Genetic Algorithms for Reliability Design Problems

Yi-Chih Hsieh
Department of Industrial Engineering
National Yunlin Polytechnic Institute
Huwei, Yunlin 63208, Taiwan

Ta-Cheng Chen and **Dennis L. Bricker**
Department of Industrial Engineering
The University of Iowa
Iowa City, IA 52242, USA

(Revised, May 1997)

Abstract

This paper presents genetic algorithms for solving various reliability design problems, which include series systems, series-parallel systems and complex (bridge) systems. The objective is to maximize the system reliability, while maintaining feasibility with respect to three nonlinear constraints, namely, cost and weight constraints, and constraints on the products of volume and weight. In this paper, both integer reliability problems (component reliabilities are given and redundancy allocation is to be decided) and mixed-integer reliability problems (both component reliabilities and redundancy allocation are to be decided simultaneously) are studied. Numerical examples show that genetic algorithms perform well for all the reliability problems considered in this paper. In particular, as reported, some solutions obtained by genetic algorithms are better than previously best-known solutions.

All correspondence should be directly addressedto Prof. Dennis L. Bricker, Department of Industrial Engineering, University of Iowa, Iowa City, IA 52242, USA. dbricker@icaen.uiowa.edu

1. Introduction

Highly reliable systems can reduce loss of money and time. Typically, there are two main approaches to enhancing the system reliability, namely, (1) increasing the reliability of the elements constituting the system and/or (2) using redundant elements in various subsystems in the system (Misra and Sharma [20]). In the former approach, while the system reliability can be improved, the required enhancement of the reliability may be beyond what is attainable even though the most reliable of currently available elements are used. Using the latter approach, the system reliability can also be enhanced, but, at the same time, the cost, weight, volume, etc. will increase as well. According to these two main approaches, two typical reliability design problems are often investigated, namely (1) integer reliability problems (component reliabilities are given and redundancy allocation is to be decided) and (2) mixed-integer reliability problems (both component reliabilities and redundancy allocation are to be decided simultaneously).

For the integer reliability problems, we may classify the problems and the approaches into two groups, namely

(a) Maximization of the system reliability subject to linear constraints. For example, Federowicz and Mazumdar [6], and Misra and Sharma [20] (using geometric programming); and Hiller and Lieberman [12] (using dynamic programming); and Misra [19] (by using a heuristic method).

(b) Maximization of the system reliability subject to nonlinear constraints. For example, Sharma and Venkateswaran [26], Luus [17], Nakagawa and Nakashima [22], Aggarwal [1], Tillman et al. [28], Kuo et al. [16], Gopal et al. [9,10], Nakagawa and Miyazaki [21], Kohda and Inoue [15], and Hikita et al. [11] (using heuristic approaches); and Park [25] (by using a fuzzy approach).

One can observe that for linearly constrained integer reliability problems, heuristic methods and geometric programming methods have often been proposed, while a fuzzy approach has been proposed when the constraints are nonlinear.

For the mixed-integer reliability problems, most efforts were devoted to nonlinearly-constrained reliability-redundancy problems. Misra et al. [20] and Nakashima et al. [21] used Lagrange multiplier and dynamic programming, respectively, to solve single-constrained reliability-redundancy problems; Hikita et al. [11], Kuo et al. [16], and Xu et al [30] used a surrogate dual, Lagrange multipliers with branch-and-bound, and a heuristic approach, respectively, to solve reliability-redundancy (mixed-integer) problems with multiple nonlinear constraints. However, most of these require derivatives for all nonlinear constraint functions, and provide only a single unique solution, that is, the design engineers are presented no other options among which to choose.

Recently, Genetic Algorithms (GAs), originally developed by Holland [13], have been widely studied and applied to solve a variety of optimization problems, usually of a combinatorial nature. This class of algorithms, inspired by the principles of natural selection and population genetics, simulates a population of individuals (potential solutions to the problem at hand) which mate, produce offspring, occasionally mutate, and (by means of "survival of the fittest") evolve into superior individuals. Owing to numerous reports of successful applications of these innovative algorithms, GAs have attracted more recent attention than most other heuristic methods in various fields, including reliability optimization problems. For example, Coit and Smith [2-5], Gen [7], Ida et al. [14], and Painton and Campbell [23,24] have used GAs to solve integer reliability problems and have reported effective and efficient solutions. For the mixed-integer reliability problems, on the other hand, Yokota et al. [29] used GAs to solve a triply-constrained reliability-redundancy problem in which four integer variables (for redundancy allocation) and four real variables (for reliabilities of components) are to be determined simultaneously. They reported that a near optimal solution was found in few seconds of CPU time. The above brief survey reveals that most efforts in the use of GAs to solve reliability design problems have been devoted to integer reliability problems, while the use of GAs to solve mixed-integer reliability problems is rare.

In this paper, nonlinearly constrained reliability problems for both integer reliability problems (component reliabilities are given and redundancy allocation is to be decided) and mixed-integer reliability problems (both the component reliabilities and redundancy allocation are to be decided simultaneously) are studied. For both integer reliability problems and mixed-integer reliability problems, three typical types of reliability problems, which includes series, series-parallel, and more complex (bridge) systems, are investigated by GAs. Numerical results show that GAs can solve these systems efficiently and effectively. Indeed, for the three typical studied systems, some solutions found by the GAs are better than previously best-known solutions.

This paper is arranged as follows: in the next section the integer reliability problems and mixed-integer reliability problems are briefly described; in Section 3 the general concept of a GA is described; numerical examples of various reliability systems are solved and discussed in Section 4. Finally, a short conclusion is provided.

2. Reliability Problems

Before introducing the problems, we list below several notations used in this paper.

m	The number of subsystems in the system.
n_i	The number of components in subsystem i, $1 \le i \le m$.
n	$\equiv (n_1, n_2, ..., n_m)$, the vector of the redundancy allocation for the system.
r_i	The reliability of each component in subsystem i, $1 \le i \le m$.
r	$\equiv (r_1, r_2, ..., r_m)$, the vector of the component reliabilities for the system.
q_i	$= 1 - r_i$, the failure probability of each component in subsystem i, $1 \le i \le m$.
$R_i(n_i)$	$= 1 - q_i^{n_i}$, the reliability of subsystem i, $1 \le i \le m$.
R_s	The system reliability.
g_i	The ith constraint function.
w_i	The weight of each component in subsystem i, $1 \le i \le m$.
v_i	The volume of each component in subsystem i, $1 \le i \le m$.

c_i The cost of each component in subsystem i, $1 \le i \le m$.

V The upper limit on the sum of the subsystems' products of volume and weight.

C The upper limit on the cost of the system.

W The upper limit on the weight of the system.

b The upper limit on the resource.

2.1 Integer Reliability Problems

For integer reliability problems, component reliabilities are already specified and only redundancy allocation is to be decided. To evaluate the performance of GAs, we consider three typical types of redundancy reliability systems which have been solved previously by other approaches. The problem of maximizing the system reliability subject to multiple nonlinear constraints can be stated as the following integer nonlinear programming problem (P1):

$$\begin{aligned} \text{(P1)}\ \max\ & R_s = f(n) \\ s.t.\ & g(n) \le b \\ & n_i \quad \text{positive integer}, \ 1 \le i \le m \end{aligned} \tag{1}$$

(1) Series systems (Hikita et al. [11] and Xu et al. [30]). There are m subsystems with subsystem i consisting of n_i elements, $1 \le i \le m$. Such a system (with $m = 5$) is shown in Figure 1, with reliability

$$\prod_{i=1}^{m} R_i(n_i) \tag{2}$$

where the reliability of subsystem i is $R_i(n_i) = 1 - q_i^{n_i}$.

Figure 1 goes here

(2) Series-parallel systems (Hikita et al. [11]). A system of this type is shown in Figure 2, with reliability

$$1 - (1 - R_1R_2)\left(1 - \left(1 - (1 - R_3)(1 - R_4)\right)R_5\right) \tag{3}$$

where the dependence of R_i on n_i has been suppressed for convenience of notation.

Figure 2 goes here

(3) Complex (bridge) systems (Hikita et al.[11]). A typical example is shown in Figure 3, with reliability

$$R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 \\ - R_1R_2R_4R_5 - R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \quad (4)$$

Figure 3 goes here

It should be noted that the series-parallel and the bridge problems were considered by Hikita et al. [11] as mixed-integer problems in which both element reliability r_i and redundancy number n_i ($i = 1,2,3,4,5$) are decision variables to be assigned values. However, in order to compare the performance of their heuristic algorithms and GA, we have here assumed that element reliability r_i has been fixed at the values which Hikita et al. report as being selected by their heuristic method. Our objective, then, is restricted to finding the optimal redundancy allocation n_i for each element i so as to maximize the system reliability.

Three nonlinear constraints which have been imposed upon the levels of redundancy by Hikita et al. [11] and others will also be imposed in our evaluation of GA. The first constraint, apparently introduced by Tillman et al. [27], restricts the sum of the products of the subsystem weights and squares of volumes:

$$g_1 = \sum_{i=1}^{m} w_i v_i^2 n_i^2 \le V \quad (5)$$

The second constraint restricts the total cost of the system:

$$g_2 = \sum_{i=1}^{m} \alpha_i \left(-1000/\ln r_i\right)^{\beta_i} \left(n_i + \exp(n_i/4)\right) \le C \quad (6)$$

where the exponential term represents the cost of interconnecting parallel elements (The parameters α_i and β_i for each element i are assumed to be given). The third and last constraint restricts the total weight of the system:

$$g_3 = \sum_{i=1}^{m} w_i n_i \exp(n_i/4) \le W \quad (7)$$

where the exponential factor adjusts for the weight of the hardware interconnecting the parallel elements. Additionally, of course, we impose the constraint

$$g_4 : n_i \quad \text{positive integers}, 1 \le i \le m. \quad (8)$$

The three problems above, namely the optimization of the system reliability as given by (2), (3), or (4), subject to constraints (5-8), will be designated P1a, P1b, and P1c, respectively, and will be used in our evaluation of GA in Section 4.1.

2.2 Mixed-Integer Reliability Problems

In the mixed-integer reliability problems, both component reliabilities and redundancy allocation are to be decided simultaneously. The problem of maximizing the system reliability subject to multiple nonlinear constraints can be expressed as the following mixed-integer nonlinear programming problem (P2):

$$\text{(P2)}\ \max\ R_s = f(r,n)$$
$$s.t.\quad g(r,n) \le b \tag{9}$$
$$0 \le r_i \le 1,\ n_i \quad \text{positive integer},\ 1 \le i \le m$$

In order to permit comparison with approaches proposed previously, the nonlinear constraints used by Hikita et al. [11], Kuo et al. [16] and Xu et al. [30] are used in our three example problems which include a series system, a series-parallel system and a complex (bridge) system. These three reliability-redundancy problems are formulated below:

1. Series system (Figure 1, Hikita et al.[11]). Problem (P2) becomes :

$$\text{(P2a)}\ \max\ f(r,n) = \prod_{i=1}^{m} R_i(n_i)$$

$$s.t.\quad g_1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 \le V \tag{10}$$

$$g_2(r,n) = \sum_{i=1}^{m} \alpha_i \left(-1000/\ln r_i\right)^{\beta_i} \left(n_i + \exp(n_i/4)\right) \le C \tag{11}$$

$$g_3(r,n) = \sum_{i=1}^{m} w_i n_i \exp(n_i/4) \le W \tag{12}$$

$$0 \le r_i \le 1,\ n_i \quad \text{positive integer},\ 1 \le i \le m. \tag{13}$$

2. Series-Parallel system (with $m = 5$, Figure 2, Hikita et al.[11]). Problem (P2) becomes :

$$\text{(P2b)}\ \max\ f(r,n) = 1 - \left(1 - R_1R_2\right)\left(1 - \left(1 - \left(1 - R_3\right)\left(1 - R_4\right)\right)R_5\right)$$

$s.t.$ (10)-(13).

3. Complex (bridge) system (with $m = 5$, Figure 3, Hikita et al.[11]). Problem (P2) becomes :

$$\text{(P2c)}\quad \max\ f(r,n) = R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_2R_4R_5 - R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5$$

$s.t.$ (10)-(13).

3. Genetic Algorithms

GAs are efficient search methods based on principles of natural selection and population genetics. They use randomized operators operating on a population of candidate solutions to generate a new population of candidates in the search space (Goldberg [8]). For any GA, a chromosome representation is needed to describe each individual in the population of interest. Each individual or chromosome is made up of a sequence of genes from a certain alphabet. Though the alphabet was limited to binary digits in Holland's original design [13], other very useful problem-specific representations of an individual or chromosome for function optimization have also been proposed.

GAs can search the solution space for optimal solutions very efficiently by using evaluation and genetic operator functions to maintain the useful schema in the population. For example, in a chromosome with a binary string representation of length eight, the string 101#####, where the # represents a "wild card", either 0 or 1, is a schema. Other types of schema are possible also. Individuals exhibiting a schema which results in higher fitness will have a higher probability of survival by the selection process in each generation and thereby will have a higher probability of being selected for mating and generating offspring which are likely to exhibit the same schema. (Mating is accomplished by the crossover operator function, in which the pair of mating chromosomes exchange substrings to produce a pair of offspring.) The new offspring usually include improved solutions since they tend to inherent the good schema, i.e., the good schema persist in the population over multiple generations. This has been discussed in detail by Michalewicz [18]. This section provides only a brief introduction of GAs; interested readers are referred to the excellent book by Goldberg [8].

3.1 GA Representation for Integer Reliability Problems

For a chromosome we use a vector of m integer numbers to represent the redundancy of the m subsystems. For instance, the chromosome 23341 represents a system with five subsystems, the first of which contains two elements, the second subsystem three elements, etc. The most interesting aspect of evolution (which includes reproduction, crossover and mutation etc.) is that of natural selection, which can be accomplished by the following steps:

Step 1. Randomly generate a population of chromosomes.

Step 2. Evaluate the fitness function for each individual in the population.

Step 3. If the stopping criteria have been achieved, then stop; else, go to the Step 4.

Step 4. Perform reproduction, crossover and mutation within the population.

Step 5. Form the new generation from the individuals resulting from Step 4. Go to Step 2.

A specified number of individuals in the initial population is randomly generated, after deciding upon an upper limit c_i on the number of elements in each subsystem i. (Such an upper limit can always be computed based upon the constraints, if no *a priori* estimate is available.) Typically, GAs evaluate individuals in the population by a so-called "fitness function" which is a composite of both the objective value (in this case, system reliability) and the penalty arising from the violation of constraints. In this paper, the fitness is defined as follows.

If one or more of the three constraints have been violated
Then Fitness=0
Else
Fitness=Objective value
End

The genetic operators include crossover and mutation. In Figure 4, the crossover point was randomly selected and the offspring generated by swapping the partial strings of parent 1 and parent 2. In the mutation operation, each individual's chromosome is mutated with a specified small, but positive, probability. A gene within the string is randomly selected to be mutated, and the selected gene's new value randomly selected within a range from 1 to the upper limit for redundancy of elements in the associated subsystem. For example, in Figure 4 two parents (23241 and 31422) are selected, and the third gene of the strings was selected for crossover, yielding the strings 32422 and 31241. With a specified probability, the resulting children are selected for mutation. In this case, the third gene of child 1 was selected to be mutated, and the new value, 1, was randomly generated for this gene, yielding the string 23122. That is, the number of the elements in the third subsystem has been decreased from 4 to 1 by the mutation operation.

Figure 4 goes here

3.2 GA Representation for Mixed-Integer Reliability Problems

In our implementation, each mixed-integer solution will be represented by a string of binary digits consisting of a substring for each subsystem. Each substring in turn consists of a binary substring representing the reliability of each component (real within [0,1]) and a second binary substring representing the level of redundancy (positive integer). This chromosome is illustrated in Figure 5.

Figure 5 goes here

Implementing the three standard genetic operators, namely, reproduction, crossover and mutation, requires the selection of the crossover point and mutation point for each string. If each gene has equal probability of selection, then the relative lengths of the reliability and redundancy substrings determines whether the search should focus primarily on the reliability or redundancy decisions. We have used 16 binary digits for representation of the reliability and only 8 for the redundancy, in order to direct the search primarily toward determining the component reliabilities. (The desired degree of precision for the real numbers (element reliabilities) should also guide our choice of substring lengths, of course.)

The representation of a real number in our GA approach is the same as that described by Michalewicz [18]. That is, the k binary digits for r_i represent an integer ρ_i within $[0, 2^k - 1]$ (where k determines the precision of the variable r_i) and the transformation $r_i = \rho_i / (2^k - 1)$ yields the component reliability within the interval [0,1]. A similar procedure, followed by rounding to the nearest integer (denoted by the [•] operation), yields the number of components in the subsystem, $n_i = \left[1 + v_i \left(\frac{c-1}{2^k - 1} \right) \right]$, within the interval $[1, c]$, where v_i is the integer (within $[0, 2^k - 1]$) specified by the binary substring for n_i. The three standard genetic operators, namely, reproduction, crossover and mutation are applied as described in Section 3.1.

4. Numerical Results

4.1 Results of Integer Reliability Problems

To evaluate the performance of GAs for integer reliability problems, the data used by Xu et al. [30], listed in Table 1, are used for the series system (P1a) and series-parallel system (P1b), while the data used by Hikita et al. [11], listed in Table 2, are used for the complex (bridge) system (P1c). For these problems, only the redundancy allocation (n_i) of each element i is to be decided.

Table 1. Data used in series system (P1a) and series-parallel system(P1b).(Xu et al. [30]).

i	r_i(P1a)	r_i(P1b)	$10^5\alpha_i$	$w_i v_i^2$	w_i	V	C	W
1	0.77939	0.83819	2.330	1	7			
2	0.87183	0.85507	1.450	2	8			
3	0.90288	0.87886	0.541	3	8	110	175	200
4	0.71139	0.91140	8.050	4	6			
5	0.78779	0.85036	1.950	2	9			

$\beta_i = 1.5,\ i = 1,2,3,4,5,\ m = 5.$

Table 2. Data used in complex systems (P1c) (Hikita et al. [11]).

i	r_i(P1c)	$10^5\alpha_i$	$w_i v_i^2$	w_i	V	C	W
1	0.79131	2.330	1	7			
2	0.81513	1.450	2	8			
3	0.90967	0.541	3	8	110	175	200
4	0.72061	8.050	4	6			
5	0.81942	1.950	2	9			

$\beta_i = 1.5,\ i = 1,2,3,4,5.$

Our GAs are coded in MATLAB® on an HP 715/75 workstation. The GA parameters which we have used for all three problems are as follows: the crossover probability is fixed to 0.65; the mutation probability is chosen to 0.03; the population size is set to 100; and the number of generations is fixed to 50. The comparison of solutions by GAs and best-known solutions are listed in Table 3. (The "slacks" given in the table are the unused resources in the constraints.)

Table 3. Comparison of results of GA with best-known solutions.

Problem	GAs	Slacks* of (5)-(7)	Best-known	Slacks* of (5)-(7)
P1a	n = (3,2,2,3,3) R_s = 0.93168	(27, 0.014, 7.519)	n = (3,2,2,3,3) R_s = 0.93168	(27, 0.014, 7.519)
P1b	n = (2,2,2,2,4) R_s = 0.999971	(40, 5.195, 1.609)	n = (3,3,1,2,3) R_s = 0.999969	(53, 0.000, 7.111)
P1c	n = (3,3,3,3,1) R_s = 0.99982	(18, 0.969, 4.265)	n = (3,3, 2,3,2) R_s = 0.99978	(27,0.124,10.573)
	n = (2,3,3,3,2) R_s = 0.99980	(17, 1.054, 7.519)		

We see from Table 3 that for the series system (P1a), the best solution obtained by GA (which appeared in 35 of the 50 generations) is n = (3,2,2,3,3) with system reliability = 0.93168. This coincides with the best-known solution reported by Xu et al. [30]. It is conjectured that this solution is optimal for this problem. Furthermore, in comparison with the previously best-known solutions of the series-parallel system (P1b) and the bridge system (P1c), reported by Hikita et al. [11], our GA has identified a better solution for the series-parallel system (P1b) and two better solutions for the complex system (P1c). It should also be noted that the CPU time of each problem for GA is about 25.6-28.7 seconds which is very competitive with other heuristic methods.

4.2 Results of Mixed-Integer Reliability Problems

To evaluate the performance of our GA for mixed-integer reliability problems, the three typical reliability-redundancy systems presented in Section 2.2 are optimized. The parameters for these problems duplicate those in Kuo et al. [16], Xu et al. [30] and Hikita et al. [11], as shown in Table 4 and Table 5. It should be reiterated that, for this class of problems, both the component reliability (r_i) and the redundancy allocation (n_i) are to be decided simultaneously for each subsystem i.

Table 4. Data used in series system (P2a) and complex system (P2c) (Hikita et al. [11]).

i	$10^5\alpha_i$	$w_i v_i^2$	w_i	V	C	W
1	2.330	1	7			
2	1.450	2	8			
3	0.541	3	8	110	175	200
4	8.050	4	6			
5	1.950	2	9			

$\beta_i = 1.5$, $i = 1,2,3,4,5$, $m = 5$.

Table 5. Data used in series-parallel system (P2b) (Hikita et al. [11]).

i	$10^5\alpha_i$	$w_i v_i^2$	w_i	V	C	W
1	2.500	2	3.5			
2	1.450	4	4.0			
3	0.541	5	4.0	180	175	100
4	0.541	8	3.5			
5	2.100	4	4.5			

$\beta_i = 1.5$, $i = 1,2,3,4,5$, $m = 5$.

Our GA for mixed-integer reliability problems is also implemented in MATLAB® on the HP 715/75 workstation and has used the following parameters: population size=200, mutation rate=0.85, crossover rate=0.03, and number of generations=500 for series problem but 100 for both series parallel and complex (bridge) problems. The numerical results are shown in Tables 6 through 8, in which the best three solutions of each problem are reported and compared with solutions reported previously.

Table 6 indicates that although the solutions of the series problem found by our GA are not better than the solution found by Xu et al. [30], all three are better than those found by Hikita et al. [11] and Kuo et al. [16]. Table 7 shows that each of the top three solutions of the series-parallel problem found by the GA is better than that reported by Hikita et al. [11]. Table 8 reports that the top three solutions of the bridge problem are all better than the solution reported by Hikita et al. [11]. In both the series-parallel and bridge problems, the solutions found by the GA vary significantly in the component reliabilities for subsystems.

This offers the design engineer a variety of options from which to choose with negligible differences in the system reliability.

Table 6. The comparison of best-3 solutions of GAs with others for series system (P2a).

	Best-3 solutions of GAs			Hikita et al.[11]	Kuo et al.[16]	Xu et al.[30]
n	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,3,2,3,2)	(3,2,2,3,3)
r	0.779427 0.869482 0.902674 0.714038 0.786896	0.773724 0.871969 0.906706 0.712657 0.784890	0.786442 0.869280 0.902171 0.712978 0.782296	0.777143 0.867514 0.896696 0.717739 0.793889	0.77960 0.80065 0.90227 0.71044 0.85947	0.77939 0.87183 0.90288 0.71139 0.78779
R_s	0.931578	0.931521	0.931517	0.931363	0.92975	0.93167
Slacks* (10)-(12)	27 0.121454 7.518918	27 0.087235 7.518918	27 0.040719 7.518918	27 0.000000 7.518918	27 0.000010 10.57248	27 0.013773 7.518918

*Slacks: the unused resources

Table 7. The comparison of solutions for series-parallel system (P2b).

	Best-3 solutions of GAs			Hikita et al. [11]
n	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(3,3,1,2,3)
r	0.785452 0.842998 0.885333 0.917958 0.870318	0.812575 0.823845 0.900929 0.866317 0.875843	0.778750 0.841456 0.855949 0.921236 0.874585	0.838193 0.855065 0.878859 0.911402 0.850355
R_s	0.99997418	0.99997313	0.99997260	0.99996875
Slacks* of (10)-(12)	40 1.194440 1.609289	40 2.441024 1.609289	40 0.916959 1.609289	53 0.000000 7.110849

*Slacks: the unused resources

Table 8. The comparison of solutions for complex (bridge) system (P2c).

	Best-3 solutions of GAs			Hikita et al. [11]
n	(3,3,3,3,1)	(3,3,3,3,1)	(3,3,3,3,1)	(3,3,2,3,2)
r	0.814090 0.864614 0.890291 0.701190 0.734731	0.758352 0.825402 0.881419 0.761758 0.751618	0.801863 0.856535 0.791371 0.747173 0.726372	0.814483 0.821383 0.896151 0.713091 0.814091
R_s	0.99987916	0.99985529	0.99984050	0.99978937
Slacks* of (10)-(12)	18 0.376347 4.264770	18 0.656451 4.264770	18 1.854075 4.264770	27 0.000000 10.572475

*Slacks: the unused resources

5. Conclusions

This paper considers three typical types of reliability problems, which include the series system, the series-parallel system, and the complex (bridge) system. The objective of these problems is to maximize the system reliability subject to various nonlinear constraints. Unlike most well-known heuristic methods, GAs are able to solve both integer reliability problems and mixed-integer reliability problems. Furthermore, their applicability is not limited to series-parallel systems. As shown in the previous section, the optimal solutions (except for problem P2a) by GAs are all superior to or tie the best solutions by other well-known heuristic methods for both integer reliability problems (in which component reliabilities are given and redundancy allocation is to be decided) and mixed-integer reliability problems (in which both the component reliabilities and redundancy allocation are to be decided simultaneously). In agreement with the success of numerous applications of GAs in various other classes of problems, our limited experience with these reliability problems has shown that GAs are very competitive with other heuristic methods. They are especially appropriate for design of nonstandard series-parallel systems. In addition, as reported in this paper, the multiple solutions found by the GA sometimes vary significantly in the component reliabilities and/or redundancy allocation for systems. This offers the design engineer a variety of options from which to choose with negligible differences in the system reliability.

References

[1] K.K. Aggarwal, "Minimum cost systems with specified reliability," *IEEE Trans. on Reliability*, R-26, 166-167, (1977).

[2] D.W. Coit and A.E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm", *IEEE Trans. on Reliability*, 45, (1996).

[3] D.W. Coit and A.E. Smith, "Use of a genetic algorithm to optimize a combinatorial reliability problem", *Proc. of 3rd IIE Res. Conf.*, 467-472, (1994).

[4] D.W. Coit and A.E. Smith, "Optimization approaches to the redundancy allocation problem for series-parallel systems", *Proc. of 4th IIE Res. Conf.*, 342-349, (1995).

[5] D.W. Coit and A.E. Smith, "Penalty guided genetic search for reliability design optimization", *Computers & Industrial Engineering*, 30(4), 895-904, (1996).

[6] A. J. Federowicz and M. Mazumdar, "Use of geometric programming to maximize reliability achieved by redundancy," *Operations Research*, 16, 948-954, (1968).

[7] M. Gen, T. Yokota, and T. Taguchi, "Optimal design of system reliability using genetic algorithm", *Proc. of 10th Fuzzy Sys. Symp.*, 181-184, (1994). (In Japanese)

[8] D.D. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning,* Addison Wesley, NY, (1989).

[9] K. Gopal, K.K. Aggarwal and J.S. Gupta, "A new method for solving reliability optimization problem," *IEEE Trans. on Reliability*, R-29, 36-37, (1980).

[10] K. Gopal, K.K. Aggarwal and J.S. Gupta, "An improved algorithm for reliability optimization," *IEEE Trans. on Reliability*, R-27, 325-328, (1978).

[11] M. Hikita, Y. Nakagawa, and H. Narihisa, "Reliability optimization of systems by a surrogate-constraints algorithm," *IEEE Trans. on Reliability*, R-41, 473-480, (1992).

[12] F. S. Hiller and G. J. Lieberman, *Operations Research*, McGraw-Hill, NY, (1995).

[13] J.H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, (1975).

[14] K. Ida, M. Gen, and T. Yokota, "System reliability optimization with several failure modes by genetic algorithm", *Proc. of 16th Int. Conf. on Comp. and Indust. Engng*, 348-352, (1994).

[15] T. Kohda and K. Inoue, "A reliability optimization method for complex systems with the criterion of local optimality," *IEEE Trans. on Reliability*, R-31, 109-111, (1982).

[16] W. Kuo and C.L. Hwang and F.A. Tillman, "A note on heuristic methods in optimal system reliability," *IEEE Trans. on Reliability*, R-27, 320-324, (1978).

[17] R. Luus, "Optimization of system reliability by a new nonlinear integer programming procedure," *IEEE Trans. on Reliability*, R-24, 14-16, (1975).

[18] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, Berlin-Heidelberg, (1994).

[19] K.B. Misra, "A simple approach for constrained redundancy optimization problem," *IEEE Trans. on Reliability*, R-21, 30-34, (1972).

[20] K. B. Misra and J. Sharma, "A new geometric programming formulation for a reliability problem," *International Journal of Control*, 18, 497-503, (1973).

[21] Y. Nakagawa and S. Miyazaki, "An experimental comparison of the heuristic methods for solving reliability optimization problems," *IEEE Trans. on Reliability*, R-30, 181-184, (1981).

[22] Y. Nakagawa and K. Nakashima, "A heuristic method for determining optimal reliability allocation," *IEEE Trans. on Reliability*, R-26, 156-161, (1977).

[23] L. Painton and J. Campbell, "Identification of components to optimize improvements in system reliability," *Proc. of the SRA PSAM-II Conf. on system-based methods for the design and operation of technological systems and processes* , 10.15-10.20, (1994).

[24] L. Painton and J. Campbell, "Genetic algorithms in optimization of system reliability," *IEEE Trans. on Reliability*, R-44, 172-178, (1995).

[25] K.S. Park, "Fuzzy apportionment of system reliability," *IEEE Trans. on Reliability*, R-36, 129-132, (1987).

[26] J. Sharma and K.V. Venkateswaran, "A direct method for maximizing reliability," *IEEE Trans. on Reliability*, R-20, 256-259, (1971).

[27] F.A. Tillman, C.L. Hwang, L.T. Fan, S.A. Balbale, "Systems reliability subject to multiple nonlinear constraints", *IEEE Trans. on Reliability*, R-17, 153-157, (1968).

[28] F.A. Tillman, C.L. Hwang and W. Kuo, "Determining component reliability and redundancy for optimum system reliability," *IEEE Trans. on Reliability*, R-26, 162-165, (1977).

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

[30] Z. Xu, W. Kuo and H.H. Lin, "Optimization limits in improving system reliability," *IEEE Trans. on Reliability*, R-39, 51-60, (1990).

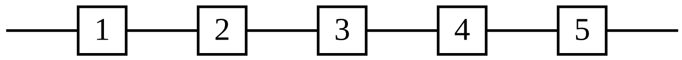

Figure 1. The series system.

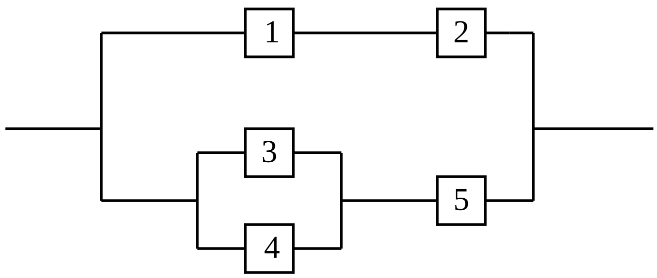

Figure 2. The series-parallel system.

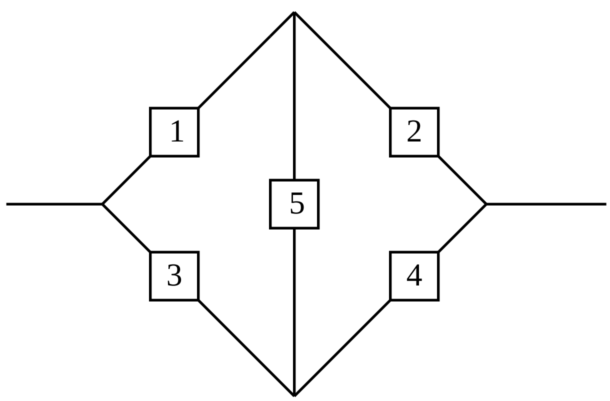

Figure 3. The complex (bridge) system.

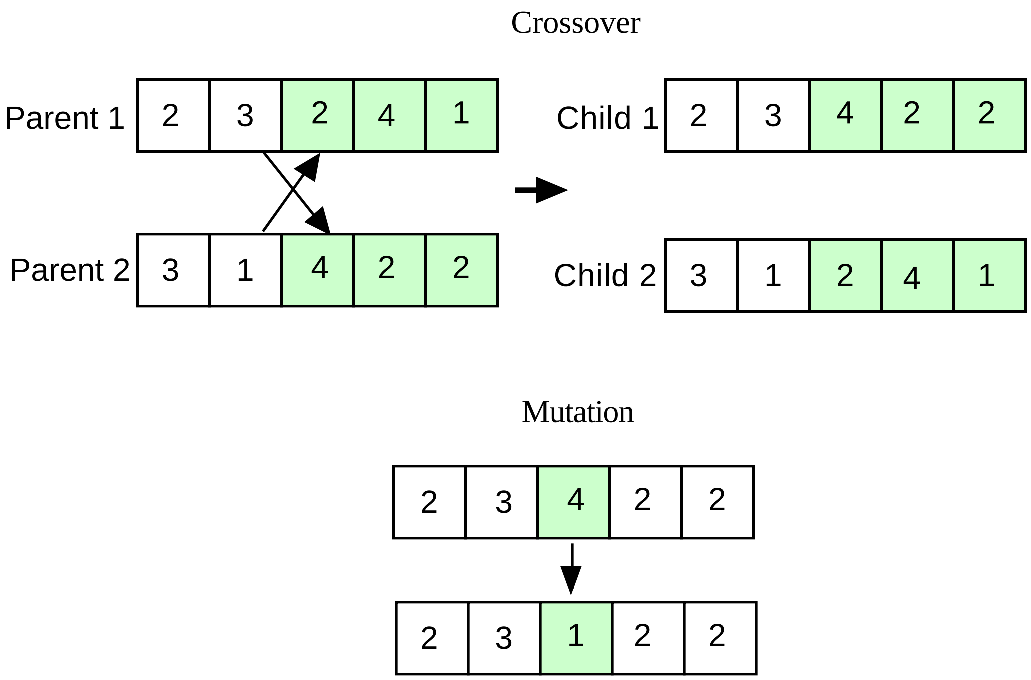

Figure 4. Crossover and Mutation for GAs.

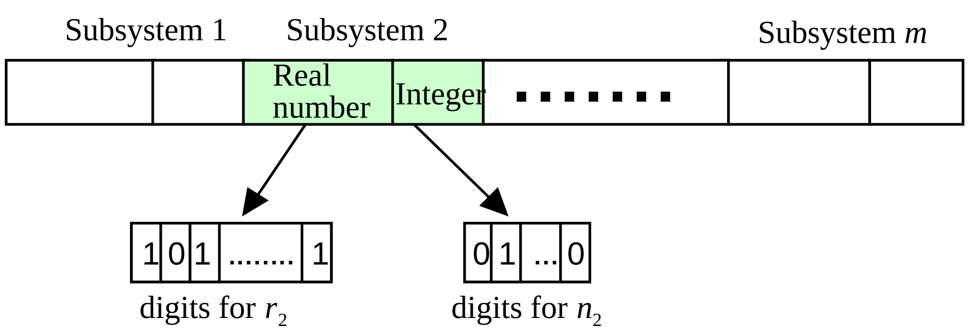

Figure 5. The solution representation of mixed-integer problem.