deviation except for $f5$.This may be attributed to the complexity of function $f5$, which makes it difficult for many algorithms to find the theoretical optimal solution. Table 5 reveals that both IWOA-PSO and IWOA can obtain the global optimal or near-optimal solution for functions $f9$-$f11$. However, in functions $f8$, $f12$ and $f13$, IWOA-PSO has a smaller standard deviation and better stability than other algorithms. The obtained results demonstrate that the proposed IWOA-PSO algorithm has certain advantages in global exploration and can effectively prevent falling into a locally optimal solution. Meanwhile, Table 6 indicates that compared with other 6 algorithms, the optimal value of IWOA-PSO is more compatible with the theoretical result. This is especially more pronounced for $f21$-$f23$ functions. Accordingly, it is concluded that IWOA-PSO is an effective scheme to reach the global optimal solution with less standard deviation.

TABLE 6. Obtained results from different algorithms on fixed dimensional multimodal functions.

Funtion		IWOA-PSO	IWOA	WOA	PSO	GWO	MPA	HHO
F14	Mean	**1.295817**	1.295817	2.111973	3.627168	4.04249	0.998000	0.998000
	Std	0.669809	0.669811	2.498594	2.560828	4.25279	7.92E-16	0.923
F15	Mean	**0.000300**	0.000518	0.000572	0.000577	0.000337	0.000307	0.000310
	Std	0.000171	0.000130	0.000324	0.000222	0.000625	3.64E-02	0.000197
F16	Mean	**-1.031627**	-1.03163	-1.03163	-1.03163	-1.03163	-1.0316	-1.03E+00
	Std	4.00E-6	0.009267	4.2E-7	6.25E-16	1.03163	5.67E-16	6.78E-16
F17	Mean	**3.98E-01**	0.398152	0.397914	0.397887	0.397889	0.397900	**3.98E-01**
	Std	1.00E-6	0.000960	2.70E-5	0.00E+00	0.397889	9.12E-15	2.54E-06
F18	Mean	**3.000000**	3.000000	3.000000	3.000000	3.000028	3.000000	3.000000
	Std	0.000002	0.024238	4.22E-15	1.33E-15	3.000000	1.95E-15	0.00E+00
F19	Mean	**-3.86E+00**	-3.862122	-3.85616	-3.86278	-3.86234	-3.8628	-3.86E+00
	Std	0.012632	0.022366	0.002706	2.58E-15	-3.25056	2.42E-15	2.44E-03
F20	Mean	**-3.320415**	-3.309784	-2.98105	-3.26634	-3.28654	-3.3220	-3.322
	Std	0.007474	0.007541	0.376653	0.060516	-3.25056	1.14E-11	0.137406
F21	Mean	**-10.15287**	-10.15162	-7.04918	-6.8651	-10.1514	-10.1532	-10.1451
	Std	1.592651	2.577641	3.629551	3.019644	-9.1401	2.53E-11	0.885673
F22	Mean	**-10.40218**	-10.40178	-8.18178	-8.45653	-10.4015	-10.4029	-10.4015
	Std	1.638015	2.661574	3.829202	3.087094	-8.5584	2.81E-11	1.352375
F23	Mean	**-10.53646**	-10.53527	-9.34238	-9.95291	-10.5343	10.5364	-10.5364
	Std	2.360446	2.538438	2.414737	1.782786	-8.5589	3.89E-11	0.927655

Figure 3 shows the convergence of various representative test functions. It is observed that WOA has low calculation accuracy and low convergence speed, and simply falls into the local optimization. Although IWOA outperforms WOA from different aspects, there are still some problems such as slow convergence speed. Therefore, IWOA is combined with the PSO algorithm to improve its performance. It should be indicated that the IWOA-PSO inherits the advantages of IWOA and PSO so that it has high accuracy, fast convergence speed, and good global search ability than other algorithms, confirming the efficiency and feasibility of the proposed algorithm.

B. EXPERIMENT 2: TIME-JERK OPTIMUM TRAJECTORY PLANNING VIA THE HYBRID IWOA-PSO ALGORITHM

A complete feeding process of the manipulator is discussed in this section. The D-H parameters of the manipulator are shown in Table 7. When the manipulator completes its motion, the pose matrix of the end effector can be expressed by Eq. (20). Figure 4 shows the trajectory curve of the manipulator end in Cartesian coordinate system.

Inserting key purposes between the starting point and the ending point can modify variations of the motion path. In this feeding process, four interpolation points divide the joint trajectory into five parts. Then interpolation points in Cartesian space can be translated into joint space angle interpolation points through an inverse solution. The obtained results in this regard are shown in Table 8.

$$T = \begin{bmatrix} -0.7468 & -0.5107 & -0.4261 & 0.3275 \\ -0.5983 & 0.2306 & 0.7657 & 01825 \\ -0.2905 & 0.8268 & -0.4818 & 0.0139 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad (22)$$

The key via points are converted into 10 control vertices through Eq. (4). Then the control vertices are allotted to Eq. (5) to get the smooth level through the quintic B-spline trajectory. The B-spline curve of the six joints describing the angle, velocity, acceleration, and jerk are presented in Figure 5-8, respectively. It should be indicated that the entire feeding time of unoptimized simulations is set to 30s.

Figure 6, 7 and 8 show that the velocity and acceleration of the initial and final points are zero and the joint jerk curves are continuous. The maximum speed of the joints are 8.05°/s, 14.21°/s, 9.60°/s, 41.07°/s, 13.26°/s, 21.34°/s, respectively, indicating that the maximum allowable speed has not yet been reached. Ignoring the kinematic constraints in the feeding process of the manipulator in the quintic B-spline indicates that the jerk of the six joints is large. This is especially more pronounced for the jerks of the last three joints, which have a great impact on the running quality of the serial manipulator. Accordingly, it is concluded that the manipulator in this situation is inefficient.

Then the hybrid IWOA-PSO algorithm was employed to solve the optimum time-jerk problem of the quintic B-spline interpolation method with the kinematic constraint. All simulations are performed in the MATLAB 2016b environment.

TABLE 7. D-H parameters of the manipulator.

Joint.no	a_i/mm	α_i/(°)	d_i/mm	θ_i
1	40	90	335	θ_1
2	280	0	0	θ_2
3	70	90	0	θ_3
4	0	-90	313	θ_4
5	0	90	0	θ_5
6	0	0	0	θ_6

TABLE 8. Via points of the six joints.

via-points	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
P0	120.34	41.07	31.09	31.59	-74.67	-9.23
P1	109.33	-13.10	26.96	57.58	-24.00	-55.12
P2	65.74	-8.27	39.03	-41.48	-38.38	34.56
P3	42.78	16.91	34.61	-53.98	-64.94	29.89
P4	30.35	29.49	5.08	-72.08	-66.00	51.75
P5	29.13	38.64	-30.00	85.31	-61.55	80.48

TABLE 9. Upper kinematic limits of the robot joints.

Joint no.	Velocity(°/s)	Acceleration （°/s^2）	Jerk(°/s^3)
1	100	45	60
2	95	40	60
3	100	50	55
4	150	70	70
5	130	50	75
6	140	80	60

TABLE 10. The comparison of time series before and after optimization.

Time interval	before optimization(s)	after optimization(s)
t_1	6	5.13
t_2	6	3.93
t_3	6	2.53
t_4	6	2.49
t_5	6	5.7
total	30	19.85

During all calculations, the population and the number of iterations were set to 30 and 100, respectively. The crossover probability is taken as 0.4.Considering the importance of work efficiency in the feeding process of the manipulator, K_T and K_J are set to 0.7 and 0.3, respectively. The kinematic limits involving velocity, acceleration, and jerk of each joint are presented in Table 9.

Then the comparison between the time series obtained after trajectory optimization by executing IWOA-PSO algorithm and the time series before optimization are shown in Table 10. It is observed that the manipulator can reach the position points that the trajectory must pass in a relatively short time (total = 19.85s), which is 10.15s shorter than the total interpolation time before the optimized quantic B-spline interpolation.

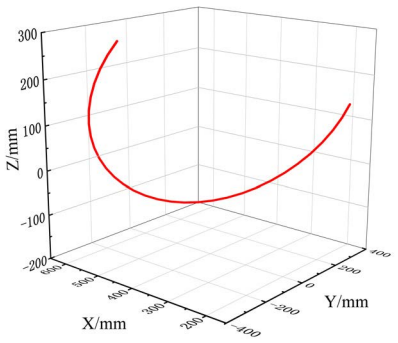

FIGURE 4. Trajectory curve in Cartesian coordinate system.

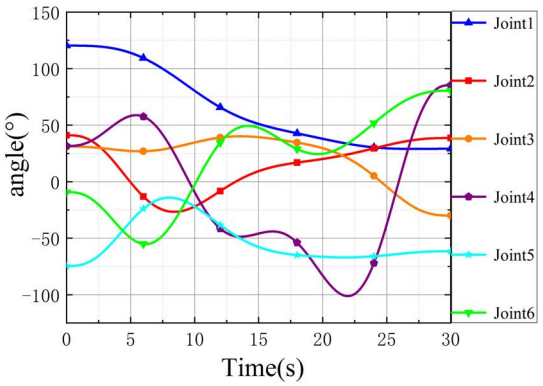

FIGURE 5. The joint angle plots of quantic B-spline interpolation.

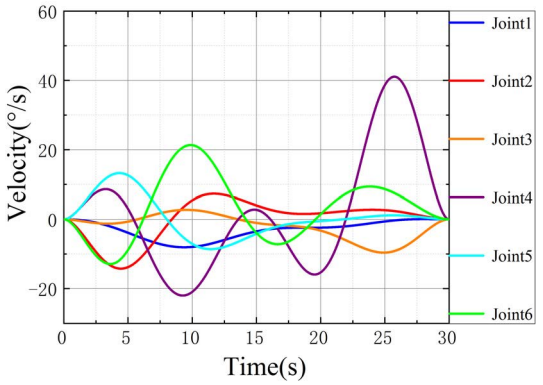

FIGURE 6. The joint velocity plots of quantic B-spline interpolation.

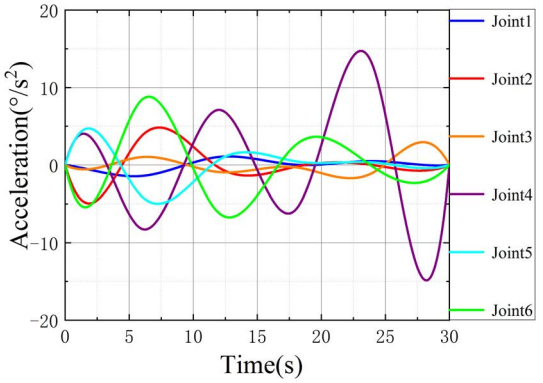

FIGURE 7. The joint acceleration plots of quantic B-spline interpolation.

Figure 9, 10, 11 and 12 present trajectories of six joints and their derivatives, including velocity, acceleration and

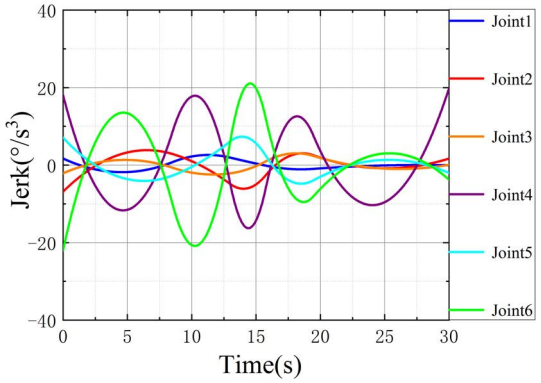

FIGURE 8. The joint jerk plots of quantic B-spline interpolation.

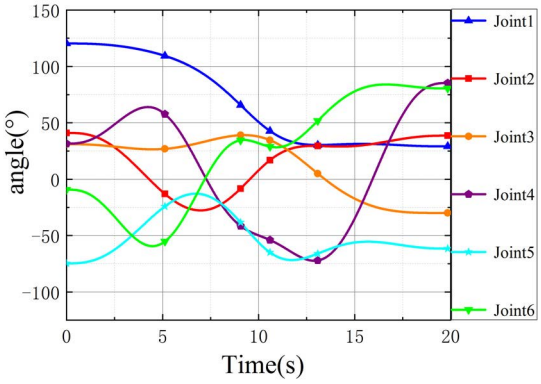

FIGURE 9. The plots of the joint angle after optimization.

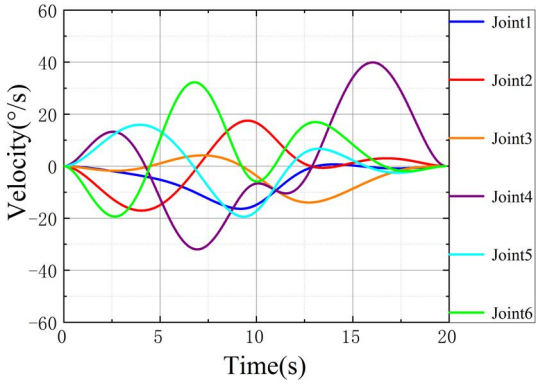

FIGURE 10. The plots of the joint velocity after optimization.

jerk after optimization. It is observed that the manipulator can reach the position points that the trajectory must pass in a relatively short time and is compatible with the position points before the optimization without the position error. The obtained results demonstrate that the proposed IWOA-PSO algorithm can improve the robot's efficiency in trajectory planning. Each joint after optimization moves with a higher acceleration at the beginning of the motion, which also ensures that the manipulator can reach a high speed lower than the limit speed in a short time that ensures the execution efficiency of the robot. The mean jerk reduces compared with the jerk before optimization. The maximum jerk of each joint before optimization is 2.627°/s^3, 6.861°/s^3, 3.051°/s^3, 19.829°/s^3, 7.339°/s^3 and 21.971°/s^3, respectively. Moreover, the maximum value of the jerk after optimization is 0.533°/s^3,5.801°/s^3, 3.166°/s^3,18.163 °/s^3, 5.655°/s^3 and 7.388°/s^3, respectively. Accordingly, it is concluded that IWOA-PSO can effectively reduce the jerk of the robot, while improving the manipulator's efficiency.

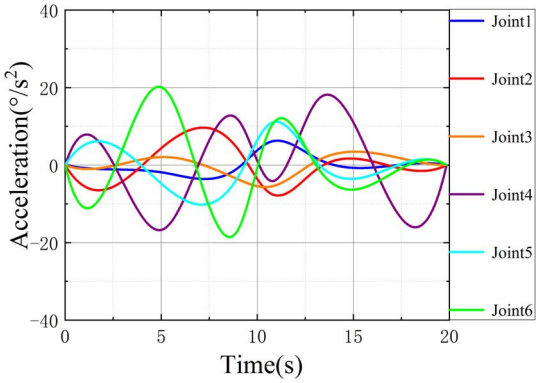

FIGURE 11. The plots of the joint acceleration after optimization.

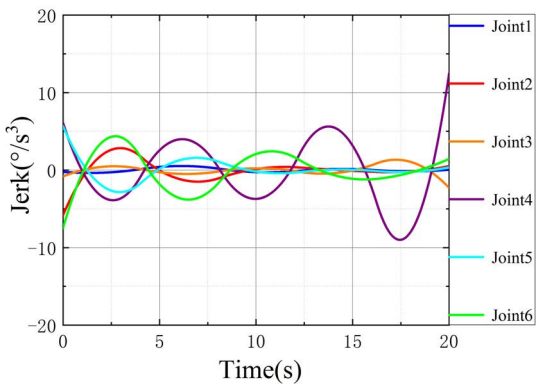

FIGURE 12. The plots of the joint jerk after optimization.

V. CONCLUSION

In the present study, the IWOA and the PSO techniques were combined and a hybrid IWOA-PSO optimization technique was proposed to get the time-jerk optimum path planning of the industrial manipulator. The proposed algorithm applies IWOA and PSO algorithms simultaneously, and then uses the crossover strategy for information exchange. Accordingly, diversity of the population is maintained in the later stages, convergence efficiency of the IWOA algorithm is improved and the problem of falling into the local optimal solution of the PSO algorithm is resolved. Then different algorithms were applied to 23 benchmark functions and the obtained results confirmed the superiority of the proposed algorithm. The optimal objective function considers both the whole running time and mean jerk combined along the whole path in each joint. The IWOA-PSO algorithm is implemented on a 6-DOF serial manipulator installed in a CNC machine tool. The obtained results from the simulation indicate that each joint after optimization through the proposed algorithm runs smoothly and satisfies the kinematics limits. Moreover, its velocity and acceleration are continuous and the jerk has been minimized significantly after optimization along with the minimum whole travel time. It is concluded that the proposed

approach is an effective scheme to maintain the continuity of the kinematic parameters for smooth traveling of the robot end-effector.

VI. FUTURE WORK

In future research, we will focus on testing the physical experimental performance of this optimal trajectory planning method presented in this paper, and consider other superior multi-objective optimization methods.

REFERENCES

[1] Q. L. Hou, Y. Dong, and S. J. Guo, "Review on energy consumption optimization methods of industrial robots," *Comput. Eng. Appl.*, vol. 54, no. 22, pp. 1–9, 2018.
[2] A. Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," *Adv. Eng. Softw.*, vol. 41, no. 4, pp. 548–556, 2010.
[3] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Mech. Mach. Sci.*, vol. 29, pp. 3–27, Jan. 2015.
[4] J. Wan, H. Wu, R. Ma, and L. Zhang, "A study on avoiding joint limits for inverse kinematics of redundant manipulators using improved clamping weighted least-norm method," *J. Mech. Sci. Technol.*, vol. 32, no. 3, pp. 1367–1378, Mar. 2018.
[5] C. S. Lin, P. R. Chang, and H. J. Lu, "Formulation and optimization of cubic polynomial joint trajectories for mechanical manipulators," *IEEE Trans. Autom. Control*, vol. AC-28, no. 12, pp. 330–335, Dec. 1983.
[6] P. Huang, Y. Xu, and B. Liang, "Global minimum-jerk trajectory planning of space manipulator," *Int. J. Control, Automat., Syst.*, vol. 4, no. 4, pp. 405–413, 2006.
[7] H.-I. Lin, "A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization," *J. Intell. Robotic Syst.*, vol. 75, nos. 3–4, pp. 379–392, 2014.
[8] D. Chen and Y. Zhang, "Minimum jerk norm scheme applied to obstacle avoidance of redundant robot arm with jerk bounded and feedback control," *IET Control Theory Appl.*, vol. 10, no. 15, pp. 1896–1903, Jun. 2016.
[9] H. Wang, H. Wang, J. H. Huang, B. Zhao, and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve," *Mech. Mach. Theory*, vol. 139, pp. 284–293, Sep. 2019.
[10] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robot. Comput.-Integr. Manuf.*, vol. 24, no. 3, pp. 415–426, 2008.
[11] X. Luo, S. Li, S. Liu, and G. Liu, "An optimal trajectory planning method for path tracking of industrial robots," *Robotica*, vol. 37, no. 3, pp. 502–520, Mar. 2019.
[12] G. G. Wang, S. Deb, and Z. H. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, Jul. 2020.
[13] A.-A. A. Mohamed, Y. S. Mohamed, A. A. M. El-Gaafary, and A. M. Hemeida, "Optimal power flow using moth swarm algorithm," *Electr. Power Syst. Res.*, vol. 142, pp. 190–206, Jan. 2017.
[14] W. Wang, Q. Tao, Y. Cao, X. Wang, and X. Zhang, "Robot time-optimal trajectory planning based on improved cuckoo search algorithm," *IEEE Access*, vol. 8, pp. 86923–86933, 2020.
[15] S. Mahapatra, B. Dey, and S. Raj, "A novel ameliorated Harris hawk optimizer for solving complex engineering optimization problems," *Int. J. Intell. Syst.*, vol. 36, no. 12, pp. 7641–7681, Dec. 2021.
[16] J. Huang, P. Hu, K. Wu, and M. Zeng, "Optimal time-jerk trajectory planning for industrial robots," *Mechanism Mach. Theory*, vol. 121, pp. 530–544, Mar. 2018.
[17] S. Lu, J. Zhao, L. Jiang, and H. Liu, "Time-jerk optimal trajectory planning of a 7-DOF redundant robot," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 25, no. 5, pp. 4211–4222, 2017.
[18] L. Feifei and L. Fei, "Time-jerk optimal planning of industrial robot trajectories," *Int. J. Robot. Autom.*, vol. 31, no. 1, pp. 1–7, 2016.
[19] A. Rout, M. Dileep, G. B. Mohanta, B. Deepak, and B. Biswal, "Optimal time-jerk trajectory planning of 6 axis welding robot using TLBO method," *Proc. Comput. Sci.*, vol. 133, pp. 537–544, Jan. 2018.
[20] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.
[21] W. Y. Guo, T. Liu, F. Dai, and P. Xu, "An improved whale optimization algorithm for forecasting water resources demand," *Appl. Soft Comput.*, vol. 86, pp. 1–18, Jan. 2020.
[22] T. Wang, Z. Xin, H. Miao, H. Zhang, Z. Chen, and Y. Du, "Optimal trajectory planning of grinding robot based on improved whale optimization algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–8, Aug. 2020.
[23] M. Liu, X. Yao, and Y. Li, "Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 105954.
[24] X. Shi, H. Fang, and L. Guo, "Multi-objective optimal trajectory planning of manipulators based on quintic NURBS," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Aug. 2016, pp. 759–765.
[25] C. de Boor, *A Practical Guide to Splines*. New York, NY, USA, 1979, pp. 81–82.
[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov. 1995, pp. 1942–1948.
[27] S. M. Bozorgi and S. Yazdani, "IWOA: An improved whale optimization algorithm for optimization problems," *J. Comput. Des. Eng.*, vol. 6, no. 3, pp. 243–259, Jul. 2019.
[28] M. M. Saafan and E. M. El-Gendy, "IWOSSA: An improved whale optimization salp swarm algorithm for solving optimization problems," *Expert Syst. Appl.*, vol. 176, pp. 1–21, Aug. 2021.
[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[30] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
[31] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

JING ZHAO was born in Jinzhong, China, in 1994. She received the bachelor's degree from the Central South University of Forestry and Technology, in 2016. She is currently pursuing the Ph.D. degree in mechanical engineering with the North University of China, Taiyuan, Shanxi, China.

Her research interests include trajectory planning and robot intelligent machining and control of robot.

XIJING ZHU was born in Qixian, Shanxi, China, in 1969. He received the B.S. degree from the North University of China, the M.S. degree in mechanical engineering from Northeastern University, and the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics, in 2007. He is a currently a Distinguished Professor of mechanical engineering with the North University of China. He is the author of two books and more than 100 articles. His research interests include nontraditional precision machining and intelligent assembly manufacturing projects.

TIANJIAO SONG is currently pursuing the bachelor's degree with the North University of China, China.

His research interests include robot dynamics and robot ultrasonic machining.