

# Joint Redundancy and Inspection-Based Maintenance Optimization for Series-Parallel System

Yuan Yao, Pan He<sup>(✉)</sup>, Zhihao Zheng, Chun Tan, and Yue Yuan

Chongqing Institute of Green and Intelligent Technology,  
Chinese Academy of Sciences, Chongqing 400714, China  
{yaoyuan, hepan, zhihao, tanchun, yuanyue}@cigit.ac.cn

**Abstract.** While researchers have concentrated on the optimization of joint redundancy and maintenance mechanism, maintenance in computing systems is quite different from that in traditional systems. Considering a routine monitoring and inspection mechanisms is conducted to detect component status and trigger repair process, this paper pays attention to the optimization problem of joint redundancy and inspection-based maintenance mechanism. After conducting steady state analysis on subsystems using inspection-based maintenance, shared repair facility and component redundancy, optimization model is built to search appropriate system structure and maintenance policy which maximizes system performance while meeting availability and cost constraints. Due to the complexity of uncertain optimization model, genetic algorithm is used to search optimal solution, using triple-element encoding mechanism and specifically designed operators. Illustrative examples are conducted to show that the optimization model and corresponding solution technique could be used to search optimal system configuration under given constraints and different cost constraints would lead to different optimization result while meeting availability constraints.

**Keywords:** Reliability optimization · Redundancy · System maintenance · Monitoring · Inspection

## 1 Introduction

As reliability has become a major concern in most engineering and computing systems, component reliability improvement and the provision of redundant components are often used together to maintain system reliability [1]. Preventive maintenance, which recovers a component from failure or degraded state, could improve the reliability of single component and is often used jointly with component redundancy for multi-performance-state systems [2]. Similar as traditional redundancy, the configuration of preventive maintenance also introduces extra cost to the whole system, so a great majority of researches have concentrated on the optimization of joint mechanism [3, 4]. Existing researches mainly assume that the repair process is taken immediately after component failure or degradation. However, in large distributed computing systems, e.g. cloud based systems, it is difficult to identify failures in various distributed components. Routine monitoring and inspection mechanisms are often conducted to detect component

status and to trigger component repair process [5]. Therefore, the inspection policy plays a key role in the whole system maintenance, but few studies have concentrated on the optimization of joint redundancy and inspection-based maintenance mechanism.

A lot of researches have been taken on the subject of joint traditional fault tolerance mechanism analysis and optimization [11, 12]. For multiple kinds of redundancy strategies, Coit [6], Tavakkoli-Moghaddam et al. [7] and Chambari et al. all [8] studied redundancy optimization problem while simulated annealing algorithm [9] and genetic algorithm [10] were used to get solutions. For maintenance mechanism, Soro et al. evaluated the reliability and performance of multi-state degraded systems with redundancy and imperfect preventive maintenance [2] and then Nourelfath et al. proposed a SP/TG heuristic approach to optimize the joint mechanism in series-parallel systems [3]. Liu et al. considered the deterioration effect after maintenance in this model and conduct optimization using genetic algorithm [4]. The mentioned researches didn't consider monitoring and inspection mechanism for repair process and were not applicable for cloud computing systems. Our previous research conducted a preliminary study on joint optimized redundancy and inspection rate [13]. Due to the computation complexity of state transition analysis, the model was only built for subsystems with two redundant components when inspection used. For systems with  $k$ -out-of- $n$  redundancy architecture, Martorell al. analyzed the optimal test interval of a two-component parallel system based on availability analysis [21]. Vaurio developed the optimization on test and maintenance interval time for  $k$ -out-of- $n$  system with four components and series systems [22, 23]. Cepin and Mavko also conducted test strategies and maintenance optimization by minimization of risk through simulated annealing [24, 25]. Torres-Echeverria et al. presented an optimization model for the maintenance and test policy design for a system using MooN voting redundancies [26]. In these researches, analytical availability and periodic test cycle analysis are conducted on probabilistic methods, but the working mechanism inside the test or maintenance strategy has not been presented.

The main objective of this paper is to define the optimal system structure and the inspection policy for each component, so that the series-parallel system performance is maximized, subject to the availability and cost budget constraints. The state transition process is first analyzed for subsystems with redundant components and a routine monitoring and inspection mechanism. Because of the recovering effect of maintenance process, the steady state reliability and performance is analyzed from the state transition process. The cost is a combination of redundancy cost and maintenance cost. While original redundancy allocation problem (RAP) has been proven to be NP-hard problem [14], the introduction of inspection-based maintenance strategy even increases the complexity of the optimization problem. Therefore, a genetic algorithm is used to search the optimal solution of redundancy and inspection routine for each subsystem and illustrative examples are presented to explain the analysis and calculation process. The following sections are organized as follows: the problem description and optimization model is listed in the following section and the solution technique is then proposed; illustrate examples are presented with experimental results analysis; conclusions are summarized at last with future work.

## 2 The Optimization Model

Notations			
$m$	Number of sequential subsystems	$A_i$	Steady state availability of subsystem $i$
$n_i$	Number of redundant components in subsystem $i$	$p_{ri}$	Work processing rate of each component in subsystem $i$
$\lambda_i$	Failure rate of each component in subsystem $i$	$P_{ri}$	Steady state performance rate of subsystem $i$
$\mu_i$	Repair rate of each component in subsystem $i$	$q_d$	Degradation rate in maintenance state
$n_{ri}$	Maximal number of components being repaired in parallel in subsystem $i$	$\mathbf{n}, \lambda_m, \mathbf{n}_r$	Corresponding vectors of $n_i, \lambda_{mi}, n_{ri}$ in all subsystems
$q_i$	Probability of successful repair for each component in subsystem $i$	$\lambda_{mi}$	Monitoring rate of maintenance mechanism in subsystem $i$
$\pi_{i,\alpha}$	Steady state probability of state $\alpha$ in subsystem $i$	$Q_{i,\alpha\beta}$	Transition rate between two states $\alpha$ and $\beta$ in subsystem $i$
$\mu_{si}$	Component detection rate for subsystem $i$	$C_{sys}$	Overall system redundancy cost
$Q_i$	The generator matrix of $Q_{i,\alpha\beta}$	$M_{sys}$	Monitoring and inspection cost per time unit
$\pi_i$	Vector of $\pi_{i,\alpha}$	$R_{sys}$	The repair facility cost
$P_{sys}$	Average performance rate of the system	$A_{sys}$	Steady-state system availability

### 2.1 Problem Description

Assume that there are  $m$  sequential subsystems in one system and for subsystem  $i$  ( $1 \leq i \leq m$ ),  $n_i$  components are running in parallel or kept as backup. The structure of such series-parallel system is shown in Fig. 1.

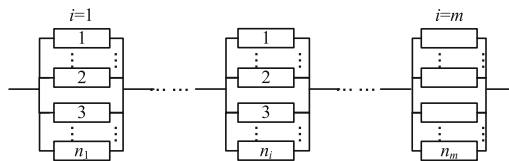
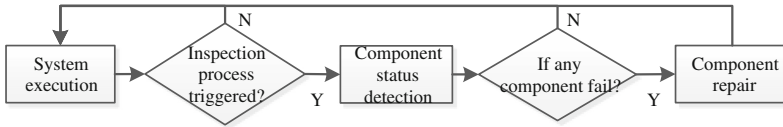
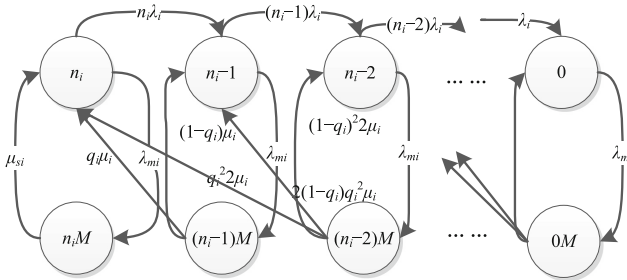


Fig. 1. Structure of general series-parallel system.

Following the fault tolerance mechanism design in researches [5],  $n_i$  components are running initially for each subsystem. A routine monitoring and inspection mechanism is used to detect the status of components. When a component fails or degrades, it is replaced by other standby backups instantly. The switched component is repaired or restarted in background and placed as backup components. A brief illustration of the process in each subsystem is shown in Fig. 2.



**Fig. 2.** Illustration of joint redundancy and inspection-based maintenance mechanism.



**Fig. 3.** State transition diagram joint redundancy and inspection-based maintenance.

To illustration the working process of each subsystem, the state transition analysis is conducted on the subsystem with the redundant components. Each subsystem is presented by a multi-state model, shown as Fig. 3, where it is assumed that:

- The working and failure pattern of each component inside one subsystem is the same. In real-world applications, there is more than one type of components inside each subsystem and the failure pattern of these components is not the same. In this case, the patterns of the component with the worst reliability and performance could be used as the common pattern for calculation. Specifically, when a cloud-computing or grid-computing system is chosen for analysis, the same CPU and memory resources would be allocated to each virtual component in one system, it is reasonable to assume that the time-to-failure of each component follows the similar distribution pattern. The time to failure of components for subsystem  $i$  is exponentially distributed with rate  $\lambda_i$ . The repair process is exponentially distributed with rate  $\mu_i$ .
- The repair process for each component is carried out in background as long as the system is working. At most  $n_{ri}$  failed components could be repaired in parallel with each repair process accomplished successfully with probability  $q_i$ .
- The routine inspection mechanism is run every certain time units. An ideal approach would be to take the time to the next inspection to be uniformly distributed. Since the difference between normal distribution and exponential distribution assumption for routine inspection mechanism is proven to be small [15], the periodical inspection/detection process is approximated to be exponentially distributed with  $\lambda_{mi}$  to reduce model complexity.
- The time used for component status detection is exponentially distributed with rate  $\mu_{si}$ . It is reasonable to assume that the time is much less than the time of repair process and is not taken into consideration when repair is conducted to avoid state explosion.

In this diagram, state  $k_{ij}(1 \leq i \leq m, 1 \leq j \leq n_i)$  represents the normal working state of  $i$ th subsystem with  $j$  active components. State  $k_{ij}M$  stands for the corresponding repair and maintenance state from  $k_{ij}$ . The performance of these states would be affected by the maintenance process running in the background while only the state of 0 and  $0M$  are failure states. State transition between states  $k_{ij}$  is triggered by the random component failure. State transition from  $k_{ij}$  to  $k_{ij}M$  is the process waiting for the next inspection. State transition from  $k_{ij}M$  back to  $k_{ij}$  is the inspection/detection and the repair process. For any state  $k_{ij}M$ ,  $n_i - k_{ij}$  components already fail in the system and repairmen process should be taken on these components. While at most  $n_{ri}$  failed components could be repaired each time, the probability of only  $w$  VM is migrated successfully is denoted by  $C_{n_{\min}}^w (q_i)^w (1 - q_i)^{n_{\min} - w}$ , where  $n_{\min}$  is the minimal value of  $n_i - k_{ij}$  and  $n_{ri}$ .

The overall state of the whole system is determined by the redundancy, inspection and maintenance policy on each subsystem. So, our main objective is to get the appropriate value of this configuration policy through optimization.

### 2.2 System Performance Evaluation Model

The reliability of a series-parallel system is determined by the reliability of each subsystem. Before constructing the optimization model for the whole system, the reliability and performance evaluation on top of the state transition diagram of subsystem  $i$  is first conducted. Since this kind of system is supposed to be running for a long time, steady state method is used to analyze this diagram as an irreducible CTMC [15].

Let  $S$  denotes the set of all states in Fig. 3  $\{0, \dots, n_i, 0M, \dots, n_iM\}$ . Let  $\pi_{i,\alpha}$  denote the steady state probability of state  $\alpha$  ( $\alpha \in S$ ) in subsystem  $i$  and  $\boldsymbol{\pi}_i$  denotes the vector  $[\pi_{ni}, \dots, \pi_0, \pi_{ni}M, \dots, \pi_0M]$ . Let  $Q_{i,\alpha\beta}$  denotes the transition rate between two states ( $\alpha \in S, \beta \in S$ ) in subsystem  $i$  and  $Q_i$  denotes the generator matrix of  $Q_{i,\alpha\beta}$ . The value of  $Q_{i,\alpha\beta}$  is

$$Q_{i,\alpha\beta} = \begin{cases} \alpha\lambda_i, 1 \leq \alpha \leq n_i, \beta = \alpha - 1 \\ \lambda_{mi}, 0 \leq \alpha \leq n_i, \beta = \alpha M \\ \mu_{si}, \alpha = n_iM, \beta = n_i \\ C_{n_{\min}}^{\beta-\gamma} q_i^{\beta-\gamma} (1 - q_i)^{w - (\beta-\gamma)} w \mu_i, w = \min\{n_{ri}, n_i - \gamma\}, \alpha = \gamma M, \gamma \leq \beta \leq n_i, 0 \leq \gamma < n_i \\ -w\mu_i, w = \min\{n_{ri}, n_r - \gamma\}, \alpha = \beta = \gamma M, 0 \leq \gamma < n_i \\ -\mu_{si}, \alpha = \beta = n_iM \\ -\alpha\lambda_i - \lambda_{mi}, 0 \leq \alpha \leq n_i, \alpha = \beta \end{cases} \tag{1}$$

Considering the Kolmogorov’s backward equation [15], the limits value of  $\pi_{i,\alpha}$  is calculated as

$$\lim_{t \rightarrow \infty} \frac{d\pi_{i,\alpha}(t)}{dt} = \left( \sum_{\beta \in S \ \& \ \beta \neq \alpha} \pi_{i,\beta} Q_{i,\beta\alpha} \right) - \pi_{i,\alpha} Q_{i,\alpha\alpha} = 0 \Rightarrow \begin{cases} \boldsymbol{\pi}_i \mathbf{Q}_i = \mathbf{0} \\ \sum_{\alpha \in S} \pi_{i,\alpha} = 1 \end{cases} \tag{2}$$

which could also be represented in matrix form.

Given the certain system structure, the failure and recover rate of each component,  $\pi_{i,\alpha}$  is a function of the number of redundant components  $n_i$ , the inspection rate  $\lambda_{mi}$  and

the maximum number of parallel repairing components  $n_{ri}$ :  $\pi_{i,\alpha} = \pi_{i,\alpha}(n_i, \lambda_{mi}, n_{ri})$ . Following the steady state analysis pattern, steady state availability  $A_i$  is used to describe subsystem reliability and the expected performance rate  $P_{ri}$  is used to describe the subsystem performance. Steady state availability  $A_i$  is the probability of the subsystem in the working state. Since only states 0 and 0  $M$  are failure states, given the value of  $n_i, \lambda_{mi}$  and  $n_{ri}$ , the value of  $A_i$  could be calculated as:

$$A_i = \sum_{\alpha \in S \ \& \ \alpha \neq 0, 0M} \pi_{i,\alpha} = A_i(n_i, \lambda_{mi}, n_{ri}) \tag{3}$$

The expected performance rate  $P_{ri}$  is the average work processing rate performed by all the working components in the subsystem. Let  $S_{i,U}$  and  $S_{i,M}$  denotes the set of normally working states  $\{\pi_{ni}, \dots, \pi_1\}$  and corresponding maintenance states  $\{\pi_{ni}M, \dots, \pi_1M\}$  for subsystem  $i$ . In the normally working state  $\alpha \in S_{i,U}$ . The work processing rate of the subsystem is the sum of processing rate of each working component. In the maintenance state  $\beta \in S_{i,M}$ , the work processing rate would be degraded due to the inspection and maintenance process running in the background. Assume that the work processing rate for each component is  $p_{ri}$  and the degradation rate is  $q_d$ . The value of  $P_{ri}$  is:

$$P_{ri} = \sum_{\alpha \in S} \pi_{i,\alpha} p_{ri,\alpha} = \sum_{\alpha \in S_{i,U}} \pi_{i,\alpha} \alpha p_{ri} + \sum_{\beta \in S_{i,D}} q_d \pi_{i,\beta} k p_{ri} = P_{ri}(n_i, \lambda_{mi}, n_{ri}), \beta = kM. \tag{4}$$

### 2.3 Optimization Model Formulation

The main objective of the optimization model is to define the optimal system structure and the maintenance policy for each component, so that the multi-state system performance is maximized, subject to steady state availability and cost budget constraints.

The objective function is evaluated as the average performance rate of the whole system  $P_{sys}$ , which is calculated from the performance rate of each subsystem. The working processing time of a series system is the sum of the processing time of each subsystem and the corresponding rate is evaluated as

$$P_{sys} = 1 / \sum_{i=1}^m (1/P_{ri}(n_i, \lambda_{mi}, n_{ri})) = f(\mathbf{n}, \boldsymbol{\lambda}_m, \mathbf{n}_r) \tag{5}$$

where  $\mathbf{n}, \boldsymbol{\lambda}_m, \mathbf{n}_r$  are the corresponding vectors for each parameter in all the subsystem.

The constraint functions include the constraint of reliability, redundancy cost and maintenance cost. The system steady state availability  $A_{sys}$  is the probability of the whole system in working state, which is the product of each subsystem's availability:

$$A_{sys} = \prod_{i=1}^m A_i(n_i, \lambda_{mi}, n_{ri}) = g(\mathbf{n}, \boldsymbol{\lambda}_m, \mathbf{n}_r). \tag{6}$$

The overall system redundancy cost  $C_{sys}$  is the cost to configure redundant components in each subsystem, which depends on the number of redundant components.

The monitoring and inspection cost is the cost to monitor and detect the status of each subsystem periodically and it is related to the monitoring rate in each subsystem. While the cost is also related to the system running time,  $M_{sys}$  stands for the monitoring and inspection cost per time unit. The repair facility cost  $R_{sys}$  is the cost used to repair failed components, which is determined by the maximal number of repair facilities prepared for each subsystem. Let  $c_i$  denotes the redundancy cost of components in subsystem  $i$ ,  $m_{di}$  stands for the cost for one inspection and detection process and the  $r_i$  is the cost of each prepared repair facility. The cost constraint functions are represented as

$$C_{sys} = \sum_{i=1}^m c_i n_i, M_{sys} = \sum_{i=1}^m m_{di} \lambda_{mi}, R_{sys} = \sum_{i=1}^m r_i n_{ri} \tag{7}$$

When redundancy and inspection-based mechanism are both used in one system, it is difficult to choose the configuration settings of each kind of strategy in each subsystem. Based on these objective and constraint functions, the parameter determination problem is cast into an optimization problem. The optimization model is defined as choosing appropriate system configuration parameters  $\mathbf{n}$ ,  $\lambda_m$ ,  $\mathbf{n}_r$  to maximize system work processing rate under the availability and cost constraints:

$$\begin{aligned} \text{Max } P_{sys} &= f(\mathbf{n}, \lambda_m, \mathbf{n}_r) \\ \text{s.t. } A_{sys} &= g(\mathbf{n}, \lambda_m, \mathbf{n}_r) \geq A_0, C_{sys} = \sum_{i=1}^m c_i n_i \leq C_0, M_{sys} = \sum_{i=1}^m m_{di} \lambda_{mi} \leq M_0, R_{sys} = \sum_{i=1}^m r_i n_{ri} \leq R_0 \\ 1 &\leq n_{ri} \leq n_i \leq n_0, 0 < \lambda_{mi} \leq \lambda_{m0} \end{aligned} \tag{8}$$

$A_0, C_0, M_0$  and  $R_0$  represents the corresponding constraints value and  $n_0, \lambda_{m0}$  are the value range of configuration parameters.

### 3 Solution Technique

Although Lagrange multiplier method could be used to solve the optimization problem by calculating partial derivative values, it asks for the closed-form representation of the optimization model. These is no closed-form formula of reliability and performance without certain value of  $n_i$  and  $n_{ri}$ , thus the proposed model is not a traditional non-linear optimization problem and could not be solved using exact method. So, in this section, a genetic algorithm is used to search the near-optimal solution.

#### 3.1 Genetic Algorithm Framework

Among several kinds of evolutionary algorithms proposed for combinatorial optimization problem, genetic algorithm (GA) conducts a random, yet directed search process wherein solutions evolves according to biological reproduction rules [16]. It is superior to gradient descent technique or random sampling algorithm and has been used in many researches for RAP [17, 18]. The general framework of this algorithm is:

Step I. Initialize population  $P_0$  with  $N$  individuals,  $P_0 = \{P_1, P_2, \dots, P_N\}$ . Set offspring population  $Q_0 = \emptyset$ , and generation counter  $t=0$ .

Step II. While  $t < t_{max}$ , do the following things:

1. Use genetic operator to generate  $Q_t$  from  $P_t$ .
  2. Combine both parent and offspring population to get a new group of population  $R_t = P_t \cup Q_t$ .
  3. Calculate the fitness value of individuals in  $R_t$  according to the objective values and constraints values.
  4. Sort  $R_t$  in descending order using fitness value.
  5. Use roulette selection operator to select  $N$  individuals in  $R_t$  and fill  $P_{t+1}$  with them.
  6. Set  $t=t+1$  and return to the beginning of Step II.
- Step III. Stop the algorithm when it reaches the maximal iteration number.

Following similar framework, the significant parts in genetic algorithm are the solution encoding mechanism, genetic operators and the fitness calculation method. The design of these parts is listed in the following subsections.

### 3.2 Solution Encoding Mechanism

Each individual in the population set represents a specific solution to the optimization problem. Since there are  $3 \times m$  elements to search in the optimization model, a triple-element encoding mechanism is used to represent each solution to the problem [19, 20]. Each solution in our algorithm is represented by a  $3 \times m$  matrix. Figure 4 illustrates an example of the encoded solution using this mechanism with  $m = 6$ .

For the matrix shown in Fig. 4, the first row represents the redundancy of components in each subsystem  $n$ , the second row stands for the number of repair facilities for each subsystem  $n_r$  while the third row is the inspection and monitoring rate of each subsystem  $\lambda_m$ . Each column of this matrix is the redundancy and inspection-based maintenance configuration of each subsystem. The initialization of a set of these individuals is accomplished by randomly generating values of each element in the matrix. The initial value of  $n_i$  or  $n_{ri}$  is a random integer which satisfies the condition of  $1 \leq n_{ri} \leq n_i \leq n_0$ . The initial value of  $\lambda_{mi}$  is real number randomly generate between 0 and  $\lambda_{m0}$ .

		1	2	3	4	5	6
Redundancy	1	2	1	4	4	5	2
Number of repair facility	2	1	1	3	2	3	1
Inspection rate	3	0.1	0.023	0.04	0.2	0.03	0.1

Fig. 4. Example of triple-element encoding mechanism.



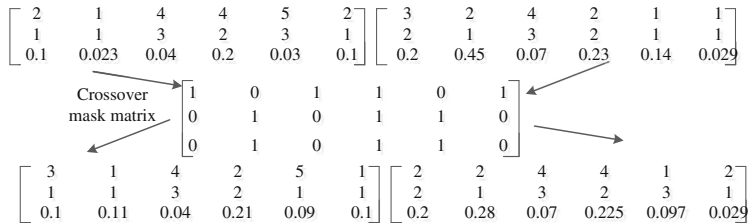


Fig. 5. Crossover operation for triple-element encoded individuals.

3.3 Genetic Operators

In the crossover operation, a  $3 \times m$  crossover mask is generated as Fig. 5. There are only 0 and 1 in this mask matrix and this matrix is randomly composed by the two values. For two randomly selected individuals  $p_x$  and  $p_y$  ( $1 \leq x, y \leq N$ ), once an element in the mask matrix equals to 1, crossover operation is conducted on the value of elements in the same position of the two solution matrixes. For integer values as  $n_i$  or  $n_{ri}$  simple swap operation is used in crossover operation. For real values of  $\lambda_{mi}$ , the simulated binary crossover (SBX) operator is used to switch two elements:

$$\begin{aligned}
 p_x \cdot \tilde{\lambda}'_{mi} &= \frac{((1-\beta)p_x \cdot \tilde{\lambda}_{mi} + (1+\beta)p_y \cdot \tilde{\lambda}_{mi})}{2} \\
 p_y \cdot \tilde{\lambda}'_{mi} &= \frac{((1+\beta)p_x \cdot \tilde{\lambda}_{mi} + (1-\beta)p_y \cdot \tilde{\lambda}_{mi})}{2}
 \end{aligned}
 , \beta = \begin{cases} (2\alpha)^{\frac{1}{\eta+1}}, & 0 \leq \alpha < 0.5 \\ \frac{1}{((2(1-\alpha))^{\frac{1}{\eta+1}})}, & 0.5 \leq \alpha < 1 \end{cases} \tag{9}$$

where  $\eta$  is the crossover index and  $p_x \cdot \tilde{\lambda}_{mi}$  is the normalized value of  $p_x \cdot \lambda_{mi}$ .

In the mutation operation, similarly, a  $3 \times m$  mutation mask is generated with randomly chosen 0 and 1, as Fig. 6. At most two elements could be set to 1 at each row to avoid dramatic change in the mutated individual. For any randomly chosen individual  $p_x$ , if any element in mutation mask is 1, the corresponding element at the same position of  $p_x$  is changed randomly. The changing process of integer element such as  $n_i$  or  $n_{ri}$  is similar as the population initialization process. For real numbers  $\lambda_{mi}$ , polynomial mutation operator is used to make changes as:

$$p_x \cdot \tilde{\lambda}'_{mi} = \begin{cases} p_x \cdot \tilde{\lambda}_{mi} + (2r)^{\frac{1}{\eta_m+1}} - 1, & 0 \leq r < 0.5 \\ p_x \cdot \tilde{\lambda}_{mi} + 1 - [2(1-r)]^{\frac{1}{\eta_m+1}}, & 0.5 \leq r < 1 \end{cases} \tag{10}$$

where  $\eta_m$  denotes the mutation index and  $p_x \cdot \tilde{\lambda}_{mi}$  is the normalized value.

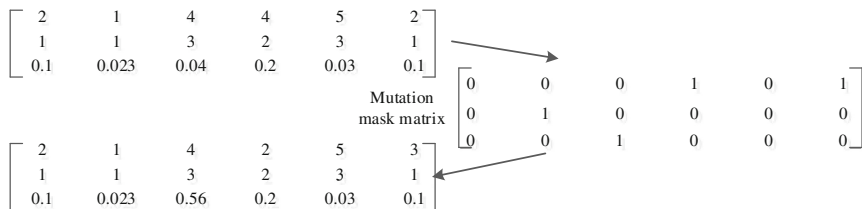


Fig. 6. Mutation operation for triple-element encoded individuals.

### 3.4 Fitness Value Calculation

The optimization problem in (8) is a single-objective multi-constraint optimization problem. The objective value represents the optimality of one individual while the constraint value represents whether this individual meets the requirement. One individual  $p_x$  outperforms another  $p_y$  if and only if the objective value of  $p_x$  is larger than the value of  $p_y$  and the constraint value of  $p_x$  meets requirement. Thus, the fitness value  $I$  of each individual is calculated as the objective value meeting constraints value:

$$I = \begin{cases} R_{sys}, A_{sys} \geq A_0, C_{sys} \leq C_0, M_{sys} \leq M_0, R_{sys} \leq R_0 \\ 0, else \end{cases} \quad (11)$$

In selection process, a weight value  $w_x$  is assigned for each individual so that individuals with larger fitness value will be assigned a higher possibility in selection:

$$w_x = \begin{cases} w_{x-1} + I_x / \sum_{y=1}^N I_y \\ 0, x = 0 \end{cases} \quad (12)$$

## 4 Illustrative Examples

In this section, numerical examples are listed to illustrate the optimization of joint redundancy and inspection-based maintenance mechanism and the process of searching near-optimal solutions. System parameters are mainly collected from a cloud-based system used for distributed and parallel processing [27]. Data from 8 series-parallel subsystems in this system are used for analysis. For each subsystem, the failure rate, repair rate, job processing rate of one component are extracted from the operation profiles and presented in Table 1. According to the maintenance schedule, the value of  $\mu_{si}$  for each component is 30 per hour and the inspection and monitoring cost of each component is set as 1 unit. The repair successful probability  $q_i$  is 0.9 and the performance degradation rate is 0.8. The redundancy cost  $c_i$  and repair facility cost  $r_i$  are estimated and shown in Table 1.

**Table 1.** Component parameters for example

Subsystem No.	Component feature					Subsystem No.	Component feature				
	$\lambda_i$ (per hour)	$\mu_i$ (per hour)	$p_{ri}$	$c_i$	$r_i$		$\lambda_i$ (per hour)	$\mu_i$ (per hour)	$p_{ri}$	$c_i$	$r_i$
1	0.00499	0.551	2	2	3	5	0.00431	0.782	1	2	4
2	0.00818	0.623	3	2	7	6	0.00567	0.445	3	3	5
3	0.00466	1.13	3	4	5	7	0.0105	1.75	2	3	6
4	0.00683	0.973	2	5	4	8	0.0150	0.237	2	3	4

### 4.1 Reliability and Performance Analysis

The reliability and performance analysis is first conducted on the first subsystem. The failure rate  $\lambda_i$  and repair rate  $\mu_i$  is 0.00499 and 0.551 per hour. The inspection and detection rate  $\underline{\mu}_{si}$  is 30 per hour. Using only one subsystem, the working processing rate  $p_{ri}$  in this experiment is set as 1 unit. First of all, the inspection rate is set as 0.5 per hour and the number of working repair facility is set as 3. The corresponding reliability and performance of this subsystem with redundancy change from 1 to 10 are shown in Fig. 7(a). Then the number of redundant component is set as 5 and inspection rate changes from 0 to 1 per hour. The result of subsystem with different number of repair facilities is included in Fig. 7(b). At last, the number of redundant component is set as 5 and the inspection rate is 0.5 per hour, the result of subsystem with the number of repair facilities changing from 1 to 10 is shown as Fig. 7(c).

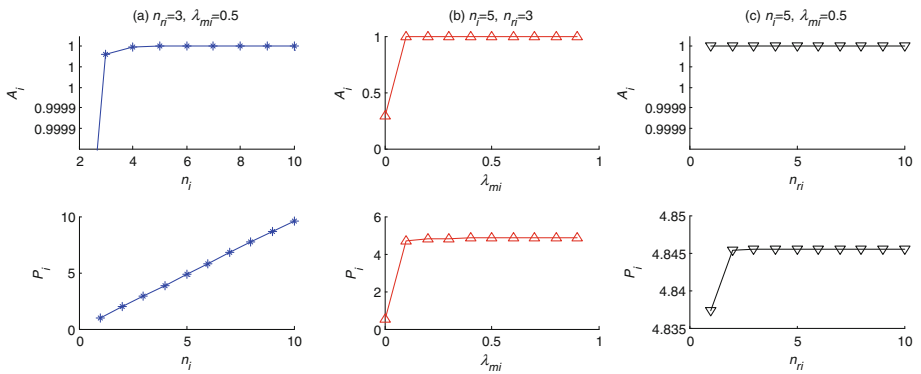
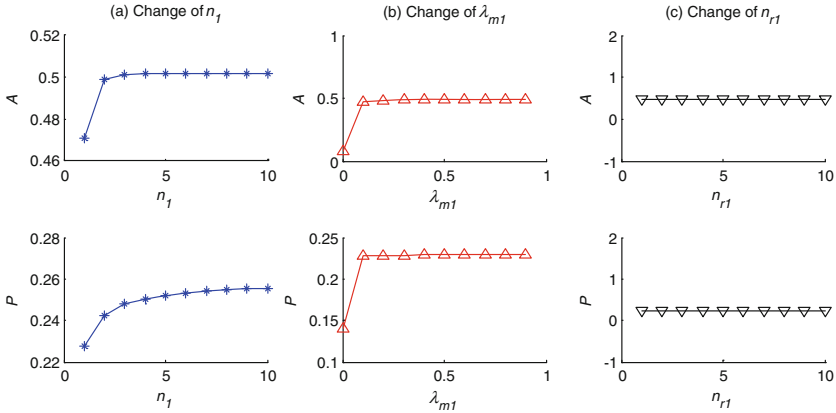


Fig. 7. Subsystem reliability and performance change with different parameters.

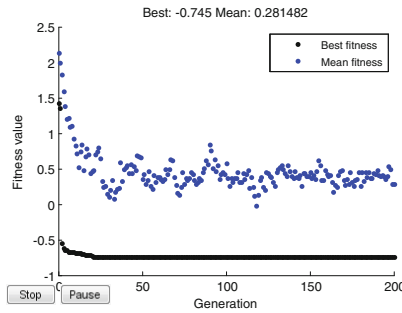
It is shown from Fig. 7 that the increase of redundant components and inspection rate has significant impact on the subsystem reliability and performance change. However, with the continuous change of these parameters, the increase trend of both reliability and performance rate declines and converges to 0, except for the performance increase with redundancy increase. The change in the number of repair facility has little impact on both reliability and performance increase. When the number of working repair facilities is larger than then number of redundant components, there is no change in both system metrics. The experiment data shows that three kinds of configuration parameters have total different impact on the overall system.

### 4.2 Mechanism Optimization

Figure 8 illustrates the overall system reliability and performance change with the configuration parameters change of the first subsystem. The initial redundancy and repair facility for each subsystem is set as 1 and the inspection rate is 0.1 per hour. It is shown that the impact of parameter change on the reliability and performance improvement generally declines dramatically as the parameters increases.



**Fig. 8.** System reliability and performance change of different groups of parameters.



**Fig. 9.** The mean and best fitness value of each generation.

So in this section, the maximum number of redundant components and repairing facilities in any subsystem is set as 10 and 5 respectively. The maximal inspection rate is set as 1. The maximum generation in the algorithm is set as 200 and the population size is 100. System parameters in Table 1 is used in this experiment for redundancy optimization. Given availability constraint 0.99, redundancy cost 70, repair facilities cost 70 and inspection cost 5, the optimal configuration of redundancy and maintenance policy using the joint mechanism is calculated and the fitness value of each generation is shown in Fig. 9. The fitness value is the negative value of performance rate:  $-P_i$ . The result in Fig. 9 indicates that the algorithm already converges within 50 generations.

The calculated optimal job performance rate is 0.7448 in this test case and the corresponding configuration parameter for each subsystem is shown in Table 2.

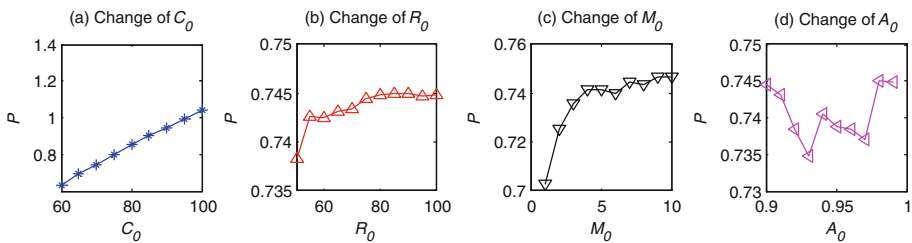
In the following experiments, four kinds of cost constraints are changed to show the different result on joint mechanism optimization. Since GA is a stochastic search algorithm, five trials are performed for each experiment and the best solution is considered as

**Table 2.** Optimal component configuration result

Subsystem No.	Component feature			Subsystem No.	Component feature		
	$n_i$	$\lambda_{mi}$ (per hour)	$n_{ri}$		$n_i$	$\lambda_{mi}$ (per hour)	$n_{ri}$
1	3	0.4637	3	5	0.6065	1	
2	3	0.7689	2	6	0.5419	2	
3	2	0.5878	1	7	0.5060	2	
4	2	0.8110	2	8	0.7092	2	

the final solution. First of all, the availability constraint is kept as 0.99 with repair facilities cost as 70 and inspection cost as 5. The result of maximized performance change with near-optimal solution is drawn in Fig. 10(a) while the redundancy cost changes from 60 to 100. Then, using the same availability constraint with redundancy cost as 70 and changing the value of repair facilities cost, the corresponding optimization result for performance rate is shown in Fig. 10(b). Keeping the cost constraints for both redundancy and repair as 70 and changing the value of inspection cost from 1 to 10, Fig. 10(c) shows the optimization result under availability constraint 0.90. Figure 10(d) presents the result for availability constraint change from 0.90 to 0.99.

It is shown in Fig. 10 that given different cost constraint, different system configuration would be get with different performance optimization result. Loosening any kind of cost constraint could all get better result on performance optimization. Specifically, the change of redundancy cost brings significant change on the optimization result while only little change is obtained for the other two kinds of cost constraints change. On the contrary, once the availability constraint could be met under the same cost, different availability constraints generally lead to similar system configuration and performance.



**Fig. 10.** The change of optimization result with different constraints.

## 5 Conclusion and Future Work

Aiming at the optimal joint redundancy and inspection-based maintenance configuration problem in series-parallel system, this paper pays attention to the routine monitoring and inspection mechanisms, which helps to detect components status and trigger repair process. Considering the shared repair facilities for different component in each subsystem, a state transition models is first built to analyze the inspection-based maintenance mechanism and Markov chain theory is used to get steady state availability and

performance metrics for the system. The optimization problem is built to search the appropriate system structure and maintenance policy maximizing system performance while meeting availability and cost constraints. Due to the complexity of uncertain optimization model, genetic algorithm is used to search optimal solution, built upon triple-element encoding mechanism. Specific genetic operators are designed for the triple-element encoding method and illustrative examples are conducted to explain the calculation and optimization process. Experiment results show that the optimization model and corresponding solution technique could be used to search optimal system configuration under given constraints and different cost constraints would lead to different optimization result while meeting availability constraints. This paper takes a preliminary study on the optimization of joint redundancy and inspection-based maintenance model. The analysis model is simplified on top of several assumptions, such as the exponential distribution time-to-failure. In the future, this model could be modified to be suited for more general failure distributions. Meanwhile, some local search operator could be combined into the genetic algorithm to improve the algorithm efficiency.

**Acknowledgments.** This work is supported by the National Natural Science Foundation of China (Grant No. 61309005) and the Frontier and Application Basic Research Program of Chongqing (Grant No. cstc2014jcyjA40015).

## References

1. Kuo, W., Wan, R.: Recent advances in optimal reliability allocation. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **37**, 143–156 (2007)
2. Soro, I.W., Nourelfath, M., Ait-Kadi, D.: Performance evaluation of multi-state degraded systems with minimal repairs and imperfect preventive maintenance. *Reliab. Eng. Syst. Saf.* **95**, 65–69 (2010)
3. Nourelfath, M., Chatelet, E., Nahas, N.: Joint redundancy and imperfect preventive maintenance optimization for series-parallel multi-state degraded systems. *Reliab. Eng. Syst. Saf.* **103**, 51–60 (2012)
4. Liu, Y., Huang, H.-Z., Wang, Z., Li, Y., Yang, Y.: A joint redundancy and imperfect maintenance strategy optimization for multi-state systems. *IEEE Trans. Reliab.* **62**, 368–378 (2013)
5. Yang, C.T., Liu, J.C., Hsu, C.H., Chou, W.L.: On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *J. Supercomputing* **69**, 1103–1122 (2014)
6. Coit, D.W.: Maximization of system reliability with a choice of redundancy strategies. *IIE Trans. (Inst. Ind. Eng.)* **35**, 535–543 (2003)
7. Tavakkoi-Moghaddam, R., Safari, J., Sassani, F.: Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliab. Eng. Syst. Saf.* **93**, 550–556 (2008)
8. Chambari, A., Rahmati, S.H.A., Najafi, A.A., Karimi, A.: A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies. *Comput. Ind. Eng.* **63**, 109–119 (2012)

9. Chambari, A., Najafi, A.A., Rahmati, S.H.A., Karimi, A.: An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies. *Reliab. Eng. Syst. Saf.* **119**, 158–164 (2013)
10. Safari, J.: Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies. *Reliab. Eng. Syst. Saf.* **108**, 10–20 (2012)
11. Levitin, G.: Optimal structure of fault-tolerant software systems. *Reliab. Eng. Syst. Saf.* **89**, 286–295 (2005)
12. Ahmadizar, F., Soltanpanah, H.: Reliability optimization of a series system with multiple-choice and budget constraints using an efficient ant colony approach. *Expert Syst. Appl.* **38**, 3640–3646 (2011)
13. He, P., Wen, J., Wu, K., Li, P., Ren, H.: Multi-objective service monitoring rate optimization using memetic algorithm. *J. Softw.* **7**, 990–997 (2012)
14. Chern, M.: On the computational complexity of reliability redundancy allocation in a series system. *Oper. Res. Lett.* **11**, 309–315 (1992)
15. Trivedi, K.S.: *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Wiley, New York (2001)
16. Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. Syst. Man Cybern.* **24**, 656–667 (1994)
17. Levitin, G., Lisnianski, A., Ben-Haim, H., Elmakis, D.: Redundancy optimization for series-parallel multi-state systems. *IEEE Trans. Reliab.* **47**, 165–172 (1998)
18. Tian, Z., Levitin, G., Zuo, M.J.: A joint reliability–redundancy optimization approach for multi-state series–parallel systems. *Reliab. Eng. Syst. Saf.* **94**, 1568–1576 (2009)
19. Tavakkoli-Moghaddam, R., Safari, J., Sassani, F.: Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliab. Eng. Syst. Saf.* **93**, 550–556 (2008)
20. Ardakan, M.A., Hamadani, A.Z.: Reliability optimization of series-parallel systems with mixed redundancy strategy in subsystems. *Reliab. Eng. Syst. Saf.* **130**, 132–139 (2014)
21. Martorell, S.A., Serradell, V.G., Samanta, P.K.: Improving allowed outage time and surveillance test interval requirements: a study of their interactions using probabilistic methods. *Reliab. Eng. Syst. Saf.* **47**, 119–129 (1995)
22. Vaurio, J.K.: The theory and quantification of common cause shock events for redundant standby systems. *Reliab. Eng. Syst. Saf.* **45**, 315 (1994)
23. Vaurio, J.K.: Optimization of test and maintenance intervals based on risk and cost. *Reliab. Eng. Syst. Saf.* **49**, 23–36 (1995)
24. Cepin, M., Mavko, B.: Probabilistic safety assessment improves surveillance requirements in technical specifications. *Reliab. Eng. Syst. Saf.* **56**, 69–77 (1997)
25. Cepin, M.: Optimization of safety equipment outages improves safety. *Reliab. Eng. Syst. Saf.* **77**, 71–80 (2002)
26. Torres-Echeverria, A.C., Martorell, S., Thompson, H.A.: Multi-objective optimization of design and testing of safety instrumented systems with MooN voting architectures using a genetic algorithm. *Reliab. Eng. Syst. Saf.* **106**, 45–60 (2012)
27. Yuwen, C., Linhong, X., Jin, Z., Leifeng, L.: Research about mobile AR system based on cloud computing. In: *Wireless and Optical Communication Conference (WOCC)*, 2013 22nd, pp. 355–359 (2013)