Available online at www.sciencedirect.com
SCIENCE DIRECT
Computers in Industry 56 (2005) 195–205
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

Immune algorithms-based approach for redundant reliability problems with multiple component choices

Ta-Cheng Chen[a,*], Peng-Sheng You[b]

[a]Department of Information Management, National Formosa University, Huwei, Yulin 632, Taiwan
[b]Graduate Institute of Transportation and Logistics, National Chia-Yi University, Chia-Yi 600, Taiwan

Received 3 March 2003; received in revised form 3 November 2003; accepted 28 June 2004
Available online 16 December 2004

Abstract

This paper considers the series–parallel redundant reliability problems in which both the multiple component choices of each subsystem and the redundancy levels of every selected component are to be decided simultaneously so as to maximize the system reliability. The reliability design optimization problem has been studied in the literature for decades, usually using mathematical programming or heuristic optimization approaches. The difficulties encountered for both methodologies are the number of constraints and the difficulty of satisfying the constraints. A penalty-guided immune algorithms-based approach is presented for solving such integer nonlinear redundant reliability design problem. The results obtained by using immune algorithms-based approach are compared with the results obtained from 33 test problems from the literature that dominate the previously mentioned solution techniques. As reported, solutions obtained by the proposed method are better than or as well as the previously best-known solutions.

Keywords: Redundant reliability problem; Immune algorithms; Optimization

1. Introduction

The system reliability optimization is very important in the real-world applications and the various kinds of systems have been studied in the literature for decades. Generally, as Misra and Sharma [1] mentioned, two main approaches are used to enhance the system reliability. One of the approaches is to increase the reliability of the elements constituted in the system, and the other is the use of redundant elements in various subsystems in the system. In the former approach, the system reliability can be enhanced to some degree, but the required enhancement of the reliability may be never attainable even though the most currently reliable elements are used. Use of the later approach is to select the optimal combination of elements and redundancy levels; the system can also be enhanced, but the cost, weight, volume, etc. will be increased as well. In addition to the above two approaches, the combination of the two approaches

* Corresponding author. Tel.: +886 5 6315740; fax: +886 5 6364127.
E-mail address: tchen@nfu.edu.tw (T.-C. Chen).

0166-3615/$ – see front matter
doi:10.1016/j.compind.2004.06.002

Nomenclature

$a_{i,j,k}$ the jth resource requirement associated with type k component of subsystem i, where $a_{i,j,k} > 0$
b_j the limitation on the jth resource
k_i the number of component choices for subsystem i, $1 \leq i \leq n$
n the number of subsystem in the system
$q_{i,k}$ the failure probability of type k component in subsystem i
$x_{i,k}$ the number of type k components in subsystem i

and reassignment of interchangeable elements are also feasible ways for increasing the system reliability [2].

Based on the above two main approaches, two main categories of reliability design problems, the integer and mixed integer problems, are investigated. The series–parallel system problem with known component reliabilities for determining the redundancy allocation belongs to integer reliability problems, in which the decision variables are constrained to integer value [3–7]. For the mixed-integer reliability problems, component reliabilities and redundancy allocation are to be decided simultaneously [1,4,8–10]. In the formulation of the series–parallel system problem considered in this paper, for each subsystem, multiple elements choices are used in parallel. The problem is then to choose the optimal combination of elements and redundancy levels to meet two constraints with cost and weight, respectively. With the known cost, reliability and weight for each element, the system design and elements selection of problem becomes a combinatorial optimization problem. Moreover, such redundancy allocation problem for series–parallel systems considered in this paper has been showed that this is an NP hard problem [11]. For solving this difficult problem, the most used integer programming techniques in literatures are generally classified into three categories that are approximate techniques, exact techniques and heuristic/meta-heuristic techniques [2,12]. The approximate techniques are such as the uses of Lagrangian multiplier and geometric programming. Kuo et al. [13] used the branch-and-bound strategy and Lagrangian multipliers, and Misra and Sharma [1] used the geometric programming for finding the nearest integers. To a problem, the exact techniques are the methods which can provide an exact optimal solution. For example, the use of dynamic programming for maximizing the system reliability with a single cost-constraint [14]. Fyffe et al. [15] used the same method to solve more difficult design problem where a system with 14 subsystems and the cost and weight constraints are considered. Furthermore, improved dynamic programming algorithm was presented by Nakagawa and Miyazaki [4] with the use of surrogate constraints for the problem with above two constraints. The heuristic techniques are the intuitive procedure for obtaining the near-optimal solutions in a reasonably short time. A majority of the recent work in the problem is devoted to developing heuristic and meta-heuristic algorithms for solving the optimal redundancy allocation problems [2]. Several heuristic methods have been suggested in literatures for the redundant allocation problems [16,17]. The meta-heuristics methods, based more on artificial intelligence than traditional mathematic programming methods, include genetic algorithms (GAs), simulated annealing, Tabu search, fuzzy optimization approach, etc. Recently, the genetic algorithm has been widely and successfully applied for solving the system reliability problem [18,5,6,10].

A new meta-heuristic optimization approach employing immune algorithms (IAs) to solve the redundant allocation problem is proposed in this paper. The merits of immune algorithms lie in pattern recognition, memorization capabilities [19] and the theory was originally proposed by Jerne [20]. Compared with other meta-heuristic approaches such as genetic algorithms and evolution strategies, the immune algorithms-based approach has very distinct characteristics: (1) the diversity is embedded by calculating the affinity and (2) the self-adjustment of the immune response is accomplished by the boost or restriction of antibody generations. These characteristics are also the advantages for solving the combinatory problems because: (1) the diversities of the feasible spaces can be better ensured, i.e., the global optimum can be more likely achieved and (2) a population of antibodies in IAs can operate simultaneously so that the possibility of paralysis in the whole process can be reduced.

This paper is arranged as follows: in the next section the series–parallel redundant reliability problem is briefly described; in Section 3, the general concept of an immune algorithms-based approach is described and numerical examples of 33 various problems are solved and discussed in Section 4. Finally, the conclusion of the paper is summarized.

2. Model description and assumptions

For integer reliability problems, both the type of component and number of the selected type of component, i.e., the redundancy allocations for each subsystem are to be decided simultaneously. The model of the series–parallel redundant reliability system with n subsystems and m separable linear constraints is considered and stated as the following integer nonlinear programming problem:

$$\max R(x|q) = \prod_{i=1}^{n}(1 - q_{i,1}^{x_{i,1}} q_{i,2}^{x_{i,2}}, \ldots, q_{i,k_i}^{x_{i,k_i}}) \quad (1)$$

$$\text{s.t.} \sum_{i=1}^{n}\sum_{k=1}^{k_i} a_{i,j,k}x_{i,k} \le b_j, \quad j = 1, 2, \ldots, m \quad (2)$$

$$x_{i,k} \in \text{non-negative integer} \quad (3)$$

It is noted that the problem generalizes the general series–parallel reliability problems when $k_i = 1$ for $i = 1, 2, \ldots, n$ [7].

In the above model of a series–parallel system problem considered in this paper, for each of n subsystems, k component choices are used in parallel. Then, the overall system is connected in series by these n subsystems with the limited resources to maximize whole system reliability. An example is shown in Fig. 1. The overall system includes 14 subsystems ($n = 14$) with weight and cost limitation are 186 and 130, respectively. The corresponding input data are described in Table 1. In Fig. 1, it shows that the first subsystem contains three components of choice 3, the second subsystem contains two components of choice 1, and so on. The reliability of the overall system is 0.9841755.

As previous investigations, the approximate techniques such as Lagrangian multiplier and geometric programming used for solving the global optimum allocation are generally time-consuming due to the complex transformation and the integer solutions are not necessarily optimal any longer. Moreover, the exact solutions for the reliability optimization problems are not necessarily desirable because it is very hard to obtain the exact solutions, and even when they are available, their utility may become marginal [2]. Because of difficulties of applying the approximate and exact techniques, a major part of the work on solving the reliability optimizations is devoted to developing heuristic/meta-heuristic algorithms. Above all, the genetic algorithms become very popular tools for solving the problem successfully [18,5,6,10]. Although genetic algorithms can be easily designed and implemented without the requirement of sophisticated mathematical treatment, the difficulties are in the determining appropriate values for the parameters. If the parameters are not assigned properly, the genetic algorithms will more likely converge to a local optimum and hard to reach the global optimum. One of the characteristics of immune algorithms-based approach mentioned in previous section, the global optimum could be more easily achieved than genetic algorithms since the diversities of the feasible spaces can be better ensured. For the above reason, the immune algorithms-based approach is applied for solving the series–parallel redundant reliability problems in this research.

3. Immune algorithms implementation

The natural immune system of all animals is a very complex system for defense against pathogenic organisms. A two-tier line of defense is in the system including the innate immune system and the adaptive immune system. The basic components are lymphocytes and antibodies [21]. The cells of the innate immune system are immediately available to combat against a wide variety of antigen without previous exposure to them. The antibody production in response to a determined infectious agent (antigen) is the adaptive immune response mediated by lymphocytes which are responsible for recognition and elimination of the pathogenic agents [22]. The cells in the adaptive system are able to develop an immune memory so that they can recognize the same antigenic stimulus when it is presented to the organism again. Also, all the antibodies are produced only in

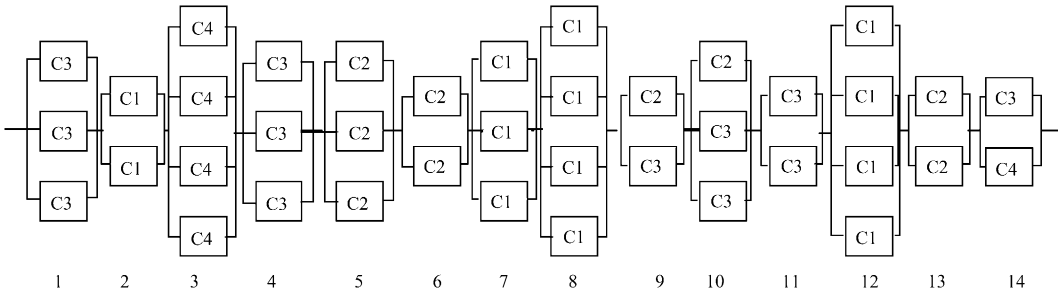

Fig. 1. A redundant reliability problem with multiple component choices.

response to specific infections. There are two main types of lymphocytes: B-lymphocytes (B-cells) and T-lymphocytes (T-cells). B-cells and T-cells carry surface receptor molecules capable of recognizing antigens. The B-cells produced by the bone marrow show a distinct chemical structure and can be programmed to make only one antibody that is placed on the outer surface of the lymphocyte to act as a receptor. The antigens will only bind to these receptors with which it makes a good fit [23].

To distinguish and eliminate the intruders of the organism is the main task of the immune system so that it must has the capability of self/non-self discrimination. As mentioned previously, various antibodies can be produced and then can recognize the specific antigens. The portion of antigen recognized by antibody is called epitope which acts as an antigen determinant. Every type of antibody has its own specific antigen determinant which is called idiotope. Moreover, in order to produce enough specific effector cells to against an infection, and activated lymphocyte has to proliferate and then differentiate into these effector cells. This process is called clonal selection [24] and followed by the genetic operations such that a large clone of plasma cell is formed. Therefore, the antibodies can be secreted and ready to bind antigens. According to above facts, Jerne [19] proposed an idiotype network hypothesis which is based on the clonal selection theory. In his hypothesis, some types of recognizing sets are activated by some antigens and produce an antibody which will then activate other types of recognizing sets. By this way, the activation is propagated through entire network of recognizing

Table 1
Component data for the example [15]

Subsystem No.	Component choices											
	Choice 1			Choice 2			Choice 3			Choice 4		
	P	*C*	*W*	*P*	*C*	*W*	*P*	*C*	*W*	*P*	*C*	*W*
1	0.90	1	3	0.93	1	4	0.91	2	2	0.95	2	5
2	0.95	2	8	0.94	1	10	0.93	1	9	*	*	*
3	0.85	2	7	0.90	3	5	0.87	1	6	0.92	4	4
4	0.83	3	5	0.87	4	6	0.85	5	4	*	*	*
5	0.94	2	4	0.93	2	3	0.95	3	5	*	*	*
6	0.99	3	5	0.98	3	4	0.97	2	5	0.96	2	4
7	0.91	4	7	0.92	4	8	0.94	5	9	*	*	*
8	0.81	3	4	0.90	5	7	0.91	6	6	*	*	*
9	0.97	2	8	0.99	3	9	0.96	4	7	0.91	3	8
10	0.83	4	6	0.85	4	5	0.90	5	6	*	*	*
11	0.94	3	5	0.95	4	6	0.96	5	6	*	*	*
12	0.79	2	4	0.82	3	5	0.85	4	6	0.90	5	7
13	0.98	2	5	0.99	3	5	0.97	2	6	*	*	*
14	0.90	4	6	0.92	4	7	0.95	5	6	0.99	6	9

sets via antigen–antibody reactions. It is noted that the antigen identification is not done by a single or multiple recognizing sets but by antigen–antibody interactions. The more details are referred to Huang [23,25]. From this point of view, for solving the combinatory optimization problems, the antibody and antigen can be looked as the solution and objection function, respectively.

3.1. Computation procedures

The computation procedures of the proposed immune algorithms-based approach illustrated in Fig. 2 work as follows and the discussion comes in sequence:

Step 1: Generate an initial population of strings (antibodies) randomly.
Step 2: Evaluate each individual in current population and calculate the corresponding fitness value for each individual.
Step 3: Select the best *n* individual with highest fitness values.
Step 4: Clone the best *n* individuals (antibodies) selected in Step 3. Note that the clone size for each select individual is an increasing function of the affinity with the antigen. In other words, the number of posterity of each antibody is proportional to their fitness values, i.e., the higher the fitness, the larger the clone size [26].
Step 5: The set of the clones in Step 4 will suffer the genetic operation process, i.e., crossover and mutation [27].
Step 6: Calculate the new fitness values of these new individuals (antibodies) from Step 5. Select those individuals who are superior to the individuals in the memory set, and then the superior individuals replace the inferior individuals in the memory set. While the memory set is updated, the individuals will be eliminated while their structures are too similar. So the individuals in the memory set can keep the diversity.
Step 7: Check the stopping criterion, if not stop then go to Step 2. Otherwise go to next step.
Step 8: Stop. The optimal or near-optimal solution(s) can be obtained from the memory set.

In our implementation, the integer solutions are represented by strings of binary digits. Each string consisting of substring includes the type of component and redundant levels for each subsystem. The details

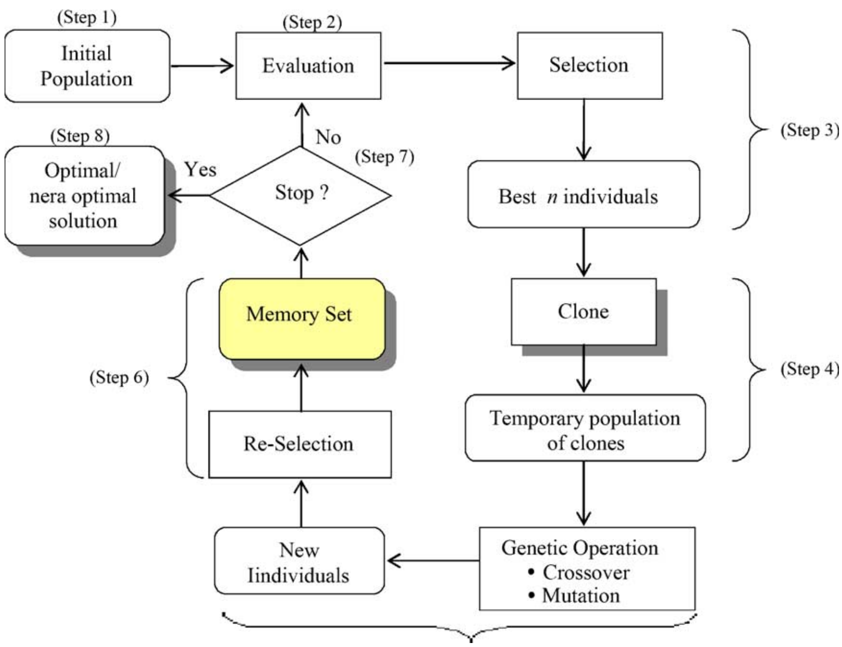

Fig. 2. The immune-based approach.

have been described in next section. In the above procedures, the clonal selection and affinity maturation processes are described in details by De Castro and Von Zuben [26]. The stopping criterion is the maximum iterations in this paper.

3.2. The representation mechanism and embodiment of diversity

The solution representation for IAs can be used in the same manner to that of genetic algorithms. In our implementation, the antibody will be represented by a binary string, each string consisting of a substring for each subsystem. Each subsystem in turn consists of a binary substring representing the type of component and the level of redundancy. A real number can be represented by a binary string and rounded to the nearest integer [28]. It is illustrated in Fig. 3.

Because of the soul of diversity in the IAs, the quality of solutions in the feasible space can be better guaranteed and obtained. So, a suppression process (diversity embodiment) is needed and shown in Step 6 in the proposed IAs procedure. In this study, for each antibody represented by a binary string can be translated into a integer string which illustrates the type of component and the corresponding redundant levels as described above. The diversity in each pair of antibody i (Ab_i) and antibody j (Ab_j) can be evaluated by calculating their affinity (f_{ij}) by following way:

$$f_{ij} = \|\text{Ab}_i - \text{Ab}_j\| \quad \text{for all } i \text{ and } j$$

While the affinity between each pair of antibodies in memory is obtained, the antibodies will be eliminated if the affinity is less than the predefined threshold. So, the diversity of the antibodies in memory is embodied. It is noted that the way of evaluating affinities of Ab–Ab and Ab–Ag are distinct. The procedure of evaluating the antibodies is to calculate the Ab–Ag affinity for each antibody that will be illustrated in the following section.

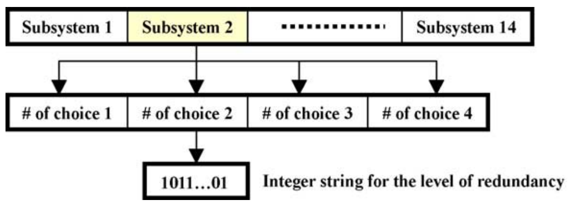

Fig. 3. Binary string represents a solution of a series–parallel reliability problem.

3.3. Constrained optimization

For breeding the superior antibodies for the next generation (iteration), to evaluate the antibody is necessary step for the immune algorithms. The goal of the algorithm is to adapt the unfeasible antibodies to the feasible antigen(s), so as to reduce the constraint violations of the search for obtaining the optimal or near-optimal solutions. Like the majority of genetic algorithms applications, for handling these constraint violations the penalty function has been defined. The penalty function increases the penalty for infeasible solutions based on the distance away from the feasible region. According to Eq. (2) in the problem formulation, the function has been defined and described as follows.

Assume the individual j within the memory of N. For each individual antibody, the constraint (2) violation value for the jth individual is defined as

$$V_j = \begin{cases} \sum_{i=1}^{n}\sum_{k=1}^{k_i} a_{i,j,k}x_{i,k} - b_j, & \text{if } \sum_{i=1}^{n}\sum_{k=1}^{k_i} a_{i,j,k}x_{i,k} > b_j \\ 0, & \text{otherwise} \end{cases}$$

Note the objective and solution are deemed as the antigen and antibody, respectively. After defining the penalty function, the fitness of each antibody to the antigen (objective) can be obtained. In other words, the affinity between each antibody and antigen is able to be determined. The affinity function (fitness function) of any Ab to Ag is described below:

$$\text{Affinity} = \frac{R(x|q)}{1 + \sum_{j=1}^{m} V_j}$$

The above affinity value is to be maximized when the penalty is minimized.

3.4. Genetic operation process

The implementation of genetic operations is the same as in genetic algorithms. It including the crossover operator and mutation operator requires the selection of the crossover point(s) and mutation point(s) for each antibody (string) under a predetermined crossover probability and mutation probability.

The crossover operator provides a thorough search of the sample space to produce good solutions. The mutation operator performs random perturbations to selected solutions to avoid the local optimum. Note the mutation rate must be small enough to avoid degrading the performance.

4. Numerical results and discussion

To evaluate the performance of our artificial immune algorithms for the integer nonlinear redundant reliability problems, 33 test problems are solved. The input data for a reliability system are described in Table 1, which includes the component choices, and the corresponding reliability of each component. The input parameters have the same values as those of Nakagawa and Miyazaki [4], Coit and Smith [5] and Hsieh [7]. These test problems based on the parameters in Table 1 are resolved with varying the available weight varied incrementally from 159 to 191 while fixing the available cost = 130. Numerical results obtained by using artificial immune algorithm are shown in Table 2, and compared with those found by Nakagawa and Miyazaki [4], Coit and Smith [5] and Hsieh [7] in Table 3. Recall that, for each problem, both the component choices and the number of the chosen component are to be decided simultaneously.

Table 2
Numerical results by artificial immune algorithm

No.	Weight	Reliability	Solution	Cost	Weight
1	191	0.9868110	333, 11, 444, 3333, 222, 22, 111, 1111, 12, 233, 33, 1111, 11, 34	130	191
2	190	0.9864161	333, 11, 444, 3333, 222, 22, 111, 1111, 11, 233, 33, 1111, 12, 34	130	190
3	189	0.9859217	333, 11, 444, 3333, 222, 22, 111, 1111, 23, 233, 13, 1111, 22, 34	130	189
4	188	0.9853297	333, 11, 444, 3333, 222, 22, 111, 1111, 13, 233, 13, 1111, 12, 34	130	188
5	187	0.9844495	333, 22, 444, 333, 222, 22, 111, 1111, 23, 233, 33, 1112, 22, 34	130	187
6	186	0.9841755	333, 11, 4444, 333, 222, 22, 111, 1111, 23, 233, 33, 1111, 22, 34	129	186
7	185	0.9834363	333, 11, 444, 333, 222, 22, 111, 1111, 23, 223, 33, 1111, 22, 34	128	185
8	184	0.9826980	333, 11, 4444, 333, 222, 22, 111, 1111, 23, 333, 33, 1111, 22, 33	129	184
9	183	0.9822062	333, 11, 444, 333, 222, 22, 111, 1111, 23, 233, 33, 1111, 22, 333	128	183
10	182	0.9815183	333, 11, 4444, 333, 222, 22, 111, 1111, 33, 333, 33, 1111, 22, 33	130	182
11	181	0.9810271	333, 11, 444, 333, 222, 22, 111, 1111, 33, 233, 33, 1111, 22, 33	129	181
12	180	0.9802902	333, 11, 4444, 333, 222, 22, 111, 1111, 33, 223, 33, 1111, 22, 33	128	180
13	179	0.9795047	333, 11, 4444, 333, 222, 22, 111, 1111, 33, 223, 13, 1111, 22, 33	126	179
14	178	0.9782085	333, 11, 444, 333, 222, 22, 33, 1111, 33, 233, 33, 1111, 22, 33	127	178
15	177	0.9772429	333, 11, 444, 333, 222, 22, 33, 133, 33, 223, 33, 1111, 22, 33	129	177
16	176	0.9766905	333, 11, 444, 333, 222, 22, 33, 1111, 33, 223, 13, 1111, 22, 33	124	176
17	175	0.9757079	333, 11, 444, 333, 222, 22, 13, 1111, 33, 223, 33, 1111, 22, 33	125	175
18	174	0.9746901	333, 11, 444, 333, 222, 22, 33, 113, 33, 223, 13, 1111, 12, 33	123	174
19	173	0.9737580	333, 11, 444, 333, 222, 22, 13, 113, 33, 233, 13, 1111, 22, 33	124	173
20	172	0.9730266	333, 11, 444, 333, 222, 22, 13, 113, 33, 223, 13, 1111, 22, 33	123	172
21	171	0.9719295	333, 11, 444, 333, 222, 22, 13, 113, 33, 222, 13, 1111, 22, 33	122	171
22	170	0.9707604	333, 11, 444, 333, 222, 22, 13, 113, 33, 222, 11, 1111, 22, 33	120	170
23	169	0.9692910	333, 11, 444, 333, 222, 22, 11, 113, 33, 222, 13, 1111, 22, 33	121	169
24	168	0.9681251	333, 11, 444, 333, 222, 22, 11, 113, 33, 222, 11, 1111, 22, 33	119	168
25	167	0.9663351	333, 11, 444, 333, 22, 22, 13, 113, 33, 222, 11, 1111, 22, 33	118	167
26	166	0.9650416	333, 11, 44, 333, 222, 22, 13, 113, 33, 222, 11, 1111, 22, 33	116	166
27	165	0.9637118	333, 11, 444, 333, 22, 22, 11, 113, 33, 222, 11, 1111, 22, 33	117	165
28	164	0.9624219	333, 11, 44, 333, 222, 22, 11, 113, 33, 222, 11, 1111, 22, 33	115	164
29	163	0.9606424	333, 11, 44, 333, 22, 22, 11, 113, 33, 222, 13, 1111, 22, 33	114	163
30	162	0.9591884	333, 11, 44, 333, 22, 22, 11, 113, 33, 222, 13, 1111, 22, 33	115	162
31	161	0.9580346	333, 11, 44, 333, 22, 22, 11, 113, 33, 222, 11, 1111, 22, 33	113	161
32	160	0.9557144	333, 11, 44, 333, 22, 22, 11, 111, 33, 222, 13, 1111, 22, 33	112	160
33	159	0.9545648	333, 11, 44, 333, 22, 22, 11, 111, 33, 222, 11, 1111, 22, 33	110	159

Note: The cost limitation is 130 for all 33 cases.

Table 3
Comparison of the proposed approach, Nakagawa and Miyazaki [4], Coit and Smith [5] and Hsieh [7] performance

No.	*W*	Nakagawa and Miyazaki			Coit and Smith			Hsieh			Chen and You			Note
		Reliability	Cost	Weight	Reliability	Cost	Weight	Reliability	Cost	Weight	Reliability	Cost	Weight	
1	191	0.9864	130	191	0.98675	130	191	0.986711	130	191	0.986811	130	191	●
2	190	0.9854	132*	189	0.98603	129	190	0.986316	130	190	0.986416	130	190	●
3	189	0.9850	131*	188	0.98556	130	189	0.985724	130	189	0.985922	130	189	●
4	188	0.9847	129	188	0.98503	130	188	0.985031	130	188	0.985330	130	188	●
5	187	0.9840	133*	186	0.98429	129	187	0.984153	129	187	0.984449	130	187	●
6	186	0.9831	129	186	0.98362	128	186	0.983879	128	186	0.984176	129	186	●
7	185	0.9829	129	185	0.98311	130	185	0.983387	127	185	0.983436	128	185	●
8	184	0.9822	126	184	0.98239	128	184	0.982204	125	184	0.982698	129	184	●
9	183	0.9815	130	182	0.98190	130	183	0.981466	124	183	0.982206	128	183	●
10	182	0.9815	130	182	0.98102	126	182	0.979690	126	182	0.981518	130	182	●
11	181	0.9800	128	181	0.98006	128	181	0.979280	125	181	0.981027	129	181	●
12	180	0.9796	126	180	0.97942	129	180	0.978327	124	180	0.980290	128	180	●
13	179	0.9792	127	179	0.97906	125	179	0.978055	123	179	0.979505	126	179	●
14	178	0.9772	123	177	0.97810	127	178	0.976878	121	178	0.978208	127	178	●
15	177	0.9772	123	177	0.97715	125	177	0.975400	122	177	0.977243	129	177	●
16	176	0.9764	125	176	0.97642	124	176	0.974975	121	176	0.976690	124	176	●
17	175	0.9744	121	174	0.97552	122	175	0.973500	122	175	0.975708	125	175	●
18	174	0.9744	121	174	0.97435	123	174	0.972328	120	174	0.974690	123	174	●
19	173	0.9723	122	173	0.97362	122	173	0.970531	119	173	0.973758	124	173	●
20	172	0.9720	123	172	0.97266	120	172	0.969232	117	172	0.973027	123	172	●
21	171	0.9700	119	170	0.97186	121	171	0.967896	118	171	0.971929	122	171	○
22	170	0.9700	119	170	0.97076	120	170	0.966776	119	170	0.970760	120	170	●
23	169	0.9675	121	169	0.96922	120	169	0.965612	117	169	0.969291	121	169	○
24	168	0.9666	120	168	0.96813	119	168	0.964150	118	168	0.968125	119	168	○
25	167	0.9656	117	167	0.96634	118	167	0.962990	116	167	0.966335	120	167	○
26	166	0.9646	116	166	0.96504	116	166	0.961210	115	166	0.965042	116	166	○
27	165	0.9621	118	165	0.96371	117	165	0.959923	113	165	0.963712	117	165	○
28	164	0.9609	116	164	0.96242	115	164	0.958601	114	164	0.962422	115	164	○
29	163	0.9602	114	163	0.96064	114	163	0.957317	112	163	0.960642	114	163	○
30	162	0.9589	112	162	0.95912	114	162	0.955547	111	162	0.959188	115	162	●
31	161	0.9565	111	161	0.95803	113	161	0.954101	112	161	0.958035	113	161	○
32	160	0.9546	110	159	0.95567	114	160	0.952953	110	160	0.955714	112	160	●
33	159	0.9546	110	159	0.95432	110	159	0.950800	108	159	0.954565	110	159	○

Note: ● represents the best solution found is superior than the solution found in literature; ○ represents the best solution found is as well as the solution found in literature.

Our artificial immune algorithm is implemented in MATLAB® on the Pentium 42.0 GHz PC with the following parameters: memory size = 120, mutation rate = 0.01, crossover rate = 0.86 and the maximum clone number = 10. Then number of generations was specified to be 3000. The determination of immune algorithm's parameters is a significant problem for the immune algorithm implementation. However, there is no any formal methodology to solve the problem because different value-combinations of the parameters result to different characteristics as well as different performance of immune algorithms. Therefore, one should note that the best values for the artificial immune algorithm parameters are case-dependent and based upon the experience from preliminary runs.

The numerical results in Table 2 reports the detailed solutions obtained by the proposed approach for each test problem. Also, they are compared with those of Nakagawa and Miyazaki [4], Coit and Smith [5] and Hsieh [7] in Table 3.

The results in Table 3 indicate that:

- compared with those of Nakagawa and Miyazaki [4], 32 solutions (1–32) obtained by immune algorithms-based approach are superior than those found by Nakagawa and Miyazaki [4]. The solution found in the 33rd test problem by both approaches is the same.

- compared with those of Coit and Smith [5], the proposed approach finds better solutions for 24 out of 33 test problems. The solutions of the left nine obtained by proposed approach are as well as those obtained by those obtained by genetic algorithms [5].
- compared with those of Hsieh [7], it is seen that the solutions found by our approach in all test problems are superior than those found by Hsieh [7].

The comparison of numerical results for 33 test problems with those in literatures is depicted in Fig. 4. In the figure, three lines illustrates this observation of comparisons, where

$$L_1 = \frac{R - R_{\mathrm{NM}}}{1 - R_{\mathrm{NM}}}, \quad L_2 = \frac{R - R_{\mathrm{CS}}}{1 - R_{\mathrm{CS}}} \quad \text{and} \quad L_3 = \frac{R - R_{\mathrm{Hsieh}}}{1 - R_{\mathrm{Hsieh}}}$$

The definition of the lines as above indicates the maximum possible improvement (MPI) which is the fraction that the best feasible solution achieved of the maximum possible improvement, considering that reliability ≤ 1 [5]. Herein, R is the reliability by the proposed IAs approach, $R_{\mathrm{N\&M}}$ the reliability by Nakagawa and Miyazaki [4], $R_{\mathrm{C\&S}}$ the reliability by Coit and Smith [5] and R_{Hsieh} the reliability by Hsieh [7]. According to the comparison of numerical results in Table 3 and Fig. 3, it shows that the proposed IAs approach performs better in those test problems with larger values of W. In general, the immune algorithms-based approach find better solutions for 24 test problems (W = 160, 162, 169 and 171–191), and tie the well-known best solutions found by other methods in the above three literatures.

Although the immune algorithm found better solutions of 24 out of 33 test problems, the improvement was extremely tiny, for instance: in test problem 32 where the difference is on the order of 10^{-5}. The differences are probably insignificant given the possible lack of precision in the constraint parameters such as in test problems 24 and 25. However, in all 33 problems, then, one could say that immune algorithms did find solutions of quality comparable to those previously published in the literature. Above all, compared with the solutions found by Nakagawa and Miyazaki [4] and Hsieh [7], the solutions found by proposed method are with more significant improvement. Nevertheless, the solution comparison between the proposed method and genetic approach [5] shows the improvement is small (less than 5%). It has to be emphasized that even very small improvements in reliability are often hard to be obtained in high reliability applications. Moreover, besides the solutions found by the proposed approach, no any of the other three approaches can dominate any other two methods.

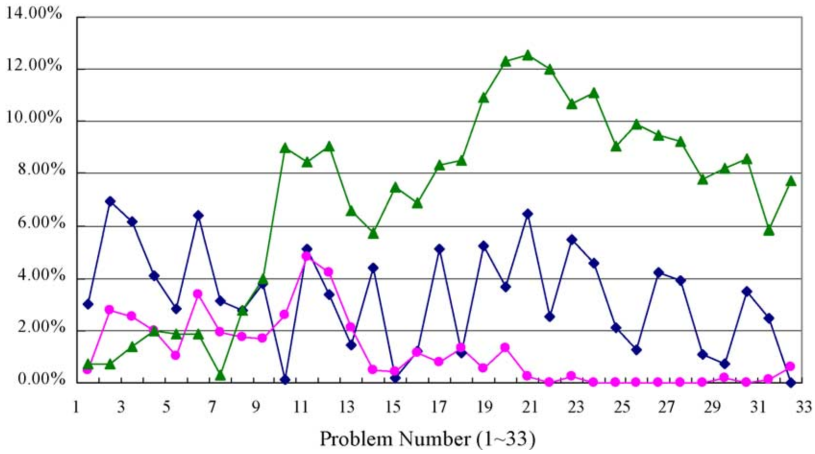

Fig. 4. The comparison of numerical results for 33 test problems. $L_1 = \frac{R-R_{\mathrm{NM}}}{1-R_{\mathrm{NM}}}$ (symbol ◆), $L_2 = \frac{R-R_{\mathrm{CS}}}{1-R_{\mathrm{CS}}}$ (symbol ●) and $L_3 = \frac{R-R_{\mathrm{Hsieh}}}{1-R_{\mathrm{Hsieh}}}$ (symbol ▲).

It seems that GAs and IAs are very similar, but there are an essential difference in the memory adopting system and the production system of various antibodies. It allows the global optimum to be acquired by using this algorithms form many optimization problems. The main reason is that the IA's diversity characteristic in memory makes the proposed approach with more probability search the global optimal solution. However, the above merit of the IAs may become its disadvantage while the CPU time is taken into account. Compared with GAs, the memory-adopting process in IAs will take slightly longer CPU time for each iteration. Although more CPU time will be taken in IAs than in GAs, it is still worth to do so since obtaining a system design with higher reliability is very difficult and important in the real-world applications.

According to above observation, it can be concluded that the performance of the proposed approach are superior than the other three methods when used to find the maximum reliability for these redundant reliability problems with multiple component choices (CPU time is ignored).

5. Conclusions

The IA-based approach to the series–parallel redundant reliability subject to multiple separable linear constraints is proposed. Unlike genetic algorithms, immune algorithms based approach preserves diversity so that it is able to discover new optima over time. Therefore, the convergence of immune algorithms-based approach is never completed and this diversity acts like a preventive measure. This notion of viability of enabling further adaptations is precisely what genetic algorithms were lacking and this may become the reason why immune algorithms-based approach provides superior solution than genetic algorithms-based approaches do. The IAs-based approach has been applied to solve the combinatory optimization engineering problems but the problem solved in this research is different than those separated in the literature, since the type of component and the component redundant levels are to be decided simultaneously for the system optimization problem. To deal with this difficulty, a solution representation and special solution procedures are proposed. Based on our limited experience, it suggests that the IAs-based approach finds solutions which are of a quality and are comparable to that of other heuristic algorithms while the CPU time is ignored. The proposed method achieves the global solution or finds a near-global solution for each problem tested.

Acknowledgments

The research is supported by grants from National Science Council, Taiwan, under contract NSC 93-2213-E-150-012. The authors also thank a number of anonymous referees for their valuable comments and useful suggestions.

References

[1] K.B. Misra, J. Sharma, A new geometric programming formulation for a reliability problem, International Journal Control 18 (1973) 497–503.

[2] W. Kuo, V. Rajendra Prasad, An annotated overview of system-reliability optimization, IEEE Transaction on Reliability 49 (2) (2000) 176–187.

[3] F.S. Hiller, G.J. Lieberman, An Introduction to Operations Research, McGraw-Hill, New York, 1995.

[4] Y. Nakagawa, S. Miyazaki, Surrogate constraints algorithm for reliability optimization problems with two constraints, IEEE Transaction on Reliability R-30 (1981) 175–180.

[5] D.W. Coit, A.E. Smith, Reliability optimization of series–parallel systems using a genetic algorithm, IEEE Transaction on Reliability 45 (1996) 254–260.

[6] D.W. Coit, A.E. Smith, Penalty guided genetic search for reliability design optimization, Computers and Industrial Engineering 30 (1996) 895–904.

[7] Y.C. Hsieh, A linear approximation for redundant reliability problems with multiple component choices, Computers and Industrial Engineering 44 (2003) 91–103.

[8] W. Kuo, C.L. Hwang, F.A. Tillman, A note on heuristic methods in optimal system reliability, IEEE Transaction on Reliability R-27 (1978) 320–324.

[9] Z. Xu, W. Kuo, H.H. Lin, Optimization limits in improving system reliability, IEEE Transaction on Reliability R-39 (1990) 51–60.

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

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

[12] K.B. Misra, U. Sharma, An efficient algorithm to solve integer-programming problems arising in system-reliability design, IEEE Transaction on Reliability 40 (1) (1991) 81–91.
[13] W. Kuo, H. Lin, Z. Xu, W. Zhang, Reliability optimization with the Lagrange multiplier and branch-and-bound technique, IEEE Transaction on Reliability R-36 (1987) 624–630.
[14] R.E. Bellman, E. Dreyfus, Dynamic programming and reliability of multicomponent devices, Operations Research 6 (1958) 200–206.
[15] D.E. Fyffe, W.W. Hines, N.K. Lee, System reliability allocation and a computational algorithm, Operations Research 17 (1968) 64–69.
[16] J. Sharma, K.V. Venkateswaran, A direct method for maximizing the system reliability, IEEE Transaction on Reliability R-20 (1) (1971) 256–259.
[17] P.M. Ghare, R.E. Taylor, Optimal redundancy for reliability in series system, ORSA 17 (1969) 838–847.
[18] L. Painton, J. Campbell, Genetic algorithms in optimization of system reliability, IEEE Transaction on Reliability 44 (1995) 172–178.
[19] N.K. Jerne, The immune system, Scientific America 229 (1) (1973) 52–60.
[20] N.K. Jerne, Clonal selection in lymphocyte network, in: G.M. Edelman (Ed.), Cellular Selection and Regulation in the Immune Response, Raven Press, New York, 1974.
[21] J.D. Farmer, N.H. Packard, A.S. Perelson, The immune system, adaptation, and machine learning, Physica D 22 (1986) 187–204.
[22] L.N. De Castro, J. Timmis, Artificial Immune Systems: A new Computational Intelligence Approach, Springer, New York, 2002.
[23] S.J. Huang, Enhancement of thermal unit commitment using immune algorithms based optimization approaches, Electrical Power and Energy Systems 21 (1999) 245–252.
[24] I.L. Weissman, M.D. Cooper, How the immune system develops, Scientific American 269 (3) (1993) 33–40.
[25] S.J. Huang, An immune-based optimization method to capacitor placement in a radial distribution system, IEEE Transaction on Power Delivery 15 (2) (2000) 744–749.
[26] L.N. De Castro, F.J. Von Zuben, The clonal selection algorithm with engineering applications, in: Workshop Proceedings of the GECCO 2000, #####, 2000, pp. 36–37.
[27] Z. Michalewicz, Genetic algorithm + Data structures = Evolution Programs, Springer-Verlag, Berlin, 1994.
[28] T.-C. Chen, G.W. Fischer, A GA-based method for the tolerance allocation problem, Artificial Intelligence in Engineering 14 (2000) 133–141.

Ta-Cheng Chen is currently an Associate Professor in the Department of Information Management at National Formosa University, Taiwan. He received his Ph.D. in industrial engineering from the University of Iowa in 1997. His research interests include AI approaches applied in optimization problems, data mining and applied operations research.

Peng-Sheng You is an Associate Professor at the Graduate Institute of Transportation and Logistics in the National Chia-Yi University. He received his Ph.D. in management science and engineering from University of Tsukuba. His research interests include yield management, inventory management and system reliability.