Reliability-Redundancy Allocation for Multi-State Series-Parallel Systems

Zhigang Tian, Ming J. Zuo, *Senior Member, IEEE*, and Hongzhong Huang

***Abstract*—Current studies of the optimal design of multi-state series-parallel systems often focus on the problem of determining the optimal redundancy for each stage. However, this is only a partial optimization. There are two options to improve the system utility of a multi-state series-parallel system: 1) to provide redundancy at each stage, and 2) to improve the component state distribution, that is, make a component in states with respect to higher utilities with higher probabilities. This paper presents an optimization model for a multi-state series-parallel system to jointly determine the optimal component state distribution, and optimal redundancy for each stage. The relationship between component state distribution, and component cost is discussed based on an assumption on the treatment on the components. An example is used to illustrate the optimization model with its solution approach, and that the proposed reliability-redundancy allocation model is superior to the current redundancy allocation models.**

***Index Terms*—Multi-state series-parallel system, optimization, reliability-redundancy allocation, state distribution.**

Notation

M	The maximum state level of the components, and the system
N	Number of subsystems (stage)
S_i	Subsystem (stage) i, $1 \le i \le N$
n_i	Redundancy in S_i
p_{ij}	Probability of component i in state j
$\boldsymbol{x}$	A vector representing the states of all components in the multi-state system
$\phi(\boldsymbol{x})$	State of the system, $\phi(\boldsymbol{x}) = 0, 1, \ldots, M$
U	System utility
C	System cost
u_s	The utility when system is in state s
r_{ik}	Defined term, $r_{ik} = p_{ik}/\sum_{l=0}^{k} p_{il}$, $k = 1, 2, \ldots, M$
c_i	Cost function of component in S_i
α_{ik}, β_{ik}	Characteristic constants with respect to state k in S_i
t	The mission time
f	Aggregate objective function in physical programming-based model
$\overline{g}_U$	Class function of system utility
$\overline{g}_C$	Class function of system cost
U_0	System utility constraint value
C_0	System cost constraint value
g_{ij}	Boundary value of preference ranges for objective i. $j = 1, 2, \ldots, 5$
$\boldsymbol{X}$	Design variable vector
h_i	Component version for stage i

Manuscript received September 24, 2004; revised May 11, 2007; accepted January 23, 2008. Associate Editor: E. Pohl.

Z. Tian is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 2W1, Canada (e-mail: tian@ciise.concordia.ca).

M. J. Zuo is with the Department of Mechanical Engineering, University of Alberta, Edmonton, AB T6G 2G8, Canada (e-mail: ming.zuo@ualberta.ca).

H. Huang is with the School of Mechanical, Electronic, and Industrial Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: hzhuang@uestc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TR.2008.920871

Acronym[1]

DM	Decision Maker

Assumptions

1) The states of the components in a subsystem are i.i.d.
2) The components, and the system may be in $M + 1$ possible states, namely, $0, 1, 2, \ldots, M$.
3) The multi-state series parallel systems under consideration are coherent systems.

I. Introduction

In binary-state system design, there are basically two options to improve the reliability of the system: to increase the component reliabilities, or to provide redundancy at various stages [5]. The term "component" here refers to an entity, with specified reliability, cost, etc., which can be connected in a certain configuration to form a subsystem, or a system. The reliability-redundancy allocation problem was first introduced by Misra & Ljubojevic [10]. They considered the problem of simultaneously determining optimal component reliabilities, and optimal redundancy levels for a series-parallel system subject to a cost constraint. The component cost was assumed to be an exponential function of component reliability at each stage. Tillman *et al.* [17] assumed a different cost-reliability relationship, and besides the cost constraint, they included weight, and volume constraints in their optimization model. Mathematically, the reliability-redundancy allocation problem is a mixed integer nonlinear programming problem. Misra & Ljubojevic [10], and Tillman *et al.* [17] used heuristic approaches to solve it. Other solution approaches include the branch-and-bound technique [4], XKL method [18], surrogate constraints method [3], and evolutionary algorithms [11].

[1]The singular and plural of an acronym are always spelled the same.

Many practical technical systems can perform their intended functions at more than two different levels, ranging from perfectly working to completely failed. These kinds of systems are called multi-state systems [5]. A multi-state system reliability model provides more flexibility for modeling equipment conditions than a binary system reliability model. System utility is a commonly used performance measure for multi-state systems [1]. In the case of a multi-state system, the concept corresponding to the concept of reliability in a binary system is the state distribution. The state distribution of a component refers to the distribution of the component in different states. Similarly, there can also be two options to improve the system utility of a multi-state series-parallel system: 1) to provide redundancy at each stage, and 2) to improve the component state distribution, that is, make a component in states with respect to higher utilities with higher probabilities.

Current studies on the optimal design of multi-state series-parallel systems mainly focus on the problem of determining the optimal redundancy for each stage. Levitin *et al.* [7] assumed that there were different versions of components for the selection for each stage. And they proposed a redundancy optimization model for multi-state series-parallel systems to determine the optimal component versions, and redundancies for various stages. The problem was later extended to including both redundancy, and maintenance optimization [6]. Ramirez-Marquez & Coit [12] proposed a heuristic approach for solving the redundancy allocation problem formulated in Levitin *et al.* [7]. Liu *et al.* [8] presented a neural network approximation approach for redundancy optimization of continuous-state series-parallel systems. Their model only had component redundancies as design variables. Tian & Zuo [15] applied the physical programming approach to the redundancy allocation of multi-state series-parallel systems, and demonstrated that it is more effective in dealing with the multiple conflicting design objectives involved in the problem.

The multi-state system structure optimizations reviewed above for multi-state systems are only partial optimization because only redundancies are regarded as design variables. Component state distributions should also be considered to be design variables. The option of selecting different versions of components provides more flexibility than just using redundancy. But this option totally depends on the products available on the market, and the amounts of available versions are always limited. By determining the optimal values of component state distributions, we can have our design optimization built into the manufacturing or scheduling process, so as to have more flexibility, and get better optimization results.

This paper presents an optimization model for multi-state series-parallel systems to jointly determine the optimal component state distribution, and optimal redundancy for each stage. We present the reason why a component state distribution can be used as a controllable design variable. The relationship between component state distribution and component cost is discussed based on an assumption on the treatments on the component. The physical programming-based optimization model is presented. An example is used to illustrate the optimization model, and its solution approach. A short version of this paper was presented in the *2005 European Safety & Reliability Conference (ESREL05)*, and has been published in the conference proceedings [16].

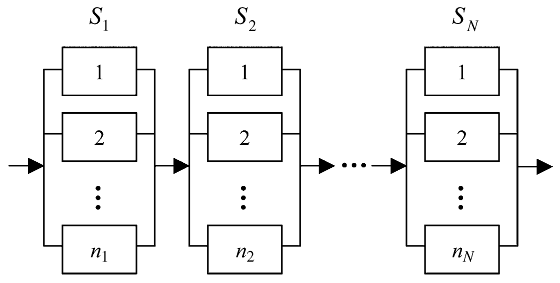

Fig. 1. The structure of a multi-state series-parallel system.

II. Problem Formulation

The structure of a multi-state series-parallel system is given in Fig. 1. A multi-state series-parallel system has N subsystems connected in series, and each subsystem i has n_i i.i.d. components connected in parallel. The probability of component i in state j is p_{ij}.

A. Design Variables

The design variables in the reliability-redundancy allocation problem for multi-state series-parallel systems are component state distributions p_{ij} ($i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$), and redundancies n_i ($i = 1, 2, \ldots, N$). It is apparent that redundancy is controllable, so that it can be used as design variable. We will justify in this part that component state distribution is also a controllable design variable.

In the binary-state case, the reliability of a component is its probability of working, and it has been used as a design variable in reliability-redundancy allocation problems of binary-state systems. In the case of multi-state systems, let us consider a three-state system where the components and system have three states {0, 1, 2}. We have the following two statements: 1) The probability that a component is in state 1 or 2 can be regarded as the reliability of this component if the word "working" means that the component's state is greater or equal to 1. 2) Similarly, the probability of a component in state 2 can be regarded as the reliability of this component if the word "working" means that the component's state is greater than or equal to 2. Therefore, just like the binary case, the state distribution of each component can be used as a design variable.

If we want to get a component with state distribution $\{p_0, p_1, p_2\}$, we can first ensure the component's reliability is p_2 under the reliability meaning of statement 2), and then ensure the component's reliability is $p_1 + p_2$ under the reliability meaning of statement 1).

Let's investigate an example of how the component state distribution is controlled. Suppose that there is a three-state component, and two treatments can be used to influence the state distribution of the component: Treatment 1 will increase the probability of the component in state 1, but will not influence the probability of the component in state 2; while Treatment 2 will increase the probability of the component in state 2. Therefore, for this three-state component, using the two treatments on the component, we can control the state distribution of the component. That is, the state distribution of this component can be regarded as a controllable design variable.

B. System Utility Evaluation

System utility is one of the most popular performance measures for multi-state systems [1]. There is a utility value related to each possible system state, and system utility is the expected utility of the multi-state system. The probability that the system state of a multi-state series-parallel system is in state s $(s = 0, 1, \ldots, M)$ or above is

$$\Pr[\phi(\boldsymbol{x}) \geq s] = \prod_{i=1}^{N}\left[1 - \left(1 - \sum_{k=s}^{M} p_{ik}\right)^{n_i}\right]. \quad (1)$$

The system utility is

$$U = \sum_{s=0}^{M} u_s \cdot \Pr[\phi(\boldsymbol{x}) = s] \quad (2)$$

where U is the system utility, and u_s is the utility when the system is in state s.

C. Formulation of System Cost

In the binary-state reliability-redundancy allocation problem, Misra & Ljubojevic [10] assumed that there is an exponential relationship between component cost, and component reliability. Tillman *et al.* [17] assumed another cost-reliability relationship, which is smoother than the one proposed by Misra & Ljubojevic [10].

The cost of subsystem i with n_i parallel components is given by Tillman *et al.* [17] as

$$C_i = c_i(r_i)\left[n_i + \exp\left(\frac{n_i}{4}\right)\right], \text{ where, } c_i(r_i) = \alpha_i\left(\frac{-t}{\ln r_i}\right)^{\beta_i} \quad (3)$$

and $c_i(r_i)$ is the cost-reliability relationship function for a component in subsystem i. $c_i(r_i)n_i$ is the cost of the components in subsystem i. The additional cost $c_i(r_i)\exp(n_i/4)$ is due to interconnecting the parallel components within the subsystem. The component cost function $c_i(r_i)$ is assumed to take the form in (3), based on a general understanding that the cost will increase exponentially with the increase of the reliability. Characteristic constants α_i and β_i are to be determined based on the collected cost, and reliability relationship data of the component.

Now let's investigate the possible cost formulation in the case of a multi-state system, where the system and components may be in $(M + 1)$ states. The state distribution of a component in subsystem i is denoted by $\{p_{i0}, p_{i1}, \ldots, p_{iM}\}$. The cost formulation of the component is based on the following assumption:

Assumption: There are M treatments that can influence the component's state distribution, and treatment k will increase the probability of the component in state k, but will not influence the probability of the component in the states above k.

Let

$$r_{ik} = \frac{p_{ik}}{\sum_{l=0}^{k} p_{il}}, \; k = 1, 2, \ldots, M. \quad (4)$$

Here, r_{ik} can be considered to be the binary-state reliability of component i under treatment k. We define the cost of the component as

$$c_i(p_{i1}, p_{i2}, \ldots, p_{iM}) = \sum_{k=1}^{M} \alpha_{ik}\left(\frac{-t}{\ln r_{ik}}\right)^{\beta_{ik}} \quad (5)$$

where α_{ik}, and β_{ik} are characteristic constants with respect to state k, $0 < \alpha_{i1} < \alpha_{i2} < \ldots < \alpha_{iM}$, $1 \leq \beta_{i1} < \beta_{i2} < \ldots < \beta_{iM}$, and t is the mission time. This cost function shown in (5) is inspired by the cost function for reliability-redundancy allocation of binary series-parallel systems, as shown in (3). Equation (5) is used to determine the cost of a component, say component i, based on its state distribution $\{p_{i0}, p_{i1}, \ldots, p_{iM}\}$.

The system cost is

$$C = \sum_{i=1}^{N} c_i(p_{i1}, p_{i2}, \ldots, p_{iM})\left[n_i + \exp(n_i/4)\right] \quad (6)$$

For each subsystem, the additional cost $c_i(p_{i1}, p_{i2}, \ldots, p_{iM})\exp(n_i/4)$ is included to represent the cost for interconnecting parallel components.

D. Characteristics of the Optimization Problem

Two objectives, system utility and system cost, are included in our optimization model. Component state distributions p_{ij} $(i = 1, 2, \ldots, N, j = 1, 2, \ldots, M)$, and redundancies n_i $(i = 1, 2, \ldots, N)$ are to be determined so as to maximize system utility, and minimize system cost. Note that $p_{i0} = 1 - \sum_{k=1}^{M} p_{ik}$. This problem is formulated mathematically as a mixed integer optimization problem in which the continuous variables represent the component state distributions, and the integer variables represent the redundancies.

The reliability-redundancy allocation problem for multi-state series-parallel systems may be formulated as a single-objective optimization problem, with system cost as a design objective, and system utility as a constraint; or with system utility as a design objective, and system cost as a constraint. If system cost is used as a design objective, like the models given by Levitin *et al.* [7], and that of Ramirez-Marquez & Coit [12], the optimization model is formulated as follows:

$$\begin{aligned}
&\min\ C \\
&\text{s.t.} \\
&\quad U \geq U_0; C \leq C_0 \\
&\quad 0 \leq p_{ij} \leq 1, \quad i = 1, 2, \ldots, N, j = 1, 2, \ldots, M \\
&\quad \sum_{j=1}^{M} p_{ij} \leq 1, \quad i = 1, 2, \ldots, N \\
&\quad 0 < n_i, \quad i = 1, 2, \ldots, N \\
&\quad n_i \text{ are integers}
\end{aligned} \quad (7)$$

Where U_0, and C_0 are the constraint values.

The reliability-redundancy allocation problem for multi-state series-parallel systems can also be formulated as a multi-objective optimization problem, because there are two design objectives, system utility and system cost. In this work, we will present a multi-objective optimization model for the joint reliability-redundancy allocation problem. Available multi-objective optimization approaches include the surrogate worth trade-off method [14], the fuzzy optimization method [13], and physical programming [9], [15], among others. Compared to other multi-objective optimization approaches, physical programming provides a better way to specify designers' preferences on different objectives, and it is easier to use in practical problems. Tian & Zuo [15] applied the physical programming approach to the redundancy allocation problem rather than the joint reliability-redundancy optimization problem. In this paper, the physical programming approach is used to model & solve this reliability-redundancy optimization problem with two design objectives: system utility, and system cost.

E. Physical Programming-Based Optimization Problem Formulation

Physical Programming, proposed by Messac [9], is a multi-objective optimization tool that explicitly incorporates the Decision Maker's (DM's) preferences on each design goal into the optimization process. It eliminates the typical iterative process involving the adjustment of the physically meaningless weights, which is required by virtually all other multi-objective optimization methods, and thus substantially reduces the computational intensities. The DMs' preferences are specified individually on each goal through physically meaningful values, which makes the physical programming method easy to use, and advantageous in dealing with a large number of objectives.

Physical programming captures the designer's preferences using class functions. A class function is a function of a design objective. The value of a class function represents the preference of the designer on the objective function value; and the smaller the class function value is, the better. Class functions are classified into four classes: smaller is better (i.e., minimization), larger is better (i.e., maximization), value is better, and range is better. There are two so-called class functions, one soft and one hard, with respect to each class. Soft class functions will become constituent parts of a single objective function, the so-called aggregate objective function, which is to be minimized in the physical programming optimization model [9]. There are four types of soft class functions, Class-1S to Class-4S, with respect to the four classes of class functions. Class-1S is a monotonously increasing function, and it is used to represent the objectives to be minimized. Class-2S is a monotonously decreasing function, and it is used to represent the objectives to be maximized. The Class-1S function is shown in Fig. 2. The value of the design objective, g_i, is on the horizontal axis; and the corresponding class function value, $\overline{g}_i$, is on the vertical axis. In the reliability-redundancy allocation problem, the utility objective has a Class-2S class function, and the cost objective has a Class-1S class function.

What the designer needs to do is to specify ranges of different degrees of desirability (highly desirable, desirable, tolerable, undesirable, highly undesirable, and unacceptable) for the class function of each objective. For example, the DM can specify ranges of degrees of desirability for the cost of a system, which is to be minimized, as follows: 1) the cost is considered to be unacceptable if it is over \$200, 2) the cost is considered to be highly undesirable if it is between \$150 and \$200, 3) the cost is considered to be undesirable if it is between \$100 and \$150, 4) the cost is considered to be tolerable if it is between \$80 and \$100, 5) the cost is considered to be desirable if it is between \$50 and \$80, and 6) the cost is considered to be highly desirable if it is below \$50. Such ranges are specified by the DM based on experience, and design purpose.

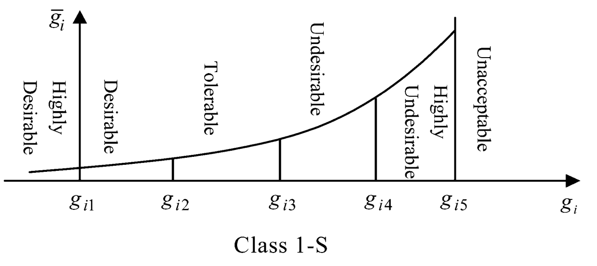

Fig. 2. Class-1S class function.

The physical programming approach solves a multi-objective optimization problem by transforming it into a single-objective optimization problem. The soft class functions of design objectives are combined into the aggregate objective function f, which is to be minimized. The physical programming-based optimization model for the multi-state reliability allocation problem is formulated as

$$\begin{aligned}
&\min f = \log_{10}\left\{\frac{1}{2}\left[\overline{g}_U(U) + \overline{g}_C(C)\right]\right\}\\
&\text{s.t.}\\
&\quad U \geq U_0\\
&\quad C \leq C_0\\
&\quad 0 \leq p_{ij} \leq 1, \quad i = 1,2,\ldots,N, j = 1,2,\ldots,M\\
&\quad \sum_{j=1}^{M} p_{ij} \leq 1, \quad i = 1,2,\ldots,N\\
&\quad 0 < n_i, \qquad i = 1,2,\ldots,N\\
&\quad n_i \text{ are integers}
\end{aligned} \tag{8}$$

where $\overline{g}_U$, and $\overline{g}_C$ are the class functions of system utility, and system cost; and they are functions of system utility U, and system cost C, which can be calculated using the formula presented in Sections II-B, and II-C. $\overline{g}_U$ is a Class-2S class function, and $\overline{g}_C$ is a Class-1S class function. U_0, and C_0 are the constraint values, and they are equal to the boundaries of the acceptable ranges of the corresponding objectives. Refer to Messac for more detailed descriptions on the physical programming approach [9].

F. Genetic Algorithm as the Optimization Solution Method

The physical programming-based model presented in (8) is a mixed integer single-objective optimization model. Algorithms such as branch-and-bound, generalized Benders decomposition, and outer approximation [2] have been used to solve this kind of problem. However, the most effective algorithm to solve mixed integer optimization problems, we believe, is the genetic algorithm (GA). The encoding method of GA enables it to directly represent continuous design variables, and discrete design variables as well, which makes the solution process much simpler.

TABLE I
SYSTEM UTILITY WITH RESPECT TO EACH SYSTEM STATE

s	0	1	2
u_s	0.0	0.5	1.0

In addition, GA has a very good global optimization capability. Therefore, GA is used to solve the physical programming-based model in this paper.

Each solution is called a chromosome in GA, and the group of chromosomes in each iteration of GA is called a population. The procedure of GA is as follows [19]:

1) Initialization. Choose the encoding method. Set the size of population, and the length of the chromosome. Specify the GA operators including the selection operator, the crossover operator, and the mutation operator. Also, specify the GA parameters mainly including the crossover rate, and the mutation rate. Set $k = 0$, and generate the initial population $P(0)$.
2) Evaluation. Calculate the fitness value for each chromosome of the current population $P(k)$. Save the chromosome $B(k)$ with the best fitness value.
3) Construction of the new population. Select chromosomes from the current population based on their fitness values to form a new population $P(k+1)$. Implement the crossover operator, and the mutation operator to generate new chromosomes in the new population $P(k+1)$. Use $B(k)$ to replace the first chromosome in $P(k+1)$ so as to keep the best chromosome in the previous iteration.
4) If the maximal iteration is reached, terminate the procedure, and output the result. Otherwise, set $k = k + 1$, and go to step 2).

III. AN EXAMPLE

A 3-stage multi-state series-parallel system is used to illustrate the basic ideas of the proposed reliability-redundancy allocation approach. The three stages (or subsystems) are connected in series, and each subsystem i has n_i i.i.d. components connected in parallel. The system is a three-state system, where the system and components can be in three states: 0, 1 and 2. The system utility u_s with respect to the corresponding system state s is shown in Table I.

A. The Joint Reliability-Redundancy Optimization Results

The cost of the component is assumed to follow the relationship given in (5). The characteristic constants used in this example are presented in Table II. The mission time is set to be $t = 1000$.

In the physical programming framework, the utility objective has a Class-2S class function (larger is better), while the cost objective has a Class-1S class functions (smaller is better). The class function settings for the three objectives are shown in Table III, where g_{i1} to g_{i5} represent the boundaries of different desirable ranges for the utility, and cost objectives [9]. The constraint values U_0, and C_0 are equal to the boundary values g_{i5}

TABLE II
CHARACTERISTIC CONSTANTS FOR THE SYSTEM

Stage i	α_{i1}	α_{i2}	β_{i1}	β_{i2}
1	1.5×10^{-5}	4.0×10^{-5}		
2	0.9×10^{-5}	3.2×10^{-5}	1.2	1.5
3	5.2×10^{-5}	9.0×10^{-5}		

TABLE III
PHYSICAL PROGRAMMING CLASS FUNCTIONS SETTING

	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}
Utility	0.99	0.97	0.92	0.85	0.75
Cost	30	50	100	150	200

of the corresponding objectives. There are 9 design variables in this problem:

$$\boldsymbol{X} = \{p_{11}, p_{12}, p_{21}, p_{22}, p_{31}, p_{32}, n_1, n_2, n_3\} \tag{9}$$

GA is used to solve the formulated single-objective nonlinear optimization model shown in (8). In this problem, the population size is chosen to be 100. The decimal encoding is used, and the chromosome length is set to be 15. We use the roulette-wheel selection scheme, one-point cross operator with crossover rate of 0.25, and even mutation operator with mutation rate of 0.1. When GA reaches the maximum epoch, which is set to 1000, the algorithm will be terminated.

Because GA is a stochastic optimization approach, we run the optimization procedure 30 times to get the overall performance. The aggregate objective function f, which is to be minimized, indicates the quality of the optimization result. Among the 30 optimization results we obtained, the best result is $f = -0.1649$, the worst result is $f = -0.1559$, and the average result is $f = -0.1617$. The optimal design variables, and objective function values with respect to the best result are given in Table IV. The optimal system utility value is 0.9728, which falls into the highly desirable range. The optimal system cost value is 88.4083, which falls into the desirable range. Such a result is satisfactory regarding the DM's preferences on these two objectives.

B. The Redundancy Optimization Results

If there are only limited versions of components with specific state distributions for each stage, the result we get will not be optimal. Suppose that there are four different versions of components available for each stage, as shown in Table V, where h_i denotes the component version for stage i. We can select components from the available versions for each stage, and determine the optimal redundancy values. We use the same characteristic constants setting, and the same approach to evaluate the system cost, as those in the reliability-redundancy allocation problem above. The physical programming class functions setting, and

TABLE IV
OPTIMIZATION RESULTS FOR THE RELIABILITY-REDUNDANCY ALLOCATION PROBLEM

Stage	State distribution		Redundancy	System	System	f
i	p_{i1}	p_{i2}	n_i	utility	cost	
1	0.2106	0.4600	7			
2	0.2226	0.4700	7	0.9728	88.4083	-0.1647
3	0.2040	0.4000	7			

TABLE V
CHARACTERISTICS OF AVAILABLE COMPONENTS

Stage i	h_i	p_{i1}	p_{i2}
1	1	0.30	0.52
	2	0.25	0.35
	3	0.12	0.45
	4	0.10	0.60
2	1	0.21	0.64
	2	0.25	0.33
	3	0.30	0.48
	4	0.14	0.40
3	1	0.28	0.40
	2	0.21	0.50
	3	0.15	0.55
	4	0.30	0.15

the system utility with respect to each system state, are also set to be the same.

The redundancy optimization problem is an integer programming problem because the design variables, the component versions, and the number of components in each stage, can only take integer values. This optimization problem is simpler than the joint reliability-redundancy optimization problem, which is a mixed integer programming problem. Here we also run the GA optimization procedure 30 times to get the overall performance. Due to the simplicity of this redundancy optimization problem, the same optimization result is generated in each of the 30 runs of GA. The optimization result is shown in Table VI, and the optimal aggregate objective function value is -0.1533. Compared to the results obtained by reliability-redundancy allocation optimization in Table IV, we can find that the optimized system utility, and system cost are both a little bit worse. The aggregate objective function value is larger, which is caused by the additional constraints. The state distributions of the selected components will not happen to be the same as the optimal state distribution obtained in the joint optimization of state distributions and redundancies presented above.

C. Sensitivity Analysis for System Cost, and System Utility

In this section, we will perform sensitivity analysis for system cost, and system utility with respect to different design variables, and model parameters. First we investigate the sensitivity of system cost. In the reliability-redundancy optimization of a three-stage system considered in this example, there are 9 design variables representing component state distributions and redundancies, as shown in (9). Component characteristic constants, α_{i1}, α_{i2}, β_1, and β_2 $(i = 1, 2, 3)$, are important model parameters that affect the system cost. In the sensitivity analysis for system cost, at the optimal design variable vector as shown in Table IV, we investigate the partial derivative of system cost C with respect to a design variable or a model parameter while keeping other design variables and model parameters constant. For instance, when studying $\partial C/\partial p_{11}$, the partial derivative of system cost C with respect to p_{11}, we keep all the other design variables, p_{12}, p_{21}, p_{22}, p_{31}, p_{32}, n_1, n_2, and n_3 as constants, as shown in Table IV; and all model parameters α_{ij}, and β_j $(i = 1, 2, 3,\ j = 1, 2)$ as constants as shown in Table II; and investigate the changes in C with respect to the changes in p_{11}.

The result of sensitivity analysis for system cost with respect to the 6 component state distribution design variables is shown in Fig. 3. The component state distribution value is on the horizontal axis. On the vertical axis, we have the partial derivative value. From Fig. 3, we can observe that in these cases, for any p_{ij}, $\partial C/\partial p_{ij}$ is always positive, which means system cost increases with the increase in p_{ij}, while other design variables are kept constant. Again for any p_{ij}, the sensitivity of system cost decreases a bit with the increase of p_{ij} when it is relatively small (less than 0.05); and after that, $\partial C/\partial p_{ij}$ always increases with the increase of p_{ij}. From Fig. 3, we can also observe that the system cost is more sensitive to the state distribution variables associated with stage 3, i.e. p_{31}, and p_{32}, because component cost characteristic constants for stage 3 are larger than those for stage 1 and 2.

The sensitivity of system cost with respect to model parameter α_{ij}, $\partial C/\partial \alpha_{ij}$ $(i = 1, 2, 3,\ j = 1, 2)$ is easy to derive analytically. Based on (5) & (6), we can get

$$\partial C/\partial \alpha_{ij} = \left(\frac{-t}{\ln r_{ij}}\right)^{\beta_{ik}} \left[n_i + \exp(n_i/4)\right]. \tag{10}$$

$\partial C/\partial \alpha_{ij}$ shown (10) is a constant value when all the design variables and other model parameters are kept as constant, except for α_{ij} itself. $\partial C/\partial \alpha_{ij}$ is also always positive, meaning the system cost increases with the increase of α_{ij}.

The sensitivity of system cost with respect to model parameter β_j, $\partial C/\partial \beta_j$ $(j = 1, 2)$ can be obtained based on (5) & (6) as well:

$$\partial C/\partial \beta_j = \sum_{i=1}^{3} \left[n_i + \exp(n_i/4)\right] \alpha_{ij} \left(\frac{-t}{\ln r_{ij}}\right)^{\beta_{ik}} \ln\left(\frac{-t}{\ln r_{ij}}\right) \tag{11}$$

$\partial C/\partial \beta_j$ shown in (11) is also always positive, meaning the system cost increases with the increase of β_j. The system cost becomes more sensitive to the β_j with the increase of β_j.

TABLE VI
OPTIMIZATION RESULTS FOR THE REDUNDANCY ALLOCATION PROBLEM

Stage i	H	State distribution p_{i1}	p_{i2}	Redundancy n_i	System utility	System cost	f
1	3	0.1200	0.4500	7			
2	3	0.3000	0.4800	7	0.9721	89.5769	-0.1533
3	1	0.2800	0.4000	7			

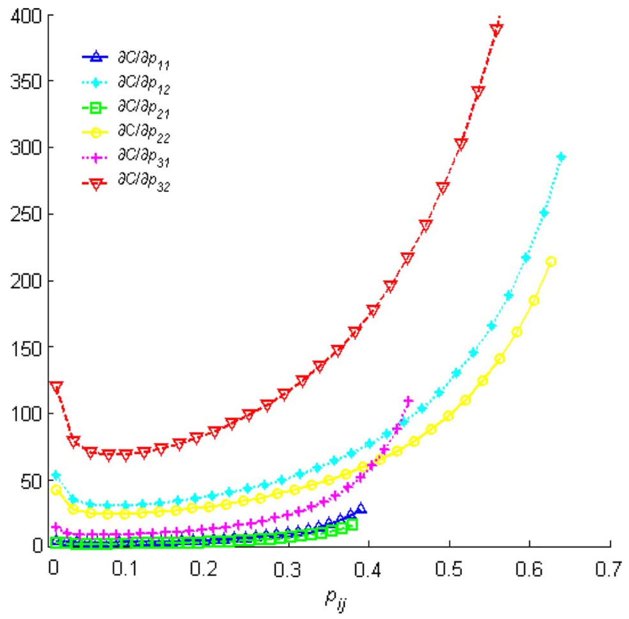

Fig. 3. Sensitivity analysis for system cost with respect to component state distributions.

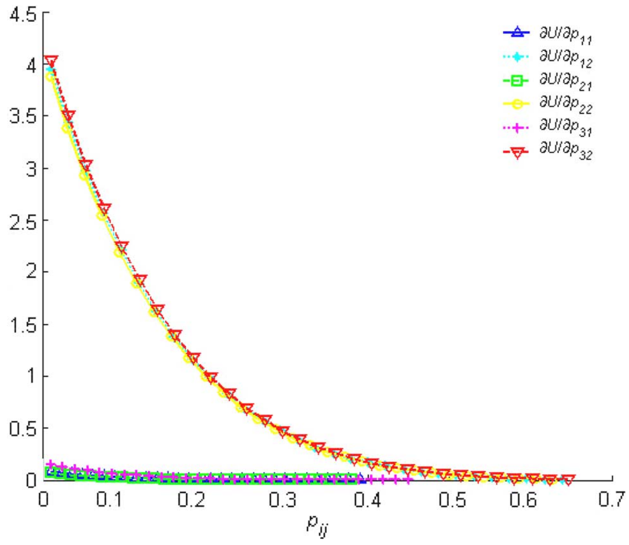

Fig. 4. Sensitivity analysis for system utility with respect to component state distributions.

Finally, we investigate the sensitivity of system utility, as shown in (1) & (2), with respect to the 6 component state distribution design variables, that is, $\partial U/\partial p_{ij}$ $(i = 1, 2, 3, j = 1, 2)$. The sensitivity analysis results are shown in Fig. 4. As we can observe, for any p_{ij}, $\partial U/\partial p_{ij}$ is always positive, meaning the system utility increases with the increase of p_{ij}. However, unlike the case of system cost, system utility becomes less sensitive to p_{ij} with the increase of p_{ij}. Another thing we observe is that system utility is more sensitive to state distribution variables associated with state 2, p_{12}, p_{22}, and p_{32}; than it is to state distribution variables associated with state 1, p_{11}, p_{21}, and p_{31}.

IV. CONCLUSIONS

There can be two options to improve the system utility of a multi-state series-parallel system: 1) to provide redundancy at each stage, and 2) to improve the component state distributions. This paper presents an optimization model for multi-state series-parallel systems to jointly determine the optimal component state distribution, and the optimal redundancy for each stage. The reason why component state distribution can be used as a controllable design variable is presented. The relationship between component state distribution, and component cost is discussed based on an assumption on the treatments on the component. The physical programming-based optimization model is presented. The example illustrates the optimization model, its solution approach, and the advantages of the proposed reliability-redundancy allocation model over the existing redundancy allocation methods. Sensitivity analysis shows the impact of different design variables, and model parameters on system cost, and system utility.

REFERENCES

[1] T. Aven, "On performance-measures for multistate monotone systems," *Reliability Engineering and System Safety*, vol. 41, no. 3, pp. 259–266, 1993.

[2] C. Floudas*, Nonlinear and Mixed-Integer Optimization: Fundamental and Applications*. New York: Oxford University Press, 1995.

[3] M. Hikita, Y. Nakagawa, K. Nakashima, and H. Narihisa, "Reliability optimization of systems by a surrogate-constraints algorithm," *IEEE Trans. Reliability*, vol. 41, no. 3, pp. 473–480, 1992.

[4] W. Kuo, H. Lin, Z. Xu, and W. Zhang, "Reliability optimization with the Lagrange multiplier and branch-and-bound technique," *IEEE Trans. Reliability*, vol. 36, no. 5, pp. 624–630, 1987.

[5] W. Kuo and M. J. Zuo*, Optimal Reliability Modeling: Principles and Applications*. Hoboken: John Wiley & Sons, Inc., 2003.

[6] G. Levitin and A. Lisnianski, "Joint redundancy and maintenance optimization for multistate series-parallel systems," *Reliability Engineering and System Safety*, vol. 64, no. 1, pp. 33–42, 1999.

[7] G. Levitin, A. Lisnianski, and H. Ben-Haim, "Redundancy optimization for series-parallel multi state systems," *IEEE Trans. Reliability*, vol. 47, no. 2, pp. 165–172, 1998.

[8] P. X. Liu, M. J. Zuo, and M. Q. H. Meng, "Using neural network function approximation for optimal design of continuous-state parallel-series systems," *Computers & Operations Research*, vol. 30, no. 3, pp. 339–352, 2003.

[9] A. Messac, "Physical programming: Effective optimization for computational design," *AIAA Journal*, vol. 34, no. 1, pp. 149–158, 1996.

[10] K. B. Misra and M. D. Ljubojevic, "Optimal reliability design of a system: A new look," *IEEE Trans. Reliability*, vol. 22, no. 5, pp. 255–258, 1973.

[11] V. R. Prasad and W. Kuo, "Reliability optimization of coherent systems," *IEEE Trans. Reliability*, vol. 49, no. 3, pp. 323–330, 2000.
[12] 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, no. 3, pp. 341–349, 2004.
[13] S. S. Rao and A. K. Dhingra, "Reliability and redundancy apportionment using crisp and fuzzy multiobjective optimization approaches," *Reliability Engineering and System Safety*, vol. 37, no. 3, pp. 253–261, 1992.
[14] M. Sakawa, "Multiobjective optimization by the surrogate worth trade-off method," *IEEE Trans. Reliability*, vol. 27, no. 5, pp. 311–314, 1978.
[15] Z. Tian and M. J. Zuo, "Redundancy allocation for multi-state systems using physical programming and genetic algorithms," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 1049–1056, 2006.
[16] Z. Tian, M. J. Zuo, and H. Huang, "Reliability-Redundancy Allocation for Multi-State Series-Parallel Systems," in *Advances in Safety and Reliability*, Kolowrocki, Ed. London: Taylor & Francis Group, 2005, pp. 1925–1930.
[17] F. A. Tillman, C. L. Hwang, and W. Kuo, "Determining component reliability and redundancy for optimum system reliability," *IEEE Trans. Reliability*, vol. 26, no. 3, pp. 162–165, 1977.
[18] Z. Xu, W. Kuo, and H. Lin, "Optimization limits in improving system reliability," *IEEE Trans. Reliability*, vol. 39, no. 1, pp. 51–60, 1990.
[19] G. Xuan and R. Cheng, *Genetic Algorithm and Engineering Design*. Beijing: Science Press, 2000.

Zhigang Tian is currently an Assistant Professor at the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Canada. He received his B.S. degree in 2000, and M.S. degree in 2003, both in Mechanical Engineering at Dalian University of Technology, Dalian, China; and his Ph.D. degree in 2007 in Mechanical Engineering at the University of Alberta, Canada. His research interests focus on reliability analysis and optimization, condition based maintenance, and time series prediction. He is a member of IIE, and INFORMS.

Ming J. Zuo is a Professor at the Department of Mechanical Engineering, University of Alberta, Canada. He received the Master of Science degree in 1986, and the Ph.D. degree in 1989, both in Industrial Engineering from Iowa State University, Ames, Iowa, U.S.A. His research interests include system reliability analysis, maintenance planning and optimization, signal processing, and fault diagnosis. He is a Department Editor, and an Editorial Board member of IIE Trans. He is an Associate Editor of IEEE Trans. Reliability. He is a senior member of IEEE, and IIE.

Hongzhong Huang is a full professor, and the Dean of the School of Mechanical, Electronic, and Industrial Engineering, University of Electronic Science and Technology of China. He received a Ph.D. degree in Reliability Engineering from Shanghai Jiaotong University, China. He is an Editorial Board Member of the *International Journal of Reliability, Quality and Safety Engineering*, *International Journal of Performability Engineering*, *Advances in Fuzzy Sets and Systems*, and *The Open Mechanical Engineering Journal*. His current research interests include system reliability analysis, warranty, maintenance planning and optimization, and computational intelligence in product design.