International Journal of Innovative
Computing, Information and Control
Volume 8, Number 10(A), October 2012

ICIC International ©2012 ISSN 1349-4198
pp. 6693–6705

AVAILABILITY ANALYSIS AND DESIGN OPTIMIZATION FOR A REPAIRABLE SERIES-PARALLEL SYSTEM WITH FAILURE DEPENDENCIES

Linmin Hu[1,2], Dequan Yue[2] and Jiandong Li[2,*]

[1]School of Economics and Management
[2]College of Science
Yanshan University
No. 438, Hebei Ave., Qinhuangdao 066004, P. R. China
linminhu@163.com; ydq@ysu.edu.cn; *Corresponding author: math@ysu.edu.cn

Received July 2011; revised January 2012

Abstract. *In this paper, the design of a repairable series-parallel system with failure dependencies is studied. The key point is that failure dependency and limited repair teams can affect the state transition rates of each subsystem, and thus affect the state distribution of the subsystem and the availability of the system. Taking this into consideration, we present a dependence function to determine the failure rate of components in each subsystem and a Markov model to determine state distribution of the subsystem. An optimal allocation problem is proposed and is aimed at minimizing the system cost consisting of costs associated with components and costs associated with repair teams subject to the system availability constraint. To solve the optimization problem, the dependence function is specified and genetic algorithm is used to determine the optimal allocation solutions due to its flexibility in representing discrete design variables and its robust search capability. A numerical example is presented to illustrate that different dependencies make the allocation strategies different.*

Keywords: Repairable system, Series-parallel, Optimization, Failure dependency, Markov model, Genetic algorithm

1. **Introduction.** Optimal design of system and reliability optimization play a key role in engineering design and have been effectively applied to enhance performance [1]. Repairable system indicates that a system can be repaired to operate normally in the event of any failure, and system availability is a concept closely related to reliability and refers to scale of measuring the reliability of a repairable system [2]. For repairable system, availability is a very meaningful measure, and achieving a high or required level of availability is an essential requisite. In general, redundant components and repair teams are used to provide a required level of system availability. While increasing the number of redundant components and the number of repair teams, the cost is also on the upswing. Therefore, system designers and decision-makers typically try to determine how many redundant components and repair teams to use in each subsystem, in order to minimize the system cost while satisfying the system availability constraint.

A series-parallel system consists of a few subsystems connected in series whereas each subsystem consists of a few components connected in parallel. A subsystem is failed if all the components in the subsystem are failed. Failure of any subsystem causes the failure of the whole system. The reliability or availability of the series-parallel system can be improved by increasing redundant components in parallel subsystems as an effective design strategy. Thus, redundancy allocation must be considered in the initial design activity. A redundancy allocation problem (RAP) of the series-parallel system refers to difficult

NP-hard combinatorial optimization problems. Much research has been investigated in the area of RAP for series-parallel system with various assumptions. Ramirez-Marquez et al. [3] showed that optimization approaches to determine optimal or very good solutions include dynamic programming, integer programming, mixed integer and non-linear programming and heuristics. Comprehensive overviews of this problem have been addressed by Kuo et al. [4] and Gen et al. [5].

Repairable series-parallel systems are frequently used in practice, e.g., power systems, telecommunications systems, manufacturing production systems, and industrial systems. The availability estimation and redundancy optimization for repairable series-parallel system have drawn continuous attention both in problem characteristics and solution methodologies [2,6-8]. In traditional RAP for a repairable series-parallel system, the repair resources are assumed to be unlimited. Recently, the research work [9] for the RAP of the system has proposed the more general case where the repair facilities are limited. This problem is more realistic than the classical RAP because it takes into account limited number of maintenance teams. In [9], the authors assumed allocating a set of maintenance personnel to each subsystem, used universal generating function and Markov chain model to evaluate the availability of each subsystem, and used a heuristic approach to solve the RAP of the system. In the studies of [2,6-9], it has been assumed that component failures occur independently in each subsystem. However, there are several situations in which this independence assumption is not valid.

In some systems, the failure rate of the operating components will increase due to the additional loading induced by the other failed components. Yu et al. [10] stated that failure dependencies consider the interactions in the failure process of a system. Pecht [11] proposed the following three types of failure dependency: common-mode failure, multi-mode failure and other failure dependencies. The optimization problem for series-parallel systems with common-mode failure was discussed in [12,13]. Levitin et al. [14,15] investigated the optimization problem for series-parallel multi-state systems with two failure modes. For the other failure dependencies, Ebeling [16] and Barros et al. [17] analyzed two-component systems, in which the failure rate of the operating component will increase due to the additional loading induced by the other failed components. In a recent study [10], an N-component redundant system with the consideration of redundant dependency was investigated and a dependency function was introduced to quantify the redundant dependency.

Though the RAP of repairable series-parallel systems and the failure dependency problem for some systems have been reported in the above research; so far very few researchers have studied the redundant components and repair teams allocation problem for repairable series-parallel system with failure dependencies. Failure dependency is often neglected in reliability analysis due to its complexity, and repair resources are often assumed to be unlimited in some system models. However, a wide variety of dependencies exist among the failure behavior of systems, and the assumption of unlimited repair resources is easily violated in practice. This motivates us to develop an optimization allocation problem for a repairable series-parallel system with failure dependencies, where limited repair teams are available for each subsystem. In this work, genetic algorithm (GA) is used to solve the optimization allocation problem. Chern [18] showed that even a simple RAP with linear constraints is NP-hard. Hence, some researchers try to develop meta-heuristic algorithms to achieve optimal or very good solutions in a reasonable computational time. As a member of meta-heuristic algorithms, GA has proved itself to be effective optimization tool for a large number of applications. Successful applications of GA to optimization allocation problem in reliability engineering are reported in [2,6-8,12-15,19].

The purpose of this paper is to present analysis of availability and the optimal design for the repairable series-parallel system with redundant dependency. A dependence function [10] is used to determine the failure rate of components in each subsystem; Markov model is developed to determine state distribution of each subsystem based on its transition rates and thus determine the subsystem availability. Then, the explicit expression of the system availability can be obtained. In addition, an optimal allocation problem is proposed and is aimed at minimizing the system cost consisting of costs associated with components and costs associated with repair teams subject to the system availability constraint. The GA is used to determine the optimal solutions due to its flexibility in representing discrete design variables and its good global optimization capability.

The structure of this paper is organized as follows. The problem is formulated in Section 2. In Section 3, the availability of the system with different redundant dependencies is analyzed. GA is used to solve the optimization problem in Section 4. A numerical example is provided in Section 5. Conclusions are given in Section 6.

Nomenclature

m number of subsystems in the repairable series-parallel system
n_i number of components in subsystem i
r_i number of repair teams in subsystem i
C_i^{c} unit cost of components in subsystem i
C_i^{r} unit cost of repair teams in subsystem i
C_S cost of the system
A_S system availability
A_0 system availability constraint value
Z^+ the space discrete of positive integers
P_j probability of a subsystem in state j
λ_j^i transition rate from state j to $j-1$ for subsystem i
μ_j^i transition rate from state $j-1$ to j for subsystem i
A_i availability of subsystem i
$g(\cdot)$ dependence function
λ_i inherent failure rate of a component in subsystem i
μ_i repair rate of a component in subsystem i
$\mathbf{n}$ vector associated with numbers of components $\mathbf{n} = (n_1, n_2, \ldots, n_m)$
$\mathbf{r}$ vector associated with numbers of repair teams $\mathbf{r} = (r_1, r_2, \ldots, r_m)$
v_k representation of chromosome
F_k individual fitness

2. **Problem Formulation.** The common structure of a series-parallel system is illustrated in Figure 1. The system consists of m subsystems with failure dependencies connected in series, and each subsystem i ($i = 1, 2, \ldots, m$) has n_i components connected in parallel. Without loss of generality, suppose that all components are identical in each subsystem. Each parallel subsystem works if and only if at least one of its components work, and the entire system works if and only if all subsystems work. In addition, there are r_i repair teams available in each subsystem i ($i = 1, 2, \ldots, m$).

Furthermore, other assumptions are given as follows:

(1) The system and component have two states: perfect functioning and complete failure.
(2) A repaired component is as good as a new one.
(3) In each subsystem, the failure rates of operating components increase with the number of other failed components.
(4) The repair rate of each component is constant.
(5) Each repair team can repair only one failed component at a time.

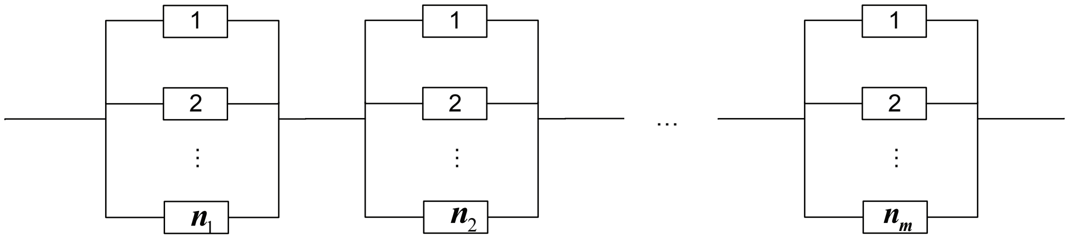

FIGURE 1. Series-parallel system structure

Our objective is to present the explicit expression of availability and to minimize the construction cost of the system consisting of the components costs and the repair teams costs while satisfying the system availability constraint. The design optimization problem to be discussed in this paper can be formulated as follows:

$$\begin{aligned}&\text{Minimize} \;\; C_S = \sum_{i=1}^{m} (n_i C_i^{\mathrm{c}} + r_i C_i^{\mathrm{r}}) \\ &\text{Subject to} \;\; A_S \geq A_0 \\ &\qquad\qquad\; n_i, r_i \in Z^+ \\ &\qquad\qquad\; r_i \leq n_i, \quad i = 1, 2, \ldots, m\end{aligned} \tag{1}$$

where C_S is the total cost of the system, A_S is the system availability. n_i is the number of components in subsystem i, and C_i^{c} is the unit cost of components in subsystem i. r_i is the number of repair teams in subsystem i, and C_i^{r} is the unit cost of repair teams in subsystem i. A_0 is the system availability constraint value.

3. **Availability and Failure Dependency Analysis.** To solve the optimization problem (1), it is important to have an effective approach to calculate the availability for the repairable series-parallel system with failure dependency. In this section, the proposed method is based on Markov model with a dependence function. We will discuss how failure dependency (redundant dependency) affects the system availability.

3.1. **Subsystem availability.** Considering a subsystem i ($i = 1, 2, \ldots, m$) composed of n_i identical component with exponential failure distribution in parallel, where any component can be viewed as a redundancy of another component. Each subsystem fails if and only if all of the components fail. r_i repair teams are available to repair the failed components, and each repair team can repair only one failed component at a time.

Let P_j ($j = 0, 1, \ldots, n_i$) be the probability of the subsystem state that only j components are working and the other $n_i - j$ components have failed during the stationary regime. It is assumed that transitions can only occur between adjacent states, and the state transition diagram of the subsystem i is shown in Figure 2. The transition rate from

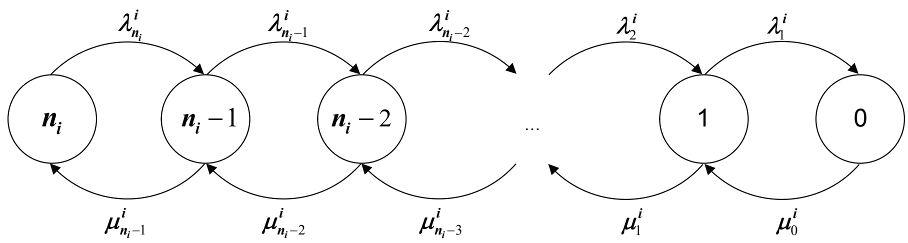

FIGURE 2. State transition diagram of the subsystem i

state j to $j-1$ is λ_j^i. The transition rate from state $j-1$ to j is μ_j^i. Based on Figure 2 and the Markov model, we have the following transition rate matrix

$$\mathrm{Q}=\begin{pmatrix} -\mu_0^i & \mu_0^i & & & & \\ \lambda_1^i & -(\lambda_1^i+\mu_1^i) & \mu_1^i & & & \\ & \lambda_2^i & -(\lambda_2^i+\mu_2^i) & \mu_2^i & & \\ & & \ddots & \ddots & \ddots & \\ & & & \lambda_{n_i-1}^i & -(\lambda_{n_i-1}^i+\mu_{n_i-1}^i) & \mu_{n_i-1}^i \\ & & & & \lambda_{n_i}^i & -\lambda_{n_i}^i \end{pmatrix} \tag{2}$$

The steady-state probability vector $\boldsymbol{P}$ is the solution of the following equations

$$\boldsymbol{P}\mathrm{Q}=\mathbf{0} \tag{3}$$

$$\boldsymbol{P}\mathbf{e}=1 \tag{4}$$

where $\boldsymbol{P}=(P_0,P_1,\ldots,P_{n_i})$ and $\mathbf{e}=(1,1,\ldots,1)^{\mathrm{T}}$. The general form solution of stationary availability of the subsystem i is

$$A_i=\sum_{j=1}^{n_i}P_j=1-P_0 \tag{5}$$

By using the theory of linear equations, we easily obtain

$$P_j=\left(\frac{\mu_0^i\mu_1^i\cdots\mu_{j-1}^i}{\lambda_1^i\lambda_2^i\cdots\lambda_j^i}\right)P_0,\quad j=1,2,\ldots,n_i \tag{6}$$

where

$$P_0=\left(1+\sum_{j=1}^{n_i}\frac{\mu_0^i\mu_1^i\cdots\mu_{j-1}^i}{\lambda_1^i\lambda_2^i\cdots\lambda_j^i}\right)^{-1} \tag{7}$$

For each parallel subsystem i, the explicit expression of availability A_i is obtained as

$$A_i=1-\left(1+\sum_{j=1}^{n_i}\frac{\mu_0^i\mu_1^i\cdots\mu_{j-1}^i}{\lambda_1^i\lambda_2^i\cdots\lambda_j^i}\right)^{-1} \tag{8}$$

3.2. **Failure dependency.** Failure dependency is often neglected in availability analysis for repairable systems. However, a wide variety of dependencies exist among the failure behavior of repairable systems in practice. See, for instance, some engineering systems with multi-component are designed to support varying amounts of load. In general, the failure rate of a component depends on the load it supports and increases with the load [20]. The types of failure dependencies among systems are different and depend on the functional and constructional configuration of the systems. Yu et al. [10] showed that the failure dependency of a system is called redundant dependency if any component can be viewed as a redundancy of another component. Here, the redundant dependency is considered for each subsystem.

To quantify the redundant dependency, we use a dependence function $g(j)$ ($j=1,2,\ldots,n_i$) [10] which is a function of the number of the redundant components. It is assumed that the failure rate of components in each redundant subsystem depends on the dependence function $g(j)$ and its inherent failure rate (failure rate at failure independency). The failure rate of components in each redundant subsystem i can be expressed as

$$\frac{\lambda_i}{g(j)},\quad j\geq 2,\quad g(1)\equiv 1 \tag{9}$$

where λ_i is the inherent failure rate of the component, $g(j)$ is the dependence function and j is the number of working components in subsystem i. Accordingly, we can obtain the transition rate λ_j^i from state j to $j-1$ for the subsystem i

$$\lambda_j^i = j\frac{\lambda_i}{g(j)}, \quad j = 1, 2, \ldots, n_i \tag{10}$$

In general, the failure rate of components in a redundant system with failure dependency is less than that of a system with failure independency, i.e., $g(j) \geq 1$. The dependence function $g(j)$ is used to shows the strength of the failure dependency. The higher the value of $g(j)$, the stronger the dependency will be.

In some redundant systems, the repair dependency may also be considered. It may be supposed that the repair rate of components is a function of the number of the redundant components. Here, we assume that the repair rate of components is independent of the number of the redundancies. Let μ_i be the repair rate of components and r_i repair teams are presented in subsystem i, and each repair team can repair only one failed component at a time. We can obtain the transition rate μ_j^i from state $j-1$ to j for the subsystem i with r_i repair teams

$$\mu_j^i = \begin{cases} r_i\mu_i, & j = 0, 1, \ldots, n_i - r_i, \\ (n_i - j)\mu_i, & j = n_i - r_i + 1, n_i - r_i + 2, \ldots, n_i - 1. \end{cases} \tag{11}$$

By using (10) and (11), the availability A_i of the subsystem i $(i = 1, 2, \ldots, m)$ is obtained as follows:

$$A_i = 1 - \left[1 + \sum_{j=1}^{n_i - r_i} \frac{r_i^j \prod_{k=1}^{j} g(k)}{j!}\left(\frac{\mu_i}{\lambda_i}\right)^j + \sum_{j=n_i-r_i+1}^{n_i} \frac{r_i^{n_i-r_i} r_i! \prod_{k=1}^{j} g(k)}{j!(n_i-j)!}\left(\frac{\mu_i}{\lambda_i}\right)^j\right]^{-1} \tag{12}$$

3.3. **System availability.** A series-parallel system can be represented by a series of parallel subsystems, as observed in Figure 1. The availability of this system considered can be determined by

$$A_S = \prod_{i=1}^{m} A_i \tag{13}$$

where A_i is the availability of the subsystem i. Considering the redundant dependency for each subsystem i, the availability of each subsystem A_i can be represented by Equation (12). Replacing A_i in Equation (13) by Equation (12), the availability of this repairable series-parallel system with redundant dependency is obtained

$$A_S = \prod_{i=1}^{m}\left\{1 - \left[1 + \sum_{j=1}^{n_i - r_i} \frac{r_i^j \prod_{k=1}^{j} g(k)}{j!}\left(\frac{\mu_i}{\lambda_i}\right)^j + \sum_{j=n_i-r_i+1}^{n_i} \frac{r_i^{n_i-r_i} r_i! \prod_{k=1}^{j} g(k)}{j!(n_i-j)!}\left(\frac{\mu_i}{\lambda_i}\right)^j\right]^{-1}\right\} \tag{14}$$

The level of redundant dependency among components is different and depends on the functional and constructional configuration of a system. In some systems, it can be expressed in terms like weak dependence, moderate dependence and strong dependence. In general, the level of redundant dependency can not be accurately estimated. Here, the dependence function introduced $g(j)$ can be used to indicate the level of the redundant dependency, different redundant dependencies can be classified through the value of $g(j)$. Based on the value of $g(j)$, four types of redundant dependencies are presented by Yu et al. [10] and are adopted in this paper as follows: 1) independence ($g(j) = 1$), 2) weak dependence ($1 < g(j) < j$), 3) linear dependence ($g(j) = j$), and 4) strong dependence ($g(j) > j$).

The following cases give the availability of the system with different types of failure dependency:

Case 1. *independence* ($g(j) = 1$)

$$A_S = \prod_{i=1}^{m} \left\{ 1 - \left[1 + \sum_{j=1}^{n_i - r_i} \frac{r_i^j}{j!} \left(\frac{\mu_i}{\lambda_i} \right)^j + \sum_{j=n_i-r_i+1}^{n_i} \frac{r_i^{n_i - r_i} r_i!}{j!(n_i - j)!} \left(\frac{\mu_i}{\lambda_i} \right)^j \right]^{-1} \right\} \tag{15}$$

Case 2. *linear dependence* ($g(j) = j$)

$$A_S = \prod_{i=1}^{m} \left\{ 1 - \left[1 + \sum_{j=1}^{n_i - r_i} r_i^j \left(\frac{\mu_i}{\lambda_i} \right)^j + \sum_{j=n_i-r_i+1}^{n_i} \frac{r_i^{n_i - r_i} r_i!}{(n_i - j)!} \left(\frac{\mu_i}{\lambda_i} \right)^j \right]^{-1} \right\} \tag{16}$$

Case 3. *weak dependence and strong dependence (as an example, $g(j)$ is uniformed as $g(j) = j^l$, $0 < l < 1$: weak dependence and $l > 1$: strong dependence)*

$$A_S = \prod_{i=1}^{m} \left\{ 1 - \left[1 + \sum_{j=1}^{n_i - r_i} r_i^j (j!)^{l-1} \left(\frac{\mu_i}{\lambda_i} \right)^j + \sum_{j=n_i-r_i+1}^{n_i} \frac{r_i^{n_i - r_i} r_i! (j!)^{l-1}}{(n_i - j)!} \left(\frac{\mu_i}{\lambda_i} \right)^j \right]^{-1} \right\} \tag{17}$$

4. Optimization Model and Genetic Algorithm.

In this section, we will present the optimization model for the repairable series-parallel system with different types of redundant dependencies. The objective is to minimize the system cost subject to the specified minimum required level of system availability. Here, GA is used to solve this optimization model.

4.1. The optimization model with redundant dependency.

Based on the discussions above, the optimization model (1) with the redundant dependency can be rewritten as follows:

$$\text{Minimize} \quad C_S(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{m} (n_i C_i^{\mathrm{c}} + r_i C_i^{\mathrm{r}})$$

$$\text{Subject to} \quad A_S(\mathbf{n}, \mathbf{r}) =$$

$$\prod_{i=1}^{m} \left\{ 1 - \left[1 + \sum_{j=1}^{n_i - r_i} \frac{r_i^j \prod_{k=1}^{j} g(k)}{j!} \left(\frac{\mu_i}{\lambda_i} \right)^j + \sum_{j=n_i-r_i+1}^{n_i} \frac{r_i^{n_i - r_i} r_i! \prod_{k=1}^{j} g(k)}{j!(n_i - j)!} \left(\frac{\mu_i}{\lambda_i} \right)^j \right]^{-1} \right\} \geq A_0 \tag{18}$$

$$n_i, r_i \in Z^+$$

$$r_i \leq n_i, \quad i = 1, 2, \ldots, m$$

where $\mathbf{n} = (n_1, n_2, \ldots, n_m)$ and $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ indicate the vectors associated with numbers of components and numbers of repair teams, respectively.

The system availability $A_S(\mathbf{n}, \mathbf{r})$ depends the following factors: 1) the number of subsystems, m; 2) the number of components and the number of repair teams in each subsystem, $\mathbf{n}$ and $\mathbf{r}$; 3) inherent failure rate and repair rate of each component, λ_i and μ_i; and 4) dependence function of the subsystem i when j component are working, $g(j)$. The number of subsystems, m, is usually determined by the system function required [21]. The inherent failure rate and repair rate of a component can be estimated based on available data. In this paper, we assume that both λ_i and μ_i are known. Moreover, to solve the optimization problem formulated in (18), we also assume that the dependence function $g(j)$

is determined through specific forms as indicated in Cases 1-3. Thus, the optimal design problem is concerned with finding the optimal solutions $\mathbf{n}$ and $\mathbf{r}$ (the optimal values of $n_1, n_2, \ldots, n_m$ and $r_1, r_2, \ldots, r_m$) to minimize $C_S(\mathbf{n}, \mathbf{r})$ subject to the system availability constraint.

4.2. **Genetic algorithm.** The optimization problem formulated in (18) is a constrained nonlinear optimization problem with discrete design variables. GA is used to obtain the optimal solutions of this optimization problem. The GA is basically an evolutionary algorithm and is a search strategy that employs random choice to guide a highly exploitative search, striking a balance between exploration of the feasible domain and exploitation of good solutions [22]. It is one of the most powerful and broadly applicable probabilistic search and optimization techniques based on concepts from evolution theory. The GA operates with "chromosomal" representation of solutions, where crossover, mutation and selection procedures are applied. Unlike various constructive optimization algorithms that use sophisticated methods to obtain a good singular solution, the GA deals with a set of solutions (population) and tends to manipulate each solution in the most simple manner [23]. There are many reports that give details on the family of GA. More details on the topic are presented in [24-26].

A simplified flow chart solving the formulated optimization problem by GA is illustrated in Figure 3. First of all, the GA randomly generates an initial population of potential

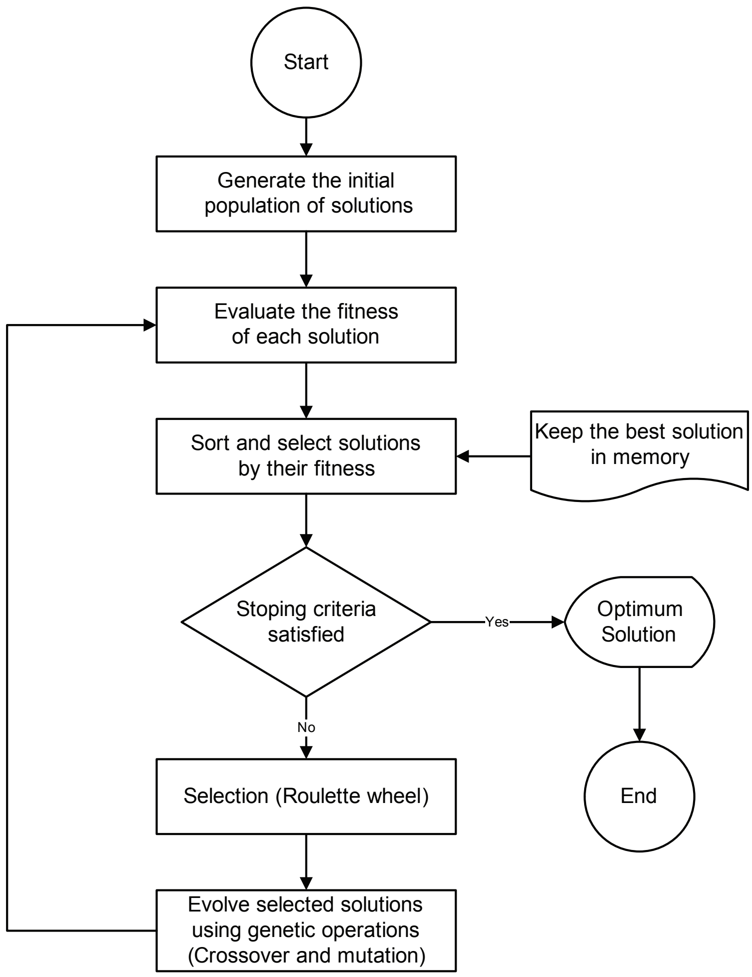

FIGURE 3. Flow chart of the procedure using genetic algorithm

solutions (individuals), referred to as the parent population, with a predefined size. The individuals of this population are evaluated using the fitness function which is determined based on the corresponding system cost and system availability. The solution with respect to the best fitness value is saved and the solutions of the current population are sorted based on their fitness values. A selection procedure is used to select the individuals candidate for reproduction on the basis of the fitness function. In order to produce a new offspring pool, the crossover operator and the mutation operator are implemented. Therefore, from the initial population a new generation is obtained. From this new generation, a second new generation is produced by the same process and so on. The stop criterion is based on the number of generations.

In the remainder of this section, the main steps of the GA we have implemented are described below in detail.

4.2.1. *Solution encoding and initial population.* A gene is defined as an ordered couple of n_{ki} and r_{ki}, just like this $v_{ki} = (n_{ki}, r_{ki})$, where the subscript k is index of chromosome that the gene belongs to, the n_{ki} denotes the number of components and r_{ki} denotes the number of repair teams in subsystem i. A chromosome is represented as

$$v_k = [(n_{k1}, r_{k1}), (n_{k2}, r_{k2}), \cdots, (n_{km}, r_{km})]$$

The size of any population is given and remains the same at each generation. Here, set the size of the population to 50. The initial population of chromosome is randomly generated. An integer between 1 and $\max_{n_i}$ (maximum number of components allowed in subsystem i) is randomly selected to represent n_{ki}, and r_{ki} is randomly selected with limits: $r_{ki} \le n_{ki}, r_{ki} \in Z^+$.

4.2.2. *Evaluation of individuals.* Each individual of the population is judged by the functional value of the fitness function. As GA follows the rule of survival-of-the-fittest candidate in nature to make a search process, so the algorithm is very suitable for solving some optimization problems. In order to provide an efficient search through the infeasible region but to assure that the final best solution is feasible, the fitness function is determined based on the corresponding system cost, system availability and a penalty function as follows:

$$F_k = \sum_{i=1}^{m} (n_{ki} C_i^{\mathrm{c}} + r_{ki} C_i^{\mathrm{r}}) + K \times \max\{0, A_0 - A_{kS}\} \tag{19}$$

where the penalty term K is a very large positive number and is used to discourage the constraint violation. When the system availability A_{kS} is less than the constraint value A_0, the individual fitness value becomes larger, and thus only solutions satisfying constraints are selected during the process of optimization. The solution with respect to the best fitness value is saved and the solutions of the current population are sorted based on their fitness values.

4.2.3. *Construction of the new population.* To form a new population, the roulette wheel selection method is used to select individuals from the current population based on their relative fitness. After the selection, the single-point crossover operator and even mutation operator are used to generate new individuals in the new population, where the crossover probability P_c is 0.5 and the mutation probability P_m is 0.1.

4.2.4. *Termination.* The new population is formed after the above steps, and runs a new genetic cycle. When the best feasible solution has not changed for 500 consecutive generations, the procedure is terminated and the results are outputted.

5. **Numerical Example.** An example is used in this section to illustrate the approach presented in this paper. In this example, a repairable system with six parallel subsystems connected in series is considered. For the available components and repair teams in each subsystem, the inherent failure rate, repair rate and the unit cost are presented in Table 1. Assume four different types of redundant dependencies (independence, weak dependence, linear dependence and strong dependence) are considered in this repairable system.

Table 1. Data for each subsystem

Subsystem i	λ_i	μ_i	C_i^{c}	C_i^{r}
1	0.03	0.10	40	15
2	0.04	0.13	50	20
3	0.05	0.14	30	10
4	0.06	0.20	70	30
5	0.07	0.18	65	25
6	0.09	0.27	80	35

In this example, the design variable vector is as follows:

$$(\mathbf{n}, \mathbf{r}) = (n_1, n_2, n_3, n_4, n_5, n_6, r_1, r_2, r_3, r_4, r_5, r_6) \tag{20}$$

where $n_i, r_i \in Z^+$ ($r_i \leq n_i, i = 1, 2, \ldots, 6$), representing the number of components and the number of repair teams in subsystem i, respectively. The maximum number of components allowed in each subsystem is 15. Our objective is to minimize the construction cost of the system considered in this example while satisfying the system availability constraint. The optimization problem can be determined using (18), dependence function $g(j)$ can be determined through specific forms as indicated in Cases 1-3. Furthermore, we fix $l = 0.5$ (representing weak dependence) and $l = 1.5$ (representing strong dependence) in Case 3. This optimization problem is solved by the GA proposed in Section 4.

We would like to investigate allocation strategies with different types of redundant dependencies when different availability requirements are considered. First, we set the availability constraint value $A_0 = 0.90$, and investigate different allocation strategies with different types of redundant dependencies. The optimal allocation solutions for independence, weak dependence, linear dependence and strong dependence are presented in Table 2, in which the number of components n_i and number of repair teams r_i are presented for each subsystem i.

It can be seen from Table 2 that the optimal solutions are different when different types of redundant dependencies are considered. For the independence type (weak dependence type, linear dependence type and strong dependence type), the optimal system contains 19 (18, 16 and 15) components and 15 (12, 12 and 11) repair teams, and the system cost is 1355 (1285, 1125 and 1060). The results show that the strongest dependence type yields the most economic system and uses the least numbers of components and repair teams.

Then we raise the availability constraint value to 0.95 and 0.99, and investigate different allocation strategies with different types of redundant dependencies. The optimal design results are listed in Tables 3 and 4, respectively. One can see that with the increase of the availability constraint value, more components and repair teams should be used, and the system cost also increases with it.

From the above results (Tables 2-4), we can notice that for stronger dependence type, less numbers of components and repair teams will be required and more economic system will be constructed. Therefore, in order to construct a more economic system, we should reinforce the dependent characters in each subsystem with redundant dependency.

Table 2. Optimal allocation solutions when availability constraint value $A_0 = 0.90$

Subsystem i	Independence (n_i, r_i)	Weak dependence (n_i, r_i)	Linear dependence (n_i, r_i)	Strong dependence (n_i, r_i)
1	(3,3)	(3,1)	(3,2)	(3,1)
2	(3,2)	(3,2)	(3,2)	(2,2)
3	(4,3)	(3,2)	(3,2)	(3,2)
4	(3,2)	(3,2)	(2,2)	(2,2)
5	(3,3)	(3,3)	(3,2)	(3,2)
6	(3,2)	(3,2)	(2,2)	(2,2)
Cost	1355	1285	1125	1060
Availability	0.9025	0.9051	0.9020	0.9031

Table 3. Optimal allocation solutions when availability constraint value $A_0 = 0.95$

Subsystem i	Independence (n_i, r_i)	Weak dependence (n_i, r_i)	Linear dependence (n_i, r_i)	Strong dependence (n_i, r_i)
1	(4,3)	(3,3)	(3,2)	(3,2)
2	(3,3)	(3,2)	(3,3)	(3,2)
3	(4,3)	(4,2)	(3,3)	(3,2)
4	(4,3)	(3,3)	(3,1)	(3,1)
5	(4,3)	(3,3)	(3,2)	(3,1)
6	(3,3)	(3,3)	(3,2)	(3,1)
Cost	1615	1410	1275	1185
Availability	0.9502	0.9504	0.9514	0.9528

Table 4. Optimal allocation solutions when availability constraint value $A_0 = 0.99$

Subsystem i	Independence (n_i, r_i)	Weak dependence (n_i, r_i)	Linear dependence (n_i, r_i)	Strong dependence (n_i, r_i)
1	(5,3)	(4,4)	(4,2)	(3,3)
2	(5,4)	(4,3)	(4,2)	(3,2)
3	(5,4)	(5,3)	(4,2)	(4,2)
4	(5,3)	(4,4)	(4,2)	(3,2)
5	(5,4)	(4,4)	(4,3)	(4,3)
6	(5,3)	(4,2)	(3,3)	(3,2)
Cost	2135	1810	1590	1410
Availability	0.9904	0.9901	0.9910	0.9901

6. **Conclusions.** In this work, we investigate the steady-state availability of a repairable series-parallel system with redundant dependency, and develop an optimal design problem for the system. The redundant dependency is quantified by introducing the dependence function. The availability of the system is analyzed by the Markov model, and the optimal allocation problem is solved by the GA. The results of optimization for the repairable series-parallel system show that the different types of redundant dependencies can affect the optimal numbers of components and repair teams, and then make the allocation strategies different. In the presence of redundant dependencies, the strongest dependence level yields the most economic system and uses the least numbers of components and

repair teams. As redundant dependency exists in many systems composed of redundant components, this study provides an important way for constructing optimal systems.

In the repairable series-parallel system we consider here, repair teams are always available. However, in many real world repairable systems, some repair teams may become unavailable for a random period of time. This random period of repair team absence (we call it as repair team vacation) can represent the time when the repair team is performing some secondary task. In future work, we will concern the development of availability and optimal design for repairable series-parallel system with failure dependency and repair teams vacation.

Acknowledgment. The research work is partially supported by the National Science Foundation of China (No. 71071133), the Natural Science Foundation of Hebei Province (No. G2012203136) and the Science Research Project of Education Department of Hebei Province (No. Z2011107), People's Republic of China.

REFERENCES

[1] S. Wang and J. Watada, Reliability optimization of a series-parallel system with fuzzy random lifetimes, *International Journal of Innovative Computing, Information and Control*, vol.5, no.6, pp.1547-1558, 2009.
[2] Y.-S. Juang, S.-S. Lin and H.-P. Kao, A knowledge management system for series-parallel availability optimization and design, *Expert Systems with Applications*, vol.34, pp.181-193, 2008.
[3] J. E. Ramirez-Marquez and D. W. Coit, A heuristic for solving the redundancy allocation problem for multi-state series-parallel systems, *Reliability Engineering and System Safety*, vol.83, pp.341-349, 2004.
[4] W. Kuo and V. R. Prasad, An annotated overiew of system-reliability optimization, *IEEE Trans. on Reliability*, vol.49, pp.176-187, 2000.
[5] M. Gen and Y. S. Yun, Soft computing approach for reliability optimization: State-of-the-art survey, *Reliability Engineering and System Safety*, vol.91, pp.1008-1026, 2006.
[6] G. Levitin and A. Lisnianski, Joint redundancy and maintenance optimization for multistate series-parallel systems, *Reliability Engineering and System Safety*, vol.64, pp.33-42, 1999.
[7] C. Elegbede and K. Adjallah, Availability allocation to repairable systems with genetic algorithms: A multi-objective formulation, *Reliability Engineering and System Safety*, vol.82, pp.319-330, 2003.
[8] H. F. Castro and K. L. Cavalca, Maintenance resources optimization applied to a manufacturing system, *Reliability Engineering and System Safety*, vol.91, pp.413-420, 2006.
[9] M. Nourelfath and D. Ait-Kadi, Optimization of series-parallel multi-state systems under maintenance policies, *Reliability Engineering and System Safety*, vol.92, pp.1620-1626, 2007.
[10] H. Y. Yu, C. B. Chu, E. Châtelet and F. Yalaoui, Reliability optimization of a redundant system with failure dependencies, *Reliability Engineering and System Safety*, vol.92, pp.1627-1634, 2007.
[11] M. Pecht, *Product Reliability, Maintainability and Supportability Handbook*, CRC Press, FL, 1995.
[12] J. E. Ramirez-Marquez and D. W. Coit, Optimization of system reliability in the presence of common cause failures, *Reliability Engineering and System Safety*, vol.92, pp.1421-1434, 2007.
[13] C. Y. Li, X. Chen, X. S. Yi and J. Y. Tao, Heterogeneous redundancy optimization for multi-state series-parallel systems subject to common cause failures, *Reliability Engineering and System Safety*, vol.95, pp.202-207, 2010.
[14] G. Levitin and A. Lisnianski, Structure optimization of multi-state with two failure modes, *Reliability Engineering and System Safety*, vol.72, pp.75-89, 2001.
[15] G. Levitin, Optimal series-parallel topology of multi-state system with two failure modes, *Reliability Engineering and System Safety*, vol.77, pp.93-107, 2002.
[16] C. E. Ebeling, *An Introduction to Reliability and Maintainability Engineering*, McGRAW-HILL, 1997.
[17] A. Barros, C. Bérenguer and A. Grall, Optimization of replacement times using imperfect monitoring information, *IEEE Trans. on Reliability*, vol.52, pp.523-533, 2003.
[18] M. S. Chern, On the computational complexity of reliability redundancy allocation in a series system, *Operations Research Letters*, vol.11, pp.309-315, 1992.

[19] Z. G. Tian, G. Levitin and M. J. Zuo, A joint reliability-redundancy optimization approach for multi-state series-parallel systems, *Reliability Engineering and System Safety*, vol.94, pp.1568-1576, 2009.
[20] G. Levitin and S. V. Amari, Optimal load distribution in series-parallel systems, *Reliability Engineering and System Safety*, vol.94, pp.254-260, 2009.
[21] P. X. Liu, M. J. Zuo and M. Meng, Using neural network function approximation for optimal design of continuous-state parallel-series systems, *Computers & Operations Research*, vol.30, pp.339-352, 2003.
[22] J. H. Holland, *Adaptation in Natural and Artificial System: An Introduction with Application to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, 1992.
[23] G. Levitin and L. Meizin, Structure optimization for continuous production systems with buffers under reliability constraints, *Int. J. Production Economics*, vol.70, pp.77-87, 2001.
[24] M. Gen and R. Cheng, *Genetic Algorithm and Engineering Design*, Wiley, New York, 1997.
[25] T. Bäck, *Evolutionnary Algorithms in Theory and Practice*, Oxford University Press, 1996.
[26] M. Schoenauer and Z. Michalewicz, Evolutionary computation, *Control Cybem*, vol.26, pp.307-338, 1997.