Hindawi Publishing Corporation
Science and Technology of Nuclear Installations
Volume 2014, Article ID 286826, 9 pages
http://dx.doi.org/10.1155/2014/286826

Research Article

PSO Based Optimization of Testing and Maintenance Cost in NPPs

Qiang Chou,[1] Daochuan Ge,[2] and Ruoxing Zhang[1]

[1] *Software Development Center, State Nuclear Power Technology Corporation, Beijing 102209, China*
[2] *School of Nuclear Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

Correspondence should be addressed to Qiang Chou; chouq07@gmail.com

Received 11 July 2014; Revised 13 November 2014; Accepted 13 November 2014; Published 9 December 2014

Academic Editor: Alejandro Clausse

Testing and maintenance activities of safety equipment have drawn much attention in Nuclear Power Plant (NPP) to risk and cost control. The testing and maintenance activities are often implemented in compliance with the technical specification and maintenance requirements. Technical specification and maintenance-related parameters, that is, allowed outage time (AOT), maintenance period and duration, and so forth, in NPP are associated with controlling risk level and operating cost which need to be minimized. The above problems can be formulated by a constrained multiobjective optimization model, which is widely used in many other engineering problems. Particle swarm optimizations (PSOs) have proved their capability to solve these kinds of problems. In this paper, we adopt PSO as an optimizer to optimize the multiobjective optimization problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Numerical results have demonstrated the efficiency of our proposed algorithm.

1. Introduction

Improvement of the availability performance for safety-related systems has drawn much attention in Nuclear Power Plant (NPP) nowadays. One way to increase availability of these systems is to improve availability of the equipment that constitutes them. In this way, people often pay attention to the more efficient testing and maintenance activities. NPPs often pursue more efficient testing and maintenance activities to control risk and cost. Actually, safety-critical equipment is normally on standby till occurrence of an accident situation which requires safety-related systems to prevent or mitigate the accident process. In order to keep safe-related systems at a high level of availability or safety, regular testing and maintenance activities are implemented. Efficient regular testing and maintenance strategy can improve the availability performance of the systems, and meanwhile it will lead to great expenditure cost. Therefore, both risk controlling and expenditure effectiveness have drawn much attention in NPP [1, 2].

Technical specifications define the limits and conditions for operating NPPs which can be seen as a set of safety rules and criteria required as a part of safety analysis report of each NPP. Both technical specifications and maintenance activities are associated with controlling risk and then with availability of safety-related systems. The resource related to risk controlling rules and criteria formally enter into optimization problems. Using a limited expenditure resource to keep safety-critical equipment at a high level of availability or safety actually is a constrained multiobjective optimization problem where the cost or the burden, that is, number of tests conducted, duration, incurred cost, and so forth, is to be minimized while the unavailability or the performance of the safety-critical equipment is constrained at a given level.

By now, some researchers have made great achievements in nuclear technology area. References [3, 4] presented a constrained multiobjective optimization model to solve this problem using genetic algorithm (GA); reference [5] first presented PSA-based techniques to solve risk-cost maintenance and testing model of an NPP using GA; reference [6] puts forward using a multiobjective approach to regulate Nuclear Power Plant (NPP); reference [7] presents using fuzzy-genetic approach to optimize the test interval of safety systems at NPP considering parameters uncertainty. In this paper, we put forward using PSO to solve the constrained multiobjective optimization problem which simulates the testing and maintenance activities. The PSO method is firstly used to solve the multiobjective optimization problem described by testing and maintenance activities in NPPs. It is a heuristic algorithm and can offer the solution by iteratively trying to improve a candidate solution with regard to a given measure of quality. Numerical results have demonstrated the reasonability of PSO method.

The plan of this paper is the following: Section 2 presents the unavailability and cost models of critical systems/components of NPP; Section 3 gives the multiobjective problem model; Section 4 reviews PSO method; Section 5 presents a case study; finally, Section 6 draws a short conclusion.

2. System Risk and Cost Function

2.1. System Unavailability Model. As to nuclear facilities, the system unavailability is classified into three types: component's unavailability, common failure, and human errors. In this paper, we just consider the component's unavailability which is caused by random failure and test and maintenance activities which are the functions of the optimization variables such as test interval, test duration, maintenance period, allowed outage time, and so on. The system unavailability is often modeled by fault tree using rare-event approximation as follows [4]:

$$U_{\text{system}}(X) \approx \sum_{j} \prod_{i} u_{ji}(X), \tag{1}$$

where X is the decision variable vector; the sum in j refers to the number of minimal cut sets generated from the considered system structure function and the product in i represents the number of the basic events belonging to the corresponding MCS. The $u_{ji}(X)$ represents the unavailability of the basic event i contained in minimal cut sequence (MCS) j.

The unavailability expressions of basic events caused by random failure are written as [4]:

$$u_r(X, T) \approx \rho + \lambda \cdot T. \tag{2}$$

$$u_r(X) \approx \rho + \frac{1}{2}\lambda \cdot T. \tag{3}$$

Equation (2) is the time-dependent unavailability evaluated at $t = T$, where ρ denotes per-demand failure probability and λ represents the failure rate. Equation (3) is the average time-dependent over a given time span T.

To reflect the effect of age, preventive maintenance, and working conditions, an averaged standby failure rate $\overline{\lambda}$ is developed [8, 9]:

$$\overline{\lambda} = \lambda_0 + \frac{1}{2}\alpha \cdot M \cdot (\Psi(Z))^2 \cdot \frac{1}{\varepsilon} \cdot (2 - \varepsilon) \tag{4}$$

$$\overline{\lambda} = \lambda_0 + \frac{1}{2}\alpha \cdot M \cdot (\Psi(Z))^2 \cdot \left[1 + (1 - \varepsilon) \cdot \left(\frac{L}{M} - 3\right)\right]. \tag{5}$$

Table 1: Meanings of the parameters used in (4)-(5).

Parameter	Meaning
λ_0	The residual standby failure rate
α	The linear aging factor
$\Psi(Z)$	Working condition factor
ε	The maintenance effectiveness factor
M	The period of a preventive maintenance
L	Overhaul period to replace a component

Note that (4) is applicable for proportional age setback (PAS) mode, and (5) is used for proportional age reduction PAR. The meanings of the parameters involved in (4) and (5) are listed in Table 1.

Therefore, one could apply (4) or (5) into (2)-(3) to account for random failure contributions considering effect of age, preventive maintenance, and working conditions.

Next, let us consider the models to account for the testing and maintenance activities effect, [9, 10] developed three expressions to characterize such effects:

$$\begin{aligned} u_t(X) &\approx f_t(X) \cdot t \cdot q_t \\ u_m(X) &\approx f_m(X) \cdot m \cdot q_m \\ u_c(X) &\approx f_c(X) \cdot d(X). \end{aligned} \tag{6}$$

The meanings of notations used in (6) are listed in Table 2.

Given that a test interval T and preventive maintenance period M, the $f_t(X)$, $f_m(X)$, $f_c(X)$ can be calculated as follows:

$$f_t(X) = \frac{1}{T} \tag{7}$$

$$f_m(X) = \frac{1}{M} \tag{8}$$

$$f_c(X) \approx \frac{1}{T} \cdot u_r(X, T). \tag{9}$$

Note that, in (9), $f_c(X)$ can be calculated from (2) replacing λ with $\overline{\lambda}$.

Additionally, $d(x)$ is often restricted by the allowed outage time (AOT) in the standard PRA and is often calculated by the following relationships:

$$\begin{aligned} d(X) &= \mu \cdot G(X) \\ G(X) &= 1 - e^{-D/u}, \end{aligned} \tag{10}$$

where D represents the allowed downtime; μ is the mean time to repair.

Table 2: Meanings of the notations in (6).

Notation	Meanings
$u_t(X)$	Unavailability caused by testing
$u_m(X)$	Unavailability caused by preventive maintenance
$u_c(X)$	Unavailability caused by corrective maintenance
$f_t(X)$	The rate of testing events
$f_m(X)$	The rate of preventive maintenance events
$f_c(X)$	The rate of corrective maintenance events
t	Mean time of testing
m	Mean time of preventive maintenance
$d(X)$	Mean downtime of corrective maintenance
q_t	Fraction of the total time t with the component down
q_m	Fraction of the total time m with the component down

2.2. System Cost Function. As to equipment, the cost expressions due to implementing test and maintenance activities are often expressed as follows [8–10]:

$$\begin{aligned} c_t(X) &= \frac{1}{T}\cdot c_{ht} \\ c_m(X) &= \frac{1}{M}\cdot c_{hm} \\ c_o(X) &= \frac{1}{L}\cdot c_r \\ c_c(X) &= f_c(X)\cdot d(X)\cdot c_{hc} \\ c_s(X) &= f_c(X)\cdot[1-G(X)]\cdot T_s\cdot c_{hs} \\ c_u &= T_s\cdot c_{hs}. \end{aligned} \tag{11}$$

The meanings of the notations used in (11) are listed in Table 3.

Then, the system cost model is easily formulated by summing up the corresponding cost contributions of the relevant components as follows:

$$C(X)=\sum_{i=1}^{n} c_i(X), \tag{12}$$

where n represents the total number of the components in the considered system. Obviously, this model can be described by a multiobjective optimization problem.

2.3. Constraints. The presence of constraints significantly affects the performance of an optimization algorithm, including evolutionary search methods. Satisfying constraints is a difficult problem in itself often. There have been a number of approaches to impose constraints including rejection of infeasible solutions, penalty functions and their variants, repair methods, use of decoders, and so on. A comprehensive review on constraint handling methods is provided by Michalewicz [11]. All the methods have limited success as they are problem dependent and require a number of additional inputs. When constraints cannot be all simultaneously satisfied, the problem is often deemed to admit no solution. The number of constraints violated and the extent to which each constraint is violated need to be considered in order to relax the preference constraints. Generally speaking, we can impose constraints over (1) the objective functions and (2) the values the decision variables in vector x can take. In our first case, we apply constraints over one of the two possible objective functions, risk or cost function, which will act as an implicit restriction function alternatively. For example, if the selected objective function to be minimized is the risk, then the constraint is a restriction over the maximum allowed value to its corresponding cost. In our second case, we handle constraints directly over the values the decision variables in vector x can take, that is, technical specification and maintenance requirements (TS&M) parameters, which are referred to as explicit constraints. Examples of this type of constraints in optimizing TS&M include limiting the STIs, AOTs, and the preventive maintenance (PM) period to take typified values because of practical considerations of planning, representing each one, for example, an hour, a day, a month, or any other realistic period in the plant, instead of only mathematical solutions that are often unpractical.

Table 3: Meanings of the notations in (11).

Notation	Meanings
$c_t(X)$	Yearly cost due to testing
$c_m(X)$	Yearly cost due to preventive maintenance
$c_o(X)$	Overhaul cost due to replacing a component with a new one
$c_c(X)$	Yearly cost due to corrective maintenance
$c_s(X)$	Outage cost due to technique specification (TS)
c_u	Constant cost due to the plant outage
T_s	Average time required to start up NPP after shutdown
c_{ht}	Hourly cost for conducting surveillance test
c_{hm}	Hourly cost for preventive maintenance
c_{hc}	Hourly cost for corrective maintenance
c_{hs}	Hourly cost for loss of production for plant downtime
c_r	The total cost of replacing a component

3. Multiobjective Optimization Model

As to a general multiobjective optimization problem, it often has n-dimensions decision variables and m-dimension optimizing objectives and can be expressed as follows:

$$\begin{aligned} \min\quad & Y=F(X)=\left(f_1(X),f_2(X),\ldots,f_m(X)\right) \\ \text{s.t.}\quad & g_i(X)\le 0,\quad i=1,2,\ldots,q \\ & h_j(X)=0,\quad j=1,2,\ldots,p \\ & X\in[X_{\min},X_{\max}], \end{aligned} \tag{13}$$

where X is an n-dimension decision space and Y is an m-dimension objective space defining m mapping functions from decision space to objective space. The $g_i(X)$ represents q

inequality constraints and $h_j(X)$ represents p equalities constraints. $X_{\min}$ and $X_{\max}$ denote the low and upper boundaries of the decision variables, respectively.

The specific multiobjective optimization model for optimizing testing and maintenance activities at NPP can be expressed as follows:

$$\begin{aligned} \min \quad & C(X) \\ & U_{\text{system}}(t) - U_g \le 0 \qquad (14) \\ & X_{\text{low}} \le X \le X_{\text{up}}, \end{aligned}$$

where U_g is a given limited unavailability level. In this paper we are using PSO based techniques to solve the minimization problem described with (14).

Multiobjective optimization has been applied in many fields of science, including engineering, economics, and logistics where optimal decisions need to be taken in the presence of tradeoffs between two or more conflicting objectives. For a nontrivial multiobjective optimization problem, there does not exist a single solution that simultaneously optimizes each objective. When one deals with multiobjective optimization problems with conflicting optimization criteria, there is not only a single solution but also a set of alternative solutions [12]. None of these solutions can be said to be better than the others with respect to all objectives. Since none of them is dominated by the others, they are called nondominated solutions. The curve formed by connecting these solutions is known as a Pareto optimal front and the solutions that lay on this curve are called Pareto optimal solutions. Classical optimization methods for multiobjective optimization suggest converting it to a single-objective optimization problem through emphasizing one particular Pareto-optimal solution at a time. This is the disadvantage of classical optimization methods, because finding Pareto-optimal solutions set is required often. Recently, a number of evolutionary algorithms (MOEAs) have been proposed [13–15]. The reason is their ability to obtain multiple Pareto-optimal solutions in one single simulation run. In these algorithms, genetic algorithm (GA) is wisdom and popular for multiobjective optimization problems and GAs are adaptive methods that can be used in searching and optimization problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The particle swarm optimization (PSO) is one type of evolutionary algorithms (EAs); PSO shares many similarities with evolutionary computation techniques such as genetic algorithms (GAs). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation but its particles have mnemonic ability of the historical optimal positions. Thus PSO is of simplicity, flexibility, and easy operation compared with GA. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. As many other EAs such as genetic algorithm (GA), Ant Colony Optimization (ACO), and so forth, PSO can find Pareto optimal solutions and have chances to find near global solutions, only with different probability and reliability in finding these solutions. The more details of PSO will be introduced later on.

4. The Multiobjective Optimization Algorithm Based on PSO

4.1. Particle Swarm Optimization. Particle swarm optimization (PSO) has been successfully used to optimize nonlinear functions, combinatorial optimization problems and multiobjective problems [16, 17] because of its simplicity, flexibility, easy operation, and fast convergence. PSO is originally attributed to Eberhart et al. [18, 19] and was first intended for simulating social behaviour, [20] as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm is simplified and it is observed to be performing optimization. The book by Kennedy and Eberhart [21] describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli [22]. The PSO's basic idea is having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formula over the particle's position and velocity. Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. However, as many other metaheuristic methods, PSOs do not guarantee that an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized; PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, and so forth.

Let N be the number of particles in the swarm, each having a position $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$ in the search-space and a velocity $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$. Let $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^T$ be the best known position of particle i. There is the best particle in the swarm labeled by g and let $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})^T$ be the best known position of the entire swarm; that is to say, P_g is the optimal position of search history before. Particles update their velocity and position as the following formula:

$$v_{ij}^{k+1} = v_{ij}^{k} + c_1 r_{1j} \left(p_{ij}^{k} - x_{ij}^{k}\right) + c_2 r_{2j} \left(p_{gj}^{k} - x_{ij}^{k}\right) \qquad (15)$$

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1}, \qquad (16)$$

where $i = 1, 2, \ldots, N$, j denotes the jth dimension of the particles, k is the number of iterations, and c_1, c_2 are the acceleration constant, whose values are often in $[0, 2]$. $r_1 \sim u(0,1)$ and $r_2 \sim u(0,1)$ are two independent identity distribution random numbers. Generally, v_{ij} is kept within the range $[V_{\min}, V_{\max}]$. The first part of formula (15) v_{ij}^k is the particle velocity of the last iteration, the second part $c_1 r_{1j}(p_{ij}^k - x_{ij}^k)$ is the *cognitive* part, and the third part $c_2 r_{2j}(p_{gj}^k - x_{ij}^k)$

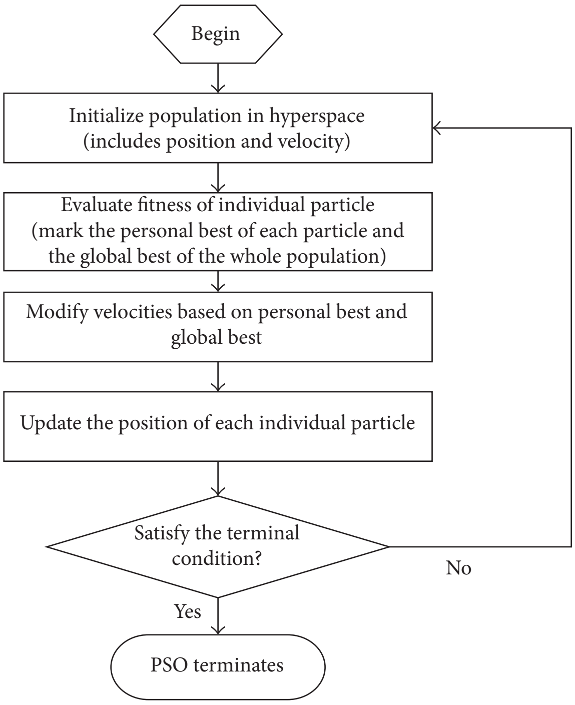

Begin
Initialize population in hyperspace (includes position and velocity)
Evaluate fitness of individual particle (mark the personal best of each particle and the global best of the whole population)
Modify velocities based on personal best and global best
Update the position of each individual particle
Satisfy the terminal condition?
No
Yes
PSO terminates

Figure 1: PSO algorithm procedure.

is the *momentum* part. A basic PSO algorithm is shown in Algorithm 1.

The summary flow chart of the basic PSO is shown in Figure 1.

The basic PSO described above has a small number of parameters that need to be fixed. One parameter is the size of the population. This is often set empirically on the basis of the dimensionality and perceived difficulty of a problem. Values in the range 20–50 are quite common.

Also there are many variant PSOs based on the basic PSO such as the PSO with inertia weight [19], the PSO with constriction coefficients [23], and so forth. The PSO with inertia weight is motivated by the desire to better control the scope of the search, reduce the importance of $V_{\max}$, and perhaps eliminate it altogether. The PSO with constriction coefficients which is based on some form of damping of the dynamics of a particle (e.g., $V_{\max}$) is necessary. These variant PSOs are applicable in different situations. In this paper, we only consider the basic PSO version to verify the validity of PSO for minimizing the testing and maintenance cost in NPP.

4.2. The Proposed Multiobjective Optimization Algorithm. The basic idea of multiobjective optimization algorithm based on PSO in this paper is that through spliting and merging the dominated set and nondominated set repeatly, we can reach a better balance between the algorithm efficiency and accuracy. This is based on the idea of fitness dominance, which is similar with the idea mentioned in literatures [23–25]. Let the initial population size be *n*. *P_Set* is one nondominated subset of the population whose size is n_1 and *NP_Set* is another n_2 size dominated subset which meets $n_1 + n_2 = n$ $(1 \leq n_1, n_2 \leq n)$. For any element $X_j \in NP\_Set$, there at least exists an element $X_i \in P\_Set$ which holds that X_i dominates X_j $(1 \leq i \leq n_1, 1 \leq j \leq n_2)$. In each iteration process, only update the elements in *NP_Set* and then compare them with the elements in *P_Set* based on fitness dominance rules. Then the dynamic switching strategy can be described as follows: for any $X_i \in P\_Set$, if there exist $X_j \in NP\_Set$, holds X_j dominates X_i, then switch the positions of these two elements. After the previous preparation, we are now ready to describe the multiobjective optimization algorithm based on basic PSO (abbreviate it as MOBPSO (see Algorithm 2)). Let the *j*th constraint for the *j*th dimension of $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$ be $x_{ij\text{-lower}} \leq x_{ij} \leq x_{ij\text{-upper}}$ $(1 \leq j \leq D)$.

5. Case Study

In this section, the high-pressure injection system (HPIS) is considered a case study; the simplified structure diagram is shown in Figure 2.

The system consists of 7 valves and 3 pumps. The function is drawing water from the refueling water storage tank (PWST) and discharges it into the cold legs of the reactor cooling system through any of the two inlets, A or B. The components reliability parameters are listed in Table 4.

The components cost information is shown in Table 5.

The constraints on the test intervals are listed in Table 6.

The constraints on components preventive maintenance intervals are listed in Table 7.

The decision variables vector is chosen as $X = \{T_1, K_2, K_3, M_1, n_1, M_2, D_1, D_2\}$ where D_1 are D_2 are

```
(1) Initialize a population array of particles with random positions and velocities on $D$
    dimensions in the search space.
(2) For loop
(3) For each particle, evaluate the desired optimization fitness function in $D$ variables.
(4) Compare particle's fitness evaluation with its $P_g$. If current value is better than $P_g$, then
    set $P_g$ equal to the current value, and $P_i$ equal to the current location $X_i$ in $D$-dimensional space.
(5) Identify the particle in the neighborhood with the best success so far, and assign its index to
    the variable $g$.
(6) Change the velocity and position of the particle according to (15)-(16).
(7) If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations),
    exit loop.
(8) end **loop**
```

Algorithm 1: Basic PSO [18].

```
(1) Initialize a population array of particles with random positions and velocities on $D$
    dimensions in the search space and the population size is $n = Pop\_Max$.
(2) For $i = 1$ to $n$
    Evaluate the fitness function in $D$ dimensional variables, namely,
    $\mathbf{F}(X_i) = \{f_1(X_i), f_2(X_i), \ldots, f_m(X_i)\}$.
    End for
(3) Divide the initial population into two subsets $P\_Set$ and $NP\_Set$. whose population sizes are
    $n_1$ and $n_2$ respectively.
(4) Update the velocity and position of each particle according to (15)-(16).
    Where $p_{gj}^k$ is selected from the subset of $P\_Set$ randomly. For the constraint-handling
    approach, update the $x_{ij}^{k+1}$ with $x_{ij}^k + x_{ij}^{k+1}$ if $x_{ij}^k + x_{ij}^{k+1}$ is in the constraint interval,
    namely $x_{ij}^k + x_{ij}^{k+1}$ is feasible.
(5) Dynamic switching strategy: Compare each particle in $NP\_Set$ with that in $P\_Set$. Let the
    particles in $NP\_Set$ be $X_1, \ldots, X_i, \ldots, X_{n_2}$ and the elements in $P\_Set$ be $X_1, \ldots, X_j, \ldots, X_{n_1}$.
    For $i = 1$ to $n_2$
        For $j = 1$ to $n_1$
            If $\mathbf{F}(X_i) < \mathbf{F}(X_j)$
            Switch $X_i$ and $X_j$ then update their index and position in sets.
            End if
        End for
    End for
    Update the two sets $P\_Set$ and $NP\_Set$.
    If there exist $k$ same particles in $P\_Set$, delete them and re-initialize $k$ particles in $NP\_Set$,
    and vice versa.
    Update $n_1$ and $n_2$.
    If $n_1 \neq n$ or not reaching the given maximum iterative number, goto Step 3.
```

Algorithm 2: MOBPSO.

Table 4: Components reliability parameters.

Equipment	λ_0 (10^{-6}/h)	ρ (10^{-3})	t/h	u/h	m/h
Valves	5.50	2.00	0.85	0.80	2.5
Pumps	3.50	0.55	5.00	4.5	25
Equipment	Age type	$\alpha = 10^{-10}/\text{h}^2$	$\Psi(z)$	ε	q_t/q_m
Valves	PAR	0.45	1	0.5	1
Pumps	PAS	0.25	1	0.5	1

Table 5: Components cost information (RMB/h).

Equipment	c_{ht}	c_{hc}	c_{hm}	c_r	c_u
Valves	150	115	120	900	11100
Pumps	150	115	120	2700	11100

the allowed outage time for valves and pumps, respectively. For the valves, the permitted range is $4\,\text{h} \le D_1 \le 24\,\text{h}$. As to the pumps, the permitted range is $24\,\text{h} \le D_2 \le 168\,\text{h}$.

Our first case of study encompasses two different single objective optimization problems, considering the optimization of the risk function and adopting the cost function as an implicit constraint or vice versa.

Firstly, we choose the system yearly cost as an objective function and the system unavailability is treated as

Table 6: Constraints on test intervals (TI).

Test interval	Valves	Pumps	Constraints	Permitted range
T_1	V1, V2	PA, PB, PC	$T_2 = K_2 \cdot T_1$ $T_3 = K_3 \cdot K_1 \cdot T_1$	$168\,\text{h} \le T_1 \le 8670\,\text{h}$
T_2	V3, V5			$1 \le K_2 \le 10$
T_3	V4, V7, V8			$1 \le K_3 \le 10$

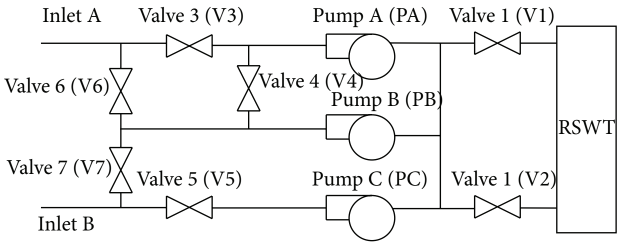


Figure 2: HPIS system of a PWR.

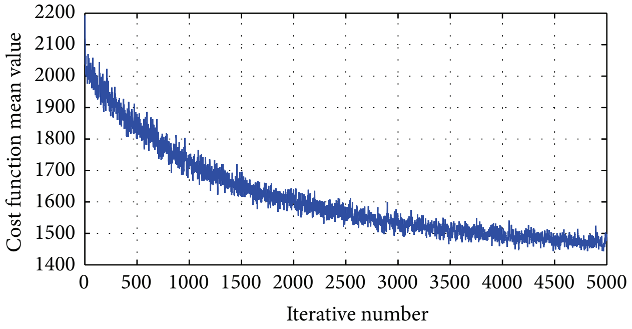


Figure 3: Convergence evolution versus cost function.

a constraint ($u_g \le 1.0e-4$). The customized basic PSO described in previous section has been used as optimizer. The optimization results are shown in Figure 3. From Figure 3, we can see the mean value of fitness (cost) in one particle decreases gradually along with the increasing of the iterative number.

Then, we choose the system yearly unavailability as the objective and treat the system yearly cost as a constraint ($c_g <$ 2460 RMB). The optimization results are shown in Figure 4. The results show that the mean value of unavailability also decreases as the increasing of the iterative number.

The optimized results are shown in Table 8.

As observed in Figures 3-4, objective functions are becoming convergent with the increase of the iteration number. PSO optimizer finally presents a valid optimized variable vector. In case 1, optimized results keep the considered system nearly at the same risk level, but greatly reduce the cost. Moreover, optimized solutions notably decrease the system risk level remaining at almost the same expenditure cost.

In the second case we consider the model as a multi-objective optimization problem and obtain the nondominated solutions with MOBPSO algorithm. Figure 5 shows the obtained nondominated solutions.

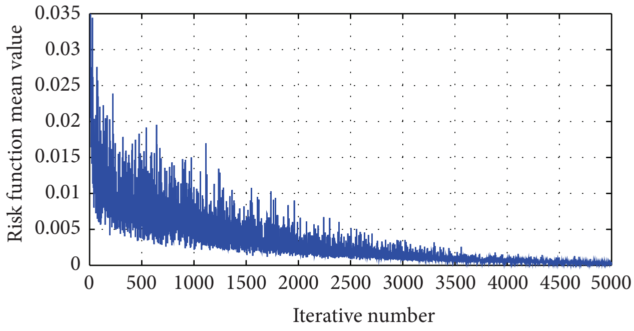


Figure 4: Convergence evolution versus risk function.

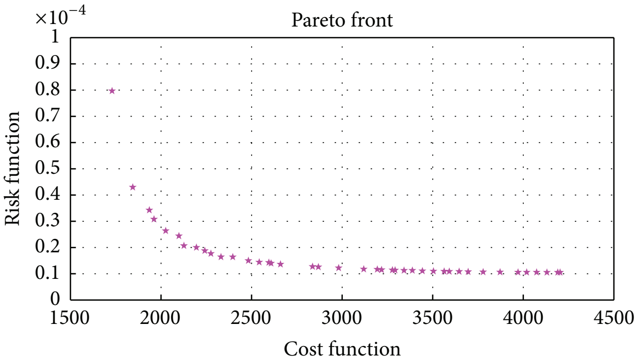


Figure 5: Obtained nondominated solutions with MOBPSO.

6. Conclusions

One important aspect of the nuclear industry is to improve the availability of safety-related equipment at NPPs to achieve high safety and low cost levels. In this paper, the multiobjective optimization algorithm based on PSO has been applied to solve the constrained multiobjective optimization of testing and maintenance-related parameters of safety-related equipment. The numerical results indicate that PSO is a powerful optimization method for finding NPP configuration resulting in minimal cost and unavailability. Also based on the MOBPSO, we obtain the nondominated solutions set. The results successfully verify that the PSO is capable of finding a nondominated solution of a constrained multiobjective problem in resource effectiveness and risk control of NPP. The multiobjective optimization algorithm based on PSO should attract more attention to apply in optimization of testing and maintenance activities of safety equipments at NPPs. Exploring the capacity (including topology structure, the optimal selection of parameters, and the integration with

Table 7: Constraint on preventive maintenance intervals.

Maintenance period	Valve	Pump	Relationship	Permitted range
M_1	V1–V7	—	$L_1 = n_1 \cdot M_1$	$2 \le n_1 \le 20$ $720 \le M_1 \le 20280$
M_2	—	PA, PB, PC	—	$720 \le M_2 \le 20280$

Table 8: Optimized results.

Variables	Initial values	Optimized results	
		Case 1	Case 2
T_1	2180	1629	2275
K_2	2	10	1
K_3	2	5	1
n_1	24	3	2
M_1	4000	14653	2507
M_2	4000	16805	10157
D_1	10	5	8
D_2	75	115	100
$U(X)$	$4.0e-4$	$1.8665e-4$	$0.97693e-5$
$C(X)$	2460	1728.7	5767.2

other EAs, etc.) of PSO for the testing and maintenance cost model is our future work.

Acronym

ACO:	Ant Colony Optimization
AOT:	Allowed outage time
GA:	Genetic algorithm
HPIS:	High-pressure injection system
MCS:	Minimal cut sequence
NPP:	Nuclear Power Plant
PAR:	Proportional age reduction
PAS:	Proportional age setback
PSWT:	Refueling water storage tank
PSO:	Particle swarm optimization.

Notation

λ:	Failure rate
$\overline{\lambda}$:	Averaged failure rate
λ_0:	The residual standby failure rate
μ:	Repair rate
X:	Decision variable vector
$F(t)$:	Cumulative distribution function
α:	Age cofactor
ε:	The maintenance effectiveness factor
$\Psi(Z)$:	Working condition factor
L:	Overhaul period.

Conflict of Interests

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

References

[1] IEAE, "Risk based optimization of technical specifications for operation of nuclear power plants," IAEA-TECDOC-729, 1993.

[2] M. Harunuzzaman and T. Aldemir, "Optimization of standby safety system maintenance schedules in nuclear power plants," *Nuclear Technology*, vol. 113, no. 3, pp. 354–367, 1996.

[3] S. Martorell, S. Carlos, A. Sánchez, and V. Serradell, "Constrained optimization of test intervals using a steady-state genetic algorithm," *Reliability Engineering and System Safety*, vol. 67, no. 3, pp. 215–232, 2000.

[4] S. Martorell, A. Sánchez, S. Carlos, and V. Serradell, "Simultaneous and multi-criteria optimization of TS requirements and maintenance at NPPs," *Annals of Nuclear Energy*, vol. 29, no. 2, pp. 147–168, 2002.

[5] T. Jiejuan, M. Dingyuan, and X. Dazhi, "A genetic algorithm solution for a nuclear power plant risk-cost maintenance model," *Nuclear Engineering and Design*, vol. 229, no. 1, pp. 81–89, 2004.

[6] A. Mishra, M. D. Pandey, and A. Chauhan, "Regulation of nuclear power plants a multi objective approach," *Annals of Nuclear Energy*, vol. 36, no. 10, pp. 1560–1573, 2009.

[7] K. Durga Rao, V. Gopika, H. S. Kushwaha, A. K. Verma, and A. Srividya, "Test interval optimization of safety systems of nuclear power plant using fuzzy-genetic approach," *Reliability Engineering and System Safety*, vol. 92, no. 7, pp. 895–901, 2007.

[8] E. Martorell, A. Munoz, and V. Serradell, "Age-dependent models for evaluating risks & costs of surveillance & maintenance of components," *IEEE Transactions on Reliability*, vol. 45, no. 3, pp. 433–442, 1996.

[9] S. Martorell, A. Sanchez, and V. Serradell, "Age-dependent reliability model considering effects of maintenance and working conditions," *Reliability Engineering and System Safety*, vol. 64, no. 1, pp. 19–31, 1999.

[10] S. Martorell, S. Carlos, A. Sanchez, and V. Serradell, "Using genetic algorithms in completion times and test intervals optimization with risk and cost constraints," in *Proceedings of the (ESREL '00), SARS and SRA-Europe Annual Conference*, May 2000.

[11] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pp. 135–155, The MIT Press, Cambridge, Mass, USA, 1995.

[12] B. Gjorgiev, D. Kančev, and M. Čepin, "Risk-informed decision making in the nuclear industry: application and effectiveness comparison of different genetic algorithm techniques," *Nuclear Engineering and Design*, vol. 250, pp. 701–712, 2012.

[13] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, 2001.

[14] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," in *Proceedings of the 5th International Conference*

on Genetic Algorithms, S. Forrest, Ed., pp. 416–423, Morgan Kauffman, San Mateo, Calif, USA, 1993.

[15] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *Parallel Problem Solving from Nature V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., pp. 292–301, Springer, Berlin, Germany, 1998.

[16] K. Parsopóulos and M. N. Vrahatis, "Particle swarm optimization method in multi-objective problems," in *Proceeedings of the ACM Symposium on Applied Computing (SAC '02)*, pp. 603–607, March 2002.

[17] C. A. C. Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1051–1056, Honolulu, Hawaii, USA, May 2002.

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

[19] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.

[20] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 303–308, April 1997.

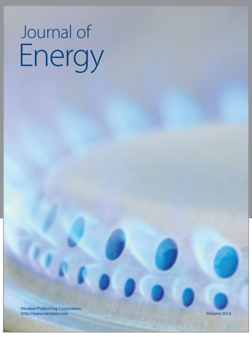
[21] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, Boston, Mass, USA, 2001.

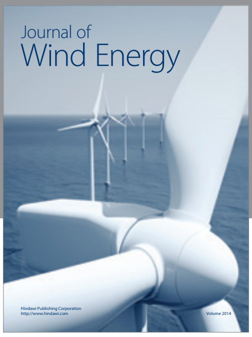
[22] R. Poli, "An analysis of publications on particle swarm optimisation applications," Tech. Rep. CSM-469, Department of Computer Science, University of Essex, Colchester, UK, 2007.

[23] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[24] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1995.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

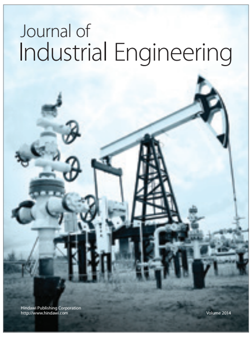

Journal of
Energy

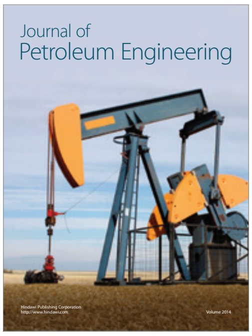

Journal of
Wind Energy

The Scientific
World Journal

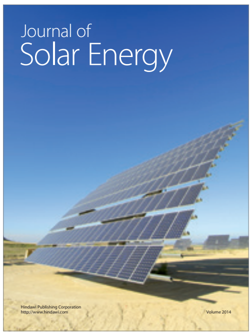
Journal of
Structures

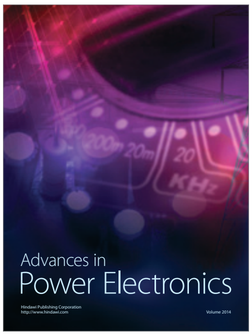
International Journal of
Rotating
Machinery


Journal of
Industrial Engineering

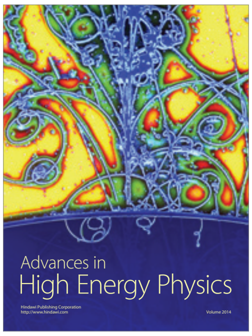

Journal of
Petroleum Engineering

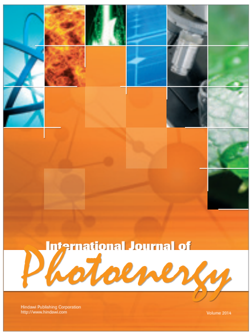
Journal of
Fuels


Journal of
Solar Energy

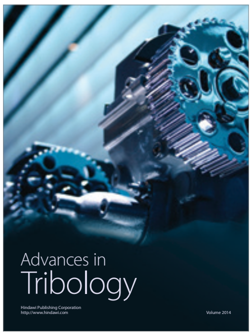

Advances in
Power Electronics

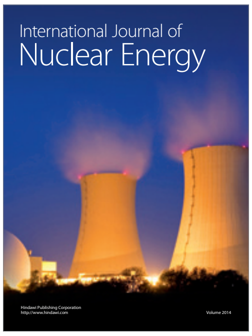
Journal of
Engineering

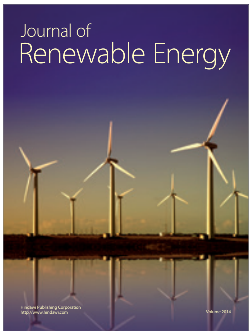

Advances in
High Energy Physics

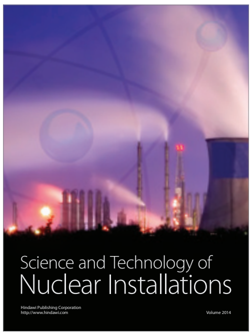

International Journal of
Photoenergy

Journal of
Combustion


Advances in
Tribology


International Journal of
Nuclear Energy


Journal of
Renewable Energy


Science and Technology of
Nuclear Installations

International Journal of
Aerospace
Engineering