IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 13, 2020, accepted December 29, 2020, date of publication January 6, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049175

Process Synthesis and Design Problems Based on a Global Particle Swarm Optimization Algorithm

CHUANHU CHEN[1] AND CHUNLIANG LI[2]
[1]School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou 221116, China
[2]Xuzhou Boyuan Technology Company Ltd., Xuzhou 221000, China

Corresponding author: Chuanhu Chen (cch@jsnu.edu.cn)

This work was supported by the Transformation Project of Xuzhou Science and Technology under Grant KC19234.

ABSTRACT Many process synthesis and design problems in engineering are actually mixed integer nonlinear programming problems (MINLP), because they contain both continuous and integer variables. These problems are generally recognized to be complex and intractable by virtue of the combinatorial characteristic. In order to effectively solve process synthesis and design problems, a global particle swarm optimization (GPSO) algorithm is proposed in this paper. GPSO algorithm makes two improvements on original particle swarm optimization (PSO) algorithm: first, it introduces a global inertia weight, which is beneficial for improving its global searching capacity during the whole optimization process; second, it adopts a mutation operation with a small probability, which enables the GPSO algorithm to get rid of the local optimum easily. Simulation results show that the GPSO algorithm has high efficiency on finding the optimal solutions, and it has stronger convergence than the other four particle swarm optimization algorithms.

INDEX TERMS Process synthesis, global particle swarm optimization algorithm, global inertia weight, mutation, convergence.

I. INTRODUCTION

In order to establish an optimal construction, the selection, arrangement, and operation should be implemented for processing units, and this procedure is defined as process synthesis [1]. More specifically, it does not dominate the optimal interconnection of processing units, but also determine the optimal type and design of units in a process system. When the system function is confirmed, the system structure and the performance of the processing units are still not ensured. The work is essentially combinatorial and open-ended and has drawn much attention from researchers recently. The process synthesis and design problem belongs to a kind of constrained optimization problem. It has both integer and real variables, and is associated with some equality and inequality constraints. To solve this kind of complex problem, some efficient methods are needed to satisfy all its constraints and minimize (or maximize) its objective function. On the other hand, particle swarm optimization algorithm (PSO) [2] is an easy and practical intelligent optimization algorithm. Moreover, the PSO and its many variants have been applied into many optimization problems, such as wastewater treatment network planning [3], image classification [4], multi-workshop facility layout problem [5], lipid extraction from microalgae [6], efficient object tracking in a video [7] and multi-compartment vehicle routing problem [8].

The PSO has two important steps including velocity updating and position updating, which can be expressed as follows:

$$v_{i,j}^{k+1} = v_{i,j}^{k} + c_1 \times r_1 \times (\mathrm{pb}_{i,j} - x_{i,j}^{k}) + c_2 \times r_2 \times (\mathrm{gb}_j - x_{i,j}^{k}), \tag{1}$$

$$x_{i,j}^{k+1} = x_{i,j}^{k} + v_{i,j}^{k+1}. \tag{2}$$

Here, $i(i = 1, \ldots, M)$ is the index of a particle, $j(j = 1, \ldots, N)$ is the index of a variable, k is the current iteration number, pb is the best position of particle i, gb is the best solution, c_1 and c_2 are learning factors, r_1 and r_2 are random numbers uniformly generated in [0, 1]. In addition, $v_{i,j}^{k}$ and $x_{i,j}^{k}$ are the velocity component and position component of the ith particle at generation k, $v_{i,j}^{k+1}$ and $x_{i,j}^{k+1}$ are the velocity component and position component of the ith particle at generation $k+1$.

II. A GLOBAL PARTICLE SWARM OPTIMIZATION ALGORITHM

In order to improve the performance of the PSO, a global particle swarm optimization algorithm (GPSO) is proposed in this paper, and the GPSO includes five steps:

Step 1 (Set Parameters): IPSO parameters include population size *PS*, the maximal iteration number *NI*, learning

The associate editor coordinating the review of this manuscript and approving it for publication was Jagdish Chand Bansal.

factors c_1 and c_2, the minimal mutation rate p_{min} and the maximal mutation rate p_{max}, the period of a cosine function T which is used to generate a inertia weight.

Step 2 (Randomly Generate PS Particles): Randomly generate the jth ($j = 1, \ldots, N$) variable of the ith ($i = 1, \ldots, PS$) particle from a uniform distribution in the ranges $[x_{dL}, x_{dU}]$ ($j = 1, 2, \ldots, N$). Here, N is the total number of variable, x_{dL}($j = 1, \ldots, N$) and x_{dU} are the lower bound and the upper bound for the jth variable.

Step 3 (Velocity Updating and Position Updating): The velocity updating of the GPSO is different from that of the original PSO, and it adopts a new strategy as follows:

$$v_{id}^{t+1} = \omega^t v_{id}^t + c_1 r_1 (p_{id} - x_{id}^t) + c_2 r_2 (p_{gd} - x_{id}^t). \tag{3}$$

In which, i is the particle index; j is the variable index; t is the iteration numbers so far; pb represents the best position of particle i so far; gb represents the best solution in the swarm so far; c_1 and c_2 are acceleration constants; r_1 and r_2 are random numbers between 0 and 1. In addition, ω^t is inertia weight, and it is given by

$$\omega^t = (\cos(\theta t) + 1)/2. \tag{4}$$

where θ is angular frequency, and $\theta = 2\pi/T$. Therefore, ω^t is a periodic function with respect to the iteration numbers t. Due to the utilization of a cosine function, ω^t always owns large values during each period, which is beneficial to improving the global searching capacity of the GPSO.

Step 4 (Mutation): In order to improve the capacity of escaping from the local optimums, a mutation operation is introduced in this paper, and it is given by

$$x_{i,d}^{t+1} = p_{i1,d}^t + F \times rand() \times (p_{i2,d}^t - p_{i3,d}^t). \tag{5}$$

Here, F is scale factor, x_{id}^{t+1} is the dth ($d = 1, \ldots, N$) dimension of the ith particle at the $(k+1)$th iteration. $p_{i1,d}^t(p_{i2,d}^t, p_{i3,d}^t)$ is the dth dimension of the $i1(i2, i3)$th personal best particle, and $i1 \neq i2 \neq i3$. This mutation operation is determined by a mutation rate which is given by

$$p^t = p_{min} + (p_{max} - p_{min}) \times t/NI. \tag{6}$$

Here, p^t is the mutation at the tth iteration, p_{min} and p_{max} are the minimal mutation rate and the maximal mutation rate, respectively. At the beginning of optimization process, the value of p^t is small, it does no harm to the convergence of the GPSO, and it can guarantee the diversity of the swarm. In the late optimization, the value of p^t is large, which is beneficial to avoiding the premature convergence of the GPSO.

Step 5 (Perturbation): The history best particles can provide useful and potential reference for all the individuals of the population, thus they are incorporated into the perturbation operator. In short, the new formulation of perturbation is given by:

$$x_{i,d}^{t+1} \leftarrow x_{i,d}^{t+1} + rand() \times (\mathrm{pb}_{i_a,d}^t - \mathrm{pb}_{i_b,d}^t). \tag{7}$$

i_a and i_b are two randomly chosen indexes, and $i_a \neq i_b \neq i$. The probability of perturbation is set to $1/(5N)$, which only adjusts each dimension with a very low probability. By using Eq.(7), GPSO is able to conduct local search and gradually improve the individuals of the population.

Step 6 (Stopping Condition): If the iteration number t reaches the maximal iteration number NI, stop running the GPSO procedure, otherwise, repeat steps 3 and 4.

Among the above steps, Step 3 uses a dynamic inertia weight to balance the global search and local search, Step 4 uses mutation operator to maintain the diversity of population, and Step 5 uses perturbation operator to improve the exploration capacity with respect to solution space.

III. PROBLEM FORMULATION

The chemical process synthesis problem aims to choose the optimal flowsheet structure and the parameters which characterize the operation of a desired process. In order to define the search space of candidate flowsheet alternatives, a superstructure is to be postulated based on preliminary screening. This superstructure is a MINLP problem [13] which is given by

$$\begin{aligned} F = \min_{x,y} \; & c^{\mathrm{T}}y + f(x), \\ \text{s.t.} \;\; & h(x) = 0, \\ & g(x) \leq 0, \\ & A(x) = a, \\ & vBy + Cx \leq d, \\ & x \in X = \{x | x \in R^n, x^L \leq x \leq x^U\}, \\ & y \in Y = \{y | y \in \{0,1\}^m, \mathrm{E}y \leq \mathrm{e}\}. \end{aligned} \tag{8}$$

Here, x denotes the vector of continuous variables in the set X, and y is the vector of $0-1$ variables which must meet linear inter constraints $Ey \leq \mathrm{e}$. $f(x), h(x) = 0$, and $g(x \leq 0)$ represent nonlinear functions involved in the objective function, equations, and inequalities, respectively. Finally, $Ax = a$ is the subset of linear equations, and $By + Cx \leq d$ is linear equalities of inequalities which contain the continuous and binary variables.

Regarding the synthesis problem, the continuous variables x are composed of temperatures, flows, pressures, and sizes. Moreover, the binary variables y represent the potential existence of process units which are embedded in the superstructure. The formulas $h(x) = 0$ and $Ax = a$ are associated with material and energy balances and design equations. Process specifications are represented by $g(x) \leq 0$ and by lower and upper bounds on the variables in x. Logical constraints that must hold for a flow sheet configuration to be chosen from in the superstructure are denoted by $By + Cx \leq d$ and $Ey \leq \mathrm{e}$. The cost function contains fixed cost charges in the term $c^{\mathrm{T}}y$ for the investment, while operating costs, revenues, and size-dependent costs for the investment are involved in the function $f(x)$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the performance of the GPSO, eight unconstrained problems are selected, and they are given by

$$f_1(x) = \sum_{i=1}^{n} x_i^2, -100 \leq x_i \leq 100 \tag{9}$$

$$f_2 = \sum_{i=1}^{n}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2),$$
$$-100 \le x_i \le 100 \tag{10}$$
$$f_3 = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10),$$
$$-10 \le x_i \le 10 \tag{11}$$
$$f_4 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1,$$
$$-600 \le x_i \le 600 \tag{12}$$
$$f_5 = 20 + e - 20\exp(-0.2\sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}})$$
$$- \exp(\frac{\sum_{i=1}^{N}\cos(2\pi x_i)}{N}), -32 \le x_i \le 32 \tag{13}$$
$$f_6 = \sum_{i=1}^{N}|x_i| + \prod_{i=1}^{N}|x_i|, -100 \le x_i \le 100 \tag{14}$$
$$f_7 = 418.9829N - \sum_{i=1}^{N}(x_i \sin(\sqrt{|x_i|})),$$
$$-500 \le x_i \le 500 \tag{15}$$
$$f_8 = \sum_{i=1}^{N}(\sum_{j=1}^{i} x_j)^2, -100 \le x_i \le 100 \tag{16}$$

Five PSO algorithms are used to solve the above eight unconstrained optimization problems with dimension size $N = 100$, and they are PSO^{T7} [9], PSO^{T7} [9], BBPSO(Bare bones particle swarm optimization) [10], BBPSO+GJ(Bare bones particle swarm optimization with Gaussian jumps) [11] and GPSO, respectively. Moreover, their parameters are set as follows: For PSO^{T7}, learning factors $c_1 = c_2 = 2$, population size $PS = 40$, the maximal iteration number $NI = 100$, the initial value of a chaotic sequence is $y(0) = 0.48$; For PSO^{T8}, $c_1 = c_2 = 2$, $PS = 40$, $NI = 100$, the initial value of a chaotic sequence is $y(0) = 0.48$; For BBPSO, $c_1 = c_2 = 2$, $PS = 40$, $NI = 100$; For BBPSO+GJ, $c_1 = c_2 = 2$, $PS = 40$, $NI = 100$, scale parameter $\eta = 1.1$; For GPSO, $c_1 = c_2 = 2$, $PS = 40$, $NI = 100$, the period of a cosine function $T = NI/10$, the minimal mutation rate $p_{min} = 0.01$, the maximal mutation rate $p_{max} = 0.1$, scale factor $F = 2$. 50 runs are carried out in each case, and the optimization results are shown in Table 1:

Here, the terms f_{min}, f_{max}, f_{aver} and f_{sd} are the minimal value, maximal value, average value and standard deviation of 50 objective function values, respectively, for each problem. According to Table 1, the GPSO is better than the other four methods in most cases. The values of f_{min}, f_{max}, and f_{aver} obtained by the GPSO are better than those obtained by the other four PSO algorithms for all unconstrained optimization problems. The values of f_{min}, f_{max}, f_{aver} and f_{sd} obtained by the GPSO are better than those obtained by the other four PSO algorithms for the first six unconstrained optimization problems. Moreover, GPSO can obtain the optimal solutions for f_1, f_3, f_4, f_5, f_6 and f_8, but any of the other algorithm fails to find an optimal solution for any problem. In addition, a statistical analysis based on t-test is used to distinguish whether the results of GPSO are statistically significantly better than those of the other. The term $>$ indicates that GPSO is significantly better than any other PSO algorithm in achieving optimization results. In short, the GPSO has shown stronger convergence and stability than the other four PSO algorithms. In order to verify and analyze the performance of the GPSO, Figs 1-8 plot the average convergence curves of five PSO algorithms for four unconstrained optimization problems.

As can be seen from Figs 1-8, the convergence rates of PSO^{T7} and PSO^{T8} are very slow, and the convergence rates of BBPSO, BBPSO+GJ are slightly faster than PSO^{T7} and PSO^{T8}. Based on a close observation from the above eight figures, the average convergence curves of GPSO decreases rapidly in the early optimization process, and its convergence rate is faster than those of the other four PSO algorithms. Furthermore, GPSO finally converges to a relatively low level for any of the eight unconstrained optimization problems compared with the the other four PSO algorithms.

To testify the performance of GPSO on solving process synthesis and design problems, four examples are considered, and they are explained as follows:

Example 1: Example 1 has one nonlinear constraint, and researchers have used different methods to solve this problem. Costa and Oliviera [14] used an evolution algorithm (EA) to solve Example 1. EA simulates the evolution process of natural system, and it only needs the information of objective functions and constraints. Cardoso *et al.* [15] used simulated annealing algorithm (SA) to solve Example. SA is a powerful tool, it summarizes the similarities of optimization problems and physical process of annealing. The mathematical model of Example 1 is given by

$$\begin{aligned} &\text{Min } f(x, y) = 2x + y \\ &\text{s.t. } 1.25 - x^2 - y \le 0, \\ &\quad x + y \le 1.6, \\ &\quad 0 \le x \le 1.6, \\ &\quad y \in \{0, 1\}. \end{aligned} \tag{17}$$

The best solution is located at $(x, y) = (0.5, 1)$, and the corresponding objective function value is equal to $f = 2$.

Example 2: Example 2 contains a nonlinear constraint. This problem was firstly proposed by Kocis and Grossmann [16], and it was solved by Costa and Oliviera [14] and Cardoso *et al.* [15]. The mathematical model of Example 2 is given by

$$\begin{aligned} &\text{Min } f(x_1, x_2, y) = -y + 2x_1 + x_2 \\ &\text{s.t. } x_1 - 2\exp(-x_2) = 0, \\ &\quad -x_1 + x_2 + y \le 0, \\ &\quad 0.5 \le x_1 \le 1.4, \\ &\quad y \in \{0, 1\}. \end{aligned} \tag{18}$$

TABLE 1. Comparison of PSO^{T7}, PSO^{T8}, BBPSO, BBPSO+GJ and GPSO on four unconstrained problems.

Problem	Algorithm	f_{min}	f_{max}	f_{aver}	f_{sd}	t-test
f_1	PSO^{T7}	2.4090e+05	2.9042e+05	2.7097e+05	9.8412e+03	>
	PSO^{T8}	2.3294e+05	2.8633e+05	2.6589e+05	1.2009e+04	>
	BBPSO	8.5648e+04	1.6825e+05	1.3690e+05	1.8766e+04	>
	BBPSO+GJ	2.3436e+04	5.1067e+04	3.5332e+04	6.1579e+03	>
	GPSO	0	1.4045e+04	3.0798e+02	1.9840e+03	–
f_2	PSO^{T7}	9.6338e+10	1.5763e+11	1.4031e+11	1.2103e+10	>
	PSO^{T8}	1.1207e+11	1.6346e+11	1.4187e+11	1.2528e+10	>
	BBPSO	4.7961e+10	1.4951e+11	9.1958e+10	2.3476e+10	>
	BBPSO+GJ	2.9402e+09	1.9034e+10	1.1858e+10	3.7125e+09	>
	GPSO	9.9000e+01	3.0712e+08	6.6312e+06	4.3487e+07	–
f_3	PSO^{T7}	3.0584e+03	3.8295e+03	3.5947e+03	1.7293e+02	>
	PSO^{T8}	3.2843e+03	3.8239e+03	3.5902e+03	1.3226e+02	>
	BBPSO	1.8973e+03	3.0327e+03	2.4068e+03	2.2955e+02	>
	BBPSO+GJ	7.3666e+02	1.1133e+03	8.9568e+02	9.1093e+01	>
	GPSO	0	9.1914e+02	3.9289e+01	1.4363e+02	–
f_4	PSO^{T7}	2.1268e+03	2.6519e+03	2.4317e+03	1.1815e+02	>
	PSO^{T8}	2.1181e+03	2.6208e+03	2.4115e+03	1.1943e+02	>
	BBPSO	9.4045e+02	1.8745e+03	1.2603e+03	1.6393e+02	>
	BBPSO+GJ	1.5630e+02	4.3077e+02	2.9513e+02	5.7215e+01	>
	GPSO	0	2.1856e+01	1.1734e+00	3.9091e+00	–
f_5	PSO^{T7}	1.9962e+01	1.9965e+01	1.9963e+01	7.9508e-04	>
	PSO^{T8}	1.9962e+01	1.9965e+01	1.9963e+01	7.5218e-04	>
	BBPSO	2.0382e+01	2.0861e+01	2.0651e+01	1.3986e-01	>
	BBPSO+GJ	1.3199e+01	1.6970e+01	1.5671e+01	8.0264e-01	>
	GPSO	0	2.5571e-01	1.1133e-02	4.9129e-02	–
f_6	PSO^{T7}	3.8808e+03	4.5983e+03	4.2736e+03	1.5462e+02	>
	PSO^{T8}	3.9742e+03	4.5304e+03	4.2632e+03	1.2155e+02	>
	BBPSO	2.0370e+03	3.1141e+03	2.5874e+03	2.5396e+02	>
	BBPSO+GJ	6.1854e+02	9.9257e+02	7.9870e+02	9.5587e+01	>
	GPSO	0	1.4962e+02	3.3824e+00	2.1282e+01	–
f_7	PSO^{T7}	2.9663e+04	3.7190e+04	3.4588e+04	1.7729e+03	>
	PSO^{T8}	2.9783e+04	3.7288e+04	3.4407e+04	1.8511e+03	>
	BBPSO	1.8809e+04	2.4690e+04	2.1811e+04	1.5188e+03	>
	BBPSO+GJ	2.0428e+04	3.0866e+04	2.3041e+04	1.7881e+03	>
	GPSO	1.1041e+03	2.2358e+04	1.6350e+04	3.3406e+03	–
f_8	PSO^{T7}	6.7910e+05	1.1444e+06	9.2397e+05	1.2963e+05	>
	PSO^{T8}	5.9046e+05	1.1348e+06	8.7236e+05	1.2916e+05	>
	BBPSO	3.4127e+05	6.5673e+05	4.6532e+05	7.8082e+04	>
	BBPSO+GJ	2.3368e+05	6.3853e+05	4.4073e+05	8.5042e+04	>
	GPSO	0	1.2788e+06	1.9208e+05	3.9057e+05	–

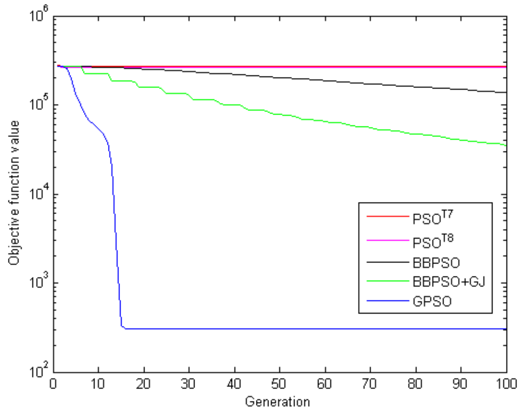

FIGURE 1. The average convergence curves of five PSO algorithms for f_1.

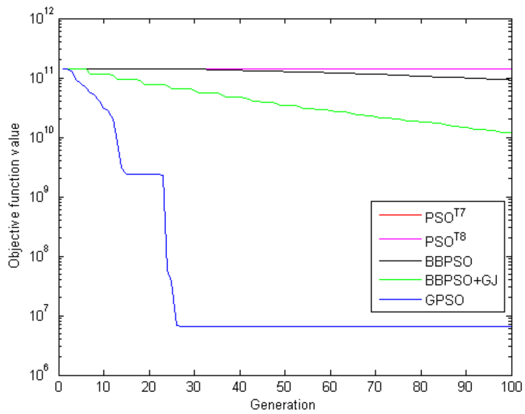

FIGURE 2. The average convergence curves of five PSO algorithms for f_2.

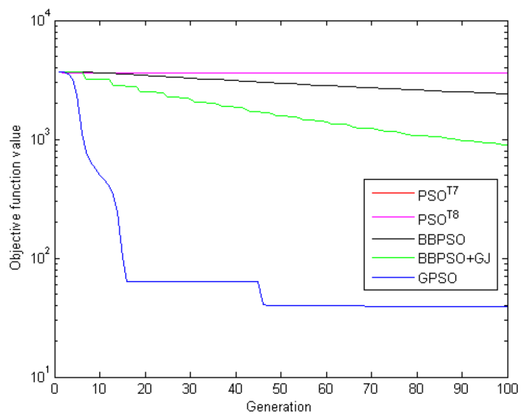

FIGURE 3. The average convergence curves of five PSO algorithms for f_3.

The best solution is located at $(x_1, x_2, y) = (1.375, 0.375, 1)$, and the corresponding objective function value is equal to $f = 2.124$.

Example 2:* By eliminating a nonlinear equality constraint, Example 2 can be simplified as follows:

$$\begin{aligned} \text{Min}\ & f(x_1, y) = -y + 2x_1 - \ln(x_1/2) \\ \text{s.t.}\ & -x_1 - \ln(x_1/2) + y \leq 0, \\ & 0.5 \leq x_1 \leq 1.4, \\ & y \in \{0, 1\}. \end{aligned} \tag{19}$$

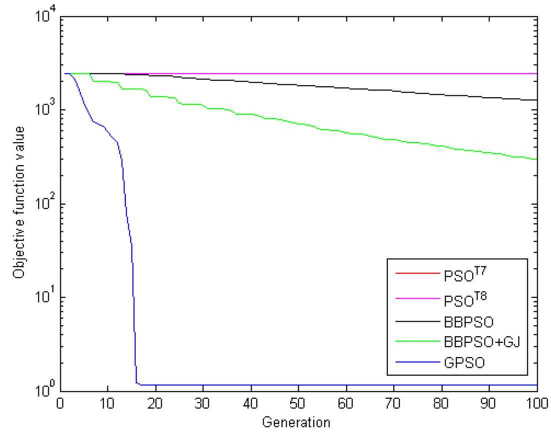

FIGURE 4. The average convergence curves of five PSO algorithms for f_4.

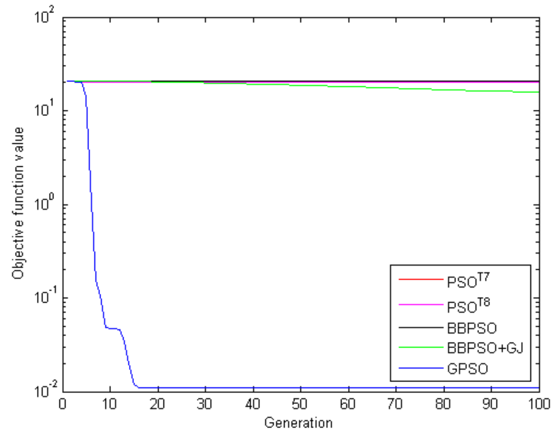

FIGURE 5. The average convergence curves of five PSO algorithms for f_5.

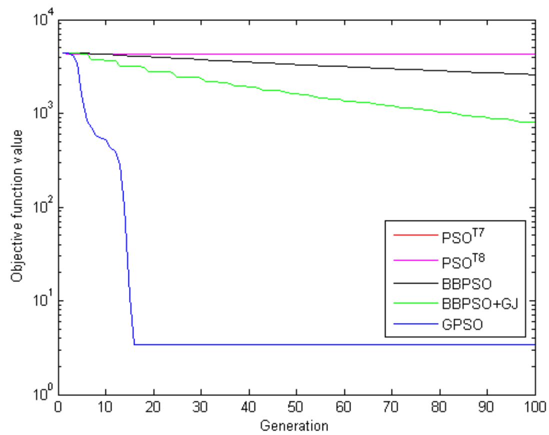

FIGURE 6. The average convergence curves of five PSO algorithms for f_6.

The best solution of Example 2* is exactly the same as that of Example 2.

Example 3: Example 3 was firstly proposed by Kocis and Grossmann [17]. This problem aims to select one between two candidate reactors so as to minimize the operating cost. In addition, this problem was solved by Diwekar and Rubin [18], Costa and Oliviera [14] and Cardoso *et al.* [15].

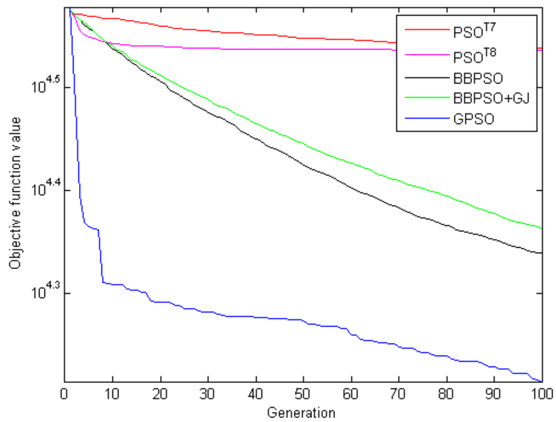

FIGURE 7. The average convergence curves of five PSO algorithms for f_7.

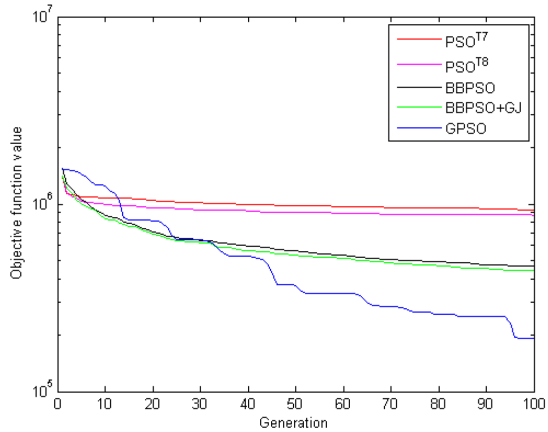

FIGURE 8. The average convergence curves of five PSO algorithms for f_8.

The mathematical model of Example 3 is given by

$$\begin{aligned}
&\text{Min } f(x, y_1, y_2, v_1, v_2) = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x\\
&\text{s.t. } y_1 + y_2 = 1,\\
&\quad z_1 = 0.9[1 - \exp(-0.5v_1)]x_1,\\
&\quad z_2 = 0.8[1 - \exp(-0.4v_2)]x_2,\\
&\quad z_1 + z_2 = 10,\\
&\quad x_1 + x_2 = x,\\
&\quad z_1y_1 + z_2y_2 = 10,\\
&\quad v_1 \le 10y_1,\\
&\quad v_2 \le 10y_2,\\
&\quad x_1 \le 20y_1,\\
&\quad x_2 \le 20y_2,\\
&\quad x_1, x_2, z_1, z_2, v_1, v_2 \ge 0,\\
&\quad y_1, y_2 \in \{0, 1\}. \qquad (20)
\end{aligned}$$

The best solution is located at $(x, y_1, y_2, v_1, v_2) = (13.36227, 1, 0, 3.514237, 0)$, and the corresponding objective function value is equal to $f = 99.245209$.

Example 3:* By eliminating nonlinear equality constraints, Example 3 can be simplified as follows:

$$\begin{aligned}
&\text{Min } f(y_1, v_1, v_2) = 7.5y_1 + 5.5(1 - y_1) + 7v_1 + 6v_2\\
&\quad + 50\frac{1 - y_1}{0.8[1 - \exp(-0.4v_2)]} + 50\frac{y_1}{0.9[1 - \exp(-0.5v_1)]}\\
&\text{s.t. } 0.9[1 - \exp(-0.5v_1)] - 2y_1 \le 0,\\
&\quad 0.8[1 - \exp(-0.4v_2)] - 2(1 - y_1) \le 0,\\
&\quad v_1 \le 10y_1,\\
&\quad v_2 \le 10(1 - y_1),\\
&\quad v_1, v_2 \ge 0,\\
&\quad y_1 \in \{0, 1\}. \qquad (21)
\end{aligned}$$

The best solution of Example 3* is exactly the same as that of Example 3. The schematic diagram of Example 3* is shown in Fig. 9

Example 4: Example 4 is a maximization problem, and it was solved by Costa and Oliviera [14] and Cardoso *et al.* [15].

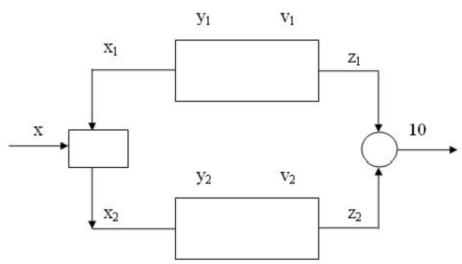

FIGURE 9. Superstructure for two-reactor problem.

TABLE 2. Coefficients for Example 4.

$a_1 = 85.334407$	$a_5 = 80.51249$	$a_9 = 9.300961$
$a_2 = 0.0056858$	$a_6 = 0.0071317$	$a_{10} = 0.0047026$
$a_3 = 0.0006262$	$a_7 = 0.0029955$	$a_{11} = 0.0012547$
$a_4 = 0.0022053$	$a_8 = 0.0021813$	$a_{12} = 0.0019085$

The mathematical model is given by

$$\begin{aligned}
&\text{Max } f(x_1, x_2, x_3, y_1, y_2) = -5.357854x_1^2\\
&\qquad - 0.835689y_1x_3 - 37.29329y_1 + 40792.141\\
&\text{s.t. } a_1 + a_2y_2x_3 + a_3y_1x_2 - a_4x_1x_3 \le 92,\\
&\quad a_5 + a_6y_2x_3 + a_7y_1y_2 + a_8x_1^2 - 90 \le 20,\\
&\quad a_9 + a_{10}x_1x_3 + a_{11}y_1x_1 + a_{12}x_1x_2 - 20 \le 5,\\
&\quad 27 \le x_1, x_2, x_3 \le 45,\\
&\quad y_1 \in \{78, \ldots, 102\}, \text{integer},\\
&\quad y_2 \in \{33, \ldots, 45\}, \text{integer}. \qquad (22)
\end{aligned}$$

The coefficients $a_i (i = 1, \ldots, 12)$ can be obtained from Table 2. The global best solution of Example 4 is $(x_1, x_3, y_1) = (27, 27, 78)$, and the other variables x_2 and y_2 can combine optionally. The optimal objective function value is equal to $f = 32217.4$.

Five PSO algorithms are used to solve the above process synthesis and design problems, and their parameters settings are the same as those of $f_1 - f_8$. Only the population size *PS* is reset to 80. 50 runs are conducted in each case, and the optimization results are shown in Table 3:

TABLE 3. The optimization results obtained by five PSO algorithms for four problems.

Problem	Algorithm	f_{min}	f_{max}	f_{aver}	f_{sd}
Case 1	PSO^{T7}	2.0000	2.2373	2.0104	4.6767e-02
	PSO^{T8}	2.0000	2.0118	2.0015	1.8406e-03
	BBPSO	2.0000	2.0000	2.0000	1.3458e-15
	BBPSO+GJ	2.0000	2.0000	2.0000	1.3458e-15
	GPSO	2.0000	2.0000	2.0000	5.0895e-08
Case 2*	PSO^{T7}	2.1245	2.1567	2.1282	1.0041e-02
	PSO^{T8}	2.1245	2.1547	2.1263	6.1148e-03
	BBPSO	2.1245	2.5578	2.1331	6.1285e-02
	BBPSO+GJ	2.1245	2.5578	2.1418	8.5781e-02
	GPSO	2.1245	2.1245	2.1245	2.0241e-10
Case 3*	PSO^{T7}	99.239635	99.239692	99.239643	1.27e-05
	PSO^{T8}	99.239635	99.239726	99.239648	1.88e-05
	BBPSO	99.239635	156.876392	101.545105	1.14e+01
	BBPSO+GJ	99.239635	156.876392	100.392370	8.15e+00
	GPSO	99.239635	99.239635	99.239635	2.98e-14
Case 4	PSO^{T7}	32217.42778	32217.42778	32217.42778	1.84e-011
	PSO^{T8}	32217.42778	32217.42778	32217.42778	1.84e-011
	BBPSO	32217.42778	32217.42778	32217.42778	1.84e-011
	BBPSO+GJ	32217.42778	32217.42778	32217.42778	1.84e-011
	GPSO	32217.42778	32217.42778	32217.42778	1.84e-011

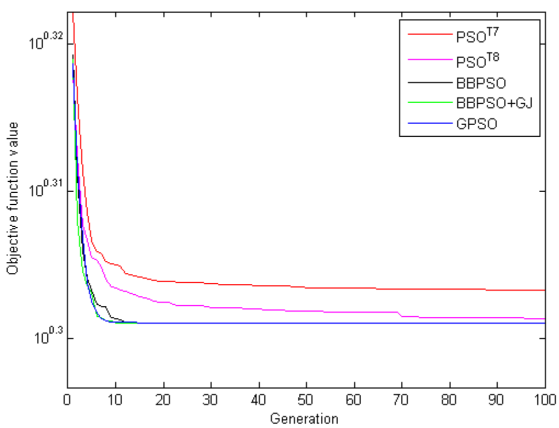

FIGURE 10. The average convergence curves of five PSO algorithms for Case 1.

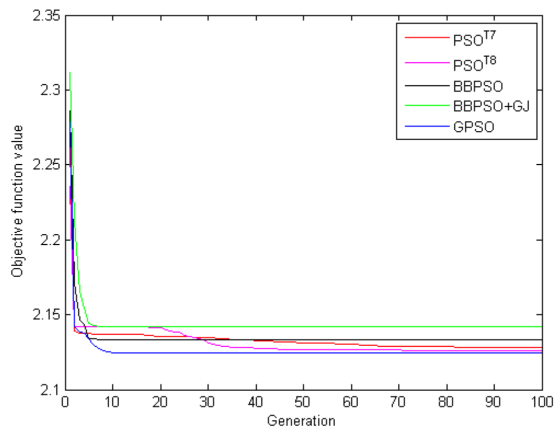

FIGURE 11. The average convergence curves of five PSO algorithms for Case 2.

According to Table 3, all the five PSO algorithms can find the best solutions for all four process synthesis and design problems. For Case 4, they find the optimum with the success rate 100%. For Case 1, GPSO, BBPSO and BBPSO+GJ can find the optimum with the success rate 100%, but PSO^{T7} and PSO^{T8} fail. For Case 2*, only GPSO can find the optimum with the success rate 100%. For Case 3*, GPSO, PSO^{T7} and PSO^{T8} can find the optimum with the success rate 100%, and the f_{aver} of GPSO is slightly smaller than those of PSO^{T7} and PSO^{T8}. In all, GPSO has exhibited stronger convergence and stability than the other four PSO algorithms.

Figs.10-13 depict the average convergence curves of five PSO algorithms for Cases 1, 2, 3 and 4. For the first three cases, GPSO can converge to the lowest levels at the end of generation. For Case 4, all five PSO algorithms can achieve the highest level at the end of generation.

By introducing a dynamic inertia weight, a new mutation operator and a novel perturbation operator, GPSO has stronger convergence and stability than the other four PSO algorithms, and is able to obtain the optimal solutions for most of the above problems.

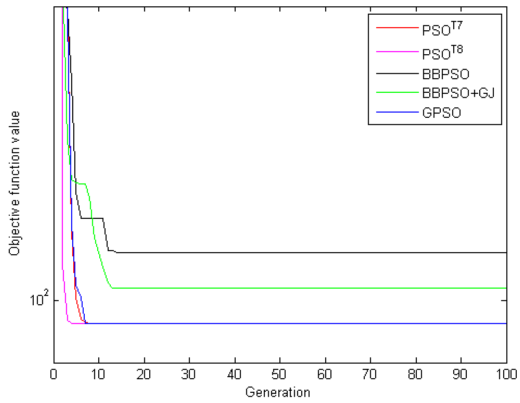

FIGURE 12. The average convergence curves of five PSO algorithms for Case 3.

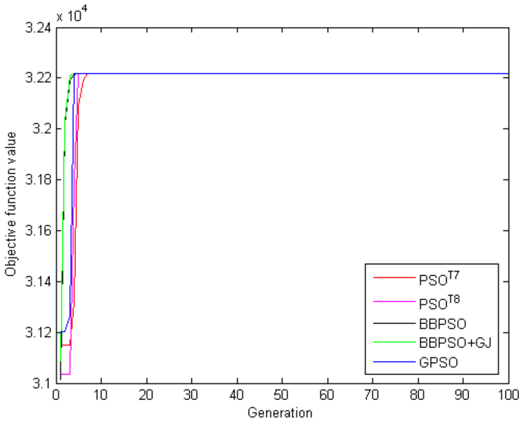

FIGURE 13. The average convergence curves of five PSO algorithms for Case 4.

V. CONCLUSION

In this paper, a global particle swarm optimization algorithm (GPSO) is designed to solve process synthesis and design problems. The GPSO introduces an inertia weight based on cosine function, and adds a mutation operation after position updating. The former enables the GPSO has strong global searching capacity over the whole optimization process; The latter can not only keep the diversity of the swarm, but also prevent the GPSO from trapping into the local optimums. Simulation results demonstrate that the GPSO has strong convergence and stability on solving most optimization problems, and it has higher efficiency of finding the global optimums. Compared to the other four PSO algorithms, the GPSO is more competent for process synthesis and design problems. Our further work will focus on the applications of the other metaheuristic algorithms [19] to process synthesis and design problems, such as chaotic krill herd algorithm [20], differential evolution algorithm [21]–[24] and monarch butterfly optimization [25].

REFERENCES

[1] R. Angira and B. V. Babu, "Optimization of process synthesis and design problems: A modified differential evolution approach," *Chem. Eng. Sci.*, vol. 61, no. 14, pp. 4707–4721, Jul. 2006.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Dec. 1995, pp. 1942–1948.

[3] X. Ye, B. Chen, L. Jing, B. Zhang, and Y. Liu, "Multi-agent hybrid particle swarm optimization (MAHPSO) for wastewater treatment network planning," *J. Environ. Manage.*, vol. 234, pp. 525–536, Mar. 2019.

[4] F. E. Fernandes, Jr., and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Swarm Evol. Comput.*, vol. 49, pp. 62–74, Sep. 2019.

[5] C. Guan, Z. Zhang, S. Liu, and J. Gong, "Multi-objective particle swarm optimization for multi-workshop facility layout problem," *J. Manuf. Syst.*, vol. 53, pp. 32–48, Oct. 2019.

[6] A. M. Nassef, H. Rezk, M. A. Abdelkareem, A. Alaswad, and A. Olabi, "Application of fuzzy modelling and particle swarm optimization to enhance lipid extraction from microalgae," *Sustain. Energy Technol. Assessments*, vol. 35, pp. 73–79, Oct. 2019.

[7] P. P. Dash and D. Patra, "Mutation based self regulating and self perception particle swarm optimization for efficient object tracking in a video," *Measurement*, vol. 144, pp. 311–327, Oct. 2019.

[8] J. Chen and J. Shi, "A multi-compartment vehicle routing problem with time windows for urban distribution—A comparison study on particle swarm optimization algorithms," *Comput. Ind. Eng.*, vol. 133, pp. 95–106, Jul. 2019.

[9] M. Eslami, H. Shareef, A. Mohamed, and M. Khajehzadeh, "An efficient particle swarm optimization technique with chaotic sequence for optimal tuning and placement of PSS in power systems," *Int. J. Electr. Power Energy Syst.*, vol. 43, no. 1, pp. 1467–1478, Dec. 2012.

[10] J. Kennedy, "Bare bones particle swarms," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2003, pp. 80–87.

[11] R. A. Krohling and E. Mendel, "Bare bones particle swarm optimization with Gaussian or cauchy jumps," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 3285–3291.

[12] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-95-012, 1995.

[13] G. R. Kocis and I. E. Grossmann, "Relaxation strategy for the structural optimization of process flow sheets," *Ind. Eng. Chem. Res.*, vol. 26, no. 9, pp. 1869–1880, Sep. 1987.

[14] L. Costa and P. Oliveira, "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems," *Comput. Chem. Eng.*, vol. 25, nos. 2–3, pp. 257–266, Mar. 2001.

[15] M. F. Cardoso, R. L. Salcedo, S. F. de Azevedo, and D. Barbosa, "A simulated annealing approach to the solution of MINLP problems," *Comput. Chem. Eng.*, vol. 21, no. 12, pp. 1349–1364, Jan. 1997.

[16] G. R. Kocis and I. E. Grossmann, "Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis," *Ind. Eng. Chem. Res.*, vol. 27, no. 8, pp. 1407–1421, Aug. 1988.

[17] G. R. Kocis and I. E. Grossmann, "A modelling and decomposition strategy for the MINLP optimization of process flowsheets," *Comput. Chem. Eng.*, vol. 13, no. 7, pp. 797–819, Jul. 1989.

[18] U. M. Diwekar and E. S. Rubin, "Efficient handling of the implicit constraints problem for the ASPEN MINLP synthesizer," *Ind. Eng. Chem. Res.*, vol. 32, no. 9, pp. 2006–2011, Sep. 1993.

[19] G.-G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 542–555, Feb. 2019.

[20] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Inf. Sci.*, vol. 274, pp. 17–34, Aug. 2014.

[21] D. Gao, G.-G. Wang, and W. Pedrycz, "Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3265–3275, Dec. 2020, doi: 10.1109/TFUZZ.2020.3003506.

[22] D. Zou, S. Li, X. Kong, H. Ouyang, and Z. Li, "Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling," *Energy*, vol. 147, pp. 59–80, Mar. 2018.

[23] X. Shen, D. Zou, N. Duan, and Q. Zhang, "An efficient fitness-based differential evolution algorithm and a constraint handling technique for dynamic economic emission dispatch," *Energy*, vol. 186, Nov. 2019, Art. no. 115801, doi: 10.1016/j.energy.2019.07.131.

[24] Q. Zhang, D. Zou, N. Duan, and X. Shen, "An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem," *Appl. Soft Comput.*, vol. 78, pp. 641–669, May 2019.

[25] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.

CHUANHU CHEN received the B.S. degree from the China University of Mining and Technology, Xuzhou, China, in 1998, and the M.S. degree in communication and information system from Soochow University, Suzhou, China, in 2004. He is currently working with the School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou. He has hosted and participated in more than ten projects of the National Natural Science Foundation of China and the Xuzhou Science and Technology Plan. His current research interests include optimization algorithms and intelligent control.

CHUNLIANG LI is currently a Senior Engineer and EMBA working with Xuzhou Boyuan Technology Company Ltd. His current research interests include enterprise management and industrial dust control and optimization technology research.