Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 34 (2008) 181–193

Expert Systems with Applications

www.elsevier.com/locate/eswa

A knowledge management system for series-parallel availability optimization and design

Ying-Shen Juang [a,c,*], Shui-Shun Lin [b], Hsing-Pei Kao [c]

[a] Department of Business Administration, Chung-Hua University, Taiwan, ROC
[b] Department of Business Administration, National Chinyi Institute of Technology, Taiwan, ROC
[c] Institute of Industrial Management, National Central University, Taiwan, ROC

Abstract

System availability is an important subject in the design field of industrial system as the system structure becomes more complicated. While improving the system's reliability, the cost is also on the upswing. The availability is increased by a redundancy system. Redundancy Allocation Problem (RAP) of a series-parallel system is traditionally resolved by experienced system designers. We proposed a genetic algorithm based optimization model to improve the design efficiency. The objective is to determine the most economical policy of components' mean-time-between-failure (MTBF) and mean time-to-repair (MTTR). We also developed a knowledge-based interactive decision support system to assist the designers set up and to store component parameters during the intact design process of repairable series-parallel system.

Keywords: Knowledge management system; Availability optimization; Genetic algorithms; Series-parallel system

1. Introduction

In 1952, the Advisory Group on the Reliability of Electronic Equipment (AGREE) defined the reliability in a broader sense: reliability indicates the probability implementing specific performance or function of products and achieving successfully the objectives within a time schedule under a certain environment (Wang, 1992). In general, a higher priority is placed on quality control rather than reliability in the process of manufacturing. Nonetheless, high quality is not equivalent to high reliability. For example, a certain component, which has passed quality control procedure in conformity to the specifications, may lead to problems when operating with other components. This involves reliability design that is related to electrical or mechanical interface compatibility among spare parts.

With the rapid technological progress and increasing complexity of system structure, any failure of any component may lead to system malfunction or serious damage. For instance, a weapon system is a precise and sophisticated system that comprises several sub-systems, components and spare parts. Failure of even a single element will likely have adverse impact upon the operability of the weapon system, or even threat the national security.

System availability, a concept closely related to reliability, refers to the scale of measuring the reliability of a repairable system. Repairable system indicates a system that can be repaired to operate normally in the event of any failure, such as computer network, manufacturing system, power plant or fire prevention system. Availability comprises "reliability" and "recovery part of unreliability after repair", indicating the probability that repairable systems, machines or components maintain the function at a specific moment" (Wang, 1992). It is generally expressed as the operable time over total time.

* Corresponding author. Address: Department of Business Administration, Chung-Hua University, Taiwan, ROC. Tel.: +886 3 5186060; fax: +886 3 5186570.

E-mail addresses: ysjuang@chu.edu.tw, yes.ebada@gmail.com (Y.-S. Juang).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.08.023

In recent years, reliability and availability have expanded their influence in various industries and fields, thus serve as an integral quality element in the organization system and manufacturing process. To maintain the reliability of sophisticated systems to a higher level, the system's structural design or system components of higher reliability shall be required, or both of them are performed simultaneously (Henley & Kumampto, 1985).

The system structure is virtually designed under the limitations such as weight, volume or other technologies, so the reliability cannot be further improved. In this case, replacing highly reliable components can improve the system reliability. While improving the reliability of systems and components, the associated cost also increases. Thus, it is a very important topic for decision-makers to fully consider both the actual business and the quality requirements. Redundancy Allocation Problem (RAP) of a series-parallel system refers to difficult NP-hard problems (Chern, 1992). Redundancy allocation is designed depending upon the experience of system designers, with the advantages (Chisman, 1998): (1) time-saving and convenient policy-making depending upon years of experience and (2) decision making via experience in the absence of information. The disadvantages include: (1) decision-making is subjective, without scientific support or evidence and (2) individual experience-based decision cannot offer an accurate or optimal design, thus leading to excessive cost. Due to potential risks, the experience-based empirical law may not be universally applied (Li, 2001). Additionally, given the fact of difficult accumulation and inheritance of design expertise, it would be very helpful to transfer, accumulate and manage design knowledge by applying systematic methods and by employing information technologies. Jeang (1999) suggested that computer-aided simulation software could contribute to system design or parameterization. Many information systems were built and a wide variety of methods were used for the reliability design (Chen & Hsu, 2006; Liu & Yang, 1999; Moon, Divers, & Kim, 1998; Varde, Sankar, & Verma, 1998). However, a well-defined knowledge system for reliability design and availability optimization was not found in the literature.

Under repairable series-parallel system framework, there are many methods to determine the optimal parameters of components, such as dynamic planning, integer programming, non-linear integer programming and heuristic or metaheuristic algorithms. As a member of metaheuristic algorithm, Genetic Algorithm (GA) has proved itself to be able to approaching optimal solution against any problem.

The purpose of this study is first to utilize Genetic Algorithms to determine MTBF (mean time between failure, MTBF) and MTTR (mean time to repair, MTTR) of various components, during the design phase of the repairable system, and to optimize availability parameters. We proposed an optimization model of repairable series-parallel system and utilized Genetic Algorithms to find solutions. We then constructed a knowledge-based information system so that the design knowledge can be stored and accumulated. The optimization model and GA procedures ensure that the cost-effective parameters of system availability can be obtained, which helps the system designers formulate optimal design policies and repair policies. The information system stored the system designs and parameters in the knowledge base and can be retrieved by significant features, which facilitates design complexity and increases design efficiency. Specifically, the objective of this study is threefold: (1) develop an optimization model of repairable series-parallel system availability and analyze the model behavior; (2) utilize genetic algorithms to obtain optimal parameter of system components in a cost-effective manner; and (3) construct a knowledge-based information system to accumulate the design knowledge.

2. Literature review

2.1. Reliability of series-parallel system

Series-parallel system indicates sub-systems in which several components are connected in parallel, and then in series, or sub-systems that several components are connected in series, and then in parallel. A series-parallel system can be improved by four methods (Wang, 1992): (1) use more reliable components; (2) increase redundant components in parallel; (3) utilize both #1 and #2; and (4) enable repeatedly the allocation of entire system framework. For the framework of series-parallel system, it is very difficult to find out an optimal solution under multiple constraint conditions (Chern, 1992). Misra Algorithm proposed by Misra and Sharma (1991) solves problems by integer programming, which serves as an algorithm searching for nearby boundary of the domain of feasible solution. Prasad and Kuo (2000) pointed out that Misra algorithm sometimes cannot yield an optimal solution, and suggested a method of searching for the upper limit of reliability's objective function. Gen, Ida, and Lee (1990, 1993) also studied how to solve the problem by integer programming.

The reliability of a series-parallel system has drawn continuous attention in both problem characteristics and solution methodologies. Nakagawa and Miyazaki (1981) utilized several examples to compare the mean failure rate of these methods. After combining Lagrange multiplier and branch-and-bound technologies, Kohda and Inoue (1982) and Kim and Yum (1993) solved the reliability of a series-parallel system by using heuristic algorithm. Kuo, Lin, Xu, and Zhang (1987) proposed a heuristic algorithm LMBB. It obtains rapidly the solution close to the optimal one via Lagrange multiplier. Other large systems, such as those placing limitation on linear resources proposed by Li and Haimes (1992), suggested a three-layer decomposition method for the optimization of system reliability. Mohan and Shanker (1998) selected system components via random selection method according to cost limitation. Hsieh, Chen, and Bricker (1998) utilized genetic algorithms to solve various reliability design problems, which include series systems, series-parallel systems and complex (bridge) systems. While considering maximum system reliability and minimal total cost, Li (2001) solved them by multiple fuzzy objective planning. Yalaoui and Chatelet (2005) formulated an approximated function for the reliability allocation problem in a series-parallel system. You and Chen (2005) proposed an efficient heuristic for series-parallel redundant reliability problems.

2.2. System availability

Under an increasingly complex and diversified system environment, some researchers tended to use simulation methods to evaluate the reliability or availability of a complex system, as common estimation methods are subjected to strict assumptions. Wang (2000) suggested two methods for the estimation of availability. The first method is applicable when the allocation of MTBF and MTTR is subjected to exponential distribution, while the second one is to estimate the interval of availability when none of them is subjected to exponential distribution. These two methods are examined and compared by the Monte Carlo Simulation. Gordan (1996) utilized statistical and graphic techniques to test data, and allowed for the data analysis of reliability and availability in cooperation with SAS statistics software.

In order to evaluate the reliability of hydroelectric generators, Lieber, Nemirovskii, and Rubinstein (1999) proposed Importance Sampling technique to improve the efficiency of traditional Monte Carlo Simulation. Hamersma and Chodos (1992) discussed the availability of telephone network system, indicating that the programs of complex large-scale network can be prepared by different simulation tools, such as Pascal and C, or the GPSS simulation language. Chisman (1998) proposed a failure-mode method and a flow method for non-series-parallel hydraulic system, utilized the GPSS simulation software to simulate the allocation of TBF and TTR of the two systems. Jeang (2001) used computer-aided simulation software VSA-3D/Pro for designing an auxiliary system, and obtained an optimal solution via statistical regression, thus providing a basis for parameterization. Mitchell and Murry (1996) simulated system structure using reliability block diagram (RBD), and forecasted the availability of a simple series-parallel system, indicating that spare parts can be applied to improve the system availability.

Meanwhile, other researches took cost factors into consideration. For example, Propst and Doan (2001) provided the system framework and electronic evaluation sheets, which may help review the availability of power system to improve the system design. In a move to resolve the conflict between maximum availability and minimum cost, Huang (1997) utilized the fuzzy multiple objective optimization and fuzzy attribute function to obtain optimal parameters, which met the cost and availability limitations. Elegbede and Adjallah (2003) employed weighed average for transforming a problem of multiple objective to single objective and solved it with GA. Martorell, Sánchez, Carlos, and Serradell (2004) proposed a general framework for multiple-objective optimization problem based on reliability, availability, maintainability, safety and resource criteria. Two GA-based methods, single-objective GA and multi-objective GA, were used to solve the optimization problem. De Castro and Cavalca (2006) presented an availability optimization of an engineering system assembled in a series configuration, with the redundancy of units and corrective maintenance resources as optimization parameters. The aim is to reach the maximum availability, considering constraints installation and corrective maintenance costs, weight and volume. They used GA as an optimization method as well.

In brief, the system availability can be calculated by approximate expression and Monte Carlo Simulation, of which approximate expression has an advantage of rapid computation along with more limitations, and the simulation method is time-consuming. For the parameterization of series-parallel system availability, it is difficult to obtain the optimal solutions within a limited range of parameters. In such case, it is possible to approach optimal solution within a limited time frame by using the features of generational evolution and parallel search of Genetic Algorithms.

2.3. Genetic algorithms

Genetic Algorithm (GA) is a probabilistic search method stimulated by genetic evolution (Holland, 1975). It was initiated from the 1970s and widely applied to many fields since 1980s. GA can efficiently solve the availability optimization problem of series-parallel, as it is suitable to the domain of feasible solution with non-linearity or discontinuity. Goldberg (1989) made a systematic study on GA mechanism, and identified three basic operators: reproduction, crossover and mutation.

When the solution space to be searched is relatively large, noisy, non-linear and complicated, the GA has higher opportunity for obtaining near-optimal solutions. The GA solely takes fitness function as its evaluation criterion. It is also a parallel processing mechanism, which searches for different areas by multiple starting points. Based on continuous evolution of generations and efficient search using the information of parent generation, it is possible to increase the speed of finding an optimal solution (Lin, Zhang, & Wang, 1995). The mutation mechanism provides more opportunities to overcome the spatial limitations of local optimum, and allows for convergence towards global optimum.

GA was applied to a wide variety of fields in recent decades (Lapa, Pereira, & De Barros, 2006; Lin, Wang, & Zhang, 1997). It was also successfully used to solve the reliability optimization problem of a series-parallel system. Painton and Campbell (1995) solved the reliability optimization problem related to personal computer design. They regarded a personal computer as a series-parallel system of twelve components, each of which has three optional packages. This study utilized GA to obtain optimal solution under budgetary limitation. Yokota, Gen, and Ida (1995) utilized GA to solve successfully the reliability optimization problem of series-parallel system with parallel components and several failure modes, which were formerly solved by the listing technique (Tillman, 1969). Coit and Smith (1996a, 1996b) used GA to solve the reliability optimization problem of series-parallel system meeting the cost and weight constraints. The results proved that GA offered more time-saving solution than the method proposed by Bulfin and Liu (1985), or N&M algorithm by Nakagawa and Miyazaki (1981). Yokota, Gen, and Li (1996) applied GA to solve non-linear mixed integer programming of reliability.

Meanwhile, Gen and Cheng (1996), Yang, Hwang, Sung, and Jin (2000), Yun and Kim (2004), Kumral (2005) also solved reliability problem via the help of GA. Besides, GA was also utilized in determining preventive maintenance schedules, while considering the reliability and system cost (Bris, Chatelet, & Yalaoui, 2003; Lapa et al., 2006; Samrout, Yalaoui, Chatelet, & Chebbo, 2005).

2.4. Computerized reliability design

In most of the industries, classical reliability-centered design is employed to decide the design strategies using reliability data without having an adequate interaction with the maintenance and operational systems. This means that the reliability-centered design process will be conducted with no or limited access to the maintenance and operational data/knowledge (Court, 1998; Hwang, Chow, & Huang, 1996). Gabbar, Yamashita, Suzuki, and Shimada (2003) stated that the commonly developed maintenance strategies were implemented and managed within the computerized maintenance management system, which was usually separated from the reliability-centered maintenance automated environment. They presented a detailed system design mechanism of improved reliability-centered maintenance process integrated with computerized maintenance management system. The proposed solution is integrated with design and operational systems and consolidates some successful maintainability approaches to formulate an effective solution for optimized plant maintenance. A prototype system is implemented by integrating with the various modules of the adopted computerized maintenance management system.

Besides, many information systems were built and a wide variety of methods were used for product reliability design. Varde et al. (1998) used probabilistic safety assessment technologies and information to develop an operator support system for research reactor operations and fault. The system was capable of improving the reliability of operator action and the reactor safety at the time of crisis as well as in normal operation. Moon et al. (1998) used Weibull plots to interpret the results of failure rate and developed an automatic early warning system utilizing expert systems and neural network to capture accurate reliability knowledge. The system led to a significant boost in productivity by at least 8 times in terms of process time. Liu and Yang (1999) developed a computer package named EASYDFQR for quality and reliability design. Design engineers can use it to obtain important design guidelines for quality and reliability. The system is a PC-based expert system. Its knowledge base contains expertise about quality and reliability, such as reliability models, design approaches, failure modes, criticality analysis, and fault tree analysis. It supports computer graphics for the explanation of design guidelines. In addition, design engineers can obtain the knowledge they need via the shortest path using the system.

Data mining usually means the methodologies and tools for an efficient new knowledge discovery from databases. Chen and Hsu (2006) provided an alternative approach by using GA-based mining approach to discover useful decision rules automatically from the breast cancer database. Their approach is capable of extracting rules, which can be further developed as a computer model for the prediction or classification of breast cancer potential like expert systems.

A well-defined knowledge system for product reliability design and availability optimization, however, was not found in the literature. We attempted to construct an optimization model and develop a knowledge-based system to overcome some of the difficulty in accumulating design knowledge.

3. Model construction

3.1. Manufacturing and repairing costs

3.1.1. Manufacturing cost

The manufacturing cost varies with different product specifications. For electronic components, a longer MTFB of manufactured components represents a lower failure (λ) and higher strength, indicating that the components feature high reliability. The product quality is thus ensured once the failure rate of components declines to a desired level. In principle, while the failure rate (λ) is lower, the components are more difficult to fabricate, leading to a sharp increase in the manufacturing cost (Li, 2001). There is a relationship between the MTBF of components and the manufacturing cost (Tillman, Hwang, & Kuo, 1980). We adopted it in this study:

$$C(\mathrm{MTBF}) = \alpha * (\mathrm{MTBF})^{\beta} + \gamma \tag{1}$$

where $C(\mathrm{MTBF})$ represents the component's manufacturing cost; α, β, γ, are constants, representing the physical property of the component, and $\beta > 1$. The relationship is illustrated in Fig. 1(a).

3.1.2. Repairing cost

For a repairable system, failure of a certain component in the system structure may lead to malfunction to some

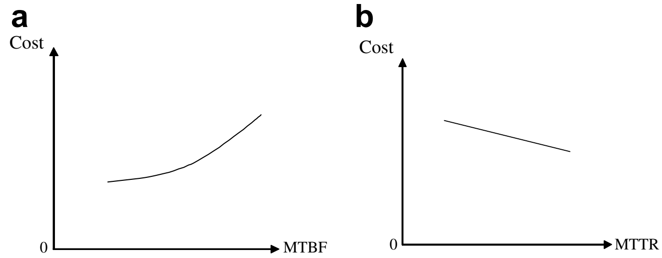

Fig. 1. Relationship between (a) MTBF, (b) MTTR and Cost.

possible extent, or even impair the overall system efficiency. In an effort to avoid such occurrence, it is necessary to repair the faulty components of the system. It is always intended for recovery as soon as possible in an event of system failure. To facilitate the repair within a limited time frame, experienced staff shall be required to work overtime or repair using the state-of-the-art equipments. In such cases, huge investment on work force and equipments will lead to a higher repair cost, despite short repair time. Assuming a linear relationship between MTTR and the repairing cost of components, a lower MTTR indicates a higher repairing cost, with the relationship shown in Fig. 1(b).

3.2. Estimation of system availability

Under the framework of series-parallel system, the availability is usually calculated by simulation method. When the components within reliability block diagram (RBD) are operated independently with each other, the series-parallel framework can be integrated and considered as a single object framework (Birolini, 1999). The estimated parameters of some basic series-parallel frameworks are listed in Fig. 2. By simplifying in sequence up to the hierarchy of entire system, it is possible to list an approximate expression availability of a series-parallel system. The system framework is illustrated in Fig. 3.

The approximate expression assumes that the components are operated independent of each other, failure rate $\lambda_i\,(=1/\mathrm{MTBF}_i)$ and repair rate $\mu_i\,(=1/\mathrm{MTTR}_i)$ are constants, $\lambda_i \ll \mu_i$, and each component is supported by a maintenance team. The simplified framework, as well as the simplified procedures, is shown in Fig. 4.

And thus:

$$
\begin{aligned}
PA_S &= 1-\frac{\lambda_S}{\mu_S} = 1-\frac{\lambda_1\lambda_6}{\mu_1\mu_6} = 1-\frac{\lambda_1}{\mu_1}[\lambda_5/\mu_5+\lambda_4/\mu_4] \\
&= 1-\frac{\lambda_1}{\mu_1}\left[\frac{\lambda_2\lambda_3}{\mu_2\mu_3}+\lambda_4/\mu_4\right] \\
&= 1-\frac{\mathrm{MTTR}_1}{\mathrm{MTBF}_1}\left[\frac{\mathrm{MTTR}_2}{\mathrm{MTBF}_2}\cdot\frac{\mathrm{MTTR}_3}{\mathrm{MTBF}_3}+\frac{\mathrm{MTTR}_4}{\mathrm{MTBF}_4}\right]
\end{aligned}
\tag{2}
$$

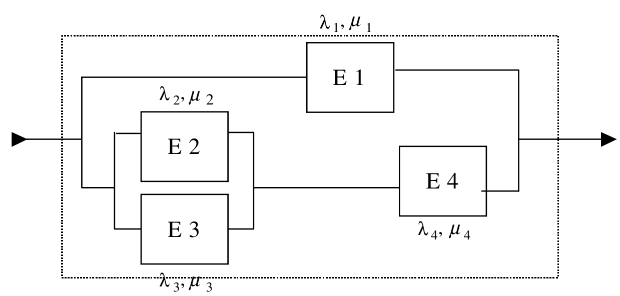

Fig. 3. Example of a series-parallel system.

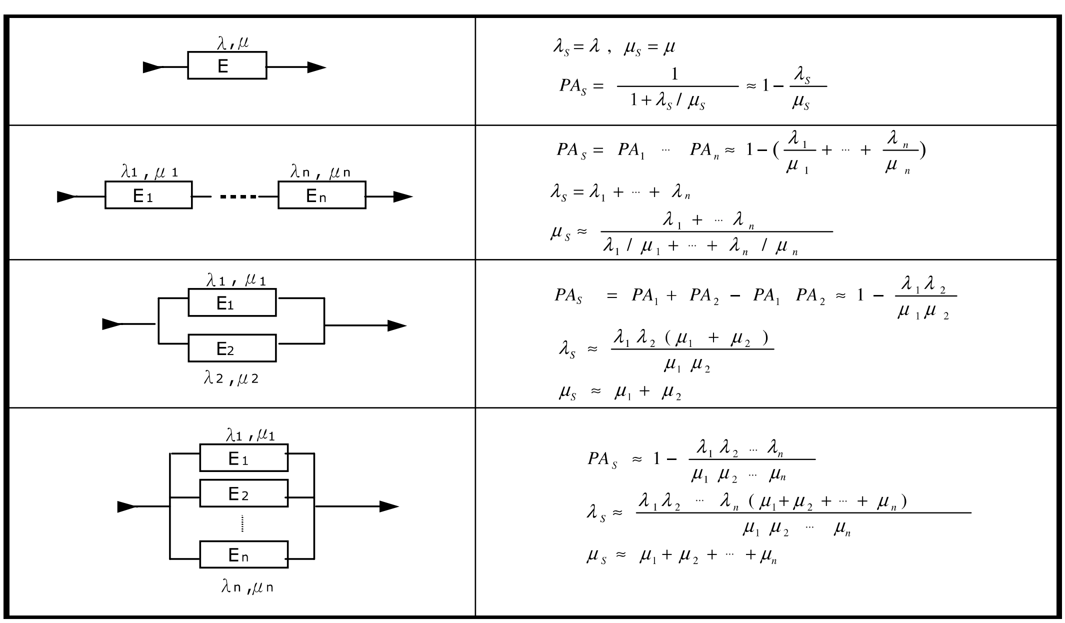

Fig. 2. Availability estimation of series-parallel systems.

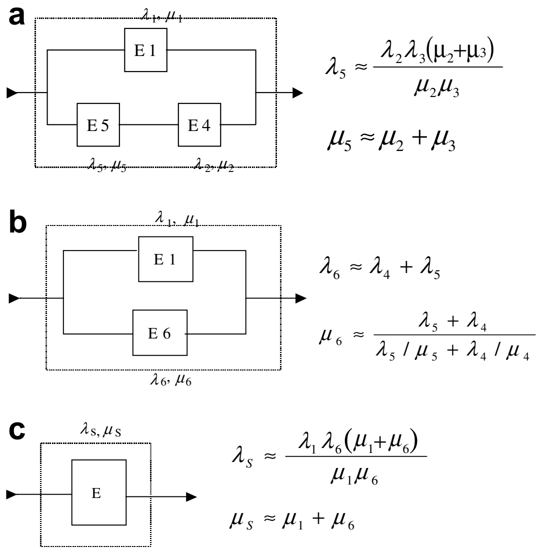

Fig. 4. Simplified framework and procedures.

3.3. Models and formulation procedures

3.3.1. Notation and assumption

Av system availability
Tc total cost of system
$MTBF_i$ MTBF of component i
$CMTBF_i$ manufacturing cost of component i
Lb_MTBF_i lower limit of MTBF of component i
Lb_CMTBF_i manufacturing cost for lower limit of MTBF of component i
Ub_MTBF_i upper limit of MTBF of component i
Ub_CMTBF_i manufacturing cost for upper limit of MTBF of component i
$MTTR_i$ MTTR of component i
$CMTTR_i$ maintenance cost of component i
Lb_MTTR_i lower limit of MTTR of component i
Lb_CMTTR_i maintenance cost for lower limit of MTTR of component i
Ub_MTTR_i upper limit of *MTT* of component i
Ub_CMTTR_i maintenance cost for upper limit of *MTT* of component i
α_i, β_i, γ_i coefficient of functional relationship between manufacturing cost and MTBF of component i, and are positive constants
a_i, b_i coefficient of functional relationship between manufacturing cost and MTTR of component i.

The assumptions are as follows. The failure rate λ and the repair rate μ of system components are constants. The failure time and the repair time are subjected to exponential distribution, and MTBF $\gg$MTTR. To approximate system availability, it is required to meet MTBF $\gg$ MTTR, namely $\mu \gg \lambda$, and assume that maintenance is independent of each other. Besides, the functional relation between MTBF or MTTR and the cost can also be clearly defined.

3.3.2. Problem essence

Repairable series-parallel system has k components. Each component i has two parameters: $MTBF_i$ and $MTTR_i$ to be determined. Their relationship with cost is shown in the following equations:

$$CMTBF_i = \alpha_i \cdot (MTBF_i)^{\beta_i} + \gamma_i \quad i = 1, 2, \ldots, k \tag{3}$$

$$CMTTR_i = a_i - b_i \cdot MTTR_i \quad i = 1, 2, \ldots, k \tag{4}$$

The objective is to locate the design parameters of MTBF and MTTR of each component so as to optimize the system availability of per capital, i.e., availability divided by total system cost. This also means that the designer aims to find out a set of component parameters in conformity to economic efficiency.

3.3.3. Formulation procedures

The proposed model and procedures are given in the following.

Step 1: List the approximate expression of system availability

$$Av = f(MTBF_1, \ldots, MTBF_k, MTTR_1, \ldots, MTTR_k) \tag{5}$$

Step 2: List the summation expression of system cost

$$Tc = \sum_{i=1}^{k} (\alpha_i \cdot (MTBF_i)^{\beta_i} + \gamma_i) + \sum_{i=1}^{k} (a_i - b_i \cdot MTTR_i) \tag{6}$$

Step 3: Construct an objective function

$$\text{Max} \cdot \frac{Av}{Tc} = f(MTBF_1, MTBF_2, \ldots, MTBF_k, MTTR_1, MTTR_2, \ldots, MTTR_k) \tag{7}$$

Step 4: formulate constraints

$$\text{s.t.}: Lb\_MTBF_i \leqslant MTBF_i \leqslant Ub\_MTBF_i$$
$$Lb\_MTTR_i \leqslant MTTR_i \leqslant Ub\_MTTR_i \quad i = 1, 2, \ldots, k \tag{8}$$

Step 5: Combine the formulations in Steps 3 and 4 to form the optimization model.
Step 6: Solve the model with solution procedures of GA.

3.4. Solution procedures of GA

The initial population was generated by GA using a set of settings for operation. Each combination of solutions in the population is called as an individual, which is presented as a chromosome. Chromosome comprises genetic factors, as genes in a series, and is represented by binary strings. A coded chromosome represents design parameters in terms of reliability design.

3.4.1. Coding scheme

We used the chromosome encoding and decoding methods as follows. Assume that there are k variables to be decided. Each variable x_i is ranged between real number a_i and b_i, with the accuracy of mth position after the radix point:

1. Identifying the length of chromosome. Find out minimal integral numberli, so $(b_i - a_i) \cdot 10^m < 2^{li} - 1$, li is the length of decimal code x_i, and the resulting length of each chromosome is:

$$l = \sum_{i=1}^{k} li \tag{9}$$

2. Decoding each variable. Cut off integrally encoded chromosome into k groups of binary bit in sequence, and convert them into the decimal system by the following formula:

$$x_i = a_i + \text{decimal}(1001\ldots001_2) \cdot \frac{b_i - a_i}{2^{li-1}} \tag{10}$$

3.4.2. Operators

As the basic GA mechanism is derived by following the evolution principle in the nature, it comprises gene framework, selection, reproduction, crossover, gene mutation and generation of new individuals within the chromosomes. The flow of GA operation is illustrated in Fig. 5.

The computational process of three GA operators, namely reproduction, crossover and mutation, are described below in detail.

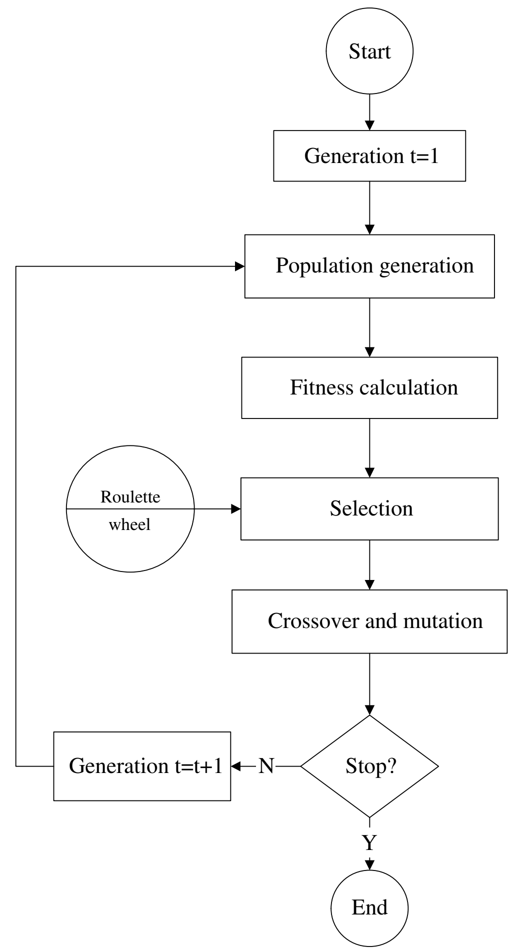

Fig. 5. Flow diagram of GA.

3.4.2.1. Reproduction. Reproduction is to reproduce the chromosome of original population into new one when generating new population. We utilized the roulette wheel to determine the probability for reproduction.

The roulette wheel method is a most frequently used reproduction principle in the process of reproduction. A chromosome encompassing a bigger slot has higher probability of reproducing new generation. The area of the slot is related to the fitness of the chromosome. The selection probability of the individual, P_j, can be represented by the following expression:

$$P_j = \frac{f_j}{\sum_{j=1}^{N} f_j} \tag{11}$$

f_j: fitness value of jth chromosome,
N: number of individuals in a population.

3.4.2.2. Crossover. Crossover exchanges the genes from two chromosomes to generate new individuals. This mechanism is primarily performed to generate chromosomes with better fitness. Basically, there are three crossover schemes, i.e. single-point, two-point and uniform. We employed the two-point crossover scheme. In the two-point crossover, the crossover points are generated randomly. The probability of performing crossover is the crossover rate.

3.4.2.3. Mutation. To approach optimal solution without trapping into local optimum, mutation is applied with a specific probability. Mutation occasionally changes certain genes in the chromosome. The selection is dependent upon mutation rate, which is usually very small. This practice is to avoid the numerous loss of excellent chromosomes, whereas not all genes of chromosomes require for mutation.

After reproduction, crossover and mutation, it is necessary to judge when to stop the evolution. There are some options: (1) maximum number of generations, (2) computation time, and (3) the extent of progress over generations.

3.4.3. GA parameters

Setting GA parameters are mostly based on empirical observation with respect to the problem variety (Elegbede & Adjallah, 2003). The GA parameters used in this study were retrieved from the availability design literature (Gen & Cheng, 1996; Yang et al., 2000; Yokota et al., 1996). Three parameters were set as listed in Table 1.

Table 1
GA parameters

Population size	Crossover rate	Mutation rate
P_s	P_c	P_m
100	0.9	0.01

3.5. Empirical data and solutions

A practical repairable system of jet fighter engine design is illustrated as an example. This system consists of eight components, with its system framework shown in Fig. 6. With reference to the availability and cost factors, it is possible to find out maximum overall efficiency of the entire system.

The range of MTBF and manufacturing cost is tabulated in Table 2. The range of MTTR and repair cost is listed in Table 3.

The solution procedures were coded and embedded in a knowledge management system. We will address the system design process and concept in the rest of this paper. As regard to the key procedure of GA solution, the users

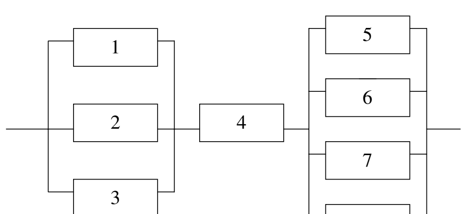

Fig. 6. Block diagram of repairable system.

Table 2
Variance range of MTBF and manufacture cost of components

Component	MTBF lower bound	MTBF upper bound
1	1400 h ($1,012,452)	1600 h ($1,029,622)
2	1250 h ($903,874)	1450 h ($917,857)
3	1300 h ($908,976)	1500 h ($924,782)
4	1850 h ($1,353,896)	2150 h ($1,373,164)
5	1150 h ($983,849)	1250 h ($990,354)
6	1000 h ($852,421)	1100 h ($860,265)
7	700 h ($677,592)	800 h ($685,559)
8	850 h ($837,584)	950 h ($846,704)

Table 3
Variance range of MTTR and repair cost of components

Component	MTTR lower bound	MTTR upper bound
1	80 h ($350,102)	100 h ($335,105)
2	65 h ($341,271)	85 h ($330,272)
3	70 h ($334,936)	90 h ($322,938)
4	2 h ($405,736)	30 h ($400,749)
5	60 h ($296,436)	70 h ($291,438)
6	50 h ($291,972)	60 h ($286,964)
7	35 h ($286,103)	45 h ($281,305)
8	40 h ($290,038)	50 h ($285,437)

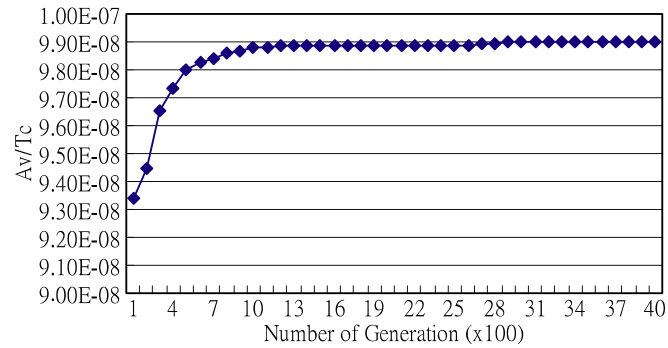

Fig. 7. Optimization process.

can configure and modify the defined parameters of the objective function according to the database of the existing design model. First, the system generated a chromosome with a length of 97 according to the required precision. Second, it configured the GA parameter according to Table 2 and set the termination criterion to a maximum number of 4000 generations. The optimization process over generations has been illustrated in Fig. 7.

The optimal objective value and design parameters are listed in Table 4. The execution time was 472.5 s. The optimal values of MTBF and MTTR for each component can also be obtained. The designers can therefore choose proper components and determine the repair policy according to the optimal design information.

4. System analysis and implementation

4.1. System analysis

The system is configured for installation at the R&D department's site with the database applications installed. The system users are primarily series-parallel system designers of the R&D department. The system database is updated while the system receives commands from users or generates computational results. While executing the optimization process, the system responds to the users with real-time result, which provides feedback for users to control the optimization process. Specifically, the system is divided into five major modules, namely, user interface module, database retrieval and storage module, design con-

Table 4
Optimal design parameter

MTBFs	MTTRs
MTBF1 = 1403.14	MTTR1 = 97.8824
MTBF2 = 1270.39	MTTR2 = 81.8689
MTBF3 = 1314.17	MTTR3 = 87.9528
MTBF4 = 1880.71	MTTR4 = 2.2205
MTBF5 = 1153.23	MTTR5 = 67.4194
MTBF6 = 1025.81	MTTR6 = 59.3548
MTBF7 = 706.67	MTTR7 = 37.0000
MTBF8 = 870.00	MTTR8 = 48.6667
Optimal objective value = 9.89709e−08	

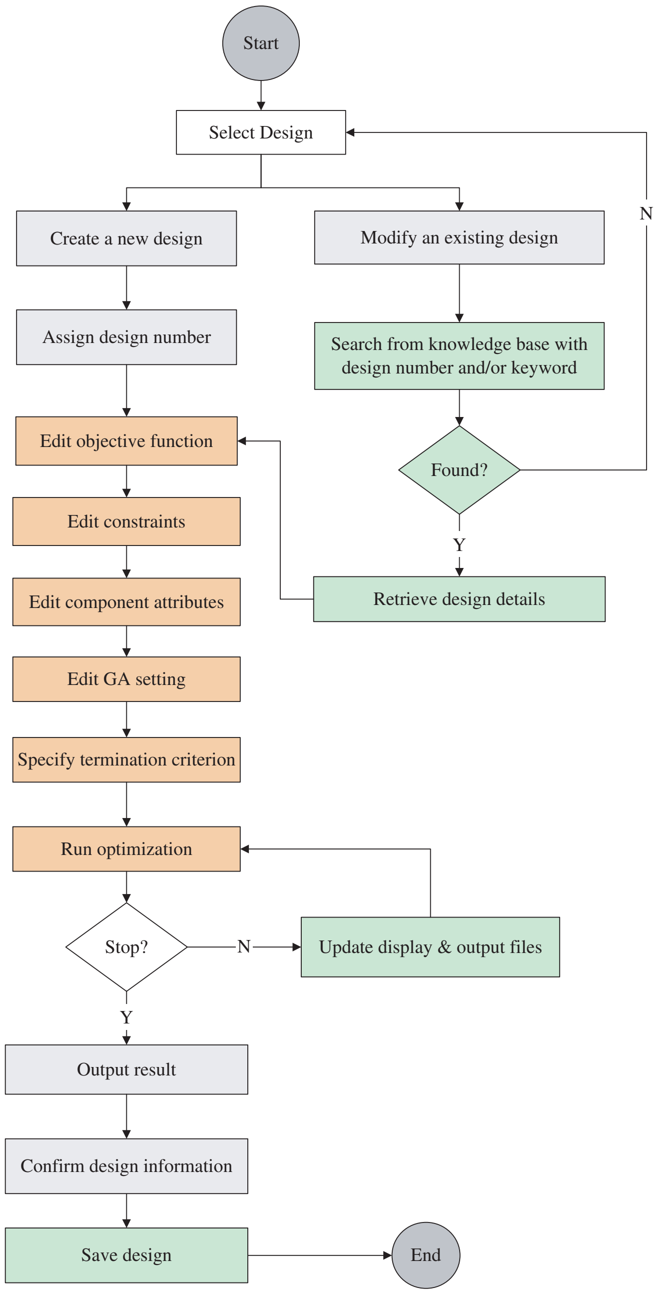

Fig. 8. System flow diagram.

figuration module, GA-based optimization module, and report generation module. The system flow diagram is illustrated in Fig. 8.

4.2. System implementation

The prototype system was developed by using C++ programming language, an object-oriented program technique, including GA procedure coding and connection of the database. The prototype was running on a web server with Windows operation system. After test activity and system refinement, the performance of the system was found promising.

4.2.1. General structure

The general system structure by a set of pseudo code is listed in the following. There are many sections in the pseudo code. The first section (steps 1 to 3) depicts design preparation such as determining the design type and retrieving the data from the knowledge base. The second section (steps 4 to 6) describes the file accessing process, such as reading component parameter file and setting GA parameters. The third section (step 7) depicts the GA based optimization process, which include reproduction, crossover and mutation operations. The fourth section (step 8) indicates the process of generating the report and storing the design project.

```
1. Define system environment
2. Receive user selection from keyboard or mouse
3. if (selection is create_new_design)
      Insert new record to basic_design_table
      else search basic_design_table for existing record by
      keyword
      if (found) retrieve record from table
         else goto 2
4. Define optimization model
   4.1. Define optimum direction
   4.2. Define objective function
   4.3. Define constraints equation
5. Specify and read component data file; open report file
6. Define optimization parameters
   6.1. Define GA parameters Pop, P_m, P_c, N_gene
   6.2. Set initial valuables: N_termi = 1, T_gene = 0, C_count =
        0, C_gene = 0, N_count = 0, m_0 = 0, StopFlag = 0
7. Execute optimization procedures
   7.1. Create randomly the initial population P(0)
   7.2. if (StopFlag = 0)/*check termination criterion*/
        case N_termi = 0: StopFlag = 1/*manually terminate
        process*/
        case N_termi = 1: if (N_count > N_gene) StopFlag =
        1/*terminated by generation number*/
        case N_termi = 2: if (N_count > T_gene) StopFlag =
        1/*terminated by allowed run time*/
        case N_termi = 3: if (N_count > N_gene and C_count <
        C_gene) StopFlag = 1/*terminated by convergence
        over generation*/
   7.3. Select Pop individuals that constitute population
        P_co to be crossed
   7.4. for (i = 1, 2, Pop-1)
        u_1 = rand(0, 1)
        if (u_1 < P_c) cross P_co(i) and P_co(i + 1) and generate
        children Chd_1 and Chd_2
          u_2 = rand (0, 1)
          if (u_2 < P_m)
            mutation(Chd_1); mutation(Chd_2)
            endif
        else
        Chd_1 = P_co(i); Chd_2 = P_co(i + 1)
        endif
```

$P_{ct}(i) = \text{Chd}_1$; $P_{ct}(i+1) = \text{Chd}_2$/* P_{ct} current population indicator*/
endfor
7.5. Store best individual to BEST(N_{count})
if (BEST(N_{count}) better than BEST(m_0))
$m_0 = N_{count}$
if ($N_{termi} = 3$ and BEST(N_{count}) < C_{gene})
C_{count} = BEST(N_{count})
endif
endif
7.6. Call plot(N_{count}, BEST(N_{count})) to update display
Write BEST(N_{count}) and time stamp to report file
7.7. on case do (N_{count}++)
goto 7.2
8. Update database and close report file.

4.2.2. Database design

The design of a database for the prototype system is mainly based on relational database with Access. The setting of primary keys and database normalization are inevitable for implementing a database application system successfully. The primary key is a field that uniquely describes each record. In the course arrangement management system, for instance, the design number was set to be the primary key in the data table storing design information. This field is unique and generated by the coding regulation. There were five major tables used in the system, including basic design table, optimization model table, component attribute table, GA parameter table, and design result table. Two file cabinets containing component setting files and report files were also embedded in the system. The optimization history is included in the report file, which can be further accessed for plotting analytical charts. Fig. 9 shows the system structure with database manipulation.

4.2.3. Interface design

The system interface was designed considering system usability. It allows users to input data by making their choice from the list or type in numbers directly. Fig. 10 presents the design of system interface. Since the system is a web-based application, it can operate with any Internet browser. The process starts from retrieving an existing design or creating a design from scratch. The central area of the display mainly contains GA optimization settings and control. Before the procedure is stopped according to the specified criterion, we allow the procedures to terminate manually. The optimization history is displayed on the right of the screen with the current optimization result. Users of the system can view the intact report after stopping the optimization process.

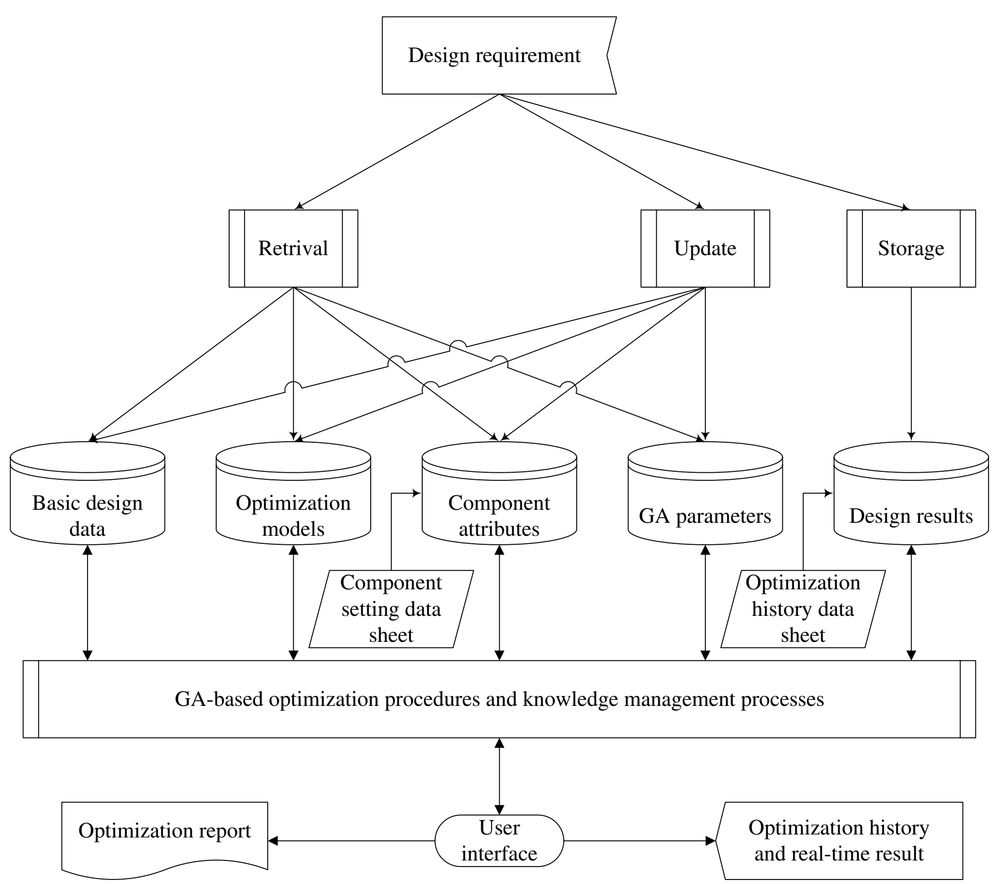

Fig. 9. System structure with database manipulation.

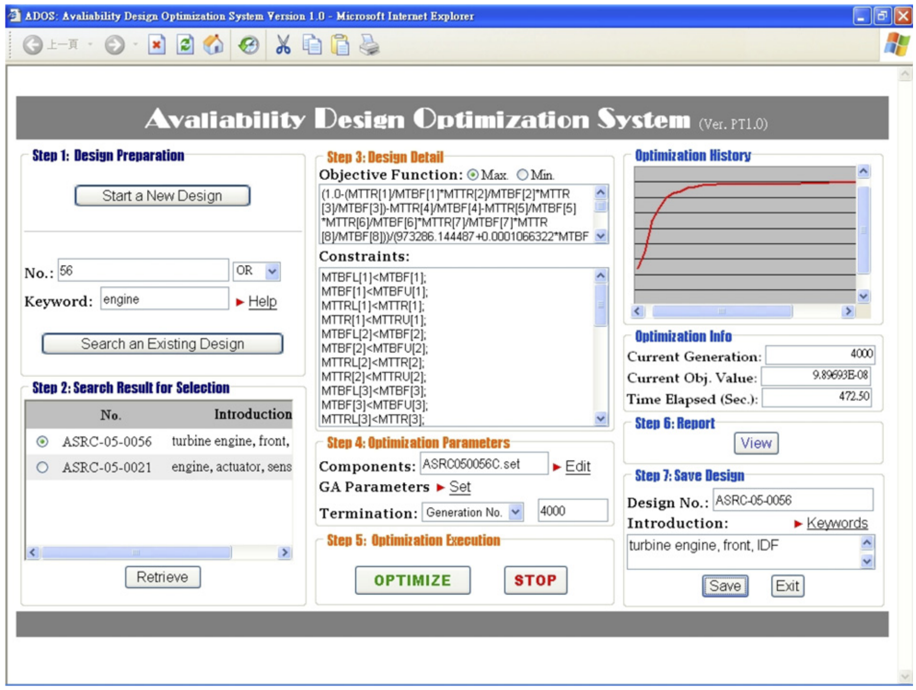

Fig. 10. Design of system interface.

4.3. Knowledge management design

We attempted to make the system more user-friendly by eliminating tedious data entry. For instance, the length of chromosome can be automatically computed and converted into the desired length when intended accuracy was retrieved from the GA setting file. Likewise, the design number and keyword of a modified design were restored from the database, so that no re-entry was needed. It also allowed the user to add phases to the keyword list before storing the design in the database. If better objective values are found, new solutions will be stored in the database and output to the result file automatically.

The operating mechanism of crossover, reproduction and mutation via Genetic Algorithms is the same, despite the framework of series-parallel system. When designing a new series-parallel system, minor modification to the optimization configuration was sufficient, such as altering objective function and constraints, adjusting component number and refining parameters. Since the design knowledge was stored and accumulated in the design library as time elapsed, the definition of optimization process for a new design can be rapidly completed with less effort. In short, the design knowledge was accumulated by a process of retrieving, modifying and storing design parameters, which reduces the burden of system designers and increases the efficiency of designing series-parallel systems from the benefit of accumulated design knowledge.

5. Conclusion

In the intellectual economy era, the design of repairable series-parallel system is inefficient if relying merely on empirical method. It tends to cause increasing design cost due to the difficulty of inheriting design experience. Applying soft computing techniques such as Genetic Algorithms to analyze and optimize the design problems of repairable series-parallel system appears to be very helpful in facilitating the decision-making of system parameter design.

We proposed an optimization model with system availability and design constraints, and then obtained optimal solution by employing Genetic Algorithms. We utilized object-oriented program technique to develop a knowledge system for the availability design of series-parallel systems, which enabled the users to retrieve, modify and fine-tune similar designs from the system database. The system provided the decision-maker with an effective tool to decide the related characteristics of each component.

Several features of the optimization model are to be adequately addressed to employ multiple criteria optimization with weighted objective functions. The GA parameters can be further optimized by experimental design as well. The system developed in this study also needs more refinement. As for the continuation of the exploitable and extendable part of this study, we are seeking ways to improve some features of the knowledge system. Further work will mainly direct to enhancing the conversion capacity for facilitating the data entry of objective function and constraints, and functions of performing sophisticated plotting and statistical analysis from the design knowledge base.

References

Birolini, A. (1999). *Reliability engineering theory and practice*. NY: Springer.

Bris, R., Chatelet, E., & Yalaoui, F. (2003). New method to minimize the preventive maintenance cost of series-parallel systems. *Reliability Engineering and System Safety, 82*, 247–255.

Bulfin, R. L., & Liu, C. Y. (1985). Optimal allocation of redundant components for large systems. *IEEE Transactions on Reliability, R-34*, 241–247.

Chen, T.-C., & Hsu, T.-C. (2006). A GAs based approach for mining breast cancer pattern. *Expert Systems with Applications, 30*, 674–681.

Chern, M. S. (1992). On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters, 11*, 309–315.

Chisman, J. A. (1998). Using discrete simulation modeling to study large-scale system reliability/availability. *Computers and Operations Research, 25*(3), 169–174.

Coit, D. W., & Smith, A. E. (1996a). Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. *Computers and Operations Research, 23*(6), 515–526.

Coit, D. W., & Smith, A. E. (1996b). Reliability optimization of series-parallel systems using genetic algorithm. *IEEE Transactions on Reliability, R45*(2), 254–260.

Court, A. W. (1998). Issues for integrating knowledge in new product development: reflections from an empirical study. *Knowledge-Based Systems, 11*(7–8), 391–398.

De Castro, H. F., & Cavalca, K. L. (2006). Maintenance resources optimization applied to a manufacturing system. *Reliability Engineering and System Safety, 91*(4), 413–420.

Elegbede, C., & Adjallah, K. (2003). Availability allocation to repairable systems with genetic algorithms: a multi-objective formulation. *Reliability Engineering and System Safety, 82*, 319–330.

Gabbar, H. A., Yamashita, H., Suzuki, K., & Shimada, Y. (2003). Computer-aided RCM-based plant maintenance management system. *Robotics and Computer-Integrated Manufacturing, 19*(5), 449–458.

Gen, M., & Cheng, R. (1996). Optimal design of system reliability using interval programming and genetic algorithms. *Computers and Industrial Engineering, 31*(1/2), 237–240.

Gen, M., Ida, K., & Lee, J. U. (1990). A Computational algorithm for solving 0-1 goal programming with GUB structures and its application for optimization problems in system reliability – Part 3. *Electronics and Communications in Japan, 73*, 88–96.

Gen, M., Ida, K., Tsujimura, Y., & Kim, C. E. (1993). Large-scale 0-1 fuzzy goal programming and its application to reliability optimization problem. *Computers and Industrial Engineering, 24*, 539–549.

Goldberg, D. E. (1989). *Genetic algorithm in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Gordan, J. (1996). Computational methods for reliability data analysis. *Annual Reliability and Maintainability Symposium*, 287–290.

Hamersma, B., & Chodos, M. S. (1992). Availability and maintenance considerations in telecommunication network design and the use of simulation. *Proceedings of AFRICON 1992 Conference*, 267–270.

Henley, E. J., & Kumampto, H. (1985). *Design for reliability and safety control*. New Jersey: Prentice-Hall.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press.

Hsieh, Y.-C., Chen, T.-C., & Bricker, D. L. (1998). Genetic algorithms for reliability design problems. *Microelectronics Reliability, 38*, 1599–1605.

Huang, H.-Z. (1997). Fuzzy multi-objective optimization decision-making of reliability of series system. *Microelectron Reliability, 37*(3), 447–449.

Hwang, Y.-J., Chow, L. R., & Huang, H. C. (1996). A knowledge-based approach to the evaluation of fault trees. *Reliability Engineering and System Safety, 52*(1), 77–85.

Jeang, A. (1999). Optimal process parameter determination for computer-aided manufacturing. *Quality and Reliability Engineering International, 15*, 3–16.

Jeang, A. (2001). Computer-aided tolerance synthesis with statistical method and optimization techniques. *Quality and Reliability Engineering International, 17*, 131–139.

Kim, J. H., & Yum, B. J. (1993). A heuristic method for solving reliability redundancy optimization problems in complex systems. *IEEE Transactions on Reliability, 42*(4), 572–578.

Kohda, T., & Inoue, K. (1982). A reliability optimization method for complex systems with the criterion of local optimality. *IEEE Transactions on Reliability, R31*(1), 109–111.

Kumral, M. (2005). Reliability-based optimisation of a mine production system using genetic algorithms. *Journal of Loss Prevention in the Process Industries, 18*, 186–189.

Kuo, W., Lin, H., Xu, Z., & Zhang, W. (1987). Reliability optimization with the Lagrange multiplier and branch-and-bound technique. *IEEE Transactions on Reliability, 36*, 624–630.

Lapa, C. M. F., Pereira, C. M., & De Barros, M. P. (2006). A model for preventive maintenance planning by genetic algorithms based in cost and reliability. *Reliability Engineering and System Safety, 91*, 233–240.

Li, Z. L. (2001). Availability allocation of series-parallel system solved from object-oriented planning, Unpublished master's thesis, Feng-Chia University, Taichung, Taiwan.

Lieber, D., Nemirovskii, A., & Rubinstein, R. Y. (1999). A fast Monte Carlo method for evaluating reliability indexes. *IEEE Transactions on Reliability, 48*(3), 256–261.

Li, D., & Haimes, Y. Y. (1992). A decomposition method for optimization of large-system reliability. *IEEE Transactions on Reliability, 41*, 183–188.

Lin, S.-S., Wang, H.-P., & Zhang, C. (1997). Statistical tolerance analysis based on beta distributions. *Journal of Manufacturing Systems, 16*(2), 150–158.

Lin, S.-S., Zhang, C., & Wang, H.-P. (1995). On mixed-discrete nonlinear optimization problems: a comparative study. *Engineering Optimization, 23*, 287–300.

Liu, T. I., & Yang, X. M. (1999). Design for quality and reliability using expert system and computer spreadsheet. *Journal of the Franklin Institute, 336*(7), 1063–1074.

Martorell, S., Sánchez, A., Carlos, S., & Serradell, V. (2004). Alternatives and challenges in optimizing industrial safety using genetic algorithms. *Reliability Engineering and System Safety, 86*(1), 25–38.

Misra, K. B., & Sharma, U. (1991). An efficient algorithm to solve integer programming problems arising in system reliability design. *IEEE Transactions on Reliability, 40*(1), 81–91.

Mitchell, B. F., & Murry, R. J. (1996). Predicting operational availability for systems with redundant, repairable components and multiple sparing levels. *Annual Reliability and Maintainability Symposium*, 301–305.

Mohan, C., & Shanker, K. (1998). Reliability optimization of complex systems using random search technique. *Microelectronics and Reliability, 28*(4), 513–518.

Moon, Y. B., Divers, C. K., & Kim, H.-J. (1998). AEWS: an integrated knowledge-based system with neural networks for reliability prediction. *Computers in Industry, 35*(2), 101–108.

Nakagawa, Y., & Miyazaki, S. (1981). An experimental comparison of the heuristic methods for solving reliability optimization problems. *IEEE Transactions on Reliability, 30*, 181–184.

Painton, L., & Campbell, J. (1995). Genetic algorithms in optimization of system reliability. *IEEE Transactions on Reliability, 44*, 172–178.

Prasad, V. R., & Kuo, W. (2000). Reliability optimization of coherent systems. *IEEE Transactions on Reliability, 49*, 323–330.

Propst, J. E., & Doan, D. R. (2001). Improvements in modeling and evaluation of electrical power system reliability. *IEEE Transactions on Industry Applications, 37*(5), 1413–1422.

Samrout, M., Yalaoui, F., Chatelet, E., & Chebbo, N. (2005). New methods to minimize the preventive maintenance cost of series-parallel systems using ant colony optimization. *Reliability Engineering and System Safety, 89*, 346–354.

Tillman, F. A. (1969). Optimization by integer programming of constrained reliability problems with several modes of failure. *IEEE Transactions on Reliability, R18*(2), 47–53.

Tillman, F. A., Hwang, C.-L., & Kuo, W. (1980). *Optimization of systems reliability*. New York: Marcel Dekker.

Varde, P. V., Sankar, S., & Verma, A. K. (1998). An operator support system for research reactor operations and fault diagnosis through a connectionist framework and PSA based knowledge based systems. *Reliability Engineering and System Safety, 60*(1), 53–69.

Wang, Z.-H. (1992). *Reliability engineering theory and practice* (5th ed.). Taipei: Quality Control Society of Republic of China.

Wang, W.-D. (2000). Confidence limits on the inherent availability of equipment. *Annual Reliability and Maintainability Symposium*, 162–168.

Yalaoui, C. C., & Chatelet, E. (2005). Reliability allocation problem in a series-parallel system. *Reliability Engineering and System Safety, 90*, 55–61.

Yang, J.-E., Hwang, M.-J., Sung, T.-Y., & Jin, Y. (2000). Application of genetic algorithm for reliability allocation in nuclear power plants. *Reliability Engineering and System Safety, 65*, 229–238.

Yokota, T., Gen, M., & Ida, K. (1995). System reliability of optimization problems with several failure modes by genetic algorithm. *Japanese Journal of Fuzzy Theory and Systems, 7*(1), 117–135.

Yokota, T., Gen, M., & Li, Y.-X. (1996). Genetic algorithm for non-linear mixed integer programming problems and its applications. *Computers and Industrial Engineering, 30*(4), 905–917.

You, P.-S., & Chen, T.-C. (2005). An efficient heuristic for series-parallel redundant reliability problems. *Computers and Operations Research, 32*, 2117–2127.

Yun, W. Y., & Kim, J. W. (2004). Multi-level redundancy optimization in series systems. *Computers and Industrial Engineering, 46*, 337–346.