The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1355-2511.htm

Reliability-redundancy optimization using simulated annealing algorithms

Ho-Gyun Kim and Chang-Ok Bae
Department of Information and Industrial Engineering, Dong-Eui University, Busan, Korea, and
Dong-Jun Park
Division of Mathematical Sciences, Pukyong National University, Busan, Korea

Abstract

Purpose – This paper aims to present a simulated annealing (SA) algorithm to search the optimal solution of reliability-redundancy allocation problems (RRAP) with nonlinear resource constraints.

Design/methodology/approach – The developed SA algorithm is coded in C++ and is applied to reliability design problems which include the series system (P1(a) and P1(b)), the series-parallel system (P2), and the complex (bridge) system (P3). The numerical experiments are executed on an IBM-PC compatible with a Pentium IV 2.0 GHz. The results are compared with those of previous studies.

Findings – The SA algorithm can find better solutions comparable to the previous studies in all problems except the problem P1(b). The difference on the order of 10^{-4} between the best and worst for all problems indicates good solution convergence of the SA algorithm. Note that the CPU times for these problems are within a few seconds by Pentium IV 2.0 GHz (P1(a) = 2.78 sec, P1(b) = 3.37 sec, P2 = 1.38 sec, and P3 = 1.40 sec).

Originality/value – The application of the SA is expanded to the RRAP, which can help reliability engineers design the system reliability.

Keywords Quality control, Parts, Design, Statistical analysis

Paper type Conceptual paper

Introduction

Reliability optimization has attracted attention due to the increasing need for highly reliable systems. In general, two typical ways of improving the system reliability are:

(1) increasing the reliability of components; and

(2) adding the redundant components in various subsystems.

Although system reliability can be improved, it is beyond our requirements even though the highest available reliable components are used. Both ways usually increase the resources such as cost, volume, weight, etc.

Reliability design problems can be mainly classified into two classes, namely for the optimal redundancy allocation problems (RAP) to decide redundancy allocation given component reliabilities and reliability-redundancy allocation problems (RRAP) to decide simultaneously both component reliabilities and redundancy allocation. Depending on linear or nonlinear constraints, these problems can also be divided.

Alot of research has been done for RAP with linear or nonlinear constraints, which is formulated as a pure-integer programming problem. For RRAP, most studies have dealt with the nonlinearly-constrained problems formulated as mixed-integer

Journal of Quality in Maintenance Engineering
Vol. 12 No. 4, 2006
pp. 354-363

1355-2511
DOI 10.1108/13552510610705928

programming problems. Some heuristic methods for RRAP have been developed. Misra and Sharma (1973) initially introduced RRAP subject to linear constraints for the new points of view of the optimal reliability design. Kuo *et al.* (1978) presented a new heuristic method for searching the optimal solution of RRAP subject to nonlinear constraints. Kuo *et al.* (1987) used the Lagrangian multipliers method and the branch-and-bound method for RRAP subject to linear or nonlinear constraints. Xu *et al.* (1990) presented a heuristic method using the Lagrangian multipliers method and the Newton method. Hikita *et al.* (1992) used the surrogate constraint method that formulates multiple constraints to only one constraint.

The heuristic methods have disadvantages, for example there exists no way to improve a solution at local optima and they should be properly developed for each problem characteristic. Recently, metaheuristics such as GA (genetic algorithm), SA (simulated annealing) and TS (Tabu search) are widely used to search the optimal solution of the combinatorial optimization problems. Painton and Campbell (1995) and Yokota *et al.* (1996) used GA to search the optimal component reliabilities and/or redundancy levels for RAP and RRAP. Hsieh *et al.* (1998) also used GA to search to find the optimal solution for RRAP and showed that GA provides better solutions within a few seconds. Furthermore, Nahas and Nourelfath (2005) presented an ant system (AS) algorithm for a variation of RAP, that is, a series system with multiple-choice and a budget constraint incorporated at each subsystem.

While several studies have used GA for the optimal reliability design, there are few studies using SA (Angus and Ames, 1997; Ravi *et al.*, 1997). Angus and Ames (1997) proposed an SA algorithm to find the optimal redundancy levels minimizing system cost subject to reliability constraints. Ravi *et al.* (1997) considered the optimal redundancy levels maximizing system reliability subject to multiple constraints. The studies of the optimal reliability design using SA are restricted to RAP. Kuo *et al.* (2001) was concerned with great potentials for SA application in the reliability design problem. In this paper, an SA algorithm is first proposed to search the optimal solution of RRAP with nonlinear resource constraints and several test problems are investigated to show its effectiveness.

This paper is organized as follows. In section 2, RRAP is formulated and three typical systems in the reliability design are briefly explained; in section 3, an SA algorithm is developed for RRAP and its parameters are described; in section 4, some numerical examples chosen from previous studies are solved and discussed. Finally, conclusions and further studies are provided in section 5.

Reliability-redundancy allocation problems

The reliability-redundancy allocation problems (RRAP) determine the optimal component reliabilities and redundancy level of components in a system to maximize the system reliability subject to several resource constraints. The RRAP is formulated as follows:

$$\max R_S = f(\mathbf{n}, \mathbf{r})$$

$$\text{subject to } g(\mathbf{n}, \mathbf{r}) \leq \mathbf{b}$$

$$n_i \in \text{positive integer},\ 1 \leq i \leq m, 0 \leq r_i \leq 1$$

In the first place, some notations for RRAP are defined. The RRAP can be a mixed-integer nonlinear programming problem, because the number of redundancy n_i is the positive integer values and the component reliability r_i is the real values between 0 and 1:

m:	the number of subsystems;
n_i:	the number of redundancy of subsystem i ($i = 1, 2, \ldots, m, 1 \leq n_i \leq u_i$);
u_i:	the upper bound of redundancy level of subsystem i;
r_i:	the component reliability of subsystem i;
n, r:	the vector of the redundancy allocation and component reliabilities of the system, $\mathbf{n} = (n_1, \ldots, n_m), \mathbf{r} = (r_1, \ldots, r_m)$;
R_S, R_i:	the system reliability and reliability of subsystem i;
g_j:	the j^{th} constraint function;
w_i, v_i, c_i:	the component weight, volume, cost of subsystem i, respectively;
W, V, C:	the upper limit on the weight, volume, cost of the system, respectively; and
b:	the upper limit on the resources of the system.

Some typical examples for RRAP include the series system, the series-parallel system, and the complex (bridge) system and have three nonlinear resource constraints which are weight, volume and cost. These examples were used by Hikita *et al.* (1992), Hsieh *et al.* (1998), Kuo *et al.* (1978), Xu *et al.* (1990) and Yokota *et al.* (1996).

Problem P1(a): Series system (Figure 1(a); Yokota *et al.* (1996)):

$$\max R_S = f(\mathbf{n}, \mathbf{r}) = \prod_{i=1}^{4} R_i(n_i)$$

$$\text{subject to } g_1(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{4} v_i n_i^2 \leq V$$

$$g_2(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{4} \alpha_i \left(-1000/\ln r_i\right)^{\beta_i} \left(n_i + \exp\left(n_i/4\right)\right) \leq C$$

$$g_3(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{4} w_i n_i \exp\left(n_i/4\right) \leq W$$

$$1 \leq n_i \leq 10, 0.5 \leq r_i \leq 1 - 10^{-6}$$

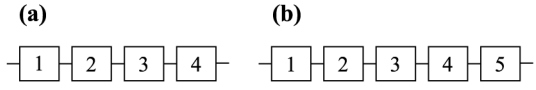

Figure 1.
The series system

Problem P1(b): Series system (Figure 1(b); Hikita *et al.* (1992), Hsieh *et al.* (1998)):

$$\begin{aligned} \max R_S &= f(\mathbf{n}, \mathbf{r}) = \prod_{i=1}^{5} R_i(n_i) \\ \text{subject to } g_1(\mathbf{n}, \mathbf{r}) &= \sum_{i=1}^{5} w_i v_i^2 n_i^2 \le V \end{aligned} \tag{1}$$

$$g_2(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{5} \alpha_i \left(-1000/\ln r_i\right)^{\beta_i} \left(n_i + \exp\left(n_i/4\right)\right) \le C \tag{2}$$

$$g_3(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{5} w_i n_i \exp\left(n_i/4\right) \le W \tag{3}$$

$$1 \le n_i \le 5, 0 \le r_i \le 1 \tag{4}$$

Problem P2: Series-parallel system (Figure 2(a); Hikita *et al.* (1992), Hsieh *et al.* (1998)):

$$\max R_S = f(\mathbf{n}, \mathbf{r}) = 1 - (1 - R_1 R_2)(1 - (1 - (1 - R_3)(1 - R_4))R_5)$$
$$\text{subject to (1)} \sim \text{(4)}$$

Problem P3: complex (bridge) system (Figure 2(b); Hikita *et al.* (1992), Hsieh *et al.* (1998)):

$$\begin{aligned} \max R_S = f(\mathbf{n}, \mathbf{r}) = {} & R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 \\ & - R_1R_2R_4R_5 - R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \end{aligned}$$
$$\text{subject to (1)} \sim \text{(4)}$$

Simulated annealing algorithm

Metaheuristic methods are developed to make up for the weak points of heuristic methods to search for near optimal solutions. Although these methods are developed with different characteristics, many optimization and decision-making fields have used them because they have simple concepts and excellent searching performance for the solution space. In recent years, metaheuristics have been successfully applied to handle a number of reliability design problems.

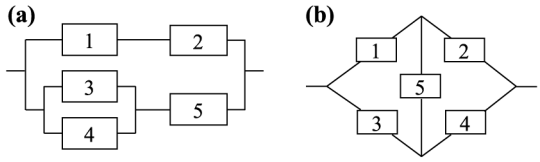

Figure 2.
The series-parallel system and complex (bridge) system

Simulated annealing (SA), introduced by Kirkpatrick *et al.* (1983) and Cerny (1985) as an alternative of the local search, is a general probabilistic method for solving combinatorial optimization problems. SA is an approach to search the global optimal solution that attempts to avoid entrapment in poor local optima by allowing an occasional uphill move to inferior solutions. This paper presents an SA algorithm to search for an (near) optimal solution of the RRAP. To apply the SA algorithm for the problem, the solution representation and the energy function are to be determined and annealing parameters (initial solution, initial temperature, cooling rate and stopping criterion) are to be initialized.

Initialization step

Solution of the RRAP is constituted of integer parts representing the redundancy level and real parts representing the components reliability, so it is described as $(\mathbf{n}, \mathbf{r}) = \{(n_1, n_2, \ldots, n_m), (r_1, r_2, \ldots, r_m)\}$. Figure 3 shows the representation of a solution that has m subsystems. The l binary digits for number of redundancy n_i represents an integer ν_i within $[0, 2^l - 1]$ that yields the number of redundancy within $[1, u_i]$ as $n_i = < 1 + \nu_i(u_i - 1)/(2^l - 1) >$. The $< \cdot >$ operator denotes rounding to the near integer. Like the l binary digits, the k binary digits for component reliability r_i represents an integer ρ_i within $[0, 2^k - 1]$ that is transformed to the component reliability within $[0, 1]$ as $r_i = \rho_i/(2^k - 1)$. As the component reliabilities are practically more than 0.5, the first position of the k binary digits is fixed as 1 to search feasible solutions efficiently. The values of l and k determine the precision of the variables, and we used $l = 8$ and $k = 16$ determined by preliminary experiments.

The energy function E that refers to an evaluation function of the performance of the SA, uses the objective function of the problem and its value will be zero if it violates the constraint functions. The initial solution is initialized by a randomly generated solution. The initial solution is evaluated by the energy function, and then the energy function value becomes the energy function values of both the current solution (X_C) and best solution (X_B). Initial and final values of the control parameter temperature referred to as T_0 and T_F, respectively, are specified. The length of iterations for each level of the current temperature T_C, referred to as L, is set as a constant. The current temperature of each iteration is adjusted by its cooling rate α. Various tests have been performed for these examples to select appropriate values of the parameters and were averaged over ten trials. All the tests were carried out while all the others were held constant. Figure 4 shows the results of problem P1(a) produced by the SA algorithm with α and L ranging from 0.9 to 0.99 and from 100 to 400, respectively and $T_0 = 50$, $T_F = 1$. It appears that α and L play an essential role on the quality of solutions obtained. The algorithm

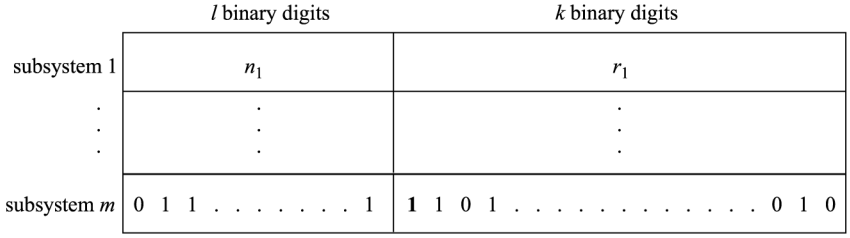

Figure 3.
Solution representation of the problem

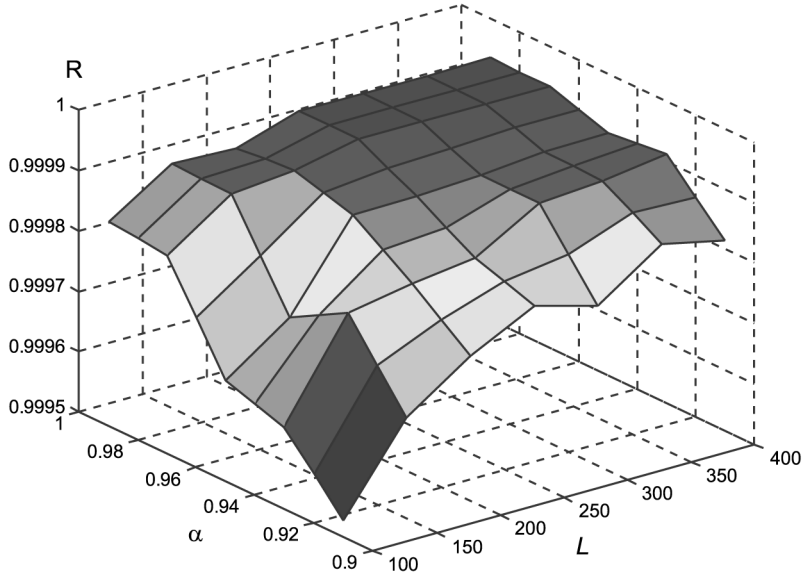

Figure 4.
Effects of α and L on the performance of the algorithm

yields the best solution with $\alpha = 0.97$ and $L = 250$. The values of $\alpha = 0.95$ and $L = 150$ yield satisfactory results for both the series-parallel (P2) and the complex system (P3).

Generation step of a feasible neighborhood solution

The solution can be divided into integer and real parts. For the integer part, since the variation effect of its element changes is greater than that for the real part, only one position is chosen and its element is changed by a random number of 0 or 1. For the real part, four positions except the first are randomly chosen and their elements in positions 1 and 2 are exchanged with those in positions 3 and 4, respectively. Figure 5 shows a sample of generation of a feasible neighborhood solution.

Evaluation step for acceptance of neighborhood solution

If the energy function value of the neighborhood solution is more than that of the current solution ($E_N > E_C$), the neighborhood solution will replace the current solution. Then compare this neighborhood solution's energy function value with that of

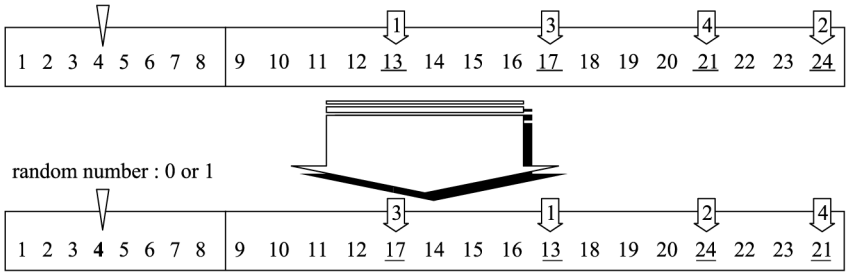

Figure 5.
A sample of the generation of a feasible neighborhood solution

the best solution found thus far (E_B). If $E_N > E_B$, then replace the best solution with E_N. Otherwise, if $E_N < E_C$ then whether or not to accept the neighborhood solution is determined by the acceptance probability $P(A) = \exp(-\Delta E/T)$, where $\Delta E = E_C - E_N$ is referred to as the difference between the energy function value of the current solution and that of the neighborhood solution.

Increment step of iteration counter

Increase the iteration counter, N, by one. And then return to the generation step of a neighborhood solution. If the value of N is greater than or equal to the maximum number of iterations for each temperature level (L), proceed to the cooling schedule step.

Cooling schedule and stopping step

The temperature is adjusted by its cooling rate α. This is calculated by $T_C = \alpha T_C - 1 (C = 1, 2, \ldots)$. If the new value of T_C is greater than or equal to the stopping value of T_F (if $T_C \geq T_F$) then reset N to one and return to the generation step of a feasible neighborhood solution. Otherwise, stop.

Numerical experiments

To evaluate the performance of the SA algorithm for the RRAP, numerical experiments for some examples (P1-P3) are conducted. Table I shows values of the parameters defining the problems, which are identical to those of Hikita *et al.* (1992), Hsieh *et al.* (1998), Kuo *et al.* (1978), Xu *et al.* (1990) and Yokota *et al.* (1996). The SA algorithm is coded in C++ and numerical experiments are executed on an IBM-PC compatible with a Pentium IV 2.0 GHz.

To evaluate the performance of the SA algorithm, we test problems P1-P3 ten times under the same conditions. The numerical results are, shown in Tables II-V, in which the best and worst solutions are reported to see the quality of solutions convergence and compared with those of previous studies.

The results of the experiment for the problem P1(a), shown in Table II, indicate that a best solution of the SA algorithm ($R_S = 0.999945$) is better than the best solution of Yokota *et al.* (1996) ($R_S = 0.999468$). In addition, the best solution of Yokota *et al.* (1996) is inferior to the worst solution of the SA algorithm ($R_S = 0.999820$) and is not a feasible solution violating the cost constraint function.

The results of the experiment for the problem P1(b) are shown in Table III. The SA algorithm ($R_S = 0.931460$) does not give a better solution than the solution of Hsieh

Table I.
Data of the problems

	P1(a)	P1(b) and P3	P2
$10^5\alpha_i$	(1.0, 2.3, 0.3, 2.3)	(2.330, 1.450, 0.541, 8.050, 1.950)	(2.500, 1.450, 0.541, 0.541, 2.100)
v_i	(1, 2, 3, 2)	(1, 2, 3, 4, 2)	(2, 4, 5, 8, 4)
w_i	(6, 6, 8, 7)	(7, 8, 8, 6, 9)	(3.5, 4.0, 4.0, 3.5, 4.5)
V	250	110	180
C	400	175	175
W	500	200	100
β_i	1.5	1.5	1.5

et al. (1998) ($R_S = 0.931578$), but it is better than the solution of Hikita *et al.* (1992) ($R_S = 0.931363$). Tables IV and V illustrate the results of the experiments of the problems P2 and P3. The solutions of the SA algorithm are better than the solution of Hsieh *et al.* (1998) for both problems.

The difference on the order of 10^{-4} between the best and worst for all problems indicates good solution convergence of the SA algorithm. Note that the CPU times for these problems are within a few seconds by Pentium IV 2.0 GHz (P1(a) = 2.78 sec, P1(b) = 3.37 sec, P2 = 1.38 sec, and P3 = 1.40 sec). We observed the CPU times for the series system is longer than that for the series-parallel and the complex (bridge)

		The SA algorithm	
	Yokota *et al.*	The best solution	The worst solution
n	(3, 6, 3, 5)	(5, 5, 5, 5)	(5, 6, 3, 5)
	0.965993	0.895644	0.872465
	0.760592	0.885878	0.801556
r	0.972646	0.912184	0.966934
	0.804660	0.887785	0.862699
R_S	0.999468	0.999945	0.999820
unused (*V*, *C*, *W*)	(92, −70.7336, 127.5832)	(50, 0.9380, 28.8037)	(76, 85.9126, 60.9789)

Table II.
Experiment result of the problem P1(a)

			The SA algorithm	
	Hikita *et al.*	Hsieh *et al.*	The best solution	The worst solution
n	(3, 2, 2, 3, 3)	(3, 2, 2, 3, 3)	(3, 2, 2, 3, 3)	(3, 2, 2, 3, 3)
	0.777143	0.779427	0.782391	0.777325
	0.867514	0.869482	0.866712	0.865980
r	0.896696	0.902674	0.901747	0.913130
	0.717739	0.714038	0.717266	0.713634
	0.793889	0.786896	0.783795	0.778836
R_S	0.931363	0.931578	0.931460	0.931050
unused (*V*, *C*, *W*)	(27, 0, 7.518918)	(27, 0.121454, 7.518918)	(27, 0.053194, 7.518918)	(27, 0.022052, 7.518918)

Table III.
Experiment result of the problem P1(b)

			The SA algorithm	
	Hikita *et al.*	Hsieh *et al.*	The best solution	The worst solution
n	(3, 3, 1, 2, 3)	(2, 2, 2, 2, 4)	(2, 2, 2, 2, 4)	(2, 2, 2, 2, 4)
	0.838193	0.785452	0.812161	0.735012
	0.855065	0.842998	0.853346	0.741222
r	0.878859	0.885333	0.897597	0.919905
	0.911402	0.917958	0.900710	0.896483
	0.850355	0.870318	0.866316	0.885176
R_S	0.99996875	0.99997418	0.99997631	0.99996786
unused (*V*, *C*, *W*)	(27, 0, 7.518918)	(40, 1.194440, 1.609289)	(40, 0.007300, 1.609289)	(40, 0.008032, 1.609289)

Table IV.
Experiment result of the problem P2

Table V.
Experiment result of the problem P3

	Hikita *et al.*	Hsieh *et al.*	The SA algorithm The best solution	The SA algorithm The worst solution
n	(3, 3, 2, 3, 2)	(3, 3, 3, 3, 1)	(3, 3, 3, 3, 1)	(2, 4, 2, 2, 2)
	0.814483	0.814090	0.807263	0.859510
	0.821383	0.864614	0.868116	0.838636
r	0.896151	0.890291	0.872862	0.927733
	0.713091	0.701190	0.712673	0.670787
	0.814091	0.734731	0.751034	0.794690
R_S	0.99978937	0.99987916	0.99988764	0.99973337
unused (V, C, W)	(27, 0, 7.518918)	(40, 1.194440, 1.609289)	(40, 0.007300, 1.609289)	(38, 0.012276, 14.09171)

system. It is conjectured that longer times result from the wider space of feasible solutions for the series system. In the Tables II-V, ten numerical experiments are run and the best and worst solutions among them are reported. In all problems except P1(b), the SA can find better solutions in comparison with the previous studies.

Conclusions

This paper considered reliability-redundancy problems (RRAP) for three typical types of reliability problems to determine simultaneous optimal reliability and redundancy level of components in system. The objective of the problems is to maximize the system reliability subject to three nonlinear resource constraints. The SA algorithm to search for the optimal solution of the problems was developed. To evaluate the performance of the SA algorithm, numerical experiments were conducted and compared with previous studies for the series system, the series-parallel system, and the complex (bridge) system. The limited examples suggests the best solution of the SA algorithm is better than the previous best solutions except the problem of series system P1(b).

We assumed the system has identical components in the subsystem and one failure mode. It is more practical to consider various choices among components and failure modes of the components in the subsystem. We also treated RRAP for small systems constituting four or five subsystems. Applying the SA to large systems needs to be investigated. We required them to be further studied.

References

Angus, J.E. and Ames, K. (1997), "A simulated annealing algorithm for system cost minimization subject to reliability constraints", *Communications in Statistics: Simulation and Computation*, Vol. 26 No. 2, pp. 783-90.

Cerny, V. (1985), "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm", *Journal of Optimization Theory and Applications*, Vol. 45, pp. 41-51.

Hikita, M., Nakagawa, Y. and Narihisa, H. (1992), "Reliability optimization of systems by a surrogate-constraints algorithm", *IEEE Transactions on Reliability*, Vol. 41 No. 3, pp. 473-80.

Hsieh, Y.C., Chen, T.C. and Bricker, D.L. (1998), "Genetic algorithms for reliability design problems", *Microelectronics Reliability*, Vol. 38 No. 10, pp. 1599-605.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), "Optimization by simulated annealing", *Science*, Vol. 220, pp. 671-80.

Kuo, W., Hwang, C.L. and Tillman, F.A. (1978), "A note on heuristic methods in optimal system reliability", *IEEE Transactions on Reliability*, Vol. 27 No. 5, pp. 320-4.

Kuo, W., Lin, H.H., Xu, Z. and Zhang, W. (1987), "Reliability optimization with the Lagrange-multiplier and branch-and-bound technique", *IEEE Transactions on Reliability*, Vol. 36 No. 5, pp. 624-30.

Kuo, W., Prasad, V.R., Tillman, F.A. and Hwang, C.L. (2001), *Optimal Reliability Design: Fundamentals and Applications*, Cambridge University Press, Cambridge.

Misra, K.B. and Sharma, J.A. (1973), "A new geometric programming formulation for a reliability problem", *International Journal of Quality Control*, Vol. 18 No. 3, pp. 497-503.

Nahas, N. and Nourelfath, M. (2005), "Ant system for reliability optimization of a series system with multiple-choice and budget constraints", *Reliability Engineering and System Safety*, Vol. 87, pp. 1-12.

Painton, L. and Campbell, J. (1995), "Genetic algorithms in optimization of system reliability", *IEEE Transactions on Reliability*, Vol. 44 No. 2, pp. 172-8.

Ravi, V., Muty, B. and Reddy, P. (1997), "Nonequilibrium simulated-annealing algorithm applied reliability optimization of complex systems", *IEEE Transactions on Reliability*, Vol. 46 No. 2, pp. 233-9.

Xu, Z., Kuo, W. and Lin, H.H. (1990), "Optimization limits in improving system reliability", *IEEE Transactions on Reliability*, Vol. 39 No. 1, pp. 51-60.

Yokota, T., Gen, M. and Li, Y.X. (1996), "Genetic algorithm for non-linear mixed-integer programming and its applications", *Computers & Industrial Engineering*, Vol. 30 No. 4, pp. 905-17.

Further reading

Metropolis, N., Rosenbluth, A., Rosenbluth, N., Teller, N. and Teller, E. (1953), "Equation of state calculations by fast computing machine", *Journal of Chemical Physics*, Vol. 21, pp. 1087-92.

Corresponding author

Ho-Gyun Kim can be contacted at: hgkim@deu.ac.kr