



**HAL**  
open science

# Redundancy optimization for multi-state series-parallel systems using ordinal optimization-based-genetic algorithm

Yishuang Hu, Yi Ding, Zhiguo Zeng

► **To cite this version:**

Yishuang Hu, Yi Ding, Zhiguo Zeng. Redundancy optimization for multi-state series-parallel systems using ordinal optimization-based-genetic algorithm. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 2021, pp.1748006X2110286. 10.1177/1748006X211028699 . hal-03464075

**HAL Id: hal-03464075**

**<https://hal.science/hal-03464075>**

Submitted on 23 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Redundancy optimization for multi-state series-parallel systems using ordinal optimization-based-genetic algorithm

XXXXX<sup>1</sup>, XXXXXXX<sup>1</sup>

## Abstract

Multi-state series-parallel systems (MSSPSs) are widely-used for representing engineering systems. In real-life cases, engineers need to design an optimal MSSPS structure by combining different versions and number of redundant components. The objective of the design is to ensure reliability requirements using the least costs, which could be formulated as a redundancy optimization problem under reliability constraints. The genetic algorithm is one of the most frequently used method for solving redundancy optimization problems. In traditional genetic algorithms, the population size need to be determined based on the experience of the modeller. Often, this ends up creating a large number of unnecessary samples. As a result, the computational burden can be huge, especially for large-scale MSSPS structures. To solve these problems, this paper proposes an optimal structure designing method named as redundancy ordinal optimization. The universal generating function technique is applied to evaluate the reliabilities of the MSSPSs. Based on the reliabilities, an ordinal optimization algorithm is adapted to update the parent populations and the stopping criterion of genetic algorithm, so that the unnecessary structure designs can be eliminated. Numerical examples show that the proposed method improves the computational efficiency while remaining satisfactorily accurate.

## Key words

Redundancy optimization, multi-state series-parallel systems, ordinal optimization, genetic algorithm

## Introduction

As an extension of binary systems, multi-state systems (MSSs) are more flexible for modeling reliabilities of real-life engineering systems <sup>1</sup>. In an MSS, the components and the system can take a range of performance levels, from perfect functioning to complete failure. System performances are calculated by considering all the combination of component states with different performance levels <sup>2</sup>. Among the MSS structures, the series-parallel structure, in which subsystems are made of components connected in parallel, and systems comprises of subsystems connected in series, is the most frequently-used one. Typical examples of multi-state series-parallel systems (MSSPSs) include electric power systems and energy transmission systems <sup>3</sup>.

To provide a required level of system reliability, redundant components are necessary. In real-life cases, several versions for the same type of redundant components might be available on the market, with different characteristics like performance levels and prices. In order to make the optimal design, the appropriate version for each component should be chosen from a list of available ones, as well as the number of redundant components. For example, suppose we have a simple power generation system comprising of thermal and hydraulic generators. The

thermal generator could be chosen from six available versions, while the hydraulic generators have five available version to choose. The maximal number of generators to be installed is 10 for the thermal while 5 for the hydraulic one. Engineers need to determine the appropriate version and number of redundant thermal generators and hydraulic generators to be installed, respectively, so that the total costs of designing the system could be minimized <sup>4</sup>. This kind of problem is well-known as the redundancy optimization <sup>5,6</sup>, which will be the focus in this paper.

Besides minimizing the system cost, maintaining high levels of reliability is also of great significance in redundancy optimization. Normally, this is done by considering system reliability as the optimization constraints. To evaluate the reliability of the MSSPS, several approaches have been established, such as the universal generating function (UGF) technique <sup>7</sup>, *Lz*-transform approach <sup>8</sup>, Markov models and bayesian method <sup>9</sup>. Among the proposed methods, UGF technique is one of the most frequently-used one <sup>10</sup>. By defining different operators, it can be flexibly applied to many different system structures, including MSSPSs <sup>11</sup>. Therefore, UGF technique is adopted in this paper for the reliability evaluation.

As the industry develops, the scale of engineering systems increases. The complexity of the redundancy optimization problem grows rapidly as the available

versions and the maximal numbers of each type of components increase dramatically. It is inefficient, if not impossible at all, to evaluate the cost and reliability for all the possible structure design one by one during the redundancy optimization. Thus, various heuristic approaches have been proposed. Genetic algorithm (GA) <sup>5</sup> and particle swarm optimization algorithm <sup>12</sup> have been widely applied on redundancy optimization. Reference 13 proposed a stochastic fractal search to allocate the redundancy level of each component. Reference 14 hybridized the cuckoo search algorithm into the redundancy optimization problem. Considering the large number of available redundant components, GA is applied in this paper, as it allows engineers to solve practical optimization problem where a variety of available designs exist <sup>15</sup>.

However, the efficiency and accuracy of GA largely depend on its parent populations and stopping criterion <sup>16</sup>, especially for redundancy optimization with large number of available components. In the traditional GA, the number of designs in the parent population for each generation is usually randomly decided. Inappropriate parent populations would lead to a local optimum and require excessive computational time. The stopping criterion in the traditional GA is also randomly decided. GA has no idea how many feasible structure designs will be found before the optimal one. Inappropriate stopping criterion may result in a great amount of unnecessary designs, which significantly increases the computational time. Furthermore, due to the enormous number of available components and system states, reliability evaluation also becomes a huge computational burden, when a large number of unnecessary designs need to be evaluated.

To solve the difficulties in redundancy optimization mentioned above, the traditional GA is integrated with ordinal optimization (OO) algorithm in this paper. OO can solve the computational problem by considering the “order” rather than the “value” <sup>17</sup>. The goal softening in OO is to relax the optimization goal from searching for the best solution to seeking a good enough solution with high probability, which can reduce the large search space into a smaller and controllable one <sup>18</sup>. OO has been widely-used in many complex optimization problems. Reference 19 incorporated the OO into piecewise linear algorithm to improve the performance optimization of distributed file systems. Reference 20 applied it to maximize core utilization in parallel computing.

In this paper, an redundancy optimization method for MSSPS, termed as redundancy ordinal optimization

(ROO), is proposed. ROO can effectively design the optimal MSSPS structure considering multiple available redundant components under reliability constraint. The ROO method combines three algorithms, which are GA, OO and UGF. With the assistance of UGF, reliabilities of structure designs generated by GA can be evaluated. Based on the calculated reliabilities, ordinal optimization-based-genetic algorithm is applied to accelerate the redundancy optimization process. Different screening rules in OO are employed to modify the parent populations and stopping criterion of traditional GA. With the appropriate parent populations and stopping criterion, an optimal MSSPS structure can be effectively designed by the modified GA. The main contributions of this paper can be summarized as follows:

1) A redundancy optimization framework is proposed for multi-state series-parallel systems considering various versions and number of multi-state components.

2) An efficient redundancy optimization method for MSSPS is proposed.

3) The parent populations in each generation of GA are determined by different screening rules in OO, in order to determine the appropriate size of parent population and improve the optimization efficiency.

4) The stopping criterion of GA is modified by OO. The total number of structure designs needed to be considered before convergence can be evaluated, which restrict the large gene pool into a calculable smaller one.

The paper is structured as follows. Section 2 introduces the MSSPS model and redundancy optimization. Section 3 presents the process of the ROO method. Section 4 analyzes the efficiency and accuracy of the ROO method. Finally, the conclusion is presented in Section 5.

## Description for the MSSPS model and redundancy optimization

This section briefly reviews the MSSPS model and the redundancy optimization problem. Assume that a MSSPS consists of  $M$  subsystems connected in series. Each subsystem, denoted as  $Subsys_i (i = 1, 2, \dots, M)$ , only contains one version of components performing the same function.  $Subsys_i$  contains  $n_i$  identical components from the same version, which are connected in parallel. The MSSPS structure is presented in Figure 1.

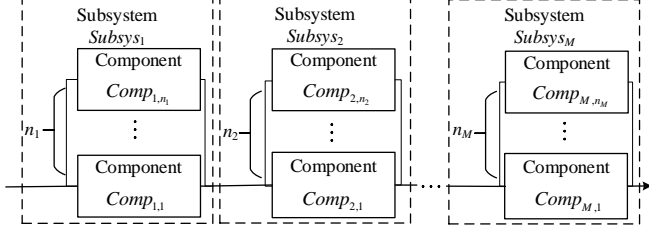


Figure 1. MSSPS structure

Assume that these components are labelled sequentially as  $1, 2, \dots, n$ , which means  $n = \sum_{i=1}^M n_i$ . The system component  $i$  ( $i = 1, 2, \dots, n$ ) has finite number of performance states, and the number is denoted as  $(k_i + 1)$ . Let  $w_{i,j}$  represents the performance level of component  $i$  in state  $j$  ( $j = 0, 1, \dots, k_i$ ). Then the set of all the performance levels of component  $i$  can be represented by  $\bar{w}_i = \{w_{i,0}, w_{i,1}, \dots, w_{i,k_i}\}$ . Let the random variable  $W_i$  represents the random performance level of component  $i$ , which takes value in the  $\bar{w}_i$ . The probabilities associated with component  $i$  in state  $j$  are  $p_{i,j} = \Pr(W_i = w_{i,j})$ .

It is assumed that the performance levels of the MSSPS are determined by the performances of its components. Suppose that the entire system has  $(K+1)$  states represented by the performance level set  $\bar{w}_{sys} = \{w_0, w_1, \dots, w_K\}$  and the system performance level  $W_{sys}$  is a random variable taking value in the  $\bar{w}_{sys}$ . The transformation  $\phi$  is the system structure function which can map from the state space of components performance levels to the state space of system performance levels. Thus,  $W_{sys}$  can be represented as  $W_{sys} = \phi(W_1, W_2, \dots, W_n)$ . Let  $w^*$  represents the system demand. For any pre-defined system demands, the probability of the system satisfying the demand levels, which is also the reliability under this demand, can be calculated as:

$$R(w^*) = \Pr(W_{sys} \geq w^*) \quad (1)$$

where  $W_{sys}$  is the system performance level,  $R(w^*)$  is the reliability of the system under  $w^*$ .

In real-life cases, to provide a required level of system reliability, the version and number of redundant components in each subsystem are two design variables to be chosen. Components in the same version have identical performance levels and purchase cost. In redundancy optimization, the number of available

versions for components in  $Subsys_i$  is  $N_i$  ( $i = 1, 2, \dots, M$ ), and is known by engineers. The available version  $j$  in  $Subsys_i$  is denoted as  $v_{i,j}$  ( $j = 1, 2, \dots, N_i$ ). The maximal number of available components in  $v_{i,j}$  is  $mn_{i,j}$ , which is known as well. Different versions and number of components can be chosen for different subsystems, but each subsystem only contains identical components from one version. The set of the available version that can be chosen for components in is represented by  $\mathbf{v}_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,N_i}\}$ . The number of components that can be chosen for version  $v_{i,j}$  is  $mn_{i,j} = \{0, 1, \dots, mn_{i,j}\}$ . The final version and number of components chosen for  $Subsys_i$  are denoted as  $v_i$  ( $v_i \in \mathbf{v}_i$ ) and  $mn_i$  ( $mn_i \in mn_{i,j}$ , if  $v_i = v_{i,j}$ ), respectively. The purchase cost for one component in  $v_i$  is  $c_i$ . In this paper, the system cost is considered as the sum of the purchase costs for all the components. Then the system cost can be calculated as  $C_{total} = \sum_{i=1}^M mn_i \cdot c_i$ .

To design an economical and reliable MSSPS structure, engineers need to determine the optimal version and number of redundant components for each subsystem, which are the values of  $v_i$  and  $mn_i$ . However, designing the optimal MSSPS structure is a complicated combinatorial optimization problem, as all the possible combination of  $v_i$  and  $mn_i$  need to be enumerated in order to derive the optimal solution. To improve the optimization efficiency, GA is applied to solve the optimization problem in this paper. The selection operator in GA can select designs as the parent population for a generation. The crossover operator in GA can generate new feasible designs while retaining good features from the parent population, and the mutation operator can enhance the diversity of new feasible designs. The general steps of GA are given in reference 15: Firstly, randomly sample  $p_s$  feasible designs as the parent population of the first generation. Secondly, within a parent population, apply crossover operator to produce new feasible designs. Then mutate these new designs with probability  $p_m$ . Repeat the operators until  $p_s$  new feasible designs are produced. Thirdly, evaluate the fitness of the  $2p_s$  feasible designs and choose  $p_s$  good designs as the parent population for the next generation. Fourthly, apply step 2 and 3 to replenish the parent population until it remains constant. Fifthly, preserve the best design in one genetic cycle. Terminate GA after  $N_c$  genetic cycles. Otherwise return

to step 1 and run a new genetic cycle. Finally, select the optimal design among  $N_c$  best designs.

Based on the general steps of GA mentioned above, the process of redundancy optimization of MSSPSs under the framework of GA can be seen in Figure 2.

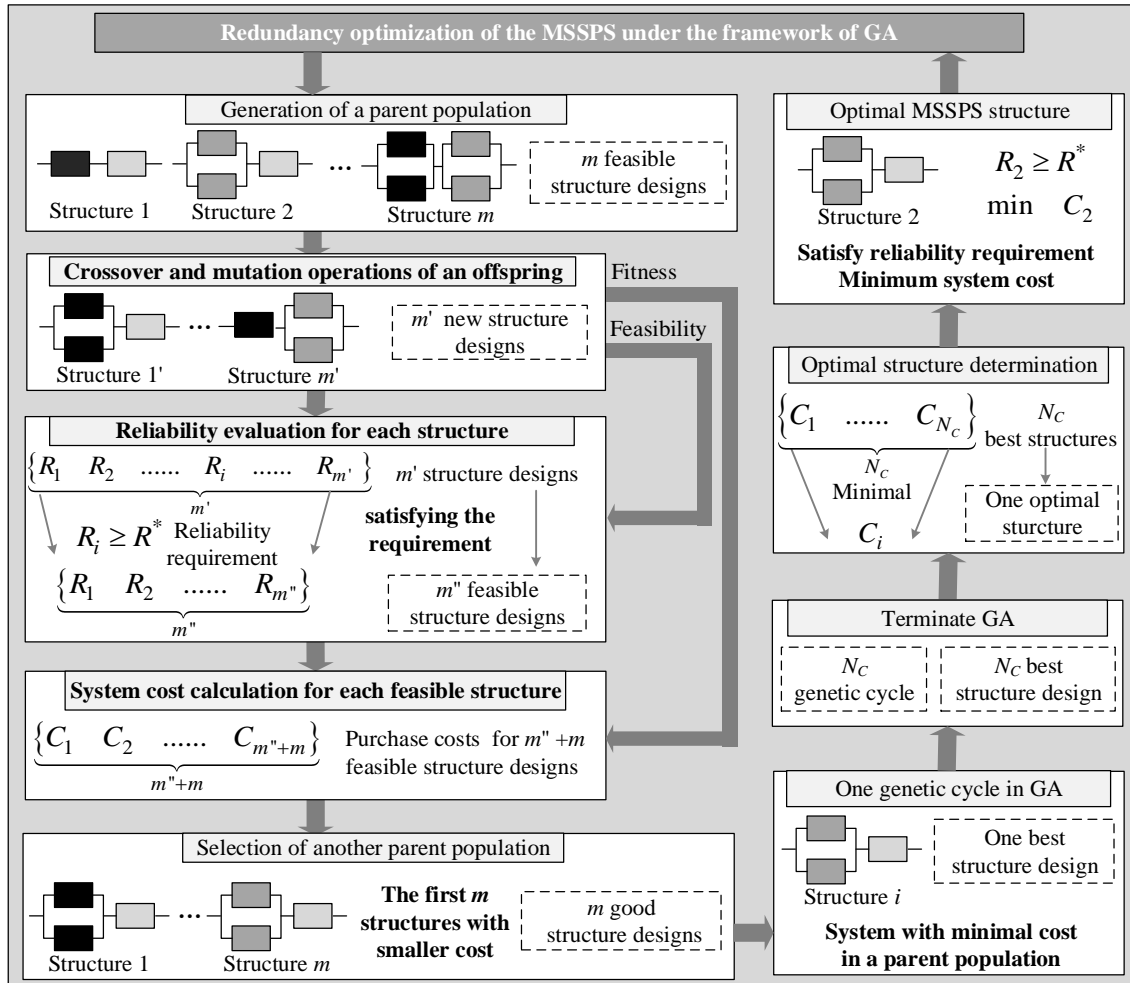


Figure 2. Redundancy optimization of MSSPSs under the framework of GA

As shown in Figure 2, the process of redundancy optimization with GA can be divided into seven steps, generation, crossover and mutation, reliability evaluation, cost assessment, selection, structure preservation and optimal structure determination. The fitness and feasibility of a structure design are defined by the system cost and system reliability, respectively. The first step of redundancy optimization is to sample  $m$  feasible structure designs as the parent population for a generation. The feasible solution is the one with a reliability no less than the reliability requirement  $R^*$ . Then, crossover and mutation operators are used to generate  $m'$  ( $m' \geq m$ ) new structure designs. The third step is to evaluate the reliabilities  $R_i$  ( $i=1,2,\dots,m'$ ) for these new structure designs, and select the  $m''$  feasible

structure designs as the offspring in this generation. The fourth step is to estimate the system cost for the  $m''+m$  feasible structure designs from the offspring and parent population. Employing the selection operator in GA, the fifth step is to find the  $m$  good structure designs, which are the top- $m$  feasible structures in terms of minimizing system costs. These  $m$  feasible structure designs are considered as the parent population for the next generation. After the constant parent population is achieved, the sixth step is to preserve the best structure design which has minimal system cost among all the generations. The final step is to determine the optimal MSSPS structure among best structure designs after the genetic cycles in GA are terminated. The optimal structure is the MSSPS structure whose system cost is

minimal and its reliability is no less than the required one.

However, as the industry develops, the computation burden for redundancy optimization is becoming unaffordable. On the one hand, both the number of available versions and components could be large for subsystems. Suppose that a MSSPS has  $M$  subsystems and each subsystem has  $V$  available versions of component. Each version has  $G$  available components. Considering the combinations of different components, there are  $(C_V^1 \cdot (C_G^1 + C_G^2 + \dots + C_G^G))^M$  structure designs which need to be evaluated. With the increasing number of available versions or components, the computational burden can grow dramatically. On the other hand, GA usually applies an arbitrarily pre-defined parent populations and stopping criterion. It may spend most of the time proving the optimality of an already found optimal design, which require massive unnecessary computational time. Furthermore, the computation time of reliability evaluation may beyond calculation considering massive unnecessary large-scale MSSPSs. To efficiently deal with the redundancy optimization task, a new method ROO for redundancy optimization of MSSPSs is proposed in the next section.

### Formulation of ROO method

In this section, ROO method is proposed to efficiently design the optimal MSSPS structure under reliability constraint.

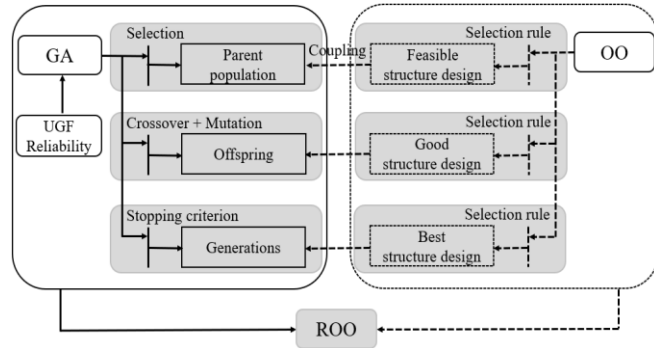


Figure 3. Process of ROO

The process of ROO method is presented in Figure 3. Firstly, apply the modified selection operator to generate the parent population for the  $i$ -th generation. Meanwhile, preserve the best structure design in the  $i$ -th generation. Secondly, apply crossover and mutation operators to generate the offspring based on the corresponding parent population. Thirdly, from the offspring and parent population, apply the modified selection operator to determine the good structure

designs as the parent population for the  $(i+1)$ -th generation. Fourthly, repeat the above steps until the total number of feasible structure designs among all the generations meets the modified stopping criterion. The optimal MSSPS structure among the preserved best structure designs can be determined after the termination of GA. In the following parts, the details of ROO are introduced.

### Feasibility and fitness evaluation of each structure

A structure design  $j$  combining different versions and number of components can be represented by  $Str_j = (C_{total}^j, R^j, v_1^j, \dots, v_M^j, mn_1^j, \dots, mn_M^j) \cdot v_i^j (i=1, 2, \dots, M)$  is the final version of components chosen for  $Subsys_i$ .  $mn_i^j$  is the number of components in  $v_i^j$  chosen for  $Subsys_i$ .  $R^j$  is the reliability, which is also considered as the feasibility under the framework of GA.  $C_{total}^j$  is the system cost, which is also the fitness in GA.

For a structure design with determined components in each subsystem, its reliability can be evaluated by UGF technique<sup>10</sup>. Based on the definitions of components mentioned in Section 2, the UGF representation to describe the probability distribution of component  $i$  is  $u_i(z) = \sum_{j=0}^{k_i} p_{i,j} \cdot z^{w_{i,j}}$ . Based on UGFs of components, the UGF of a system with  $n$  components is calculated as:

$$U(z) = \Omega_\phi(u_1(z), \dots, u_n(z)) \\ = \sum_{j_1=0}^{k_1} \dots \sum_{j_n=0}^{k_n} \left( \prod_{i=1}^n p_{i,j_i} z^{\phi(w_{1,j_1}, \dots, w_{n,j_n})} \right) \quad (2)$$

Where  $\Omega_\phi$  is the composition operators and  $\phi$  is the structure functions.

In the MSSPS, two basic structure functions are concerned: the series structure function  $\phi_S$  and the parallel structure function  $\phi_P$ , which are defined as:

$$\phi_S(W_1, \dots, W_n) = \min(W_1, \dots, W_n) \\ \phi_P(W_1, \dots, W_n) = \sum_{i=1}^n W_i \quad (3)$$

By applying system structure functions into Eq. (2), the reliability  $R(w^*)$  can be calculated by Eq. (4) considering system demand  $w^*$ :

$$R(w^*) = \delta_A(U(z), w^*) = \sum_{i=0}^K p_i \cdot \chi(w_i - w^*) \quad (4)$$

where  $(K+1)$  is the number of system states.  $p_i$  is the probability associated with the performance level  $w_i$  in state  $i$ .  $\chi$  is defined as an indicator function  $\chi(x) = 1(x \geq 0)$ , which equals to 0 if the variable  $x$  is less than 0 and equals to 1 otherwise.

*Example 1:* Consider a multi-state electric generation system with two parallelly connected generator units  $G1$  and  $G2$ . Every generating unit has three possible states: 0, 1, 2. When the total weights of electric power are no less than 17MW, the  $G_i$  is considered to be in state 2; when it is less than 17MW but no less than 10MW, it is in state 1; Otherwise, it is in state 0. The probabilities for the system to be in states 2, 1 and 0 are 0.8, 0.1 and 0.1, respectively. The UGF of each unit  $G_i$  is:

$$u_{G1}(z) = u_{G2}(z) = 0.1 \cdot z^0 + 0.1 \cdot z^{10} + 0.8 \cdot z^{17} \quad (5)$$

Applying the system structure functions, the reliability of the system attaining the system demand of 20MW is:

$$\begin{aligned} R(w^*) &= \delta_A(U(z), 20) = \delta_A(\Omega_\phi(u_{G1}(z), u_{G2}(z)), 20) \\ &= \delta_A(0.01 \cdot z^0 + 0.02 \cdot z^{10} + \dots + 0.64 \cdot z^{34}, 20) \quad (6) \\ &= 0.01 + 0.16 + 0.64 = 0.81 \end{aligned}$$

For a structure design, its system cost can be evaluated as follows. By employing UGF, reliability of a structure can be obtained. Considering the reliability requirement  $R^*$ , a feasible structure design represented by  $Strf_j = (C_{total}^j, R^j \geq R^*, v_1^j, \dots, v_M^j, mn_1^j, \dots, mn_M^j)$  can be selected. Assume the cost for one component in  $v_i^j$  is  $c_i^j$ . Then the system cost  $C_{total}^j$  for  $Strf_j$  is calculated as:

$$\text{System cost } C_{total}^j = \sum_{i=1}^M n_i^j c_i^j \quad (7)$$

According to the calculated system costs among all the feasible structure designs, the good structure design with smaller system cost can be determined, which is  $Strg_j = (\text{smaller } C_{total}^j, R^j \geq R^*, v_1^j, \dots, v_M^j, mn_1^j, \dots, mn_M^j)$ .

#### Parent population selection of the first generation

In the traditional GA, the size of the initial parent population needs to be assigned artificially, based on the experience of the modeller. However, in redundancy optimization, the number of available versions and redundant components can be large. Inappropriate size of parent population may result in unnecessary structure designs, which leads to severe computational burden. Therefore, this paper integrates OO into GA to determine a suitable size for initial parent population. OO solves the computational problem by considering the “order” rather than the “value”. The basic idea of OO is simple: Instead of finding the best solution, OO attempts to find a “good enough solution” within a significantly reduced computational time<sup>17</sup>.

We apply Blind Picking rule (BP) in OO as the screening rule in this section<sup>18</sup>. Let  $F$  be the population of all the feasible designs. The set  $G$  consists of top- $\beta$  truly best designs among  $F$ .  $p$  is the population proportion of top- $\beta$  truly best designs among  $F$ , which is  $|G|/|F|$ . If we randomly sample a design solution, the probability that this design is in  $G$  is  $p$ . If we randomly sample  $g$  feasible designs in  $F$ , the probability  $p_{one}$  that at least one of the sampled designs is in  $G$  is  $p_{one} = 1 - (1 - p)^g$ . Thus, the value of  $g$  can be calculated as:

$$g = \ln(1 - p_{one}) / \ln(1 - p) \quad (8)$$

**Table 1.** Required number of feasible designs for different  $p_{one}$  and  $p$

Case	$p=1\%$	$p=0.1\%$	$p=0.01\%$
$p_{one}=50\%$	69	692	6932
$p_{one}=90\%$	229	2301	23025
$p_{one}=99\%$	459	4603	46049
$p_{one}=99.9\%$	688	6905	69075
$p_{one}=99.99\%$	917	9206	92099
$p_{one}=99.999\%$	1146	11508	115124

The determination of  $g$  for any different  $p_{one}$  and  $p$  is named as BP. Table 1 shows how to interpret  $g$  for different  $p_{one}$  and  $p$ . For example, if engineers want the best design generated to lie within the top-0.1% of all the feasible designs with a probability of at least 99%, which means  $p_{one} = 0.99$  and  $p = 0.001$ . Only  $g = \ln(1 - 0.99) / \ln(1 - 0.001) \approx 4603$  feasible designs should be sampled. This number indicates that if 4603 feasible designs are sampled during optimization, the probability that the best design among these 4603 designs is within the top-0.1% of all the feasible designs is at least 99%.

In redundancy optimization, BP is applied to obtain a more rational population size of GA. Once the values of  $p_{one}$  and  $p$  are determined by engineers, the total number of sampled feasible structure designs for the first generation can be calculated by Eq. (8). The sample  $g = \lambda_1$  feasible structure designs during the optimization can ensure that the identified best structure design among the sampled designs is within the top- $p$  of all the feasible structure designs with probability at least  $p_{one}$ . The structure designs among the other generations are inherited and mutated from the parent population of the first generation. The appropriate feasible structure designs from the first generation provide an upper bound for the final optimal structure.

Thus, the modified initial population by OO offers a more suitable parent population than the randomly sizing in traditional GA.

*Example 2:* Our goal is to ensure that the best structure design of the first generation found is in the top-1% of all the feasible structure design with probability at least 99%. Therefore, after initialization, we sample 459 feasible structure designs. If a sampled structure design is not feasible, we resample until 459 feasible structure designs satisfying the reliability requirement are found. These 459 feasible structure designs are considered as the parent population for the first generation in GA.

Based on OO,  $\lambda_1$  feasible structure designs can be determined as the initial population. With these  $\lambda_1$  designs, we can obtain  $2\lambda_1$  feasible structure designs by the crossover and mutation operators mentioned below.

#### Offspring selection of the corresponding generation

After the determination of the parent population for the  $i$ -th generation, we use a crossover operator to generate new structure designs that retain the good features of the parent population. Assume that there are  $\lambda_i$  feasible structure designs in the parent population for the  $i$ -th generation. The crossover and mutation operators will use these  $\lambda_i$  designs to produce  $\lambda_i$  new feasible structure designs as the offspring. The detail explanations for crossover and mutation operators can be seen in reference 12. To ensure that all the new structure designs produced by crossover and mutation operators are feasible, the reliability of every structure design is evaluated every time a crossover or a mutation occurs.

After the crossover and mutation operators, the  $i$ -th generation consisting of  $2\lambda_i$  feasible structure designs are produced, whose feasible structure designs come from the parent population and offspring. For the  $i$ -th generation, preserve the best structure design  $Strb_i = (\min C_{total}^i, R^i \geq R^*, v_1^i, \dots, v_M^i, mn_1^i, \dots, mn_M^i)$  which has the minimal system cost among the feasible ones. Moreover, with these  $2\lambda_i$  designs, we can determine the parent population for the  $(i+1)$ -th generation by the selection operator mentioned below.

#### Parent population Selection of the next generation

After determining the parent population and offspring for the  $i$ -th generation, the selection operator should be

used. It chooses  $\alpha_i$  good structure designs from feasible structure designs in the  $i$ -th generation. The good structure designs represent the feasible structure designs which have top- $\alpha_i$  smaller system cost. In traditional GA, the value of  $\alpha_i$  is usually randomly defined as 1000 or 5000. However, the traditional GA may choose a large amount of unnecessary good structure designs, which lead to severe computational burden for the next generation. Thus, to provide a more rational value of  $\alpha_i$  for each generation, the selection operator is modified by the OO in this paper.

As mentioned above, the number of feasible structure designs in the  $i$ -th generation is  $2\lambda_i$  after the crossover and mutation operations. Evaluate the system costs of  $2\lambda_i$  feasible structure designs and sort the feasible structure designs from smallest to largest based on the system cost. The new order of  $2\lambda_i$  feasible structure designs  $Strfi^{ORD}$  and its corresponding system costs  $Ci_{totalORD}$  are represented as:

$$Strfi^{ORD} = \{Strf_1, Strf_2, \dots, Strf_{2\lambda_i}\} \quad (9)$$

$$Ci_{totalORD} = \{C_{total}^1, C_{total}^2, \dots, C_{total}^{2\lambda_i}\} \quad (10)$$

And the system costs satisfy the condition

$$C_{total}^{2\lambda_i} \geq \dots \geq C_{total}^2 \geq C_{total}^1$$

where  $Strf_j$  is the feasible structure design  $j$  whose system cost is  $C_{total}^j$ .

In this paper, BP is applied to obtain a rational number of good structure designs for each generation. Similar to the first generation, the number of good structure designs should ensure that the best structure design among the sampled good structure designs is within the top- $p$  of all the feasible structure designs with probability at least  $p_{one}$ . Once the values of  $p_{one}$  and  $p$  are determined, the number of good structure designs that should be chosen from the  $i$ -th generation can be calculated by Eq. (8). These  $g = \alpha_i$  good structure designs  $Strgi = \{Strg_1, Strg_2, \dots, Strg_{\alpha_i}\}$  are the top- $\alpha_i$  structure designs with smaller costs from  $Strfi^{ORD}$ , which are regarded as the parent population for the  $(i+1)$ -th generation. The first generation provides the upper bound for the optimal structure design, while next generations only need to increase the diversity of structure designs so that the optimality of final determined structure design can be further improved.



Thus, the confidence level  $p_{one}$  of the other generations can be lower than the first generation.

*Example 3:* Our goal is to ensure that a top-1% feasible structure design can be found with at least a 90% confidence level for the other generations. Thus, we need to select the top-229 feasible structure designs with small system costs from  $Strfi^{ORD}$ . These 229 feasible structure designs are considered as the good structure designs  $Strgi = \{Strg_1, Strg_2, \dots, Strg_{229}\}$  in the  $i$ -th generation. These  $Strgi$  are also regarded as the parent population for the  $(i+1)$ -th generation, which means  $\lambda_{i+1} = 229$ .

In the modified selection operator,  $\alpha_i$  good feasible structure designs can be selected from  $2\lambda_i$  feasible structure designs in the  $i$ -th generation, which is also the parent population for the  $(i+1)$ -th generation. Increasing the generation number by 1 and continue with the crossover, mutation and selection operators, preserve the best structure design  $Strb_i$  for each generation until the stopping criterion mentioned below is satisfied.

#### *Modified stopping criterion of GA*

In traditional GA, the stopping criterion depends on the number of genetic cycles. Once the pre-defined  $N_c$  genetic cycles are reached, GA is terminated. The value of  $N_c$  is randomly decided as well, such as 20 or 50. However, a small value of  $N_c$  would lead to a local optimum and a large value of  $N_c$  may result in massive computational time. An appropriate stopping criterion with calculable number of feasible structure designs is essential in redundancy optimization. Thus, this paper integrates OO with GA to provide a suitable stopping criterion. The screening rule in OO can tell the exact number of feasible structure designs that need to be found before the final optimal one, which forms a new stopping criterion.

The goal of redundancy optimization is to ensure that one of the top- $p$  feasible structure designs can be found with at least a probability  $p_{one}$ . The probability of reaching this goal should be larger than the first generation, so that the optimal solutions could be improved through iterations. Once the number of all the newly generated feasible structure designs is no less than  $g$  calculated by Eq. (8), GA should be terminated. However, as mentioned above, once  $p_{one}$  is fixed, the value of  $g$  is dependent on  $p$ .  $p$  is the population proportion of top- $\beta$  truly best structure designs among

all the feasible structure designs. An unsuitable  $p$  may lead to a large value of  $\beta$  if the total number of feasible structure designs is huge. Then one of the top- $p$  feasible structure designs found by the modified GA cannot be regarded as the “good enough solution”. Therefore, a variable  $n_0$  is introduced to determine a propriate value of  $p$  for stopping criterion. The total number of structure designs in redundancy optimization is defined as  $Q$ .  $n_0$  is the number of elements in set  $G$ , which can be calculated as  $n_0 = pQ$ .  $p$  can be determined once  $n_0$  satisfies:

$$\exists k \in \{1, 2, \dots, 10\} \quad s.t. 100 \cdot (k-1) \leq n_0 = pQ \leq 100 \cdot k \quad (11)$$

It can be noticed from the three operators in GA that the first generation produces  $2\lambda_1$  new feasible structure designs, while the other generations generate  $\lambda_i$  new feasible structure designs. Thus, GA should be terminated after the  $s$ -th generation, once  $2\lambda_1 + \lambda_2 + \dots + \lambda_s \geq g$  and  $2\lambda_1 + \lambda_2 + \dots + \lambda_{s-1} < g$  are satisfied. After termination,  $s$  best structure designs  $Strb = \{Strb_1, Strb_2, \dots, Strb_s\}$  can be preserved.

Through the modified stopping criterion, GA should be terminated after  $s$  generations. The optimal MSSPS structure can be determined through  $Strb$  as mentioned below.

#### *Optimal structure determination*

After the termination of GA, best structure designs from  $s$  generations are determined. Each best structure design contains the appropriate versions and number of components for each subsystem. Pick up the optimal MSSPS structure among these best structure designs. This optimal structure is the MSSPS structure which has minimal system cost and satisfies the reliability requirement.

#### *Summary of the proposed ROO method*

The process of the ROO method is illustrated in Figure 4.

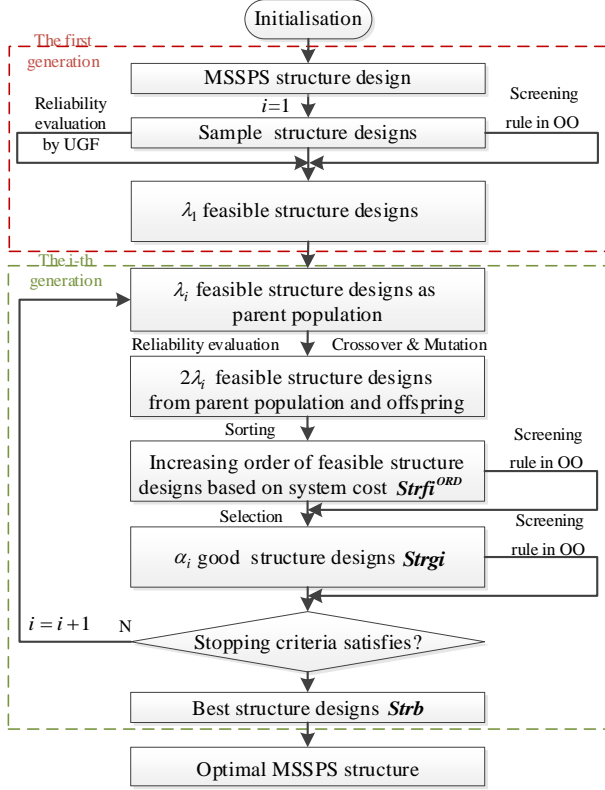


Figure 4. Flow chart for the construction of ROO

**Step 1) Selection of the first generation:** Sample  $\lambda_1$  feasible structure designs as the parent population for the first generation.  $\lambda_1$  is determined by OO. The feasibility of structure designs is measured by UGF.

**Step 2) Crossover and mutation of the offspring:** Apply crossover and mutation operators to generate  $\lambda_i$  new feasible structure designs, which is named as offspring for the  $i$ -th generation.

**Step 3) Determine the best structure design:** Sort the system cost  $C_{i\_totalORD}$  from largest to smallest of  $2\lambda_i$  feasible structure designs  $Strfi^{ORD}$ . Select the best structure design  $Strb_i$  for the  $i$ -th generation.

**Step 4) Select the other generations:** Based on the modified selection operator, select the top- $\alpha_i$  good structure designs  $Strgi$  from  $Strfi^{ORD}$ . These good structure designs are regarded as the parent population for the  $(i+1)$ -th generation, where  $\lambda_{i+1} = \alpha_i$ .  $\alpha_i$  is determined by OO

**Step 5) Terminate the GA:** According to the modified stopping criterion, GA is terminated after  $s$  generations. Determine the best structure designs  $Strb$  from all the generations and go to step 6. Otherwise, set  $i=i+1$  and continue with step 2.

**Step 6) Determine the optimal structure  $Strb$ .**

### Numerical studies

In this section, ROO method is applied in the redundancy optimization of real-life electric power systems, which is a typical example of MSSPS. Through theoretical analysis and numerical experiments, the accuracy and efficiency performance of ROO and GA are compared.

An electric power system can be divided into three subsystems: generation, transmission and transformation subsystems. The block diagram of power system structure can be seen in Figure 5. The electric power is transmitted from generation subsystem to its load system through two transformation subsystems and two transmission subsystems.  $Subsys_1$  represents the generation subsystem consisting of  $n_1$  generators.  $Subsys_2$  and  $Subsys_4$  represent the transformation subsystems made up of  $n_2$  and  $n_4$  transformers, respectively.  $Subsys_3$  and  $Subsys_5$  represent the transmission subsystems with  $n_3$  and  $n_5$  transmission lines, respectively.

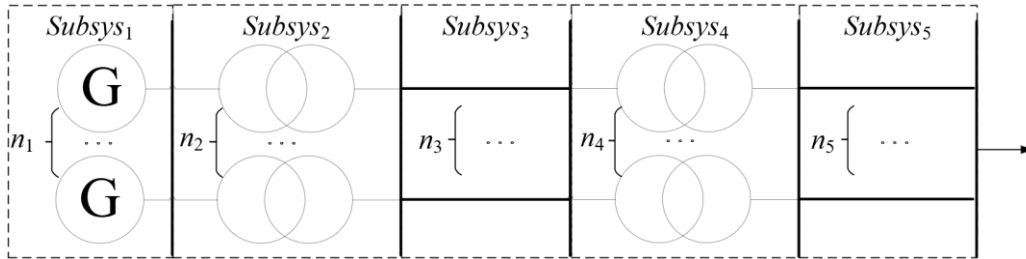


Figure 5. Block diagram of MSSPS structure

Each transformer in  $Subsys_2$  and  $Subsys_4$  is modeled as a two-state component. Each transmission line in

$Subsys_3$  and  $Subsys_5$  is also considered as binary-state component. While each generator in  $Subsys_1$  has five

states. The predefined demand is 5 MW. The number of available versions and components in each subsystem, along with the parameters of each components can be seen in Appendix. The required reliability  $R^*$  is 0.88. To construct an optimal MSSPS structure, we need to select the appropriate version and number of components for each subsystem. Since the total number of structure designs is larger than 1 million, an exhaustive examination of such an enormous structure designs is not feasible. Therefore, we apply ROO method to efficiently determine the optimal one.

### Application of ROO method

To ensure that a top-0.1% feasible structure designs can be found with at least a 99% probability in the first generation, we sample 4603 feasible structure designs as the parent population for the first generation. Similarly, to ensure that a top-0.1% feasible structure designs can be found with at least a 90% probability in the other generations, 2301 good structure designs are selected as the parent population for the other generations. Moreover, when  $p$  equals to 0.01%, the stopping criterion mentioned in Section 4 is satisfied. Thus, to ensure that a top-0.1% feasible structure designs can be found with at least a 99.999% probability, the total number of all the feasible designs is equal to 11508. The mutation probability  $p_m$  is 0.2.

After three generations, ROO is terminated. The characteristics of versions and components in the optimal MSSPS structure is presented in Table 2. As can be seen from Table 2,  $Subsys_1$  has one generator from version 2.  $Subsys_2$  has one transformer from version 4.  $Subsys_3$  has three transmission lines from version 2.  $Subsys_4$  has one transformer from version 4.  $Subsys_5$  has one transmission line from version 2. The total system cost is equal to \$ 597 million.

**Table 2.** Characteristics of versions and components in the optimal structure

Subsystem	1	2	3	4	5
$v_i$	$v_2$	$v_4$	$v_2$	$v_2$	$v_4$
$n_i$	1	1	3	1	1

### Comparisons of accuracy and efficiency between ROO and GA

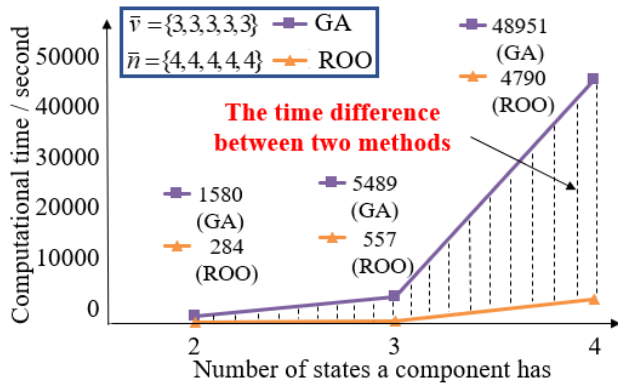
To demonstrate the accuracy and efficiency of ROO method, two comparisons are made between ROO and GA in terms of precision and computational time.

We apply GA to determine the optimal MSSPS structure under reliability requirement 0.88. The population size of each generation, mutation probability and genetic cycles are 5000, 0.2 and 10, respectively. The optimal MSSPS structure calculated by GA is the same optimal structure as ROO, where  $Subsys_1$  has one component from version 2,  $Subsys_2$  has one component from version 4,  $Subsys_3$  has three components from version 2,  $Subsys_4$  has one component from version 4 and  $Subsys_5$  has one component from version 2. Thus, ROO method exhibits highly precision in redundancy optimization.

Experiment below shows the high efficiency of the ROO in computational time. The experiments were developed by MATLAB2017b, in a computer with 2.60GHz CPU and 4 GB of RAM. In the previous case study, most of the components are considered as binary-state components. To observe the performance of ROO method in large-scale systems, multi-state components instead of the simple binary components are used. Despite the component states, we also enlarge the number of available versions and components in each subsystem.  $N = \{N_1, N_2, \dots, N_5\}$  represents the set of the number of available versions for five subsystems, where  $N_i (i = 1, 2, \dots, 5)$  is the number of available versions in  $Subsys_i$ . Suppose that the number of available components for each version in the same subsystem are equal.  $maxn = \{maxn_1, maxn_2, \dots, maxn_5\}$  is defined as the set of the maximum number of available components for each version in five subsystems, where  $maxn_i$  is the maximum number of available components for each version in  $Subsys_i$ . The CPU time for ROO and GA is compared in Table 3 and Figure 6. In Figure 6, the x-axis represents the number of states a component has. The y-axis shows the CPU time by GA and ROO, respectively.

**Table 3.** CPU time for two methods

CPU time (seconds)	Number of states a component has when $N = \{3, 3, 3, 3, 3\}$ $maxn = \{4, 4, 4, 4, 4\}$			Number of states a component has when $N = \{4, 4, 4, 4, 4\}$ $maxn = \{5, 5, 5, 5, 5\}$		
	2	3	4	2	3	4
GA	1580	5489	----	1891	7274	----
ROO	284	557	4790	485	741	8754



**Figure 6.** Calculation time when  $\bar{v}=\{3,3,3,3,3\}$  and  $\bar{n}=\{4,4,4,4,4\}$

From Table 3 and Figure 7, the calculation time of ROO method is nearly one-tenth the time of GA. As there are too many unnecessary structure designs, traditional GA requires huge computational time to generate and eliminate these structures. Particularly, the efficiency is more obvious when the number of available redundant components are larger. Therefore, it can be concluded that ROO can effectively shorten the computational time in redundancy optimization.

## Conclusion

Reducing computational burdens becomes a crucial problem for redundancy optimization considering multiple available redundant components. In order to efficiently design the optimal MSSPS structure, this paper proposes a ROO method to reduce the computational time and guarantee the optimization accuracy. OO is applied as a desirable tool to limit the gene pool in GA and improve the computational efficiency. The selection operator in GA is modified by the screening rule in OO, which can determine the appropriate parent population of each generation. Moreover, the stopping criterion of GA is also improved by OO to rationally terminate the optimization process. The efficiency and accuracy performance of ROO method are compared with the traditional optimization method. Results show that ROO produces the same optimal structure design with a 10-fold reduction in computing time compared with traditional GA. This advantage would be more obvious when the scale of redundancy optimization is larger.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China [Grant number 71871200].

## References

1. Bao M, Ding Y, Singh C, Shao C. A multi-state model for reliability assessment of integrated gas and power systems utilizing universal generating function techniques. *IEEE Transactions on Smart Grid* 2019; 10(6): 6271-6283.
2. Contreras J, Rivas F, Song J, Guardado J. Multi-state system reliability analysis of HVDC transmission systems using matrix-based system reliability method. *International Journal of Electric Power Energy System* 2018; 100: 265-278.
3. Wang G, Duan F, Zhou Y. Reliability evaluation of multi-state series systems with performance sharing," *Reliability Engineering & System Safety* 2018; 173: 58-63.
4. Levitin G, Lisnianski A, Haim H, Elmakis D. Redundancy optimization for series-parallel multi-state systems. *IEEE Transactions on Reliability* 1998; 47(2): 165-172.
5. Devi S, Garg D. Hybrid genetic and particle swarm algorithm: redundancy allocation problem. *International Journal of System Assurance Engineering and Management* 2020; 11(2): 313-319.
6. Wei W, Ming L, Yong F. Multi-objective optimization of reliability-redundancy allocation problem for multi-type production systems considering redundancy strategies. *Reliability Engineering & System Safety* 2020; online. <https://doi.org/10.1016/j.res.2019.106681>.
7. Li Y, Zio E. A multi-state model for the reliability assessment of a distributed generation system via universal generating function. *Reliability Engineering & System Safety* 2012; 106(5): 28-36.
8. Lisnianski A, Haim B. Short-term reliability evaluation for power stations by using Lz-transform. *Journal of Modern Power Systems and Clean Energy* 2013; 1(2): 110-117.
9. Liu Y, Lin P, Li Y. Bayesian reliability and performance assessment for multi-state systems. *IEEE Transactions on Reliability* 2015; 64(1): 394-409.
10. Ding Y, Zuo M, Lisnianski A. A framework for reliability approximation of multi-state weighted k-out-of-n systems. *IEEE Transactions on Reliability* 2010; 59(2): 297-308.
11. Zhou X, Shi K. Capacity failure rate based opportunistic maintenance modeling for series-parallel multi-station manufacturing systems. *Reliability Engineering & System Safety* 2019; 181: 46-53.
12. Yang Z, Liu Y, Ruan S, Jiang T. An improved particle swarm optimization algorithm for reliability redundancy allocation problem with mixed redundancy strategy and heterogeneous components. *Reliability Engineering & System Safety* 2019; 180: 62-74.

13. Huang Z, Coolen F, Maturi T. A heuristic survival signature-based approach for reliability-redundancy allocation. *Reliability Engineering & System Safety* 2019; 18: 511-517.
14. Kanagaraj G, Ponnambalam S, Jawahar N. A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems. *Computers & Industrial Engineering* 2013; 66(4): 1115– 1124.
15. Tavakkolimoghaddam R, Safari J, Sassani F. Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliability Engineering & System Safety* 2017; 93(4): 550-556.
16. Peiravi A, Karbasian M, Ardakan M, David C. Reliability optimization of series-parallel systems with K-mixed redundancy strategy. *Reliability Engineering & System Safety* 2019; 183: 17-28.
17. Ho Y, Lee L, Lan E. Explanation of goal softening in ordinal optimization. *IEEE Transactions on Automatic Control* 1999; 44(1): 94-99.
18. Ho Y., Zhao Q, Jia Q. Ordinal optimization: soft optimization for hard problems. Berlin, Germany: Springer 2006.
19. Ma T, Tian F, Dong B. Ordinal optimization-based performance model estimation method for HDFS. *IEEE Access* 2020; 8: 889-899.
20. Ni E, Ciocan D, Henderson S, Hunter S. Efficient ranking and selection in parallel computing environments. *Operational Research* 2017; 65(3): 821-836.

- MSSPS Multi-state series-parallel system  
 UGF Universal generating function  
 OO Ordinal optimization  
 GA Genetic algorithm  
 ROO Redundancy ordinal optimization

**Nomenclature**

- $M$  total number of subsystems connected in series in a MSSPS  
 $Subsys_i$  subsystem  $i(i = 1, 2, \dots, M)$  in a MSSPS  
 $Strf_j$  feasible structure design  $j$   
 $Strg_j$  good structure design  $j$   
 $Strbi$  best structure design in the  $i$ -th generation  
 $w_{i,j}$  performance level of component  $i$  in state  $j$   
 $P_{i,j}$  probability of component  $i$  in state  $j$   
 $N_i$  total number of available versions of components in  $Subsys_i$   
 $v_{i,j}$  available version  $j(j = 1, 2, \dots, M_i)$  in  $Subsys_i$   
 $mn_{i,j}$  total number of available components in  $v_{i,j}$   
 $C_{total}$  system cost  
 $v_i$  final version of components chosen for  $Subsys_i$   
 $mn_i$  final number of components chosen for  $Subsys_i$

**Appendices**

*Abbreviations*

**Table 4.** Characteristics of available versions and components in five subsystems

$Subsys_1$	Parameters of a component						Cost of a component	Number of available components
	State	0	1	2	3	4		
Version 1	$w_{i,j}$	0MW	5MW	8MW	10MW	20MW	\$ 200million	3
	$P_{i,j}$	0.02	0.2	0.1	0.3	0.48		
Version 2	$w_{i,j}$	0MW	5MW	8MW	10MW	20MW	\$ 185million	3
	$P_{i,j}$	0.01	0.1	0.15	0.1	0.64		
Version 3	$w_{i,j}$	0MW	4MW	9MW	15MW	20MW	\$ 175million	3
	$P_{i,j}$	0.02	0.2	0.1	0.2	0.48		
$Subsys_2$	Parameters of a component				Cost of a component	Number of available components		
	State	0		1				
Version 1	$w_{i,j}$	0MW		5MW		\$ 160million	3	
	$P_{i,j}$	0.02		0.98				
Version 2	$w_{i,j}$	0MW		5MW		\$ 150million	2	
	$P_{i,j}$	0.2		0.8				
Version 3	$w_{i,j}$	0MW		5MW		\$ 140million	2	

	$P_{i,j}$	0.01	0.99		
Version 4	$w_{i,j}$	0MW	5MW	\$ 130million	2
	$P_{i,j}$	0.1	0.9		
<i>Subsys<sub>3</sub></i>	Parameters of a component			Cost of a component	Number of available components
	State	0	1		
Version 1	$w_{i,j}$	0MW	5MW	\$ 40million	3
	$P_{i,j}$	0.02	0.98		
Version 2	$w_{i,j}$	0MW	5MW	\$ 35million	3
	$P_{i,j}$	0.1	0.9		
<i>Subsys<sub>4</sub></i>	Parameters of a component			Cost of a component	Number of available components
	State	0	1		
Version 1	$w_{i,j}$	0MW	5MW	\$ 180million	3
	$P_{i,j}$	0.02	0.98		
Version 2	$w_{i,j}$	0MW	5MW	\$ 160million	2
	$P_{i,j}$	0.01	0.99		
Version 3	$w_{i,j}$	0MW	5MW	\$ 147million	2
	$P_{i,j}$	0.2	0.8		
Version 4	$w_{i,j}$	0MW	5MW	\$ 135million	2
	$P_{i,j}$	0.1	0.9		
<i>Subsys<sub>5</sub></i>	Parameters of a component			Cost of a component	Number of available components
	State	0	1		
Version 1	$w_{i,j}$	0MW	5MW	\$ 45million	3
	$P_{i,j}$	0.01	0.99		
Version 2	$w_{i,j}$	0MW	5MW	\$ 42million	3
	$P_{i,j}$	0.2	0.8		