International Journal of Industrial Engineering, 18(3), 120-129, 2011.

A TABU SEARCH FOR MULTIPLE MULTI-LEVEL REDUNDANCY ALLOCATION PROBLEM IN SERIES-PARALLEL SYSTEMS

Kil-Woong Jang and Jae-Hwan Kim

Department of Data Information, Korea Maritime University, Busan, 606-791, South Korea

Corresponding author: Jae-Hwan Kim, jhkim@hhu.ac.kr

The traditional RAP (Redundancy Allocation Problem) is to consider only the component redundancy at the lowest-level. A system can be functionally decomposed into system, module, and component levels. Modular redundancy can be more effective than component redundancy at the lowest-level. We consider a MMRAP (Multiple Multi-level Redundancy Allocation Problem) in which all available items for redundancy (system, module, and component) can be simultaneously chosen. A tabu search (TS) of memory-based mechanisms that balances intensification with diversification via the short-term and long-term memory is proposed for its solution. To the best of our knowledge, this is the first attempt to use a TS for MMRAP. Our algorithm is compared with the existing genetic algorithm(GA) for MMRAP on the new composed test problems as well as the benchmark problems in the literature. Computational results show that the TS outstandingly outperforms the GA for all test problems.

Significance: To evaluate the performance of metaheuristic approaches for the MMRAP, the existing GA was coded in C/ C++ programming language and a TS algorithm was developed. From computational results, we noticed that the proposed TS substantially outperformed the existing GA.

Keywords: tabu search, module, component, redundancy allocation, reliability optimization.

(Received 1 May 2010; Accepted in revised form 27 Feb 2011)

1. INTRODUCTION

The reliability of a system can be increased by properly allocating redundant units to subsystems under various resource and technological constraints. A well-known Redundancy Allocation Problem (RAP) is to determine the optimal number of redundant component in order to maximize the system reliability constrained to resource restrictions or system-level constraints for cost and weight, etc. The RAP has become an important issue for system designs such as semiconductor integrated circuits, nanotechnology, and most electronic systems. The RAP is typically classified into the five categories such as series, parallel, series-parallel, parallel-series, and complex systems. In this study, we only focus on the series-parallel systems. Solutions for the series-parallel RAP have been suggested by many authors. Fyffe *et al*. (1968) originally set up the problem and suggested a solution algorithm utilizing a dynamic programming approach. Nakagawa and Miyazaki (1981) developed 33 variations of Fyffe's problem, where the weight constraint varied its value from 159 to 191. Coit and Liu (2000) proposed zero-one integer programming for small size of problems. They constrained the solution space so that only the identical component type can be allowed for each subsystem.

On the contrary, Coit and Smith (1996) extended Fyffe's problem in such a way that the parallel system could be more flexible. They allowed a mixing of component types within a subsystem and employed a genetic algorithm to obtain optimal solutions. Better solutions have been presented by tabu search (Kulturel-Konak *et al*., 2004), ant colony optimization (Liang & Smith, 2004), variable neighborhood search (Liang & Chern, 2007), and hybrid metaheuristics (Nahas *et al*., 2007; Ouzineb *et al*., 2010). The above all methods proposed for only best solutions for 14-subsystem problems (Coit & Smith, 1996) without referring to their global optimal solutions. However, Kim and Kim (2006) suggested the global optimal solutions for these problems by the transformation of binary integer programming, and also showed that all solutions suggested by Kulturel-Konak *et al*. (2004) for the 14-subsystem problems were the global optimum. For lager problems with up to 56-subsystem, Bae *et al*. (2007) compared metaheuristics with global optimal solutions.

In the meanwhile, Yun and Kim (2004) proposed a new kind of RAP, that is, MRAP (Multi-level RAP). The traditional RAP is to consider only the component redundancy at the lowest-level. A system can be functionally decomposed into system, module, and component levels. Modular redundancy can be more effective than component redundancy at the lowest-level. MRAP is to consider all cases of the redundancy for system, module, and component levels. MRAP, however, has a strong restriction that only one level among component, module and system can be an alternative for redundancy in RAP. Yun *et al*. (2007) extended MRAP to the multiple MRAP (MMRAP) in which all available items for redundancy (system, module, and component) can be simultaneously chosen, by relaxing the strong restriction of MRAP, and also proposed a simple GA in order to obtain the optimal solution for MMRAP. In this paper, we propose a TS of memory-based mechanisms, balancing intensification with diversification, for MMRAP. To the best of our knowledge, this is the first attempt to use a TS for MMRAP. Our TS is also compared with the existing GA (Yun *et al*., 2007) for MMRAP on the new composed test problems as well as the benchmark problems in the literature. Computational results show that the TS outstandingly outperforms the GA (Yun *et al*., 2007) for all test problems.

The rest of this paper is organized as follows: in section 2 we describe the MMRAP. In section 3, we propose a TS for MMRAP. In section 4 and 5, numerical examples and computational results for the performance of TS are provided with the benchmark problems. Finally, conclusions are discussed in section 6.

2. MULTIPLE MULTI-LEVEL REDUNDANCY ALLOCATION PROBLEM

2.1 Acronym

RAP	redundancy allocation problem
MRAP	Multi-level RAP
MMRAP	multiple MRAP
GA	genetic algorithm
TS	tabu search
TSMMRAP	tabu search for MMRAP

2.2 Notations

n	the number of basic items
m	the number of constraints
x_i	the number of redundancy allocated to the *i*th basic item
$R(x)$	the system reliability
$R_p(x)$	the penalty function of the system reliability
$g_j(x)$	the *j*th constraint function
r_i	the reliability of the *i*th basic item
c_i	the cost of the *i*th basic item
b_j	the amount of allowable resource *i*
l_i	the lower bound of x_i
u_i	the upper bound of x_i

2.3 Assumptions

In MMRAP, the number of available redundancy structure increases exponentially as the size of problem becomes large. In this paper, two assumptions are set up to exclude impossible redundancy structures (Yun *et al*., 2007).

(1) The combination of the basic items for redundancy should satisfy the function at the system level. If a basic item is used, all its sibling items should be used or its function should be satisfied by corresponding child items.

(2) The basic items for redundancy should be used in parallel at one combination.

2.4 Problem Formulation

The MMRAP optimization model can be generally formulated as the following nonlinear integer programming problem:

Maximize $R(x)$ ... (1)

subject to $g_j(x) \le b_j$ for j=1, 2, …, m

$l_i \le x_i \le u_i$

x_i is a positive integer number for i =1,2, …, n

This problem was proven to be a NP-hard problem (Chern, 1992). We refer interested readers to Yun *et al*. (2007) for the detailed formulation of MMRAP.

3. TS ALGORITHM

The TS algorithm, first proposed by F. Glover (1986), is a metaheuristic method to expand its search beyond local optimality using adaptive memory. The adaptive memory is a mechanism based on the tabu list of prohibited moves. The tabu list is one of the mechanism to prevent cycling and guide the search towards unexplored region of the solution space. The TS generally adopts the penalty function to allow to explore the search towards the attractive infeasible region.

The TS has been successfully applied to many combinatorial optimization problems such as vehicle routing problems, travelling salesman problems, time tabling problems, and resource allocation problems, etc. In RAP, Kultruel-Konak *et al.* (2004) developed the TS algorithm for the RAP of series-parallel system, and it is hitherto known that their solutions are the best among the other metaheuristics such as genetic algorithm (Coit & Smith, 1996), ant colony optimization (Liang & Smith, 2004), variable neighborhood search (Liang & Chern, 2007), and hybrid metaheuristics (Nahas *et al*., 2007). Recent hybrid metaheuristic (Ouzineb *et al*., 2010) obtained the same results of Kultruel-Konak *et al.* (2004).

In this paper, we develop a kind of TS called TSMMRAP which is based on the TS of Kultruel-Konak *et al* . (2004). The TSMMRAP consists of 4 parts which are construction of initial solutions, the tabu list, the penalty function, and the structure of generating the neighborhood solutions. Among them, the structure of generating the neighborhood solutions is usually different according to the characteristics of the decision variables for the problem.

The TSMMRAP can then be summarized as follows:

Step 0. Generate randomly an initial feasible solution.

Step 1. Explore all the neighborhood solutions for the defined moves.

Step 2. If the best solution of the step 1 is in the tabu list and is not better than the current best,
then repeat step 1 to find the next best. Otherwise, select the solution as the next best move.

Step 3. If the stopping criterion is satisfied, then stop. Otherwise, go to step 1.

The initial solutions are randomly generated, and the scheme of randomly constructing the initial solutions is nearly identical to the general scheme. In our experiments, we try to find the optimal solution 10 times with different initial solutions for each problem. The stopping criterion of TSMMRAP was defined as 500 iterations without finding an improvement in the best feasible solution. For the size of a tabu list, Kultruel-Konak *et al*. (2004) showed that the dynamic size of the tabu list plays a critical role in finding the better solutions for RAP. In our experiment, the size of the tabu list is reset every 20 iterations to the value of between [n, 3n] uniformly distributed for the long-term memory. Once the list is full, the oldest element of the tabu list should be removed as a new one is added.

The TS generally adopts the penalty function to allow to explore the search towards the promising infeasible region. The the same type of the penalty function of Kultruel-Konak *et al*. (2004) is adopted in our TSMMRAP. The penalty function for only one cost constraint is as follows:

$$R_p(x) = R(x) - \left(R_{all} - R_{feas}\right)\left(\frac{\Delta c}{NFT_c}\right)^k \quad \cdots \quad (2)$$

where R_{all} is the unpenalized (feasible or infeasible) system reliability of the best solution found so far, R_{feas} is the system reliability of the best feasible solution found so far, and Δc represent the magnitude for the violation of the cost constraint. The value of NFT_c is used to control the range of explored infeasible region. Namely, If the current move is feasible, NFT_c has a property of encouraging to search the neighborhood solutions towards the infeasible region by decreasing the penalized value of (2). Reversely, if the current move is infeasible, NFT_c is controlled to increase the penalized value for exploring the search towards the feasible region. The initial value of NFT_c is set to 1% of the constraint limit for the cost and *k* is set to 1, though computational results are insensitive to these values.

In TSMMRAP, the structure of the neighborhood solutions is generated by simple two moves. The first move is to add one for all the allocated redundant number of the module or component. The second move is to subtract one for all the allocated number of the module or component. For example, let us consider the following system structure of multi-level in Table 1. The cost function is $g(x) = c_i x_i + \lambda_i^{xi}$ for *i*=1, 2,…, *n.* The constraint limit for the cost is set to 95. The system structure is shown in Fig. 1. Let us assume the current solution during the iterations of TSMMRAP to obtain the optimal solution for our problem to be given in Table 2, that is, the encoding status of (0, 2, 1, 1, 1, 1, 1). From the current solution, all possible neighborhood solutions by the first and second move are 7 and 5 cases shown in Table 2, respectively. Specifically, when the value of any component or module equals to 1, the second move is somewhat complicated. Namely, if the value of any component is 1, then the values of all components in the same subsystem is set to zero and the value of the module in the same subsystem is compensated to the appropriate values to maintain the feasible solution, and vice versa. For example, in Table 2, when the value of component (112) is 1, it is replaced with the appropriate values (1 or 2) of module (11). That is, the redundant value of module (11) can be assigned up to the maximum redundant value of all components ((111), (112), and (113)). Reversely, when the value of module (12) equals to 1, it is replaced with the value of all components added to each component by 1, (121, 122) = (2, 2). Namely, it is allowed to add the redundant value of all components up to the redundant value of module (12).

Table 1. The input data for an illustration

Basic Item	Parent unit	r_i	c_i	λ_i
1(system)	-	0.40029	72	2
11	1	0.72675	26	2
12	1	0.76500	19	3
111	11	0.90000	5	3
112	11	0.95000	6	4
113	11	0.85000	5	4
121	12	0.90000	6	4
122	12	0.85000	7	4

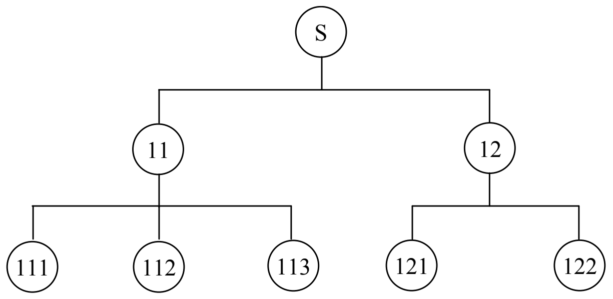


Figure 1. The system structure of Table 1

In Table 2, the system reliability for each neighborhood solution is evaluated by the penalty function of the equation (2). The neighborhood solution (1, 2, 1, 1, 1, 1, 1), which has the maximum value of 0.8930 among the 12 cases, is selected for the next best move in our TSMMRAP.

Table 2. All possible neighborhood solutions by two moves

	x=(11, 111, 112, 113, 12, 121, 122)	$g(x)$	$R_p(x)$
current solution	(0, 2, 1, 1, 1, 1, 1)	81	0.7553
first move	(1, 2, 1, 1, 1, 1, 1)	109	0.8930
	(0, 3, 1, 1, 1, 1, 1)	104	0.7621
	(0, 2, 2, 1, 1, 1, 1)	99	0.7930
	(0, 2, 1, 2, 1, 1, 1)	98	0.8686
	(0, 2, 1, 1, 2, 1, 1)	106	0.7891
	(0, 2, 1, 1, 1, 2, 1)	99	0.7696
	(0, 2, 1, 1, 1, 1, 2)	100	0.7768
second move	(0, 1, 1, 1, 1, 1, 1)	70	0.6866
	(1, 0, 0, 0, 1, 1, 1)	71	0.6866
	(2, 0, 0, 0, 1, 1, 1)	99	0.8742
	(0, 2, 1, 1, 0, 2, 2)	96	0.7736
	(0, 2, 1, 1, 2, 0, 0)	85	0.7553

4. NUMERICAL EXAMPLES

We consider two examples to evaluate the performance of the existing GA (Yun *et al*., 2007) and the proposed TSMMRAP. Two algorithms are coded in C/C++ programming language, and the numerical experiments are executed on an IBM-PC

Table 3. The input data for the first example

Basic Item	Parent unit	r_i	c_i	λ_i
1(system)	-	0.4003	72	2
11	1	0.7268	26	2
12	1	0.7650	19	3
13	1	0.7200	21	2
111	11	0.9000	5	3
112	11	0.9500	6	4
113	11	0.8500	5	4
121	12	0.9000	6	4
122	12	0.8500	7	4
131	13	0.9000	8	3
132	13	0.8000	7	4

compatible with a Pentium IV 3.0 GHz. Two benchmark examples are taken from Yun *et al.* (2007). The data for the first and the second example are shown in Table 3 and 4, respectively. The first example consists of 11 basic items and 20 test problems varying the cost limit from 150 to 340. The second example consists of 16 basic items and 10 test problems varying the cost limit from 260 to 350.

For the two examples, we compared the TSMMRAP with the existing GA (Yun *et al*., 2007). Computational results for the first and second example are shown in Table 5 and 6, respectively. As the results in Table 5 indicate, GA and TSMMRAP obtain the same results for 20 test problems for the first example. For larger size of the second example, however, TSMMRAP obtains the better optimal solutions for 7 cases of 10 test problems than the GA (Yun *et al*., 2007).

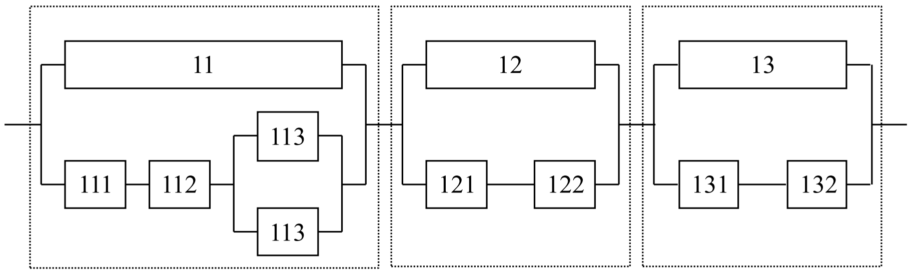
In addition, to show the difference between MMRAP and MRAP, the system structure of the optimal solution for MMRAP and MRAP for the cost limit of 160 of the first example are shown in Fig. 2(a) and Fig. 2(b), respectively. In this case, the system reliability for MMRAP and MRAP are given by 0.8316 and 0.8309, respectively. This result indicates that MMRAP considering the redundancy for the module and component simultaneously, provides the possibility of having the better system reliability than MRAP.

Table 4. The input data for the second example

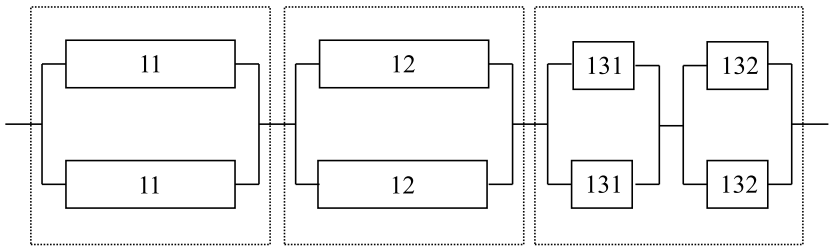
Basic Item	Parent unit	r_i	c_i	λ_i
1(system)	-	0.3338	130	2
11	1	0.7650	15	2
12	1	0.8572	32	3
13	1	0.7650	14	2
14	1	0.8200	25	2
15	1	0.8114	24	3
111	11	0.9000	6	3
112	11	0.8500	5	3
121	12	0.9600	8	5
122	12	0.9500	7	4
123	12	0.9400	6	3
131	13	0.9000	5	3
132	13	0.8500	4	3
151	15	0.9200	7	4
152	15	0.9000	5	3
153	15	0.9800	3	3

Table 5. Computational results for the first example

No.	Cost limit	Metaheuristic methods	
		GA	TSMMRAP
1	150	0.8057	0.8057
2	160	0.8316	0.8316
3	170	0.8576	0.8576
4	180	0.8773	0.8773
5	190	0.8920	0.8920
6	200	0.9136	0.9136
7	210	0.9319	0.9319
8	220	0.9457	0.9457
9	230	0.9535	0.9535
10	240	0.9587	0.9587
11	250	0.9641	0.9641
12	260	0.9694	0.9694
13	270	0.9739	0.9739
14	280	0.9773	0.9773
15	290	0.9808	0.9808
16	300	0.9835	0.9835
17	310	0.9861	0.9861
18	320	0.9888	0.9888
19	330	0.9903	0.9903
20	340	0.9918	0.9918



(a) MMRAP



(b) MRAP

Figure 2. The system structure for MMRAP and MRAP

Table 6. Computational results for the second example

No.	Cost limit	Metaheuristic methods	
		GA	TSMMRAP
1	260	*0.8632	*0.8632
2	270	*0.8780	*0.8780
3	280	0.8939	*0.8955
4	290	0.9021	*0.9129
5	300	0.9165	*0.9188
6	310	0.9221	*0.9248
7	320	0.9291	*0.9380
8	330	0.9440	*0.9450
9	340	*0.9502	*0.9502
10	350	0.9546	*0.9554

*: the case of obtaining the best solution

5. COMPUTATIONAL RESULTS

To additionally evaluate the performance of the existing GA (Yun *et al*., 2007) and the TSMMRAP, the new test problems for lager size of system structure are designed. They are composed of combining the first example with the second example. The resulting data for the composed problem is shown in Table 7. The system structure is shown in Fig. 3. This problem consists of 26 basic items and the system cost constraints ranges from 700 to 880 in step size of 20. They are composed of 10 sets of 10 test problems. Totally, 100 test benchmark problems are designed.

Two algorithms are coded in C/C++ programming language, and experiments are performed on a Pentium IV 3.0 GHz PC. Performances of GA and TSMMRAP are assessed in terms of average relative error (A), maximum relative error (M), optimality rate (O) and average execution time (sec.) of 100 problems (T) defined as follows.

$$A = \frac{1}{10}\sum_{j=1}^{10}\frac{\left(R_j^* - R_j\right)}{R_j^*}$$

$$M = max\left\{\frac{\left(R_j^* - R_j\right)}{R_j^*}\right\}, \text{ for } j=1, 2 \ldots, 10$$

O = the number of times (out of 10 problems) that each method yields the best solution.

R_j = the system reliability obtained by each method for each test problem j.

R_j^* = the best system reliability obtained by both of GA and TSMMRAP.

In our experiments, the stopping criterion of TSMMRAP was defined as 500 iterations without finding an improvement in the best feasible solution, and TSMMRAP was applied 10 times with different starting initial solutions. The computational results for evaluating the performance between the GA (Yun *et al*., 2007) and TSMMRAP are summarized in Table 8. As the results in Table 8 indicate, TSMMRAP outstandingly outperformed the GA. TSMMRAP obtained the higher system reliability for all test problems than the GA. The GA failed to obtain the best solution for all cases of 100 test problems. Even for the best case of cost limit (880), the average relative error (A) of GA is very poor, scoring the value of 0.0299. In particular, for the case of cost limit (700), GA has the worst value of maximum relative error (M=0.042). In terms of computing time, however, we noticed that TSMMRAP requires almost 2 times as much as the computing time of GA.

6. CONCLUSIONS

In this study, a new version of TS called TSMMRAP was proposed for MMRAP (Multiple Multi-level Redundancy Allocation Problem) in which all available items for redundancy (system, module, and component) can be simultaneously chosen. To the best of our knowledge, this is the first attempt to use a TS for MMRAP. The TSMMRAP was also compared with the existing GA (Yun et al., 2007) on the new composed test problems as well as benchmark problems in the literature.

Table 7. The input data for larger system

Basic Item	Parent unit	r_i	c_i	λ_i
1(system)	-	0.1336	210	2
11	1	0.72675	26	2
12	1	0.7650	19	3
13	1	0.7200	21	2
14	1	0.7650	15	2
15	1	0.8572	32	3
16	1	0.7650	14	2
17	1	0.8200	25	2
18	1	0.8114	24	3
111	11	0.9000	5	3
112	11	0.9500	6	4
113	11	0.8500	5	4
121	12	0.9000	6	4
122	12	0.8500	7	4
131	13	0.9000	8	3
132	13	0.8000	7	4
141	14	0.9000	6	3
142	14	0.8500	5	3
151	15	0.9600	8	5
152	15	0.9500	7	4
153	15	0.9400	6	3
161	16	0.9000	5	3
162	16	0.8500	4	3
181	18	0.9200	7	4
182	18	0.9000	5	3
183	18	0.9800	3	3

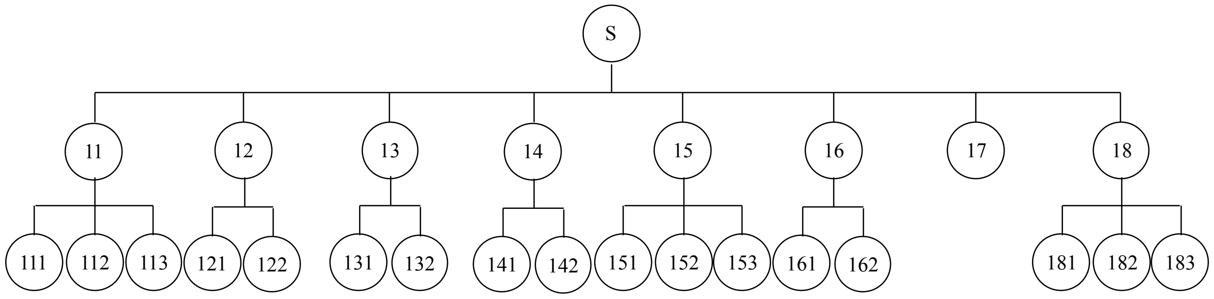


Figure 3. System structure

From the computational results, TSMMRAP substantially outperformed the GA (Yun *et al.*, 2007). The TSMMRAP obtained the higher system reliability for all test problems than the GA, even though TSMMRAP required almost 2 times as much as the computing time of GA. In terms of solution quality, our TSMMRAP is recommended as a promising metaheuristic in this field. To achieve the better solution quality, the development of more powerful metaheuristics for MMRAP would be performed in the future research.

Table 8. Computational results

No.	Cost limit	GA			TSMMRAP		
		A	M	O	A	M	O
1	700	0.0510	0.0742	0/10	0.0	0.0	10/10
2	720	0.0487	0.0739	0/10	0.0	0.0	10/10
3	740	0.0452	0.0692	0/10	0.0	0.0	10/10
4	760	0.0415	0.0589	0/10	0.0	0.0	10/10
5	780	0.0363	0.0590	0/10	0.0	0.0	10/10
6	800	0.0381	0.0551	0/10	0.0	0.0	10/10
7	820	0.0376	0.0536	0/10	0.0	0.0	10/10
8	840	0.0309	0.0518	0/10	0.0	0.0	10/10
9	860	0.0332	0.0454	0/10	0.0	0.0	10/10
10	880	0.0299	0.0420	0/10	0.0	0.0	10/10
			T 5.24(sec.)			T 9.67(sec.)	

7. REFERENCES

1. Bae, C.O., Kim, H.G., Kim, J.H., Son, J.Y., and Yun, W.Y.(2007). Solving the redundancy allocation problem with multiple component choices using metaheuristics. International Journal of Industrial Engineering 14(4): 315–323.
2. Chern, M.S. (1992). On the computational complexity of reliability redundancy allocation in a series system. Operations Research Letters, 11: 309–315.
3. Coit, D.E. and Smith, A.E. (1996). Reliability optimization of series-parallel systems using a genetic algorithm. IEEE Transactions on Reliability, 45(2): 254–260 (1996).
4. Coit, D.E. and Liu, J. (2000). System reliability optimization with k-out-of-n subsystems. International Journal of Reliability, Quality and Safety Engineering, 7(2): 129–142.
5. Fyffe, D.E., Hines, W.W., and Lee, N.K. (1968). System reliability allocation and a computational algorithm. IEEE Transactions on Reliability, 17(2): 64–69.
6. Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research, 13: 533-549.
7. Kim, J.H. and Kim, J.S.(2006). Globally solving the redundancy allocation problem for the case of series-parallel systems. Proceedings of the 2nd Asian International Workshop(AIWARM 2006), 150–157.
8. Kulturel-Konak, S., Norman, B.A., Coit, D.W., and Smith, A.E. (2004). Exploiting tabu search memory in constrained problems. INFORMS Journal on Computing, 16(3): 241–254.
9. Liang, Y.C. and Smith, A.E. (2004). An ant colony optimization algorithm for the reliability allocation problem (RAP). IEEE Transactions on Reliability, 53(3): 417–423.
10. Liang, Y.C. and Chen, Y.C. (2007). Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm. Reliability Engineering and System Safety, 92: 323-331.
11. Nahas, N., Nourelfath, M., and Ait-Kadi D.(2007). Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems. Reliability Engineering and System Safety, 92: 211-222.
12. Nakagawa, Y. and Miyazaki, S. (1981). Surrogate constraints algorithm for reliability optimization problems with two constraints. IEEE Transactions on Reliability, 30(2): 175–180.
13. Ouzineb, M., Nourelfath, M., and Gendreau, M.(2010). An efficient heuristic for reliability design optimization problems. Computers & Operations Research, 37: 223-235.
14. Yun, W.Y. and Kim, J.W. (2004). Multi-level redundancy optimization in series systems. Computers & Industrial Engineering, 46: 337-346.
15. Yun, W.Y., Song, Y.M. and Kim, H.G. (2007). Multiple multi-level redundancy allocation in series systems. Reliability Engineering and System Safety, 92: 308-313.

BIOGRAPHICAL SKETCH

Kil-Woong Jang received the B.S., M.S. and Ph.D. in Computer Engineering from Kyungpook National University, Korea in 1997, 1999 and 2002, respectively. Since 2003, he has been Associate Professor of Data Information at Korea Maritime University. His research interests include the development of metaheuristic algorithms and wireless network protocols.

Jae-Hwan Kim is Professor and Chair of Data Information at Korea Maritime University. He received his B.S. in Industrial Engineering from Korea University and his M.S. & Ph.D. in Industrial Engineering from Korea Advanced Institute of Science and Technology (KAIST). He will be a visiting scholar of Industrial and Systems Engineering at Auburn University from August 2011 to August 2012. His research centers on reliability optimization, application and development of metaheuristic algorithms, and combinatorial optimization.