Hindawi
Journal of Sensors
Volume 2021, Article ID 7833641, 12 pages
https://doi.org/10.1155/2021/7833641

Research Article

Solving Cold-Standby Reliability-Redundancy Allocation Problems with Particle-Based Simplified Swarm Optimization

Chia-Ling Huang,[1] Yunzhi Jiang ,[2] and Wei-Chang Yeh[3]

[1]*Department of International Logistics and Transportation Management, Kainan University, Taoyuan 33857, Taiwan*
[2]*School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*
[3]*Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 300, Taiwan*

Correspondence should be addressed to Yunzhi Jiang; jiangyunzhi@foxmail.com

Received 22 April 2021; Revised 15 July 2021; Accepted 10 August 2021; Published 21 August 2021

Academic Editor: Wei-Chiang Hong

Particle swarm optimization (PSO) and simplified swarm optimization (SSO) are two of the state-of-the-art swarm intelligence technique that is widely utilized for optimization purposes. This paper describes a particle-based simplified swarm optimization (PSSO) procedure which combines the update mechanisms (UMs) of PSO and SSO to determine optimal system reliability for reliability-redundancy allocation problems (RRAPs) with cold-standby strategy while aimed at maximizing the system reliability. With comprehensive experimental test on the typical and famous four benchmarks of RRAP, PSSO is compared with other recently introduced algorithms in four different widely used systems, i.e., a series system, a series-parallel system, a complex (bridge) system, and an overspeed protection system for a gas turbine. Finally, the results of the experiments demonstrate that the PSSO can effectively solve the system of RRAP with cold-standby strategy and has good performance in the system reliability obtained although the best system reliability is not obtained in all four benchmarks.

1. Introduction

The reliability-redundancy allocation problem (RRAP) is the best known reliability design problem and is a classical optimization problem that seeks to maximize system reliability. To optimize system reliability for RRAP, the development of the system designs involves the selection of the reliability and the redundancy levels of the components. Hence, RRAP belongs to the category of mixed-integer programming problems because the components' reliabilities are denoted as continuous values that fall between zero and one, while the redundancy levels are integer values. RRAP formulations generally involve system constraints on allowable cost, weight, volume, etc.

Based on the system's required functions, the entire system is made up of a specific number of subsystems. The goal of the RRAP is to select the best combination of components and their reliabilities in each subsystem to maximize system reliability, R_s, given constraints such as cost, weight, and volume. In the RRAP literature, the objective is aimed at maximizing the system reliability subjected to several nonlinear constraints [1–4]. The mixed-integer nonlinear optimization programming model for RRAP is formulated as follows to maximize the system reliability by determining the number of components and the component reliabilities in each subsystem:

$$\begin{aligned} \text{Maximize} \quad & R_s = f(\mathbf{R}, \mathbf{N}), \\ \text{Subject to} \quad & g_j(\mathbf{R}, \mathbf{N}) \le u_j, \\ & 1 \le j \le \text{the number of constraints}, \end{aligned} \tag{1}$$

where R_s is the system reliability; $\mathbf{R} = (r_1, r_2 \cdots, r_{\text{nsu}})$ and $\mathbf{N} = (n_1, n_2, \cdots, n_{\text{nsu}})$ are the component reliability vector and the redundancy allocation vector of the system, respectively, where r_i and n_i are, respectively, the reliability of

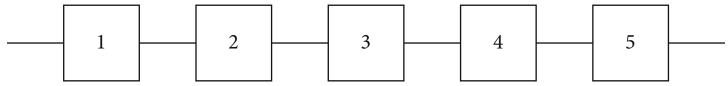

Figure 1: The series system.

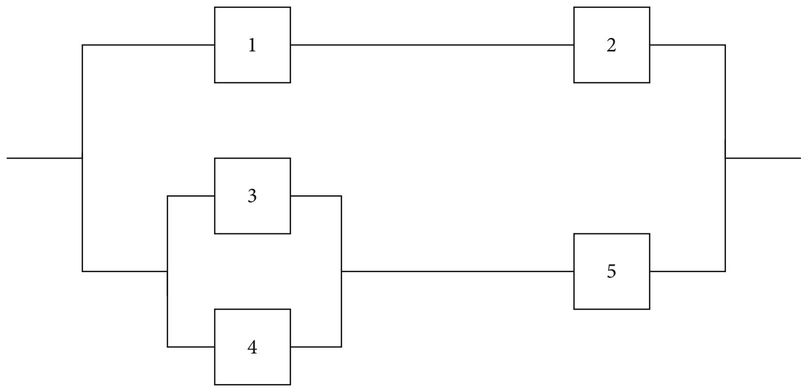

Figure 2: The series-parallel system.

each component and the number of components in subsystem i for $i = 1, 2, \cdots, n_{su}$; $f(\bullet)$ is the objective function for the system reliability; $g_j(\bullet)$ and u_j are the j^{th} constraint function and its resource limitation, respectively.

The main goal of reliability engineering is to increase the system reliability. Two different strategies, i.e., active and cold-standby, are usually used to meet this goal. All components simultaneously start to operate from time zero, for the active strategy, although only one is required at any particular time. The cold-standby strategy first developed and studied on redundancy allocation problem (RAP) by Coit in 2001 [5]; the redundant components are protected from stresses associated with system operation so that no component fails before its start.

There has been much research on different solution algorithms for RRAP. For example, particle swarm optimization (PSO) [3, 6], nondominated sorting genetic algorithm II (NSGA-II) [7–9], artificial bee colony algorithm (ABC) [10], genetic algorithms (GA) [1, 2, 4, 11], simplified swarm optimization (SSO) [12, 13], nest cuckoo optimization algorithm [14], a hybrid of PSO and SSO (PSSO) [15], and stochastic fractal search (SFS) [16] have been employed to study for RRAP.

Most previous research of RRAP in the literature has been devoted to the active strategy [1–4, 8, 10, 13, 15]. Several of these researches of RRAP using the cold-standby strategy can be distinguished which are aimed at solving the multiobjective [9] and focusing on the single objective of maximizing the system reliability [11, 12, 14, 16]. In addition, some researches of RRAP adopt the mixed strategy of active and cold-standby [6, 7]. In this work, the research of RRAP using the cold-standby strategy with single objective of system reliability that experimented on the four typical and famous benchmarks of RRAP including a series system (Figure 1), a series-parallel system (Figure 2), a complex (bridge) system (Figure 3), and an overspeed protection system for a gas turbine (Figure 4) as shown in Section 2.2 is studied. The research in [11, 12, 14, 16] studied RRAP using the cold-standby strategy with single objective of system reliability but only experimented on the first three famous benchmarks of RRAP including a series system, a series-parallel system, and a complex (bridge) system. Therefore, in this paper, the cold-standby strategy is used to increase system reliability in the RRAP formulation while focusing on the single objective of maximizing system reliability, and a solution methodology is presented to optimize system reliability for comprehensive experiments on all four famous benchmarks of RRAP.

Since the early 1990s, soft computing (SC) has been utilized to obtain optimal or good-quality solutions to difficult optimization problems. Swarm intelligence (SI) is a newly developed branch of SC that belongs in the category of population-based stochastic optimization. Particle swarm optimization (PSO) that was first developed by Kennedy and Eberhard in 1995 [17] and simplified swarm optimization (SSO) that was originally exploited by Yeh in 2009 [18] are two of the most well-known algorithms in SI. In recent years, we have seen an increasing interest both in PSO [3, 6, 15, 19–23] and in SSO [12, 13, 15, 24–28] for solving larger problems in science and technology.

The goal of this paper is to optimize the system reliability using RRAP with cold-standby strategy that belongs to the mixed-integer optimization programming model. Therefore, the merits of PSO and SSO, which are used to search for optima in real and discrete numbers, respectively, are adopted in this work. That is, a hybrid algorithm of PSO and SSO (PSSO) [15], which has only been used in RRAP with active strategy, is the first time used to optimize the system reliability using RRAP with cold-standby strategy. To demonstrate the efficiency of PSSO, a comprehensive comparative performance study with another recently introduced algorithm is presented for four different widely used systems. In summary, the novelty and contributions of this work are the RRAP using the cold-standby strategy with

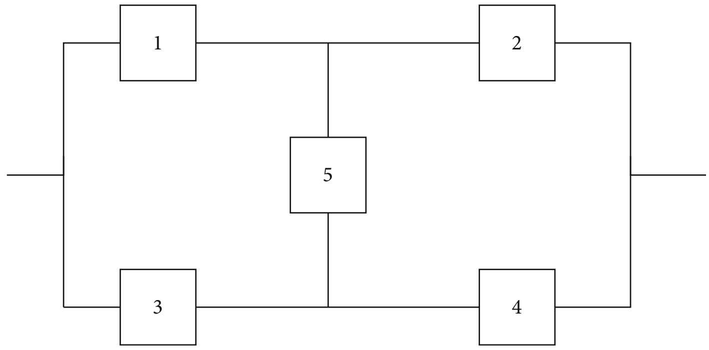

Figure 3: The complex (bridge) system.

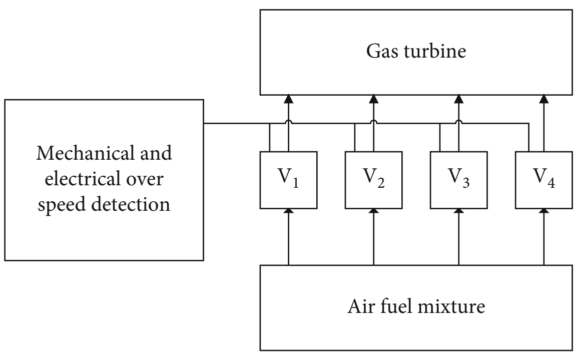

Figure 4: The overspeed protection system for a gas turbine.

single objective of system reliability that has comprehensive experiments on all the four famous benchmarks of RRAP.

This paper is organized as follows. Section 2 presents the mathematical formulation of the cold-standby redundancy strategy for RRAP and four systems. Section 3 provides respective descriptions of PSO and SSO and the orthogonal array test. The PSSO and related UM are discussed in Section 4. A comprehensive comparative study of the performances of PSSO optimizing the four systems is given in Section 5. Finally, the discussion and conclusion are given in Section 6.

2. The Cold-Standby Redundancy RRAP and Four Systems

2.1. The Cold-Standby Redundancy RRAP. Cold-standby redundancy is more difficult to implement than active redundancy because of the necessity to detect failures as they occur and activate the redundant component. If more than one component is used ($n_i > 1$), then there is one initially operating component and $n_i - 1$ components in cold standby waiting to be activated. The subsystem reliability for any distribution of component times-to-failure can be modeled as follows [5, 11].

$$\mathrm{R_i(t) = r_i(t)} + \sum_{\mathrm{k=1}}^{\mathrm{n_i-1}} \int_0^{\mathrm{t}} \mathrm{r_i(t-u) f_i^{(k)}(u) du}. \qquad (2)$$

A detection and switching mechanism is required to sense the occurrence of the component failure and to activate (switch to) a redundant component for cold-standby redundancy. However, the switch itself may fail. For the two imperfect operations, detection and switching, the subsystem reliability for any component time-to-failure distribution with imperfect failure detection and switching can be modeled as follows [5, 11].

Fact 1: continual detection and switching mechanism

$$\mathrm{R_i(t) = r_i(t)} + \sum_{\mathrm{k=1}}^{\mathrm{n_i-1}} \int_0^{\mathrm{t}} \rho_{\mathrm{i}}(\mathrm{u})\, \mathrm{r_i(t-u) f_i^{(k)}(u) du}. \qquad (3)$$

Fact 2: detection and switching mechanism only at time of failure

$$\mathrm{R_i(t) = r_i(t)} + \sum_{\mathrm{k=1}}^{\mathrm{n_i-1}} \int_0^{\mathrm{t}} \rho_{\mathrm{i}}^{\mathrm{k}}\, \mathrm{r_i(t-u) f_i^{(k)}(u) du}. \qquad (4)$$

In this study, we investigate the continual detection and switching mechanism. It is difficult to determine a closed form of Equation (3). A convenient lower bound on subsystem reliability can be determined as follows because $\rho_{\mathrm{i}}(\mathrm{u}) \geq \rho_{\mathrm{i}}(\mathrm{t})$ for all $\mathrm{u} \leq \mathrm{t}$.

$$\mathrm{R_i(t)} \geq \tilde{\mathrm{R}}_{\mathrm{i}}(\mathrm{t}) = \mathrm{r_i(t)} + \rho_{\mathrm{i}}(\mathrm{t}) \sum_{\mathrm{k=1}}^{\mathrm{n_i-1}} \int_0^{\mathrm{t}} \mathrm{r_i(t-u) f_i^{(k)}(u) du}. \qquad (5)$$

The limit of $\mathrm{R_i(t)} - \tilde{\mathrm{R}}_{\mathrm{i}}(\mathrm{t})$ is zero as $\rho_{\mathrm{i}}(\mathrm{t})$ approaches one. Hence, $\rho_{\mathrm{i}}(\mathrm{t})$ is usually close to 1.0 [5].

If the probability distribution of a component's time-to-failure is exponential, then Equation (5) can be expressed by treating the probability of subsystem failure as a homogeneous Poisson process prior to the n_i^{th} failure. In this case, the reliability of the subsystem is the probability that there are strictly less than n_i failures, which is a Poisson distribution with parameter λ_i. Hence [5, 11],

$$\int_0^{\mathrm{t}} \mathrm{r_i(t-u) f_i^{(k)}(u) du} = \frac{\mathrm{e}^{-\lambda_{\mathrm{i}} \mathrm{t}} (\lambda_{\mathrm{i}} \mathrm{t})^{\mathrm{k}}}{\mathrm{k!}}. \qquad (6)$$

A convenient lower bound on subsystem reliability is determined as follows:

$$\tilde{\mathrm{R}}_{\mathrm{i}}(\mathrm{t}) = \mathrm{r}_{\mathrm{i}}(\mathrm{t}) + \rho_{\mathrm{i}}(\mathrm{t}) \sum_{\mathrm{x}=1}^{\mathrm{n_i}-1} \frac{\mathrm{e}^{-\lambda_{\mathrm{i}}\mathrm{t}}(\lambda_{\mathrm{i}}\mathrm{t})^{\mathrm{x}}}{\mathrm{x}!}. \tag{7}$$

In the mathematical formulation of cold-standby redundancy for the RRAP, λ_i and n_i are the two decision variables and r_i is obtained on the basis of λ_i from Equation (6).

2.2. The Four Systems. This paper applies RRAP with cold-standby strategy to four systems: a series system (Figure 1), a series-parallel system (Figure 2), a complex (bridge) system (Figure 3), and an overspeed protection system for a gas turbine (Figure 4).

Due to their structures, the four systems have different objective functions to maximize their reliabilities but are subject to similar multiple nonlinear constraints. The respective RRAPs with cold-standby redundancy are formulated as follows.

System 1. The series system as in Figure 1 [11, 29]

$$\begin{aligned}
\max \quad & f(\mathbf{R}, \mathbf{N}) = \prod_{i=1}^{N_{\mathrm{su}}} R_i(n_i, t), \\
s.t \quad & g_1(\mathbf{R}, \mathbf{N}) = \sum_{i=1}^{N_{\mathrm{su}}} w_i v_i^2 n_i^2 \le V, \\
& g_2(\mathbf{R}, \mathbf{N}) = \sum_{i=1}^{N_{\mathrm{su}}} \alpha_i \left(\frac{-1000}{\ln r_i(t)} \right)^{\beta_i} \left(n_i + \exp\left(\frac{n_i}{4} \right) \right) \le C, \\
& g_3(\mathbf{R}, \mathbf{N}) = \sum_{i=1}^{N_{\mathrm{su}}} w_i n_i \exp\left(\frac{n_i}{4} \right) \le W, \\
& 0 \le r_i(t) \le 1,\ r_i(t) \in \text{real number}, \\
& n_i \in \text{positive integer},\ i = 1, \cdots, N_{\mathrm{su}}.
\end{aligned} \tag{8}$$

System 2. The series-parallel system as in Figure 2 [11, 29]

$$\begin{aligned}
\max \quad & f(\mathbf{R}, \mathbf{N}) = 1 - (1 - R_1(t)R_2(t)) \\
& \qquad \cdot \{1 - [1 - (1 - R_3(t))(1 - R_4(t))]R_5(t)\}, \\
s.t \quad & g_1(\mathbf{R}, \mathbf{N}) \le V, \\
& g_2(\mathbf{R}, \mathbf{N}) \le C, \\
& g_3(\mathbf{R}, \mathbf{N}) \le W, \\
& 0 \le r_i(t) \le 1,\ r_i(t) \in \text{real number}, \\
& n_i \in \text{positive integer},\ i = 1, \cdots, N_{\mathrm{su}}.
\end{aligned} \tag{9}$$

System 3. The complex (bridge) system as in Figure 3 [11, 29]:

$$\begin{aligned}
\max \quad f(\mathbf{R}, \mathbf{N}) = {} & R_1(t)R_2(t) + R_3(t)R_4(t) + R_1(t)R_4(t)R_5(t) \\
& + R_2(t)R_3(t)R_5(t) - R_1(t)R_2(t)R_3(t)R_4(t) \\
& - R_1(t)R_2(t)R_3(t)R_5(t) - R_1(t)R_2(t)R_4(t)R_5(t) \\
& - R_1(t)R_3(t)R_4(t)R_5(t) - R_2(t)R_3(t)R_4(t)R_5(t) \\
& + 2R_1(t)R_2(t)R_3(t)R_4(t)R_5(t), \\
s.t \quad & g_1(\mathbf{R}, \mathbf{N}) \le V, \\
& g_2(\mathbf{R}, \mathbf{N}) \le C, \\
& g_3(\mathbf{R}, \mathbf{N}) \le W, \\
& 0 \le r_i(t) \le 1,\ r_i(t) \in \text{real number}, \\
& n_i \in \text{positive integer},\ i = 1, \cdots, N_{\mathrm{su}}.
\end{aligned} \tag{10}$$

System 4. The overspeed protection system [29]

An RRAP with a cold-standby redundancy formulation of an overspeed protection system with a time-related cost function [29] for a gas turbine is introduced for the first time. The model is formulated as follows.

$$\begin{aligned}
\max \quad & f(\mathbf{R}, \mathbf{N}) = \prod_{i=1}^{N_{\mathrm{su}}} [1\text{-}(1\text{-}r_i(t))^{n_i}], \\
s.t \quad & h_1(\mathbf{R}, \mathbf{N}) = \sum_{i=1}^{N_{\mathrm{su}}} v_i n_i^2 \le V, \\
& h_2(\mathbf{R}, \mathbf{N}) = \sum_{i=1}^{N_{\mathrm{su}}} \alpha_i \left(\frac{-1000}{\ln r_i(t)} \right)^{\beta_i} \left(n_i + \exp\left(\frac{n_i}{4} \right) \right) \le C, \\
& h_3(\mathbf{R}, \mathbf{N}) = \sum_{i=1}^{N_{\mathrm{su}}} w_i n_i \exp\left(\frac{n_i}{4} \right) \le W, \\
& 0.5 \le r_i(t) \le 1\text{-}10^{-6},\ r_i(t) \in \text{real number}, \\
& 1 \le n_i \le 10,\ n_i \in \text{positive integer},\ i = 1, \cdots, N_{\mathrm{su}}.
\end{aligned} \tag{11}$$

3. Preliminaries

The PSO and SSO ought to be expounded at first because the PSSO is the hybrid of PSO and SSO. In addition, an orthogonal array test (OA) is introduced in this study to help improve solution quality and a penalty function is used to deal with constraints.

3.1. The PSO. PSO belongs to the family of swarm intelligence algorithms that was originally developed by Kennedy and Eberhard [17]. A population of random particles is initialized with random positions and velocities in the solution space; these are to be optimized by the fitness function to guide the direction of the solution. In each generation, pBest, denoted as P_i^{l-1}, is the local best solution among Y_i^0, Y_i^1, …,Y_i^{l-1}; each solution has its own pBest. P_{gBest}^{l-1} is the global best, which is the best solution of all existing solutions among all pBests; there is only one gBest at a time. In the l^{th} generation, each solution Y_i^l moves towards pBest P_i^{l-1} and gBest P_{gBest}^{l-1} [19, 20] for $l = 0,1,2,\cdots,N_g$ and $i = 1,2,\cdots,N_{\text{so}}$. The velocities and positions are updated according to the following equation after both P_i^{l-1} and P_{gBest}^{l-1} are found.

$$D_i^l = c_0 \cdot D_i^{l-1} + c_g \cdot \rho_1 \left(P_{\text{gBest}}^{l-1} - Y_i^{l-1}\right) + c_p \cdot \rho_2 \left(P_i^{l-1} - Y_i^{l-1}\right), \tag{12}$$

$$Y_i^l = Y_i^{l-1} + D_i^l, \tag{13}$$

where D_i^l and Y_i^l are the velocity and position of the i^{th} solution at the l^{th} generation, c_0 usually is equal 0.9999, $c_g\rho_1$ and $c_p\rho_2$ are the weights of the search directions, and 4 is the upper bound of $c_g + c_p$ [17].

3.2. The SSO. The SSO belongs to the swarm intelligence family and is a population-based dynamic optimization algorithm that was originally developed by Yeh [18]. It is also initialized with a population of random solutions inside the problem space and then searches for optimal solutions by updating subsequent generations. Let c_w, c_p, c_g, and c_r be the probabilities of the new variable value updated from the variable in the same position of the current solution; the sum of c_w, c_p, c_g, and c_r equals one. The fundamental concept of SSO is that to maintain population diversity and enhance the capacity to escape from a local optimum [18], each variable of any solution needs to be updated to a value related to its current value, its current pBest, the gBest, or a random feasible value. A random movement of SSO is based on the following model after c_w, c_p, and c_g are given:

$$x_{ij}^l = \begin{cases} x_{ij}^{l-1} & \text{if } \rho_{[0,1]} \in [0, C_w = c_w), \\ p_{ij}^{l-1} & \text{if } \rho_{[0,1]} \in \left[C_w, C_p = C_w + c_p\right), \\ g_j & \text{if } \rho_{[0,1]} \in \left[C_p, C_g = C_p + c_g\right), \\ x & \text{if } \rho_{[0,1]} \in \left[C_g, 1\right), \end{cases} \tag{14}$$

where $i = 1,2,\cdots,\text{N}_{\text{so}}$, $j = 1,2,\cdots,N_{\text{su}}$, $l = 0,1,2,\cdots,N_g$, and x is a random number between the lower and upper bounds of the j^{th} variable.

3.3. The Orthogonal Array Test (OA). This paper adopts the orthogonal array test (OA) to improve solutions because the OA is helpful to systematically and efficiently produce a potentially good approximation [30, 31]. Table 1 illustrates the class of the three-level OA where the numbers 1, 2, and 3 in each column indicate the levels of factors, and an equal number of 1s, 2s, and 3s is contained in each column. Columns 1-3 are the factors of the parameters c_w, c_p, and c_g in Equation (14), and column 4 is the factor of the parameter c_0 in Equation (12).

Table 1: The orthogonal array.

Tests	Columns			
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table 2: The parameter values of SSO.

c_w	c_p	c_g	Reference
0.1	0.3	0.5	[39]
0.15	0.25	0.35	
0.1	0.3	0.5	[32]
0.1	0.3	0.3	
0.1	0.3	0.5	[34]
0.15	0.25	0.35	[35]
0.15	0.25	0.35	[30]
0.15	0.35	0.25	[33]
0.15	0.35	0.25	[36]
0.1	0.3	0.5	[37]
0.2	0.1	0.19	[38]

Table 3: The 3 most frequent parameter values of SSO.

c_w	Frequency	c_p	Frequency	c_g	Frequency
0.10	5	0.25	3	0.25	2
0.15	5	0.30	5	0.35	3
0.20	1	0.35	2	0.50	4

Table 4: The combinations of the parameter values.

	Factors			
		SSO		PSO
Tests	c_g	c_p	c_w	c_0
1	0.25	0.25	0.10	0.9999
2	0.25	0.30	0.15	Equation (15)
3	0.25	0.35	0.20	Equation (16)
4	0.35	0.25	0.15	Equation (16)
5	0.35	0.30	0.20	0.9999
6	0.35	0.35	0.10	Equation (15)
7	0.50	0.25	0.20	Equation (15)
8	0.50	0.30	0.10	Equation (16)
9	0.50	0.35	0.15	0.9999

The numbers of parameter values that were set according to the published papers of the SSO algorithm [30, 32–39] are arranged in Table 2. The top three parameter values that are most frequently set are $c_w = 0.1, 0.15, 0.2$, $c_p = 0.25, 0.3, 0.35$, and $c_g = 0.25, 0.35, 0.5$, as presented in Table 3.

The three parameter values that are most frequently set in PSO are 0.9999, linearly decreasing as in Equation (15) and exponentially decreasing as in Equation (16).

$$c_0 = 0.9999 - g * \frac{0.1}{N_g}, \tag{15}$$

$$c_0 = c_0 \cdot \sqrt[N_g]{\frac{.9}{c_0}}. \tag{16}$$

The nine combinations of the parameter values are presented in Table 4 from the information above.

3.4. The Penalty Function. A penalty function as shown in Equation (17) is used to deal with constraints. That is, the penalty function in Equation (17) is a penalty mechanism for system reliability if any constraint exceeds the upper limit of cost, weight, or volume.

$$R_{\text{penalty}} = \begin{cases} R_s & \text{if } X = (N, \Lambda) \text{is a feasible solution,} \\ R_s\left(\min\left\{\left[\dfrac{V}{g_1(R,N)}\right], \left[\dfrac{C}{g_2(R,N)}\right], \left[\dfrac{W}{g_3(R,N)}\right],\right\}\right)^{\pi} & \text{otherwise,} \end{cases} \tag{17}$$

where R_{penalty} is the system reliability confirmed by the penalty function.

4. The PSSO

In this section, the PSSO is used together with an all-variable-UM (here termed n-UM and λ-UM) that retains the merits of PSO and SSO that are beneficial in searching for the optima in real and discrete numbers, respectively [15, 17–20, 32–39]. This study considers how the simultaneous application of those merits is conducive to computing the RRAP with cold-standby strategy, which is a mixed-integer programming model. Therefore, the key characteristics and merits of PSO and SSO are applied in this paper to optimize RRAP with cold-standby strategy.

The parameters λ_i of the Poisson distribution in Equation (7) and the numbers of all components need to be determined in cold-standby redundancy RRAP. Each number of components is an integer, and each parameter λ_i of a component is a real number. Hence, two different UMs, termed n-UM and λ-UM, are proposed to update the numbers of all components and the parameters λ_i of all components.

4.1. The n-UM. Applying the SSO, the proposed n-UM updates the number of components, i.e., n_{su} for $\text{su} = 1, 2, \cdots, N_{\text{su}}$, in each subsystem. Let $\text{so} = 1, 2, \cdots, N_{\text{so}}$, $\text{su} = 1, 2, \cdots, N_{\text{su}}$, $g = 1, 2, \cdots, N_g$, and n be a random number between the lower and upper bounds of the su$^{\text{th}}$ variable, and the mathematical model is as follows:

$$N_{\text{so,su}}^{g} = \begin{cases} \hat{n}_{\text{gBest},su} & \text{if } \rho_{[0,1]} \in \left[0, C_g = c_g\right), \\ \hat{n}_{\text{so,su}} & \text{if } \rho_{[0,1]} \in \left[C_g, C_p = C_g + c_p\right), \\ n_{\text{so,su}}^{g\text{-}1} & \text{if } \rho_{[0,1]} \in \left[C_p, C_w = C_p + c_w\right), \\ n & \text{if } \rho_{[0,1]} \in [C_w, 1]. \end{cases} \tag{18}$$

4.2. The λ-UM. Applying the velocity function of PSO, the proposed λ-UM updates the parameters λ_i of the Poisson distributions (Equation (7)) of the components in each subsystem. The mathematical model is as follows:

$$\Lambda_{\text{so}}^{g} = c_0 \cdot \Lambda_{\text{so}}^{g\text{-}1} + c_g \cdot \rho_1\left(\hat{\Lambda}_{\text{gBest}}^{g\text{-}1} - \Lambda_{\text{so}}^{g\text{-}1}\right) + c_p \cdot \rho_1\left(\hat{\Lambda}_{\text{so}}^{g\text{-}1} - \Lambda_{\text{so}}^{g\text{-}1}\right). \tag{19}$$

4.3. Pseudo-Code for PSSO. The pseudo-code of PSSO is as follows.

Step 0. Generate $X_{\text{so}}^0 = (N_{\text{so}}^0, \Lambda_{\text{so}}^0)$ randomly, calculate $F(X_{\text{so}}^0)$, and let $P_{\text{so}} = X_{\text{so}}^0$ and $F(P_{\text{so}}) = F(X_{\text{so}}^0)$ for $\text{so} = 1, 2, \cdots, N_{\text{so}}$.

Step 1. Find gBest such that $F(X_{\text{so}}^0) \le F(X_{\text{gBest}}^0)$ for $\text{so} = 1, 2, \cdots, N_{\text{so}}$.

Step 2. Let $g = 1$.

Step 3. Update N_{so}^g and Λ_{so}^g based on Equations (18) and (19).

Step 4. If $F(P_{\text{so}}) < F(X_{\text{so}}^g)$, let $P_{\text{so}} = X_{\text{so}}^g$ and $F(P_{\text{so}}) = F(X_{\text{so}}^g)$ and go to Step 5. Otherwise, go to Step 6.

Step 5. If $F(P_{\text{gBest}}) < F(P_{\text{so}})$, let $\text{gBest} = \text{so}$.

Step 6. If $\text{so} < N_{\text{so}}$, let $\text{so} = \text{so} + 1$ and go to Step 3.

Step 7. If $g < N_g$, let $g = g + 1$ and go to Step 2. Otherwise, halt.

Table 5: The parameter values used in Systems 1 and 3.

Subsystem i	$10^5\alpha_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.330	1.5	1	7			
2	1.450	1.5	2	8			
3	0.541	1.5	3	8	110	175	200
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

Table 6: The parameter values used in System 2.

Subsystem i	$10^5\alpha_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.500	1.5	2	3.5			
2	1.450	1.5	4	4.0			
3	0.541	1.5	5	4.0	180	175	100
4	0.541	1.5	8	3.5			
5	2.100	1.5	4	4.5			

Table 7: The parameter values used in System 4.

Subsystem i	$10^5\alpha_i$	β_i	v_i	w_i	V	C	W
1	1	1.5	1	6			
2	2.3	1.5	2	6	250.0	400.0	500.0
3	0.3	1.5	3	8			
4	2.3	1.5	2	7			

Table 8: The reliability of System 1 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.96957732	0.96385617	0.91263728	0.00815955
2	0.96956482	0.96500015	0.95140225	0.00434131
3	0.99700404	0.92272200	0.82028328	0.04904667
4	0.99656129	0.91877856	0.76842776	0.05594605
5	0.96957924	0.95894940	0.87581632	0.01622886
6	0.96957612	0.96423395	0.93441124	0.00538507
7	0.96957233	0.94684495	0.81660930	0.02901255
8	0.99651026	0.92788871	0.81550324	0.05106236
9	0.96957926	0.88761690	0.68355049	0.07417057
Average	0.97861385	0.93954342	0.84207124	0.03259477

5. Experimental Results

While this paper aims at optimizing the system reliability, the studied RRAP with cold-standby strategy comprehensively applied to the typical and well-known four systems described in Section 2.2: a series system, a series-parallel system, a complex (bridge) system, and an overspeed protection system for a gas turbine is solved by PSSO [11, 29].

The PSSO implemented for RRAP with cold-standby strategy including the four systems was coded in the C^{++} programming language and run on an Intel Core i7 3.07 GHz PC with 6 GB memory. The experiments used 1000 generations ($N_g = 1000$), the number of solutions was $N_{so} = 100$, the mission time $t = 1000$, and the convenient lower bound on subsystem reliability $\rho_i(t) = 0.99$.

Four systems are provided to evaluate the performance of PSSO for cold-standby RRAP, which is the mixed-integer nonlinear reliability design. The corresponding input

Table 9: The reliability of System 2 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.99998828	0.99998567	0.99997026	0.00000451
2	0.99998827	0.99998617	0.99997358	0.00000360
3	0.99998826	0.99998677	0.99997358	0.00000332
4	0.99998826	0.99998517	0.99996740	0.00000534
5	0.99998828	0.99997617	0.99957038	0.00004333
6	0.99998827	0.99998245	0.99994913	0.00000724
7	0.99998827	0.99995877	0.99971734	0.00004601
8	0.99998826	0.99997295	0.99981548	0.00002908
9	0.99998828	0.99944028	0.99314011	0.00138911
Average	0.99998827	0.999919378	0.999119696	0.000170171

Table 10: The reliability of System 3 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.99997538	0.99997142	0.99990226	0.00000963
2	0.99997535	0.99997288	0.99993734	0.00000603
3	0.99997532	0.99997305	0.99990998	0.00000703
4	0.99997532	0.99997176	0.99992335	0.00000656
5	0.99997538	0.99995309	0.99930499	0.00006959
6	0.99997536	0.99996981	0.99991011	0.00001123
7	0.99997538	0.99986639	0.99850974	0.00020563
8	0.99997535	0.99995631	0.99979033	0.00003090
9	0.99997538	0.99936835	0.98981897	0.00141941
Average	0.999975358	0.999889229	0.998556341	0.000196223

Table 11: The reliability of System 4 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.99679154	0.99678469	0.99674692	0.00001589
2	0.99679122	0.99678877	0.99676663	0.00000382
3	0.99679135	0.99678779	0.99678178	0.00000189
4	0.99679140	0.99678700	0.99674419	0.00000507
5	0.99679154	0.99677863	0.99674708	0.00002024
6	0.99679144	0.99678831	0.99674626	0.00000772
7	0.99679149	0.99677851	0.99669826	0.00002050
8	0.99679121	0.99678736	0.99674322	0.00000985
9	0.99679154	0.99320681	0.90823357	0.01352766
Average	0.99679141	0.99638754	0.98691199	0.00151251

Table 12: Comparison of PSSO with previous work for System 1.

	PSO	SSO	GA, 2014 [11]	New SSO, 2019 [12]	ENCOA, 2020 [14]	SFS, 2019 [16]	PSSO
R_s	0.96268903	0.95989227	0.96957758	0.96957924	0.99999	0.969579	0.99700404
N	(2,3,2,3,3)	(3,2,2,4,2)	(3, 2, 2, 3, 3)	(3, 2, 2, 3, 3)	(1, 3, 3, 2, 2)	(3, 2, 2, 3, 3)	(3,4,4,3,3)
λ_1	0.00015369	0.00030518	0.00026841	0.00026587	0.000596	0.000266	0.00003271
λ_2	0.00036900	0.00015259	0.00011931	0.00011924	0.000661	0.000119	0.00008385
λ_3	0.00006705	0.00009572	0.00008840	0.00008856	0.000546	0.000089	0.00002787
λ_4	0.00036014	0.00073242	0.00036600	0.00036728	0.000624	0.000367	0.00006836
λ_5	0.00026303	0.00009942	0.00025356	0.00025387	0.000559	0.000254	0.00008259
r_1	0.85754177	0.73699381	0.76459335	0.76653	0.742386	0.766498	0.96782041
r_2	0.69142591	0.85848344	0.88752892	0.88759	0.816868	0.887576	0.91957190
r_3	0.93514551	0.90871712	0.91539527	0.91525	0.713963	0.915293	0.97251610
r_4	0.69757716	0.48074328	0.69350544	0.69261	0.575015	0.692567	0.93392478
r_5	0.76871676	0.90536223	0.77603145	0.77579	0.738721	0.775825	0.92072882
MPI	91.97%	92.53%	90.15%	90.15%		90.15%	

Table 13: Comparison of PSSO with previous work for System 2.

	PSO	SSO	GA, 2014 [11]	New SSO, 2019 [12]	ENCOA, 2020 [14]	SFS, 2019 [16]	PSSO
R_s	0.99998418	0.99996929	0.99998824	0.99998827	0.99999999	0.99998827	0.99998828
N	(3,3,1,2,3)	(3,3,1,2,3)	(3,3,2,1,3)	(3,3,1,2,3)	(3,2,1,3,3)	(3,3,1,2,3)	(3,3,1,2,3)
λ_1	0.00020286	0.00030518	0.00019255	0.00019156	0.00042564	0.00019186	0.00019217
λ_2	0.00018530	0.00027466	0.00017100	0.00016417	0.00017187	0.00016498	0.00016433
λ_3	0.00023161	0.00006104	0.00009632	0.00010691	0.00033426	0.00010705	0.00010766
λ_4	0.00007707	0.00005276	0.00010680	0.00009639	0.00033807	0.00009509	0.00009650
λ_5	0.00016042	0.00018311	0.00014449	0.00014839	0.00025940	0.00014847	0.00014768
r_1	0.81638904	0.73699381	0.82484672	0.82567	0.68245868	0.82542444	0.82516974
r_2	0.83085895	0.75983179	0.84281657	0.84860	0.86473794	0.84791146	0.84846299
r_3	0.79325626	0.94079016	0.90817308	0.89860	0.69972896	0.89848373	0.89793059
r_4	0.92582234	0.94860880	0.89869900	0.90811	0.73210357	0.90929419	0.90800596
r_5	0.85178644	0.83268033	0.86546301	0.86209	0.70098038	0.86202871	0.86270308
MPI	25.91%	61.83%	0.34%	0.09%		0.09%	

data and parameters are the same as in [11, 29] and are presented in the supplementary file "Data.docx" (available here) and Tables 5–7, respectively.

The experimental results solved by PSSO in terms of the statistical analysis for the maximum (the best), mean, minimum (the worst), and standard deviation of the related nine combinations, using the OA introduced in Section 3.3, are presented in Tables 8–11. The best solutions for the system reliability are 0.99700404, 0.99998828, 0.99997538, and 0.99679154 for the four systems, respectively.

Finally, Tables 12–14 illustrate the best performances of PSSO in comparison with previous results [11, 12, 14, 16], the PSO, and the SSO algorithms for the first three systems, respectively. The cold-standby strategy for RRAP is applied to the fourth system, i.e., overspeed protection of a gas turbine, for the first time, considering the PSSO applies the combined merits of PSO and SSO. Hence, the best performance of PSSO in comparison with the PSO and the SSO algorithms for the fourth system is illustrated in Table 15.

In Tables 12–15, the second row illustrates the solution to the system reliability. Other rows containing N, λ_i, and r_i indicate the number of components, the parameters of the Poisson distribution in Equation (7), and the reliability of components in each subsystem, respectively. Finally, the MPI rows give the improvements of the solutions found by the proposed solution over those of the best known previous solutions; the calculation equation is $(R_{s\_\text{PSSO}} - R_{s\_\text{other}})/(1 - R_{s\_\text{other}})$, where R_{s_PSSO} indicates the system reliability obtained by PSSO and R_{s_other} indicates the system reliability obtained by a previous algorithm.

The results demonstrate that the PSSO performs better than the PSO and the SSO in terms of system reliability for the fourth system. The results of the first three systems in terms of system reliability obtained by ENCOA [14] are

Table 14: Comparison of PSSO with previous work for System 3.

	PSO	SSO	GA, 2014 [11]	New SSO, 2019 [12]	ENCOA, 2020 [14]	SFS, 2019 [16]	PSSO
R_s	0.99996602	0.99994009	0.99997413	0.99997537	0.99999995	0.99997538	0.99997538
N	(3,3,3,3,1)	(3,3,2,3,2)	(3,3,3,3,1)	(3,3,2,4,1)	(2,3,2,3,3)	(3,3,2,4,1)	(3,3,2,4,1)
λ_1	0.00026983	0.00033569	0.00021744	0.00019913	0.00048053	0.00019954	0.00020096
λ_2	0.00018939	0.00009709	0.00015412	0.00015582	0.00035308	0.00015658	0.00015568
λ_3	0.00016766	0.00006783	0.00014232	0.00006808	0.00058080	0.00006839	0.00006777
λ_4	0.00030117	0.00054932	0.00031802	0.00049070	0.00052360	0.00048775	0.00048910
λ_5	0.00017940	0.00045776	0.00026897	0.00027934	0.00028347	0.00027585	0.00027874
r_1	0.76351099	0.71484227	0.80457234	0.81944	0.70647573	0.81911028	0.81794334
r_2	0.82746317	0.90747281	0.85717305	0.85571	0.68327513	0.85506338	0.85583287
r_3	0.84564286	0.93442369	0.86734683	0.93418	0.88812138	0.93389858	0.93447620
r_4	0.73994868	0.57734434	0.72759162	0.61219	0.61431138	0.61400415	0.61317957
r_5	0.83577161	0.63269698	0.76416666	0.75628	0.70242105	0.75892956	0.75673635
MPI	27.54%	58.90%	4.83%	0.04%			

Table 15: Comparison of PSSO with PSO and SSO for System 4.

	PSO	SSO	PSSO
R_s	0.99670304	0.99677964	0.99679154
N	(3,3,3,3)	(3,3,3,3)	(3,3,3,3)
λ_1	0.00007028	0.00006779	0.00007003
λ_2	0.00009144	0.00009527	0.00009313
λ_3	0.00005319	0.00004520	0.00004502
λ_4	0.00009473	0.00009376	0.00009315
r_1	0.93213157	0.93446076	0.93236397
r_2	0.91261420	0.90913042	0.91107606
r_3	0.94820411	0.95580962	0.95597900
r_4	0.90962070	0.91050480	0.91105944
MPI	2.68%	0.36%	

the best but the results found by PSSO are the second best. The detailed comparisons are as follows.

(1) The system reliabilities R_s of 0.99999, 0.99999999, and 0.99999995 obtained by ENCOA [14] for the first three systems are better than PSSO, those of the previous work, the PSO, and the SSO

(2) The system reliability R_s of 0.99679154 obtained by PSSO for the fourth system is better than the PSO and the SSO

(3) However, the system reliabilities R_s of 0.99700404, and 0.99998828 obtained by PSSO for the first two systems are better than those of the previous work, the PSO, and the SSO except ENCOA [14], and the system reliability R_s of 0.99997538 obtained by PSSO for the third system is better than those of the previous work, the PSO, and the SSO except ENCOA [14] and SFS [16].

6. Conclusion and Future Work

A particle-based version of SSO called PSSO with a new UM to enhance the ability of traditional SSO is used to solve RRAP with cold-standby strategy. The RRAP cold-standby effectively maximizes the system reliability with the PSSO. Moreover, the UM is an important part of soft computing. This paper presents significant and novel modifications to SSO to optimize the cold-standby redundancy RRAP.

A comprehensive comparative study of the performances of the PSSO and previous work has been made. The system reliability obtained by the PSSO is better than the PSO and SSO for the fourth system. The system reliability obtained by the PSSO is the second best; those are second to ENCOA [14] for the first three systems. Roughly speaking, the PSSO based on UM has the ability to optimize the mixed-integer programming model and can be used to solve cold-standby redundancy RRAP efficiently. In future research, we will focus on strengthening SSO performance and will apply it to different optimization problems and solve practical engineering problems with larger-scale systems.

Acronyms

PSO: Particle swarm optimization
SSO: Simplified swarm optimization
PSSO: Particle-based simplified swarm optimization
RAP: Redundancy allocation problem
RRAP: Reliability-redundancy allocation problem
pBest: Local best
gBest: Global best
n-UM: Proposed update mechanism for the number variables of all components
λ-UM: Proposed update mechanism for the λ variables of all components
MPI: Maximum possible improvement.

Notations

n_{su}, n_{so}, n_g: The number of subsystems in the system, solutions, and generations, respectively

$\mathbf{N}$: $\mathbf{N} = (n_1, n_2, \cdots, n_{su})$ is the redundancy allocation vector of the system, where n_i is the number of components in subsystem i for $i = 1, 2, \cdots, N_{su}$

$\mathbf{R}$: $\mathbf{R} = (r_1, r_2, \cdots, r_{Nsu})$ is the component reliability vector of the system, where r_i is the reliability of each component in subsystem i for $i = 1, 2, \cdots, N_{su}$

$R_i(\bullet), q_i$: $R_i(\bullet) = 1 - q_i^{n_i}$ is the reliability of subsystem i, where $q_i = 1 - r_i$ is the failure probability of each component in subsystem i for $i = 1, 2, \cdots, N_{su}$

$\tilde{R}_i(\bullet)$: The convenient lower-bound reliability of subsystem i

$r_i(t)$: The reliability of each component in subsystem i at the mission time t

$f_i^{(k)}$: The pdf of subsystem i at the k^{th} failure arrival for $k = 0, 1, \cdots, n_{su} - 1$

$\rho_i(\bullet)$: The failure detection/switching reliability

λ_i: The parameter of the Poisson distribution in subsystem i

R_s: The system reliability

$g_j(\mathbf{R}, \mathbf{N})$: The j^{th} constraint function with respect to $\mathbf{R}$ and $\mathbf{N}$

α_i, β_i: The physical feature of each component in subsystem i for $i = 1, 2, \cdots, N_{var}$

u_j: The resource limitation for the j^{th} constraint function

v_i, c_i, w_i: The volume, cost, and weight, respectively, of each component in subsystem i, $i = 1, 2, \cdots, N_{var}$

V, C, W: The upper limits on the volume, cost, and weight of the system, respectively

$f(\mathbf{R}, \mathbf{N})$: The fitness function with respect to $\mathbf{R}$ and $\mathbf{N}$

$N_{so}^g, n_{so,su}^g$: $N_{so}^g = (n_{so,1}^g, n_{so,2}^g, \cdots, n_{so,Nsu}^g)$ is the redundancy allocation vector of the so$^{\text{th}}$ solution at the g^{th} generation, where $n_{so,su}^g$ is the su$^{\text{th}}$ variable for su $= 1, 2, \cdots, N_{su}$

$\Lambda_{so}^g, \lambda_{so,su}^g$: $\Lambda_{so}^g = (\lambda_{so,1}^g, \lambda_{so,2}^g, \cdots, \lambda_{so,Nsu}^g)$ is the λ vector of the component parameters of the so$^{\text{th}}$ solution at the g^{th} generation, where $\lambda_{so,su}^g$ is the su$^{\text{th}}$ variable for su $= 1, 2, \cdots, N_{su}$

X_{so}^g: $X_{so}^g = (N_{so}^g, \Lambda_{so}^g)$ is the so$^{\text{th}}$ solution at the g^{th} generation

$\bullet^\wedge, \bullet^\wedge_{gBest}$: The related pBest and gBest.

Data Availability

Data are available in the supplementary information file.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Conceptualization, software, investigation, resources, and writing—original draft preparation were contributed by all authors; methodology was contributed by Chia-Ling Huang; validation was contributed by Yunzhi Jiang; formal analysis was contributed by Yunzhi Jiang and Chia-Ling Huang; data curation was contributed by Chia-Ling Huang; supervision was contributed by Yunzhi Jiang and Chia-Ling Huang; project administration was contributed by Chia-Ling Huang; funding acquisition was contributed by Yunzhi Jiang. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China and Ministry of Science and Technology, Taiwan, R.O.C., under grant 61702118, MOST 105-2221-E-424-003, and MOST 109-2221-E-424-002.

Supplementary Materials

The corresponding input data and parameters are the same as in [11, 29] and are presented in the supplementary file "Data.docx" and Tables 5–7, respectively. *(Supplementary Materials)*

References

[1] L. Sahoo, "Reliability redundancy allocation problems under fuzziness using genetic algorithm and dual-connection numbers," in *Nature-Inspired Computing Paradigms in Systems*, pp. 111–123, Elsevier, 2021.

[2] H. A. Khorshidi, I. Gunawan, and M. Y. Ibrahim, "A value-driven approach for optimizing reliability-redundancy allocation problem in multi-state weighted k-out-of-n system," *Journal of Manufacturing Systems.*, vol. 40, no. 1, pp. 54–62, 2016.

[3] E. Zhang and Q. Chen, "Multi-objective reliability redundancy allocation in an interval environment using particle swarm optimization," *Reliability Engineering & System Safety.*, vol. 145, pp. 83–92, 2016.

[4] H. Kim and P. Kim, "Reliability-redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm," *Reliability Engineering & System Safety.*, vol. 159, pp. 153–160, 2017.

[5] D. W. Coit, "Cold-standby redundancy optimization for non-repairable systems," *IIE Transactions.*, vol. 33, no. 6, pp. 471–478, 2001.

[6] Z. Ouyang, Y. Liu, S. J. Ruan, and T. Jiang, "An improved particle swarm optimization algorithm for reliability-redundancy allocation problem with mixed redundancy strategy and heterogeneous components," *Reliability Engineering & System Safety.*, vol. 181, pp. 62–74, 2019.

[7] W. Wang, M. Lin, Y. Fu, X. Luo, and H. Chen, "Multi-objective optimization of reliability-redundancy allocation problem for multi-type production systems considering redundancy strategies," *Reliability Engineering & System Safety.*, vol. 193, p. 106681, 2020.

[8] P. K. Muhuri, Z. Ashraf, and Q. M. D. Lohani, "Multiobjective reliability redundancy allocation problem with interval type-2 fuzzy uncertainty," *IEEE Transactions on Fuzzy Systems.*, vol. 26, no. 3, pp. 1339–1355, 2018.

[9] M. Abouei Ardakan and M. T. Rezvan, "Multi-objective optimization of reliability-redundancy allocation problem with cold-standby strategy using NSGA-II," *Reliability Engineering & System Safety.*, vol. 172, pp. 225–238, 2018.

[10] W. C. Yeh and T. J. Hsieh, "Solving reliability redundancy allocation problems using an artificial bee colony algorithm," *Computers & Operations Research.*, vol. 38, no. 11, pp. 1465–1473, 2011.

[11] M. Abouei Ardakan and A. Zeinal Hamadani, "Reliability-redundancy allocation problem with cold-standby redundancy strategy," *Simulation Modelling Practice and Theory.*, vol. 42, pp. 107–118, 2014.

[12] W. C. Yeh, "Solving cold-standby reliability redundancy allocation problems using a new swarm intelligence algorithm," *Applied Soft Computing.*, vol. 83, p. 105582, 2019.

[13] W. C. Yeh, Y. Z. Su, X. Z. Gao, C. F. Hu, J. Wang, and C. L. Huang, "Simplified swarm optimization for bi-objection active reliability redundancy allocation problems," *Applied Soft Computing Journal.*, vol. 106, p. 107321, 2021.

[14] M. ArezkiMellal and E. Zio, "System reliability-redundancy optimization with cold-standby strategy by an enhanced nest cuckoo optimization algorithm," *Reliability Engineering & System Safety.*, vol. 201, p. 106973, 2020.

[15] C. L. Huang, "A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems," *Reliability Engineering & System Safety.*, vol. 142, pp. 221–230, 2015.

[16] M. N. Juybari, M. A. Ardakan, and H. Davari-Ardakani, "A penalty-guided fractal search algorithm for reliability–redundancy allocation problems with cold-standby strategy," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 233, no. 5, pp. 775–790, 2019.

[17] J. Kennedy and R. C. Eberhard, "Particle swarm optimization," in *proceedings of IEEE international conference on neural networks*, pp. 1942–1948, Publishing, Piscataway, NJ, USA, 1995.

[18] W. C. Yeh, "A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems," *Expert Systems with Applications.*, vol. 36, no. 5, pp. 9192–9200, 2009.

[19] G. R. You, Y. R. Shiue, W. C. Yeh, X. L. Chen, and C. M. Chen, "A weighted ensemble learning algorithm based on diversity using a novel particle swarm optimization approach," *Algorithms.*, vol. 13, no. 10, p. 255, 2020.

[20] L. Li, L. Chang, T. Gu, W. Sheng, and W. Wang, "On the norm of dominant difference for many-objective particle swarm optimization," *IEEE Transactions on Cybernetics.*, vol. 51, no. 4, pp. 2055–2067, 2021.

[21] X. F. Song, Y. Zhang, D. W. Gong, and X. Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[22] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Evolutionary Computation.*, vol. 25, no. 4, pp. 794–808, 2021.

[23] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Transactions on Cybernetics.*, vol. 51, no. 2, pp. 874–888, 2021.

[24] W. C. Yeh, "A new harmonic continuous simplified swarm optimization," *Applied Soft Computing.*, vol. 85, p. 105544, 2019.

[25] W. C. Yeh and J. S. Lin, "New parallel swarm algorithm for smart sensor systems redundancy allocation problems in the Internet of Things," *Journal of Supercomputing.*, vol. 74, no. 9, pp. 4358–4384, 2018.

[26] X. Zhang, W. C. Yeh, Y. Jiang, Y. Huang, Y. Xiao, and L. Li, "A case study of control and improved simplified swarm optimization for economic dispatch of a stand-alone modular microgrid," *Energies*, vol. 11, no. 4, p. 793, 2018.

[27] W. C. Yeh, "A novel boundary swarm optimization method for reliability redundancy allocation problems," *Reliability Engineering & System Safety.*, vol. 192, p. 106060, 2019.

[28] W. C. Yeh, C. L. Huang, P. Lin, Z. Chen, Y. Jiang, and B. Sun, "Simplex simplified swarm optimisation for the efficient optimisation of parameter identification for solar cell models," *IET Renewable Power Generation.*, vol. 12, no. 1, pp. 45–51, 2018.

[29] T. C. Chen, "IAs based approach for reliability redundancy allocation problems," *Applied Mathematics and Computation.*, vol. 182, no. 2, pp. 1556–1567, 2006.

[30] W. C. Yeh, "Novel swarm optimization for mining classification rules on thyroid gland data," *Information Sciences.*, vol. 197, pp. 65–76, 2012.

[31] S. J. Ho, S. Y. Ho, and L. S. Shu, "OSA: orthogonal simulated annealing algorithm and its application to designing mixed<tex>$rm H_2 /rm H_infty$</tex>optimal controllers," *Man and Cybernetics, Part A: Systems and Humans.*, vol. 34, no. 5, pp. 588–600, 2004.

[32] W. C. Yeh, "Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans.*, vol. 42, no. 1, pp. 250–261, 2012.

[33] W. C. Yeh, "New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series," *IEEE Transactions on Neural Networks and Learning Systems.*, vol. 24, no. 4, pp. 661–665, 2013.

[34] C. Bae, W. C. Yeh, N. Wahid, Y. Y. Chung, and Y. Liu, "A new simplified swarm optimization (SSO) using exchange local search scheme, International Journal of Innovative Computing," *Information and Control*, vol. 8, no. 6, pp. 4391–4406, 2012.

[35] W. C. Yeh, C. M. Lin, and S. C. Wei, "Disassembly sequencing problems with stochastic processing time using simplified swarm optimization, International Journal of Innovation," *Management and Technology.*, vol. 3, no. 3, pp. 226–231, 2012.

[36] W. C. Yeh, Y. M. Yeh, P. C. Chang, Y. C. Ke, and V. Chung, "Forecasting wind power in the Mai Liao Wind Farm based on the multi-layer perceptron artificial neural network model with improved simplified swarm optimization," *Electrical Power and Energy Systems.*, vol. 55, pp. 741–748, 2014.

[37] C. Bae, N. Wahid, Y. Y. Chung, W. C. Yeh, N. W. Bergmann, and Z. Chen, "Effective audio classification algorithm using swarm-based optimization, International Journal of Innovative

Computing," *Information and Control*, vol. 7, no. 1, pp. 1–10, 2011.

[38] W. C. Yeh, "Orthogonal simplified swarm optimization for the series–parallel redundancy allocation problem with a mix of components," *Knowledge-Based Systems.*, vol. 64, pp. 1–12, 2014.

[39] C. Bae, W. C. Yeh, Y. Y. Chung, and S. L. Liu, "Feature selection with intelligent dynamic swarm and rough set," *Expert Systems with Applications.*, vol. 37, no. 10, pp. 7026–7032, 2010.