Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 1516271, 12 pages
http://dx.doi.org/10.1155/2016/1516271

Research Article

Hybrid Optimization Algorithm of Particle Swarm Optimization and Cuckoo Search for Preventive Maintenance Period Optimization

Jianwen Guo,[1] Zhenzhong Sun,[1] Hong Tang,[2] Xuejun Jia,[2] Song Wang,[3] Xiaohui Yan,[1] Guoliang Ye,[1] and Guohong Wu[4]

[1] *School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China*
[2] *Dongguan Neutron Science Center, Dongguan 523890, China*
[3] *College of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, China*
[4] *Dongguan Hengli Mould Technology Development Limited Company, Dongguan 523460, China*

Correspondence should be addressed to Zhenzhong Sun; sunzz@dgut.edu.cn

Received 4 September 2015; Accepted 17 December 2015

Academic Editor: Elsayed A. Elsayed

All equipment must be maintained during its lifetime to ensure normal operation. Maintenance is one of the critical roles in the success of manufacturing enterprises. This paper proposed a preventive maintenance period optimization model (PMPOM) to find an optimal preventive maintenance period. By making use of the advantages of particle swarm optimization (PSO) and cuckoo search (CS) algorithm, a hybrid optimization algorithm of PSO and CS is proposed to solve the PMPOM problem. The test functions show that the proposed algorithm exhibits more outstanding performance than particle swarm optimization and cuckoo search. Experiment results show that the proposed algorithm has advantages of strong optimization ability and fast convergence speed to solve the PMPOM problem.

1. Introduction

Maintenance refers to restoring aging or faulty equipment parts to a satisfactory operating condition. The primary goal of maintenance is to avoid or mitigate the consequences of failure of equipment. Maintenance is a critical role in the success of manufacturing enterprises [1]. Manufacturing enterprises require cost-effective and adaptive production and maintenance strategies to capture market share [2].

Preventive maintenance (PM) is one of the most popular maintenance policies recommended for the maintenance of manufacturing systems because it is executed with a planned interval aiming at improving machine conditions and preventing unforeseen failures [3]. PM includes a set of activities to improve the overall reliability and availability of a system. PM activities are inspection, testing, diagnosis, disassembly, assembly, cleaning, repair, and replacement. The ideal PM would prevent all equipment failure before it occurs [4].

Preventive maintenance involves a trade-off between the production losses caused by occupying production time and the cost savings achieved by preventing system failure [5]. However, as a result of lack of experience in testing equipment, methods, and maintenance personnel, the accuracy of the PM period is still a key issue [6]. To solve this problem, many PM optimization models have been developed to search the optimal PM period under various conditions [7].

Optimization model is a mathematical model that refers to choosing the best solution from all feasible solutions. Finding an optimal PM period (optimal solution) is difficult because of complexity of the optimization model. Recently, many metaheuristic algorithms provide new solutions to various complex optimization problems by imitating the self-organization mechanism of natural biological communities and the adaptive ability of evolution [8]. Metaheuristic algorithms also have been proposed to solve the combinatorial explosion problem of PM optimization models recently [9].

Particle swarm optimization (PSO) [10] is a metaheuristic algorithm inspired by the social behavior of populations with collaborative properties. The PSO imitates this species collaboration and is widely used in solving mathematical optimization problems. PSO exhibits easy understanding, simple operation, and rapid searching. It has been successfully applied to several fields [11]. Cuckoo search (CS) [12] is a metaheuristic optimization algorithm inspired by the reproduction strategy of cuckoo species. Cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover strange eggs in its nest, and it will either destroy the eggs or abandon the nest to build a new one. This algorithm is enhanced by the so-called Lévy flights rather than by simple isotropic random walks. The effectiveness of CS over other methods such as GA and PSO has been validated on benchmarked functions [12].

PSO has several advantages, such as fast convergence speed, but it also has some defects, such as premature convergence, and it easily falls into local optima. CS has several advantages, such as few control parameters and high efficiency, but it also has some defects, such as slow convergence speed and low accuracy. A PSO and CS hybrid algorithm should be developed as a hybrid algorithm with an outstanding performance because of the complementation of PSO and CS.

In this paper, a preventive maintenance period optimization model (PMPOM) is proposed. PMPOM takes the cost of shutdown caused by breakdown maintenance, preventive maintenance, and inspection maintenance as evaluation indexes. PSO and CS hybrid optimization algorithm PSOCS is proposed to solve PMPOM. The test functions and application examples show that the proposed algorithm has advantages of strong optimization ability and fast convergence speed that can effectively solve the PMPOM.

The remainder of the paper is organized as follows. Section 2 introduces related works. Section 3 introduces PMPOM. Section 4 introduces PSOCS hybrid algorithm and carries out algorithm performance test. Section 5 uses PSOCS algorithm to solve PMPOM problem. Section 6 provides the conclusions and further works of the study.

2. Related Works

2.1. Particle Swarm Optimization. PSO is a population-based metaheuristic algorithm that is inspired by the social behavior of populations with collaborative properties. The PSO imitates this species collaboration and is widely used in solving mathematical optimization problems. The flow of PSO is shown in Figure 1.

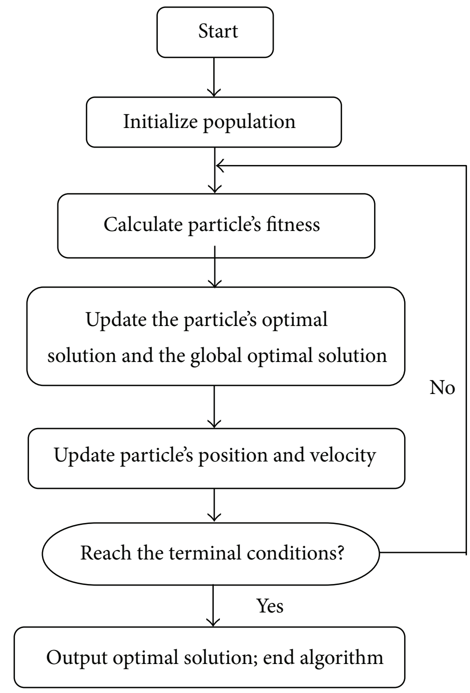

Figure 1: Flowchart of PSO.

Population in PSO is represented as $X = \{X_1, X_2, \ldots, X_N\}$, where X_i is a particle that moves within a multidimensional search space and strives for the optimal solution. A particle's property includes position and velocity. The position of a particle is a solution candidate. The velocity of a particle is the information about direction and varying rate. A particle that moves within a d dimensional search space is represented as $X_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$. A particle can adjust its position x_{id} toward the best positions according to its experience (its best position p_{id}) and that of their neighboring particles (the best position of population p_{gd}). In this manner, all particles are expected to gradually approach the global optimum.

Particle's velocity is updated by using

$$\begin{aligned} v_{id}(k+1) = {} & \omega v_{id}(k) + \eta_1 r_1 \left(p_{id} - x_{id}(k)\right) \\ & + \eta_2 r_2 \left(p_{gd} - x_{id}(k)\right), \end{aligned} \tag{1}$$

where $v_{id}(k+1)$ is the dth component of particle's velocity after the $(k+1)$th update; p_{id} is currently the particle's best solution of dth component after the kth update; p_{gd} is currently the population's best global solution of the dth component after the kth update; η_1 and η_2 are positive constant parameters called acceleration coefficients, controlling the movement steps of particles; ω is inertia weight that controls the effect of previous values of particle's velocity on next one. r_1 and r_2 are random variables with a range $[0, 1]$.

To avoid the particle's position beyond the search space, maximum search velocity $v_{\max}$ is introduced. If $v_{id}(k+1) > v_{\max}$, $v_{id}(k+1) = v_{\max}$, and if $v_{id}(k+1) < -v_{\max}$, then $v_{id}(k+1) = -v_{\max}$.

Particle position is updated by using

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1), \tag{2}$$

where $x_{id}(k+1)$ is the dth component of particle's position after the $(k+1)$th update and $v_{id}(k+1)$ is the dth component of particle's velocity after the $(k+1)$th update.

The update procedure consecutively iterates until a predetermined terminal condition is reached. Thereby, the best solution is obtained. Using formulas (1) and (2), three factors have a major effect on a particle's update speed: (1) the distance between particle's current position and particle's best solution, (2) the distance between particle's current position and population's best global solution, and (3) the speed before this iteration. The importance of the three factors is determined by weight coefficients η_1, η_2, and ω, respectively.

2.2. Cuckoo Search. CS is a new and efficient population-based heuristic evolutionary algorithm for solving optimization problems. CS has the advantages of simple implementation and few control parameters. This algorithm is based on the obligate brood parasitic behavior of some cuckoo species combined with the Lévy flight behavior of some birds and fruit flies. It has been applied to solve a wide range of real-world optimization problems, such as structural optimization problem [13], shop scheduling problem [14, 15], nonconvex economic dispatch problem [16], and short-term hydrothermal scheduling problem [17].

Below are the approximation rules during the search process [18].

(1) Each cuckoo lays one egg (solution) at a time and dumps its egg in a randomly chosen nest.

(2) The best nests with a high-quality egg (better solution) will be carried over to the next generation.

(3) The number of available host nests is fixed. A host bird can discover an alien egg with a probability P_a $[0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest and build a completely new nest.

From the implementation point of view, we can say that each egg in a nest represents a solution, and each cuckoo can lay only one egg (thus representing one solution). In this case, no distinction exists among an egg, a nest, or a cuckoo, because each nest corresponds to one egg, which also represents one cuckoo.

In CS, each nest's position or egg's position can be regarded as a solution, because each nest corresponds to one egg. In the initial process, each solution is generated randomly when generating the ith solution in the $(i + 1)$th generation. Position is updated by using

$$x_i(k+1) = x_i(k) + \alpha \otimes L(\lambda), \tag{3}$$

where $x_i(k + 1)$ is the nest's ith position of the $k + 1$ generation in the population; $x_i(k)$ is the nest's ith position of the k generation in the population; α is a real number that denotes the step size, which should be proportional to the scales of the optimization problem; $\otimes$ represents entry-wise multiplications; and $L(\lambda)$ is the random search vector produced by Lévy distribution.

The use of Lévy flights [19] for local and global searching is a vital part of CS [12]:

$$L(\lambda) \sim u = t^{-\lambda} \quad (1 < \lambda < 3). \tag{4}$$

Here, the consecutive jumps/steps of a cuckoo essentially form a random walk process, which obeys a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by Lévy walk around the best solution, which will accelerate the local search. However, a substantial fraction of the new solutions should be generated by far field randomization whose locations should be far enough from the current best solution. This approach will ensure the system will not be trapped in a local optimum.

CS updates each generation solution by Lévy flight, and a better solution is retained. Then, the retained solution is eliminated randomly by discover probability P_a and is updated. The above process is repeated until the algorithm ends. The steps of the cuckoo algorithm are as follows.

Step 1. The parameters, number of the bird's nest, termination condition of algorithm, search space dimension of algorithm d, step size α, and P_a are set.

Step 2. The population (randomly determining the position of the nest) $p^{(0)} = [x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}]^T$ is randomly initialized.

Step 3. The initial population fitness value is calculated by using objective function, and the optimal position of the nest $x_{gb}^{(0)}$ is determined.

Step 4. The position of the nest is updated by using formula (16), and a new position is created. Then, the fitness value is calculated by using the objective function. A better nest is chosen between the new and the old nests as an individual in the population. A new group of nest's position is as follows:

$$g_{k+1} = [x_1(k+1), x_2(k+1), \ldots, x_n(k+1)]^T. \tag{5}$$

Step 5. Randomize elimination mechanism. The parameter P_a reflects the probability whether the nest will be abandoned or be updated. Thus, for the eggs that may be found by the host, the location of the nest should be changed. Initially, an n-dimensional vector $R_n = [r_1, r_2, \ldots, r_n]$ is produced. For r_i in R_n, each component of a vector follows a uniform distribution with $[0, 1]$ when $r_i > P_a$, randomly changing the ith nest position. Through a comparison between the fitness values of the old and new nests, the better nest will be chosen as a new generation of individuals in the population. The new group of nest's position is produced as $p_{k+1} = [x_1(k+1), x_2(k+1), \ldots, x_n(k+1)]^T$.

Step 6. The best nest position is updated by using p_{k+1}.

Step 7. If the termination condition of the algorithm is satisfied, the optimal output position of nest is achieved, and the algorithm is terminated; otherwise, Step 4 is performed.

2.3. Preventive Maintenance. PM is one of the most important strategies for equipment maintenance and has been a concern of most scholars. In the past years, many period optimization models have been established. Chareonsuk et al. [20] established the PMPOM to achieve the target with minimum operation costs and then studied the effect of equipment operation cost and reliability by different maintenance period. Vaurio [21] established an optimization model to achieve a target with efficiency and low operating costs, and to study the effect of different maintenance period of the model. Jiang and Ji [22] treated maintenance period optimization as a multiobjective optimization problem, in which the equipment operation cost, equipment life, effectivity, and reliability are the optimization objective. Kalir [23] established preventive maintenance for semiconductor manufacturing.

PM optimization problem can be treated as a constrained optimization problem. Recently, several new metaheuristic algorithms have been implemented to solve the problem. Samrout et al. [24] presented an ant colony to optimize the maintenance periods. Moghaddam et al. [25] presented a new multiobjective optimization model to determine the optimal preventive maintenance and replacement schedules in a repairable and maintainable multicomponent system. Yare et al. [26] introduced multiple swarm concepts for the modified discrete PSO to form a robust algorithm for solving the preventive maintenance schedule problem. Abdulwhab et al. [27] used the GA optimization technique to maximize the overall system reliability for a specified future time period, in which a number of generating units are to be removed from service for preventive maintenance. Berrichi et al. [28] presented an algorithm based on the ant colony optimization paradigm to solve the joint production and maintenance scheduling problem. Verma and Ramesh [29] viewed the initial scheduling of preventive maintenance as a constrained nonlinear multiobjective decision making problem. The optimization problem is solved using an elitist GA, and maintenance domain knowledge is effectively incorporated in its implementation.

Each optimization algorithm has its own advantages and disadvantages. Thus, many hybrid optimization algorithms have been developed to solve the optimization problem. Lin and Wang [30] presented a hybrid genetic algorithm to optimize the periodic preventive maintenance model in a series-parallel system. Leou [31] proposed a novel algorithm for determining a maintenance schedule for a power plant. This algorithm combines the GA with simulated annealing to optimize maintenance periods and minimize maintenance and operational cost. Ma et al. [32] proposed a hybrid swarm intelligence algorithm to optimize the preventive maintenance period. Kim and Woo [33] presented a methodology for optimal maintenance scheduling of generating units using a hybrid algorithm that combines a scatter search and a GA. Samuel and Rajan [34] presented a hybrid PSO-based GA and hybrid PSO-based shuffled frog leaping algorithm to solve the long-term generation maintenance scheduling problem.

3. Preventive Maintenance Period Optimization Model

Suppose that equipment is newest at time 0. $[0, t]$ is the limited time zone in which equipment is running nonstop; $[0, t]$ is divided into n preventive maintenance period, in which each time span of the preventive maintenance period is not necessarily equal. t_i is the ith preventive maintenance period, where $i \in \{1, 2, \ldots, n\}$.

Minor maintenance [35] is a strategy used in case of equipment failure during the preventive maintenance period t_i. This strategy cannot change equipment failure rate and reliability. Minor maintenance time is very small and can be neglected, unlike t_i.

The objective function of PM is as follows:

$$\min \quad f = \sum_{i=1}^{n} C_r F_i + \sum_{i=1}^{n} C_p(\alpha_i, t_i) + \sum_{i=1}^{n} C_l \theta_i, \tag{6}$$

where $\sum_{i=1}^{n} C_r F_i$ is the cost of equipment shutdown caused by breakdown maintenance, C_r is the average maintenance cost of the equipment after the failure occurs, F_i is failure probability functions within t_i, $\sum_{i=1}^{n} C_p(\alpha_i, t_i)$ is the cost of equipment shutdown caused by preventive maintenance, $C_p(\alpha_i, t_i)$ is preventive maintenance cost function within t_i, α_i is age reduction factor [36], $\sum_{i=1}^{n} C_l \theta_i$ is the cost of equipment shutdown caused by inspection and maintenance, C_l is the cost of equipment shutdown caused by overhaul in unit time, and θ_i is the time of preventive maintenance in t_i.

In this paper, failure probability function F_i follows the Weibull distribution. Its probability density function is calculated by using

$$f(t) = \frac{m}{\eta} \left(\frac{t}{\eta}\right)^{m-1} \exp\left[-\left(\frac{t^m}{\eta}\right)\right], \tag{7}$$

where η and m are the scale parameter and shape parameter of the inherent attribute of the equipment, respectively. The value of η and m can be calculated by using statistics and analysis of historical fault data.

Failure probability is calculated by using

$$\lambda(t) = \frac{m}{\eta} \left(\frac{t}{\eta}\right)^{m-1}. \tag{8}$$

Failure probability within T_i is calculated by using

$$F_i = \int_0^{T_i} \lambda_i(t)\, dt, \tag{9}$$

where $\lambda_i(t)$ is failure probability within t_i and is a function of age reduction factor α_i. The mathematical expression of $\lambda_i(t)$ can be obtained as follows:

$$\begin{aligned} \lambda_1(t) &= \lambda(t), \\ \lambda_2(t) &= \lambda(t + t_1 - \alpha_1 t_1), \\ \lambda_3(t) &= \lambda(t + t_2 - \alpha_2 t_2). \end{aligned} \tag{10}$$

And then

$$\begin{aligned} \lambda_i(t) &= \lambda(t + t_{i-1} - \alpha_{i-1} t_{i-1}) \\ &= \lambda\left(t + \sum_{k=1}^{i-1} t_k - \sum_{k=1}^{i-1} \alpha_k t_k\right) \\ &= \lambda\left(t + \sum_{k=1}^{i-1} (1 - \alpha_k) t_k\right), \quad i = 1, 2, \ldots, n. \end{aligned} \tag{11}$$

Therefore

$$F_i = \int_0^{t_i} \lambda_i(t)\, dt = \int_0^{t_i} \lambda\left(t + \sum_{k=1}^{i-1} (1 - \alpha_k) t_k\right) dt, \qquad i = 1, 2, \ldots, n. \tag{12}$$

A finite time PMPOM as formula (13) is the cost of equipment shutdown caused by breakdown maintenance, preventive maintenance, and inspection:

$$\min \quad C_r \sum_{i=1}^{n} \int_0^{t_i} \lambda \left(t + \sum_{k=1}^{i-1} (1-\alpha_k)\, t_k \right) dt + \sum_{i=1}^{n} C_p(\alpha_i, t_i) + \sum_{i=1}^{n} C_i \theta_i \tag{13}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} t_i + \sum_{i=1}^{n} \theta_i < T, \quad i = 1, 2, \ldots, n \tag{14}$$

$$T_l \le \theta_i, \quad i = 1, 2, \ldots, n, \tag{15}$$

where T_l is the minimum time for the preventive maintenance.

The optimization objective is the minimized maintenance and production costs caused by downtime. The constraint is preventive maintenance operation time. The number of decision variables $(2n+1)$ and the mathematical expressions (14) and (15) of the constraint conditions are also dynamic. Making the optimization problem is more complex than making a general multiobjective nonlinear optimization problem because of the effect of dynamic change. Therefore, using a high-efficiency method is necessary to solve this problem.

4. PSO-CS Hybrid Algorithm

4.1. PSOCS Hybrid Algorithm Model. The PSO has advantages such as easy understanding, simple operation, and rapid searching. However, in solving a large complex problem, PSO becomes easily trapped in local optimum. This weakness must be overcome to extend the practicability of PSO. CS has advantages such as few control parameters and high efficiency, but it also has some defects, such as slow convergence speed and low accuracy. In CS, high randomness of the Lévy flight makes the search process quickly jump from one area to another area. Thus, the global search ability of the algorithm is very strong. However, given the high randomness of the Lévy flight, the algorithm initiates a blind search process, convergence speed becomes slow, and the searching efficiency is significantly reduced close to the optimal solution.

To improve the performance of CS, PSO is introduced in the update process of CS. Thus, a PSOCS hybrid algorithm is developed. PSOCS first uses Lévy flights in the search space to search, and then it uses the position of the PSO update mode to accelerate the particles to the optimal solution convergence. At the same time, the random elimination mechanism of CS can successfully escape local optima, thereby improving the performance of searching for the optimal solution.

Algorithm terms are defined as follows.

(1) Population and Population Size (sizepop). The population is composed of a certain number of individuals; the total number of individuals is the population size, with sizepop.

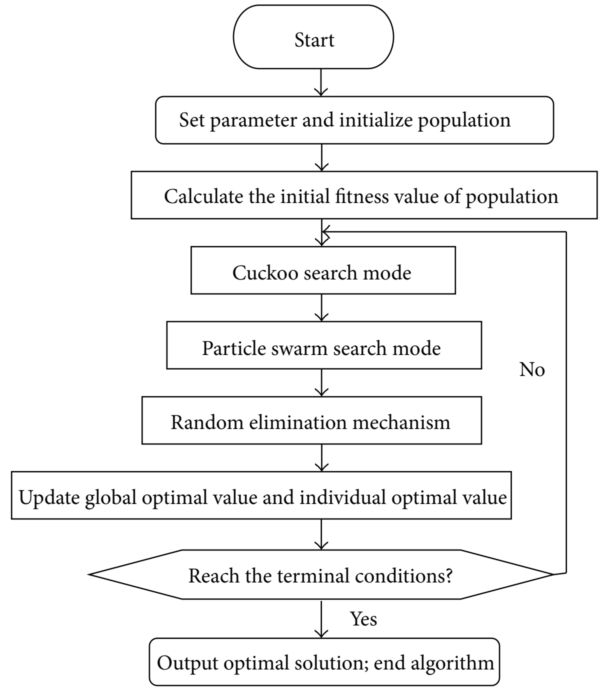

Figure 2: Flowchart of PSOCS.

(2) Fitness. Fitness is an index of individual quality. In general, a large fitness value corresponds to a good result, and vice versa.

(3) Search Space Upper Bound (Ub) and Search Space Lower Bound (Lb). Ub and Lb are the upper bound and lower bound, respectively, of the search space for the optimization problem.

(4) Maximum Search Velocity ($v_{\max}$) and Minimum Search Velocity ($v_{\min}$). Speed is limited as the algorithm performs a search. Consider $v_{\max} = a * \text{Ub}$, where a is the adjustment coefficient in the range of (0, 1). Consider $v_{\min} = b * \text{Lb}$, where b is also the adjustment coefficient in the range of (0, 1).

(5) PSO Search Mode. In this mode, an individual updates its position and velocity by using the process of PSO.

(6) Cuckoo Search Mode. An individual updates its position by using the process of CS. An individual in CS has no speed and velocity updating formula, whereas an individual in PSO search mode has both position and velocity. Individual velocity in the cuckoo search mode is not updated, and the current velocity of the individual is the velocity updated by PSO search mode.

(7) Discovery Probability. Through the random elimination mechanism in cuckoo search mode, the host has probability P_a of finding foreign eggs.

A flowchart of PSOCS is shown in Figure 2. Its procedure is as follows.

Step 1. The parameter is set, and the population is initialized. The parameters sizepop, run, Ub, Lb, a, b, P_a, η_1, η_2, $\omega_{\max}$,

TABLE 1: Test functions.

ID	Function expression	Search range	Theoretical optimal value
1	$f_1(x) = \sum_{i=1}^{10} x_i^2$	$[-100, 100]$	$\min(f_1(x)) = 0$
2	$f_2(x) = \sum_{i=1}^{10} \lvert x_i\rvert + \prod_{i=1}^{10} \lvert x_i\rvert$	$[-100, 100]$	$\min(f_2(x)) = 0$
3	$f_3(x) = \sum_{i=1}^{10}\left(\sum_{j=1}^{i} x_i\right)^2$	$[-100, 100]$	$\min(f_3(x)) = 0$
4	$f_4(x) = 0.5 + \dfrac{\sin^2\sqrt{x_1^2+x_2^2} - 0.5}{[1+0.001(x_1^2+x_2^2)]^2}$	$[-5, 5]$	$\min(f_4(x)) = 0$

and $\omega_{\min}$ are set. Population is initialized randomly, which includes initialization position p and velocity v of individual.

Step 2. The initial fitness value of the population is calculated by using the objective function, and the fitness value and position of the global optimal individual are determined.

Step 3. Cuckoo search mode is initiated. The position of the individual in Lévy flight search is updated by using formula (16), and a new individual is produced. The fitness values of new and old individuals are compared; the better result is selected as a new-generation individual.

Step 4. PSO search mode is initiated. The position and velocity of the individual are updated, and then a new individual is produced. The position is updated by using formula (1), and the velocity is updated by using formula (2). Before updating the velocity, the inertia weight coefficient needs to be updated by using

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * \frac{\text{iter}}{\text{run}}, \tag{16}$$

where iter and run are the current iteration times and maximum iteration times of the algorithm, respectively. $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum inertia weights, respectively. In a comparison of the fitness values of new and old individuals, the one with the better result is selected as a new individual, and the global optimal individual is updated.

Step 5. An n-dimensional vector $R_n = [r_1, r_2, \ldots, r_n]$ is produced, and r_i obeys a uniform distribution with $[0, 1]$. When $r_i > P_a$, a new individual is randomly produced by using formula (17). In a comparison between the fitness values of the old and new nests, the better one will be selected as a new generation of individuals in the population:

$$\text{Temp}_i = \text{Lb} + (\text{Ub} - \text{Lb}) * \text{rand}\,n(1, d). \tag{17}$$

Step 6. The global and individual optimal values are updated. The optimal positions of all the individuals and whole populations are updated.

Step 7. If the end condition of the algorithm is satisfactory, then the optimal position of the nest is outputted, and the algorithm is terminated; otherwise, Step 3 is performed.

4.2. Algorithms Test. In this paper, four test functions [37] are chosen to test the performance of the algorithm, and the results are compared with those of PSO and CS to verify the performance of the algorithm. The selected four test functions are shown in Table 1. The environment of the simulation experiment is as follows: CPU is an Intel Core i5-3470 @ 3.20 GHz with 4 GB of RAM, the computer system platform is a Windows 7 32-bit operating system, and the program is written in Matlab.

The basic parameters of the algorithm are as follows: population size is 25, the maximum number of iterations of function $f_1(x)$ is 1000, the maximum number of iterations of function $f_2(x)$ is 1000, the maximum number of iterations of function $f_3(x)$ is 2000, the maximum number of iterations of function $f_4(x)$ is 5000, $\omega_{\max} = 0.7$, $\omega_{\min} = 0.4$, $\eta_1 = 0.2$, $\eta_2 = 0.5$, $\omega_{\max} = 0.3 * \text{Ub}$, and $\omega_{\min} = 0.3 * \text{Lb}$.

The tests use PSO, CS, and PSOCS to optimize the functions in Table 1, and every function is optimized by repeating the tests 50 times. The average optimization results are shown in Table 2. After optimization was repeated 50 times, the minimum fitness value is the best value (optimal individual of algorithm), the maximum fitness value is the worst value, and the average fitness value is the average value. Fitness value square deviation is the mean square deviation that was obtained after optimization was performed 50 times and is shown in

$$\sigma = \sqrt{\frac{1}{50}\sum_{i=1}^{50}(s_i - \bar{s})^2}, \tag{18}$$

where s_i denotes the results after 50 optimization iterations and $\bar{s}$ is the average value.

Table 2 shows that for the test functions $f_1(x)$, $f_2(x)$, $f_3(x)$, and $f_4(x)$, the global optimal solution found by PSO and CSO is not ideal with the current population size and number of iterations. To obtain better results, the size of the larger population needs to be set, and more iteration is needed. The global optimal solution found by PSOCS is infinitely close to the theoretical optimum of the test function.

Table 2: Average fitness value of optimization results for PSO, CS, and PSOCS.

Test function	Algorithms	Minimum fitness value	Maximum fitness value	Average fitness value	Fitness value square deviation
$f_1(x)$	CS	4.1797×10^3	2.2880×10^4	1.5607×10^4	3.7549×10^3
	PSO	5.3129×10^3	2.3924×10^4	1.5416×10^4	4.1414×10^3
	PSOCS	4.4090×10^{-24}	1.2366×10^{-20}	2.0111×10^{-21}	3.0248×10^{-21}
$f_2(x)$	CS	433.4732	722.1931	555.8759	62.9874
	PSO	7.0584×10^9	2.9650×10^{14}	2.5924×10^{13}	5.3873×10^{13}
	PSOCS	1.6394×10^{-14}	1.7799×10^{-11}	6.5228×10^{-13}	2.5319×10^{-12}
$f_3(x)$	CS	4.2811×10^3	2.0117×10^4	1.5079×10^4	3.7231×10^3
	PSO	9.6441×10^3	4.2375×10^4	2.1081×10^4	7.4645×10^3
	PSOCS	1.4565×10^{-13}	1.2702×10^{-8}	9.2612×10^{-10}	2.2172×10^{-9}
$f_4(x)$	CS	1.3793×10^{-5}	0.0097	0.0020	0.0022
	PSO	4.8710×10^{-5}	0.0107	0.0084	0.0033
	PSOCS	0	0.0097	0.0019	0.0033

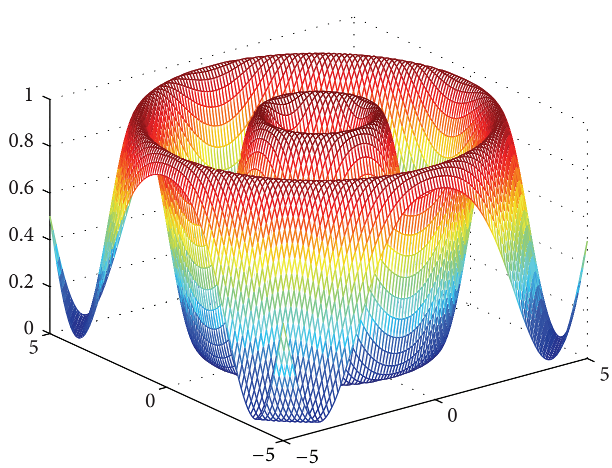

Figure 3: Value distribution map of $f_4(x)$.

The optimization results of $f_4(x)$ indicate poor optimization performance of PSO, CSO, and PSOCS under current parameter settings. The value distribution map of $f_4(x)$ in search space is shown in Figure 3. The figure shows that many local optimum points exist, and being trapped in local optima is easy. The average fitness value and fitness value square deviation are the two key indicators of the stability of the algorithm. Table 2 shows that the average fitness value of the test functions is close to the minimum fitness value, and fitness value square deviation is within the ideal range. Therefore, the algorithm is more stable than PSO and CSO.

4.3. Algorithm Performance Comparison. With the average fitness value, average optimal fitness value, and average standard deviation fitness value taken as the evaluating indicators, this paper tests PSOCS performance with changes in iteration times. The average fitness value is the average fitness value of results obtained after 50 times of optimization. The average optimal fitness value is the average optimal value obtained after 50 times of optimization. The average standard deviation fitness value is the average standard deviation fitness value obtained after 50 times of optimization. The change trends of the three evaluating indicators with the increase of iteration times are shown in Figures 4, 5, and 6, respectively. Charts (a), (b), (c), and (d) in Figures 4, 5, and 6 show the performance of the evaluation indicators in the test functions $f_1(x)$, $f_2(x)$, $f_3(x)$, and $f_4(x)$, respectively.

It can be seen from Figure 4 that the optimization ability of PSOCS in the four test functions is better than that of CS and PSO, and the optimal solution is infinitely close to the theoretical optimal solution.

The performance of CS is different from the test function. It can find the optimal solution in the test function $f_2(x)$ quickly. However, the performance of the other three test functions is not satisfactory, and more iteration is needed to find a satisfactory result.

The optimization ability of PSO is the worst of the three algorithms on the test functions. The result that was obtained after 1000 iterations is almost the same as the optimal values of the initial population. As the number of iterations increase, PSO is unable to effectively improve the quality of the optimization results.

The stability and convergence rate of the algorithm can be tested based on the average fitness value and average standard deviation fitness value with the increased number of iterations. Figures 4, 5, and 6 show that PSO shows good stability and convergence rate in $f_2(x)$, but not in the other test functions. PSOCS has a fast convergence speed and is close to 0 with fewer iterations. The performance of CS in the test functions is between that of PSO and PSO.

The above analysis indicates that the performance of PSOCS is improved greatly by integrating CS and PSO. Moreover, the stability, convergence speed, and optimization ability of PSOCS are better than those of CS and PSO. Furthermore, PSOCS requires few iterations to search for the optimal results. PSOCS is efficient, stable, and fast and can effectively solve the problem of continuous space optimization. Therefore, it can also be used to solve the problem of the optimization of the preventive maintenance period.

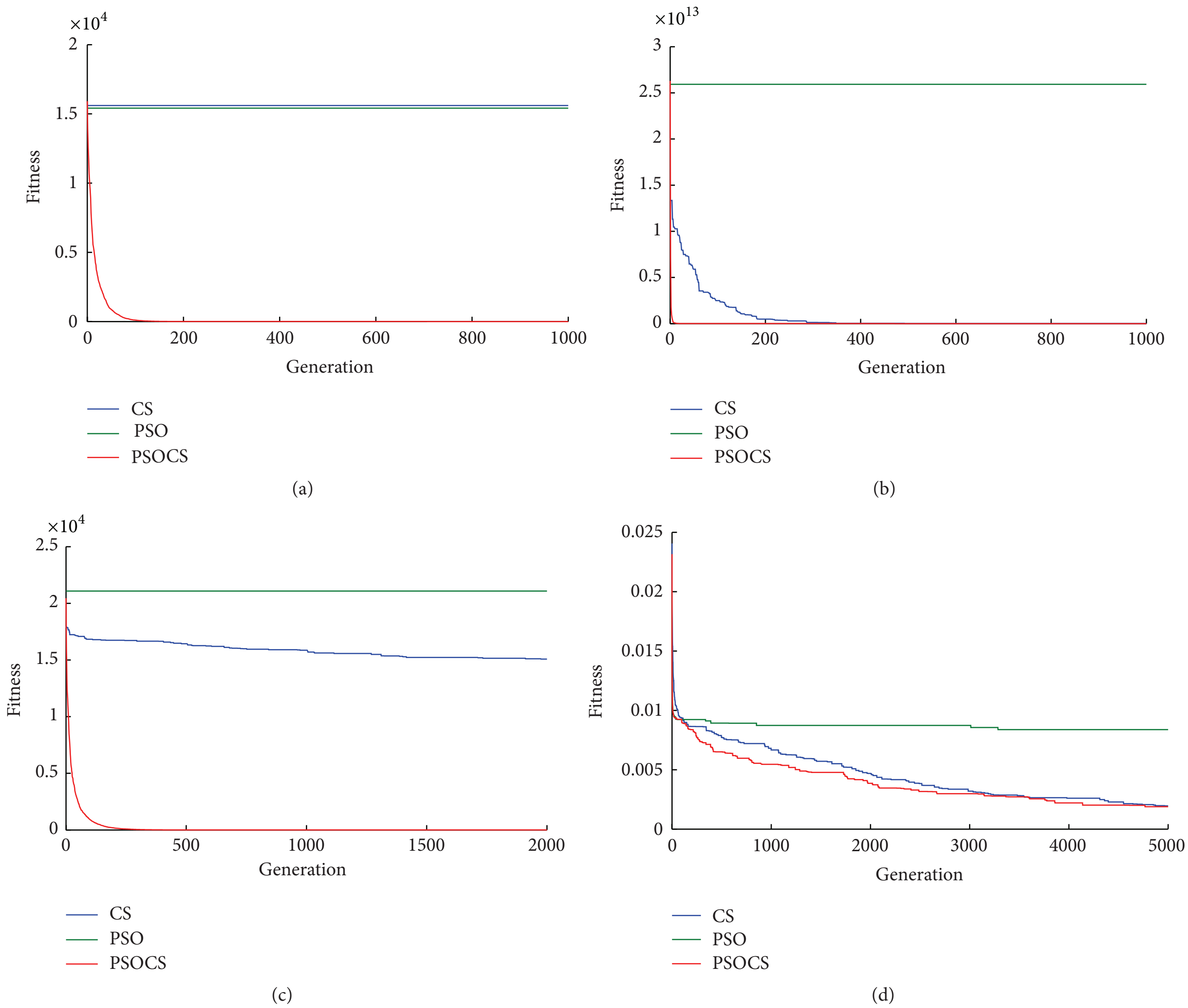

FIGURE 4: Change trend of average optimal fitness value.

TABLE 3: Parameter setting.

Parameters	T	C_l	C_r	T_l	$C_p(\alpha_i, t_i)$	α_i, $i = 1, 2, \ldots, n$
Value	1000 h	2000	560	3 h	$200 + 25\alpha_i t_i$	0.95

5. PSOCS for PMPOM

We use PSOCS to solve the preventive maintenance period optimization problem. Historical fault data analysis shows that the equipment fault time approximately follows the Weibull distribution with the parameters $m = 2$ and $\eta = 50$. The rest of the parameters are shown in Table 3. With the above data and (9), (12), and (13), the constrained preventive equipment within the $[0, 1000]$ maintenance model can be obtained.

The PSOCS parameters are as follows: sizepop = 25, run = 300, $\omega_{\max} = 0.7$, $\omega_{\min} = 0.4$, $\eta_1 = 0.2$, $\eta_2 = 0.5$, $P_a = 0.25$, search range is $[0, 1000]$, and speed range is $[-300, 300]$.

TABLE 4: Optimization results.

PM times	Cost	PM operation time θ_i
0	224000	—
1	168340	3
2	137620	3
3	111640	3
4	103570	3
5	102820	3
6	106320	3
7	110490	3

The optimization results are shown in Table 4. The curve of the optimization results is shown in Figure 7. The above optimization results indicate that the minimum total cost is

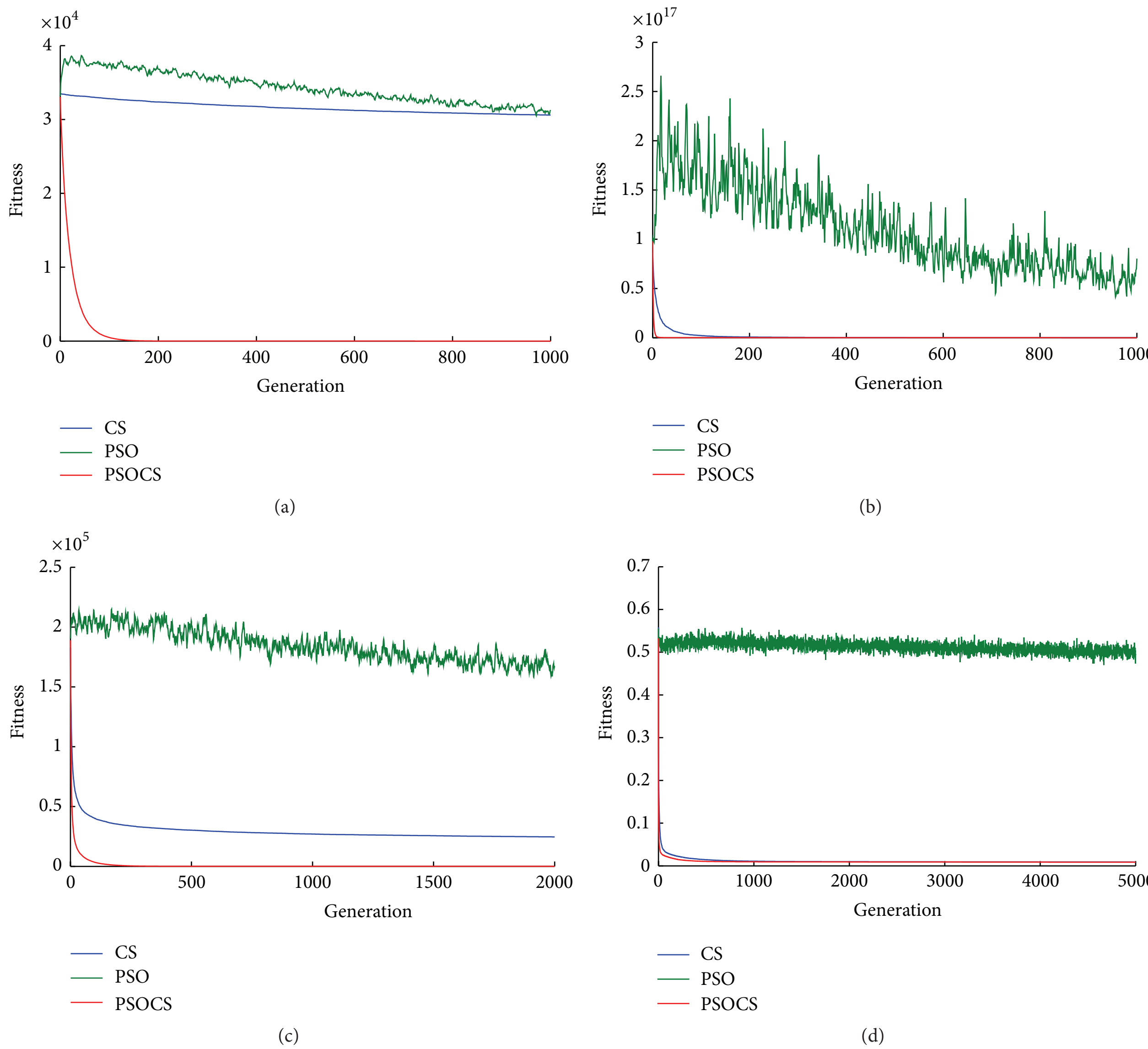

Figure 5: Change trend of average fitness value.

102820 and the four preventive maintenance period *gBest* are 185.98, 216.63, 225.62, and 212.05.

When the maintenance time is 4, PSOCS is applied to solve the problem. The average fitness and optimal fitness of the population that changes with iteration times are shown in Figure 8. The standard deviation of population fitness that changes with iteration times is shown in Figure 9.

Figure 8 shows that PSOCS can search the optimal solution in fewer than 100 iterations. The curves of the average and optimal fitness almost coincide, which shows that the algorithm has converged. Figure 9 shows that the standard deviation of the population fitness declined rapidly and reaches down to 0. Figure 10 shows the optimal fitness of the population (sizepop = 25). From Figure 10, the optimal fitness of each individual is 102820. This result indicates that the algorithm has a fast convergence rate and good stability. The test proves that PSOCS can effectively solve the maintenance period optimization problem based on the optimization model proposed in this paper.

6. Conclusions and Further Works

In this paper, a PSOCS hybrid algorithm is developed to solve a finite time PMPOM in which the costs of equipment shutdown caused by breakdown maintenance, preventive maintenance, and inspection are used as evaluation indexes. Compared with PSO and CS, PSOCS has the advantages of fast convergence speed, strong searching ability, and the ability to solve the problem of multidimensional continuous space optimization by using test functions. A test example shows that PSOCS can effectively solve the maintenance period optimization problem based on the proposed optimization

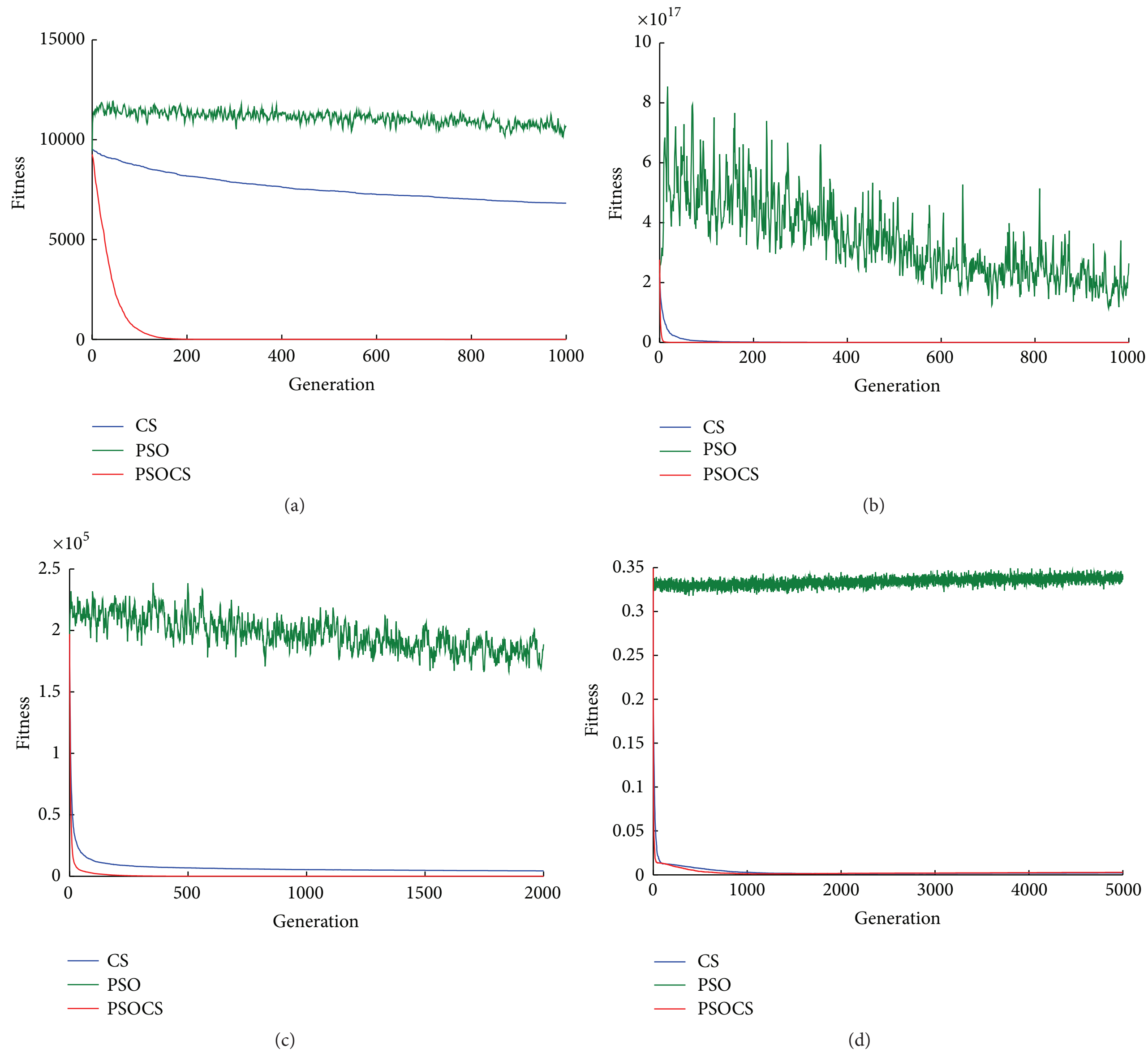

Figure 6: Change trend of average standard deviation fitness value.

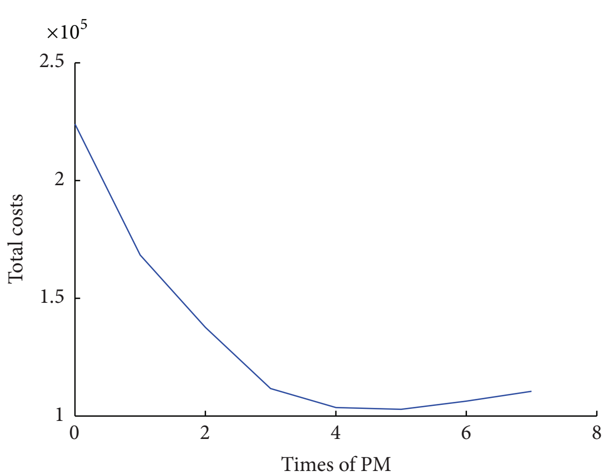

Figure 7: Optimization results of test.

model. Furthermore, we established the PMPOM based on economic and reliability indicators. This model can be considered a multiobjective optimization problem, which is addressed by using PSOCS.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The study was supported by the National Natural Science Foundation of China (Grant no. 71201026), the Natural Science Foundation of Guangdong (no. 2015A030310274, no. 2015A030310415, and no. 2015A030310315), the Project of

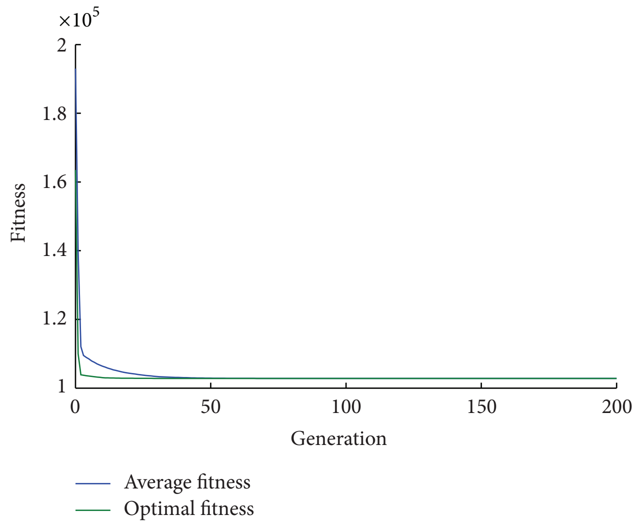

Figure 8: Average fitness and optimal fitness of the population changing with iteration times.

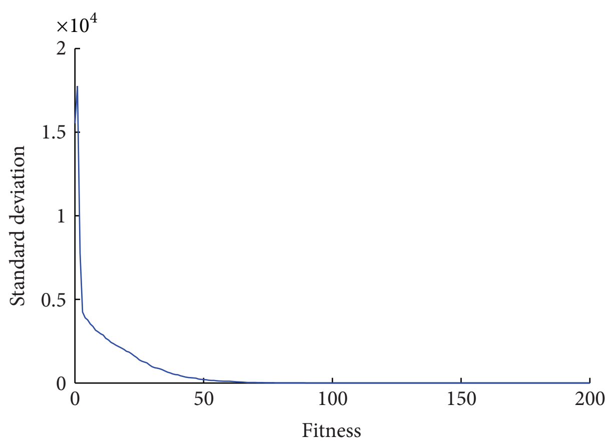

Figure 9: Standard deviation of the population fitness.

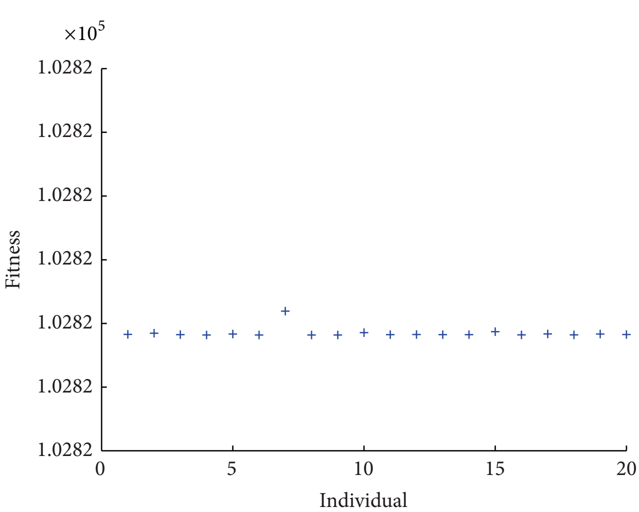

Figure 10: The optimal fitness of the population (sizepop = 25).

Department of Education of Guangdong Province (no. 2013KJCX0179, no. 2014KTSCX184, and no. 2014KGJHZ014), the Development Program for Excellent Young Teachers in Higher Education Institutions of Guangdong Province (no. Yq2013156), the Dongguan Universities and Scientific Research Institutions Science and Technology Project (no. 2014106101007), and the Dongguan Social Science and Technology Development Project (no. 2013108101011).

References

[1] S. Takata, F. J. A. M. Van Houten, E. Westkämper et al., "Maintenance: changing role in life cycle management," *CIRP Annals—Manufacturing Technology*, vol. 53, no. 2, pp. 643–655, 2004.

[2] X.-M. Jin and J. Ni, "Joint production and preventive maintenance strategy for manufacturing systems with stochastic demand," *Journal of Manufacturing Science and Engineering*, vol. 135, no. 3, Article ID 031016, 2013.

[3] A. N. Das and S. P. Sarmah, "Preventive replacement models: an overview and their application in process industries," *European Journal of Industrial Engineering*, vol. 4, no. 3, pp. 280–307, 2010.

[4] J. Levitt, *Complete Guide to Predictive and Preventive Maintenance*, Industrial Press, 2013.

[5] K.-S. Moghaddam and J.-S. Usher, "Sensitivity analysis and comparison of algorithms in preventive maintenance and replacement scheduling optimization models," *Computers & Industrial Engineering*, vol. 61, no. 1, pp. 64–75, 2011.

[6] R. Yang, J. Kang, and Z. Quan, "An enhanced preventive maintenance optimization model based on a three-stage failure process," *Science and Technology of Nuclear Installations*, vol. 2015, Article ID 193075, 13 pages, 2015.

[7] M.-B. Biggs, B. Christianson, and M. J. Zuo, "Optimizing preventive maintenance models," *Computational Optimization and Applications*, vol. 35, no. 2, pp. 261–279, 2006.

[8] X.-J. Yu and M.-S. Gen, *Introduction to Evolutionary Algorithms*, Springer, Berlin, Germany, 2010.

[9] S.-H. Ding and S. Kamaruddin, "Maintenance policy optimization—literature review and directions," *International Journal of Advanced Manufacturing Technology*, vol. 76, no. 5–8, pp. 1263–1283, 2015.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, December 1995.

[11] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: an overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[12] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing*, pp. 210–214, IEEE, Coimbatore, India, December 2009.

[13] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[14] M. K. Marichelvam, T. Prabaharan, and X. S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Applied Soft Computing Journal*, vol. 19, pp. 93–101, 2014.

[15] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," *International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5-8, pp. 951–959, 2013.

[16] D. N. Vo, P. Schegner, and W. Ongsakul, "Cuckoo search algorithm for non-convex economic dispatch," *IET Generation, Transmission & Distribution*, vol. 7, no. 6, pp. 645–654, 2013.

[17] T. T. Nguyen and D. N. Vo, "Modified cuckoo search algorithm for short-term hydrothermal scheduling," *International Journal of Electrical Power & Energy Systems*, vol. 65, pp. 271–281, 2015.

[18] X.-T. Li and M.-H. Yin, "A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem," *International Journal of Production Research*, vol. 51, no. 16, pp. 4732–4754, 2013.

[19] G. Zumofen, J. Klafter, and M.-F. Shlesinger, "Lévy flights and Lévy walks revisited," in *Anomalous Diffusion From Basics to Applications: Proceedings of the XIth Max Born Symposium Held at Lądek Zdrój, Poland, 20–27 May 1998*, vol. 519 of *Lecture Notes in Physics*, pp. 15–34, Springer, Berlin, Germany, 1999.

[20] C. Chareonsuk, N. Nagarur, and M. T. Tabucanon, "A multicriteria approach to the selection of preventive maintenance intervals," *International Journal of Production Economics*, vol. 49, no. 1, pp. 55–64, 1997.

[21] J. K. Vaurio, "Availability and cost functions for periodically inspected preventively maintained units," *Reliability Engineering & System Safety*, vol. 63, no. 2, pp. 133–140, 1999.

[22] R. Jiang and P. Ji, "Age replacement policy: a multi-attribute value model," *Reliability Engineering and System Safety*, vol. 76, no. 3, pp. 311–318, 2002.

[23] A. A. Kalir, "Segregating preventive maintenance work for cycle time optimization," *IEEE Transactions on Semiconductor Manufacturing*, vol. 26, no. 1, pp. 125–131, 2013.

[24] M. Samrout, F. Yalaoui, E. Châtelet, and N. Chebbo, "New methods to minimize the preventive maintenance cost of series-parallel systems using ant colony optimization," *Reliability Engineering & System Safety*, vol. 89, no. 3, pp. 346–354, 2005.

[25] K. S. Moghaddam, S. Kamran, and J. S. Usher, "A new multi-objective optimization model for preventive maintenance and replacement scheduling of multi-component systems," *Engineering Optimization*, vol. 43, no. 7, pp. 701–719, 2011.

[26] Y. Yare, G. K. Venayagamoorthy, and U. O. Aliyu, "Optimal generator maintenance scheduling using a modified discrete PSO," *IET Generation, Transmission and Distribution*, vol. 2, no. 6, pp. 834–846, 2008.

[27] A. Abdulwhab, R. Billinton, A. A. Eldamaty, and S. O. Faried, "Maintenance scheduling optimization using a genetic algorithm (GA) with a probabilistic fitness function," *Electric Power Components and Systems*, vol. 32, no. 12, pp. 1239–1254, 2004.

[28] A. Berrichi, F. Yalaoui, L. Amodeo, and M. Mezghiche, "Bi-objective ant colony optimization approach to optimize production and maintenance scheduling," *Computers and Operations Research*, vol. 37, no. 9, pp. 1584–1596, 2010.

[29] A. K. Verma and P. G. Ramesh, "Multi-objective initial preventive maintenance scheduling for large engineering plants," *International Journal of Reliability, Quality and Safety Engineering*, vol. 14, no. 3, pp. 241–250, 2007.

[30] T.-W. Lin and C.-H. Wang, "A hybrid genetic algorithm to minimize the periodic preventive maintenance cost in a series-parallel system," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1225–1236, 2012.

[31] R.-C. Leou, "A new method for unit maintenance scheduling considering reliability and operation expense," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 7, pp. 471–481, 2006.

[32] S.-S. Ma, H. Zhang, T. Feng, and J. Xue, "Optimization of preventive maintenance period based on hybrid swarm intelligence," in *Proceedings of the 6th International Conference on Natural Computation (ICNC '10)*, vol. 5, pp. 2656–2659, IEEE, Yantai, China, August 2010.

[33] J. Kim and G. Z. Woo, "Optimal scheduling for maintenance period of generating units using a hybrid scatter-genetic algorithm," *IET Generation, Transmission and Distribution*, vol. 9, no. 1, pp. 22–30, 2015.

[34] G. G. Samuel and C. C. A. Rajan, "Hybrid: particle swarm optimization-genetic algorithm and particle swarm optimization-shuffled frog leaping algorithm for long-term generator maintenance scheduling," *International Journal of Electrical Power & Energy Systems*, vol. 65, pp. 432–442, 2015.

[35] N. Montgomery, D. Banjevic, and A. K. S. Jardine, "Minor maintenance actions and their impact on diagnostic and prognostic CBM models," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 303–311, 2012.

[36] I. T. Dedopoulos and Y. Smeers, "An age reduction approach for finite horizon optimization of preventive maintenance for single units subject to random failures," *Computers & Industrial Engineering*, vol. 34, no. 3, pp. 643–654, 1998.

[37] E. Vilian, S. Mohanna, and S. Tavakoli, "Improved cuckoo search algorithm for global optimization," *International Journal of Communications and Information Technology*, vol. 1, no. 1, pp. 31–44, 2011.