International Journal of Applied Operational Research
Vol. 2, No. 2, pp. 49-57, July 2012

Journal homepage: www.ijorlu.ir

A Memtic genetic algorithm for a redundancy allocation problem

M. EbrahimNezhad Moghadam Rashti*, B. M. Vishkaei, R. Esmaeilpour

Received: October 24, 2011 ; **Accepted:** April 18, 2012

Abstract In general redundancy allocation problems the redundancy strategy for each subsystem is predetermined. Tavakkoli- Moghaddam presented a series-parallel redundancy allocation problem with mixing components (RAPMC) in which the redundancy strategy can be chosen for individual subsystems. In this paper, we present a bi-objective redundancy allocation when the redundancy strategies for subsystems are considered as a variable of the problem. As the problem belongs to the NP-hard class problems, we will present a new approach for the non-dominated sorting genetic algorithm (NSGAII) and Memtic algorithm (MA) with each one to solve the multi-objective model.

Keywords Redundancy Allocation Problems, Series-Parallel Problems, Redundancy Strategies, Non Dominated Sorting Genetic Algorithms (NSGAII), Memtic Algorithms.

1 Introduction and literature review

In this paper the series-parallel redundancy allocation problem is considered. This kind of problem can be categorized into redundancy allocation problems without component mixing (RAPCM) and redundancy allocation problems with a mix of components. Fyff [1] studied the redundancy allocation problem and used goal programming to solve it. You [2] presented a workable heuristic algorithm for series-parallel redundancy allocation problems. Coit [3] presented a new model for series-parallel systems (RAPCM) in which cold strategy is used for all subsystems. Coit [4] studied the case with k-out-of-n subsystems when the redundancy strategy for each subsystem is predetermined and includes active and cold strategies. Coit [5] considered the redundancy allocation problem without component mixing when either active or cold-standby redundancy strategy can be selectively chosen for individual subsystems. Moghaddam [6] extended this model using genetic algorithm to solve it.

Coit and Smith [7,8] presented a genetic algorithm for a redundancy allocation problem with a mix of components with k-out-of-n subsystems. But the proposed algorithm cant be used for large problems because of its chromosome. Coit [9] presented a combined neural network and genetic algorithm (GA) approach to solve a redundancy allocation problem (RAPMC) to reduce the total cost of the system. Moreover, the redundancy allocation

* **Corresponding Author.** (✉)
E-mail: M.Ebrahimnazhad@gmail.com, (M. EbrahimNezhad Moghadam Rashti.)

M. EbrahimNezhad Moghadam Rashti
Young Researchers club, Qazvin Branch, Islamic Azad University, Qazvin, Iran.

B. M. Vishkaei
Department of Industrial Engineering, Islamic Azad University, Qazvin Branch, Iran.

R. Esmaeilpour
Department of Department of Management, University of Guilan, Iran.

problems with a mix of components have been solved with different methods. Liang [10] used variable neighborhood descend algorithm to solve this kind of problems. This method is simpler than variable neighborhood search algorithm that is describe by Liang [11]. Onishi [12] presented an exact method based on improved surrogate constraint (ISC) approach to solve these problems. Tavakkoli- Moghaddam [13], Ebrahimnezhad Moghadam Rashti [14] considered each subsystem’s strategy as a variable for redundancy allocation problems with a mix of components. The model that is studied in this paper is based on the proposed model.

The structure of this paper is organized as follows; Sections 2 and 3 present the problem description and the problem formulation, respectively. In Section 4, a new approach for the non-dominated sorting genetic algorithm (NSGAII) will be present to solve the multi-objective model. In section 5 the algorithm parameters are tuned by Taguchi method and will be used for different sizes of problems. Finally, conclusion is presented.

2 Problem definition

The problem that is studied in this paper is the result of adding a new objective function to the series-parallel allocation problem of integer programming type that is presented by Moghadam [13], Ebrahimnezhad [14]. This new objective function calculates the pure profit that is gained during a limit period of time considering different costs like purchasing cost, penalties during downtime and damage cost of the components. Therefore, the objective is to determine the strategy, kind of the components and the number of the components that are allocated to each subsystem such that the reliability and the profit of the system are maximized while the cost and weight constraints are satisfied.

3 Problem modeling

To formulate the problem, first the parameters are defined and finally the mathematical model is developed in section 3.2.
The parameters of the model
The parameters of the model are defined as follows:

A set of all subsystems with active redundancy,
S set of all subsystems with cold-standby redundancy,
N set of all subsystems with no redundancy.
s number of subsystems
n_i number of components used in subsystems i (i=1,2,3,…,s)
n set of $\mathrm{n_i} (n_1, n_2, ..., n_s)$
$\mathrm{z_i}$ index of the component that is dedicated to subsystem i
z set of $\mathrm{z_i}$ $(z_1, z_2, ..., z_s)$
T mission time
$\mathrm{R(t, z_i)}$ system reliability at time t
$\mathrm{r_{ij}(t)}$ reliability at time t for component j in subsystem i
$\lambda_{\mathrm{ij}}, k_{ij}$ scale and shape parameters for the Erlangen distribution; $\mathrm{f_{ij}}(t) = (\lambda_{ij}^{\kappa_{ij}} t^{\kappa_{ij}-1} e^{-\lambda_{ij} t}) / \Gamma(\kappa_{ij})$
C, W system constraint limits for cost and weight

c_{ij}, w_{ij} purchasing cost and weight for the jth available component for the subsystem i

$\rho_i(t)$ failure switching reliability at time t

p_t the amount of money per unit time paid by the customer for the plan service

TP total profit from plant operation

B_i installation cost per each component

TC_p total purchase and installation costs

c_{NS} penalty cost during downtime, due to missed delivery of agreed service

Tc_{NS} total penalties during downtime for period T

c_d damage cost per each component

Tc_d total damaged cost during period T

3.1 Objective Functions

The profit function contains the plant profit, purchasing and installation cost, penalties during downtime and the damage cost.

$$TP = P_t . \int_{\circ}^{T} R(t) \tag{1}$$

Equation 1 is the plant profit in which p_t represents the amount of money per until time paid by the customer for the plant service, and R(t) is the instantaneous plant reliability.

$$TC_p = \sum_{i=1}^{s} n_i (C_{iz_i} + B_{iz_i}) \tag{2}$$

Equation 2 is the purchasing and installation cost of the s nodes in which the ith of them constituted of n_i components.

$$TC_{NS} = C_{NS} . \int_{\circ}^{T} (1 - R(t)) \tag{3}$$

Equation 3 is the amount of money to be paid to the customer because of missed delivery of the agreed service when the plant is unavailable.

$$TC_d = \sum_{i=1}^{S} \min\{n_i, \lambda_{iz_i} K_{iz_i} T\} C_s = \sum_{i=1}^{S} U_i \tag{4}$$

Equation 4 is the total damage cost during period T. The component time – to-failure is distributed according to Erlangen distribution, so $\lambda_{iz_i} \kappa_{iz_i} T$ is the average failed components during period T for the i^{th} subsystem. As all the dedicated components may fail before finishing period T, the number of damaged components for subsystem i would be $\min\{n_i, \lambda_{iz_i} \kappa_{iz_i} T\}$

Equation (4) can be changed as equation (5):

$$TC_d = \sum_{i=1}^{S} U_i \tag{5}$$

In which u_i can be shown as:

$$U_i \leq \lambda_{iz_i} \kappa_{iz_i} Tc_s \ \ \& \ \mathrm{U_i} \leq n_i.c_s \tag{6}$$

The net profit objective function can be then written as follows:

$$\mathrm{G} = P_t \int_{\circ}^{T} R(t) \text{ - } (\sum_{i=1}^{s} n_i (C_{iz_i} + B_{iz_i}) + C_{NS} \int_{\circ}^{T} (1 - R(t)) + \sum_{i=1}^{S} \min \ \{n_i, \lambda_{iz_i} K_{iz_i} T\} C_s) \tag{7}$$

According to equations (5):

$$\mathrm{G} = P_t \int_{\circ}^{T} R(t) \text{ - } (\sum_{i=1}^{s} n_i (C_{iz_i} + B_{iz_i}) + C_{NS} \int_{\circ}^{T} (1 - R(t)) + \sum_{i=1}^{S} U_i)$$

St :

$\mathrm{U_i} \leq \lambda_{iz_i}.k_{iz_i}.T.c_s \qquad \forall \mathrm{i}$

$\mathrm{U_i} \leq n_i.c_s \qquad \forall \mathrm{i}$

3.2 The Mathematical Model

The proposed mathematical model is described as follow:

$$Max \quad \mathrm{R(T)} = \prod_{\mathrm{i \in S}} (\mathrm{r_{iz_i}}(T) + \rho(T) e^{-\lambda_{iz_i} T} \times \sum_{l=k_{iz_i}}^{k_{iz_i} n_i - 1} \frac{(T \times \lambda_{iz_i})^l}{l!}) \times \prod_{\mathrm{i} \in A} (1\text{-}(1\text{-}\mathrm{r_{iz_i}}(T))^{n_i} \times \prod_{\mathrm{i \in N}} \mathrm{r_{iz_i}}(T) \tag{1}$$

$$Max \quad \mathrm{G} = P_t \int_{\circ}^{T} R(t) \text{-} (\sum_{i=1}^{s} n_i (C_{iz_i} + B_{iz_i}) + C_{NS} \int_{\circ}^{T} (1 - R(t)) + \sum_{i=1}^{S} U_i) \tag{2}$$

$st.$

$$\sum_{\mathrm{i=1}}^{\mathrm{S}} \mathrm{C_{iz_i}} n_i \leq C, \tag{3}$$

$$\sum_{\mathrm{i=1}}^{\mathrm{S}} \mathrm{w_{iz_i}} n_i \leq W, \qquad \mathrm{i} = 1, 2, ..., \mathrm{S}, \tag{4}$$

$$U_i \leq n_i C_s, \qquad \mathrm{i} = 1, 2, ..., \mathrm{S}, \tag{5}$$

$$\mathrm{U_i} \leq C_s \lambda_{iz_i} K_{iz_i} T, \qquad \mathrm{i} = 1, 2, ..., \mathrm{S}. \tag{6}$$

(1): Maximize the reliability of the system in which the first, second and third term of the equation denotes the reliability for the subsystem with cold redundancy strategy, active strategy and no redundant Coit [9].

(2): Maximize the net profit value.

(3): Purchasing cost constraint.

(4): Weight constraints.

(5), (6): $u_i = \min\{ n_i, \lambda_{iz_i} \kappa_{iz_i} T \} C_s$ is replaced with the two constraint $u_i \geq n_i C_s, u_i \geq \lambda_{iz_i} \kappa_{iz_i} T.C_s$

This model belongs to NP-hard class and solving it by exact methods is not possible, a small example considering two subsystems will be solved with exact method in the next section. All possible amounts for the variables are considered and the problem will be solved for all the possible combinations of these variable amounts. Finally, the set of pareto optimal solution will be calculated among these possible solutions and a unique solution will select among the set of solution using normalize method.

As this model belongs to NP-hard class, solving it by exact methods is not possible. In this paper we use NSGAII considering a new approach for producing the chromosomes of the first population to solve the model.

4 Solving algorithm

The new approach that is used for producing the first population of NSGAII is described as follow:
1. Produce one chromosome considering the possible dedicating components with the most reliability for each subsystem.
2. Produce N-1 (N is the size of parent population) random chromosomes.
3. Check the N-1 random chromosomes by following process:

If the chromosome denies cost constraint:
Find the subsystem with the most cost and reduce one of its components and replace the old chromosome with the new one.

Else if the chromosome denies weight constraint:
Find the subsystem with the most weight and reduce one of its components and replace the old chromosome with the new one.

Else:
There is no need to change the chromosome.
Consider this chromosome for the next step.

Other steps of the algorithm except the last one are the same as the steps that are described by Coit [15]. In last step we obtain the best optimal solution using DIS method. In multiple-objective problems a number of efficient are yielded. Sometimes these solutions can be so various that it is not possible for the decision maker to select the final decision easily. Displaced ideal solution (DIS) [16] is a kind of filtering approach that is helpful in decreasing this collection (the number of solutions). Therefore, the proposed algorithm can be described as follow:

Step 1: Create a parent population of size N.
Set $t = \circ$.
Step 2: Apply crossover and mutation to $P_\circ$ to create offspring population $Q_\circ$ of size N.
Step 3: If the stopping criterion is satisfied, stop and return to P_t .
Step 4: Set $R_t = P_t \cup Q_t$.
Step 5: Using the fast non-dominated sorting algorithm, identify the non-dominated fronts $F_1, F_2, ..., F_k$ in $\mathrm{R_t}$
Step 6: For $i = 1, 2, ..., k$ do following steps:
Step 6.1: Calculate crowding distance of the solutions in F_i

Step 6.2: Create P_{t+1} as follows:

Case 1: If $|P_{t+1}|+|F_i|\le N$ then set $P_{t+1}=P_{t+1}\cup F_i$;

Case 2: If $|P_{t+1}|+|F_i|>N$ then add the least crowded $N-|P_{t+1}|$ solutions from F_i to P_{t+1} .

Step7: Use binary tournament selection based on the crowding distance to select parents from P_{t+1} , Apply crossover and mutation to P_{t+1} to create offspring population Q_{t+1} of size N.

Step8: Set t = t + 1 , and go to Step3.

Step9: normalize the first rank's solutions to filter them and gain the best answer.

In the last step, Considering X as the best gained reliability and Y as the best gained net profit value, the set of Pareto optimal solutions can be normalized by; $\sqrt{(x_i-X)^2+(y_i-Y)^2}$ in which the x_i and y_i are the amounts of the first and second objective for the ith optimal solution in the best gained rank, respectively. The best solution has the lowest normalized amount.

The crossover and mutation operations and the chromosome are the ones which are described by Moghaddam [13]. Moreover, to assure the best solution is feasible, the dynamic penalty function proposed by Coit et al. [15, 16] is adopted.

5 Numerical examples

Based on the CPU time, the examples are divided into three categories of small, medium and large problems which are presented in Table 1. Taguchi approach is used to estimate the factors of the proposed algorithm. As we face a rank of solutions, we will use Mean Ideal Distance (MID) approach to gain unique answers for algorithm to use Taguchi approach.

$$MID=\frac{\sum_{i=1}^{n}c_i}{n}$$

where n is the number of non-dominated se and $c_i=\sqrt{(1-f_{1i})^2+(P_tT-f_{2i})^2}$. The lower value of MID, the better of solution quality we have.

$$SNS=\sqrt{\frac{\sum_{i=1}^{n}(MID-c_i)^2}{n-1}}$$

The higher value of SNS, the better of solution quality we have (more diversity in obtained solution).

Table 2 shows the 5 levels of the parameters in Taguchi approach. The results of the Taguchi method are shown in Table3, 4 and 5. Using these tuned parameters all the examples are solved again and the final answers are available in table6. The small amount of the deviation shows the algorithm is stable.

Table 1 Categories of examples

	Number of subsystems	CPU Time
Small	2 , 4 , 7 , 9	0 to 2
Medium	11 , 14 , 17 , 20	2 to 6
Large	22 , 25 , 28 , 31	6 to up

Table 2 Levels of the parameters for Taguchi method

Factor	Symbol	Level	Type				
Pc	A	5	A1:0.8	A2:0.75	A3:0.65	A4:0.55	A5:0.45
Pm	B	5	B1:0.075	B2:0.05	B3:0.04	B4:0.03	B5:0.01
Pop-size	C	5	C1:50	C2:80	C3:120	C4:200	C5:300

Table 3 The results of Taguchi method for small examples

Level	A	B	C
1	85.54	84.51	85.41
2	84.45	85.17	84.18
3	84.75	85.02	85.72
4	85.24	85.57	85.12
5	85.71	85.4	85.24
Delta	1.26	1.06	1.54
Rank	2	3	1

Table 4 The results of Taguchi method for small examples

Level	A	B	C
1	77.8	73.04	74.06
2	74.68	77.77	77.74
3	75.29	77.88	77.4
4	76.32	74.15	77.64
5	76.21	77.47	73.47
Delta	3.12	4.84	4.27
Rank	3	1	2

Table 5 The results of Taguchi method for small examples

Level	A	B	C
1	43.63	46.19	47.16
2	45.67	47.59	44.74
3	40.65	41.25	45.4
4	48.94	49.8	42.87
5	50.49	44.56	49.2
Delta	9.84	8.55	6.33
Rank	1	2	3

Table 6 The results of the three examples after running the parameters

Small Examples				Medium Examples				Large Examples			
Run	Example1			Run	Example1			Run	Example1		
	R	G	CUPTime		R	G	CUPTime		R	G	CUPTime
1	0.2466542	1999768	0.5704722	1	0.9969337	1983662	2.26740537	1	0.963749	974062	7.6604054
2	0.3142853	1999912	0.2290881	2	0.9973484	1983369	2.41373835	2	0.9146874	1801296	6.2025662
3	0.2466542	1999154	0.4334892	3	0.9973484	1985756	2.09331005	3	0.9999668	1968157	6.6557303
4	0.3142853	1997920	0.381713	4	0.9999999	1995139	1.58542243	4	0.9333317	849993	6.9247886
Deviation	0.0390468	907.29543	0.1410356	Deviation	0.0014085	5541.236	0.36096568	Deviation	0.03731	567968	0.6106796
Run	Example2			Run	Example2			Run	Example2		
	R	G	CUPTime		R	G	CUPTime		R	G	CUPTime
1	0.9163339	1830830	0.275265	1	0.999999	1990662	2.29083142	1	0.9980558	1963621	6.446572
2	0.8883064	1773129	0.3355297	2	0.9984291	1986711	2.84317727	2	0.999999	1978181	6.9044156
3	0.8883064	1772878	0.2452757	3	0.999999	1989778	2.4251877	3	0.9907291	1929301	7.5606973
4	0.9163339	1831653	0.2468299	4	0.9996195	1986832	2.8224484	4	0.999729	1976965	6.5821473
Deviation	0.0161817	33625.56	0.0421857	Deviation	0.0007435	2024.039	0.27968959	Deviation	0.0043518	22786.2	0.4967729
Run	Example3			Run	Example3			Run	Example3		
	R	G	CUPTime		R	G	CUPTime		R	G	CUPTime
1	0.999999	1997329	0.5447996	1	0.999999	1987032	2.80511773	1	0.999999	1990284	7.7927327
2	0.9997016	1989581	0.4417642	2	0.999398	1976976	2.35364178	2	0.999989	1991668	8.1590108
3	0.9475343	1892343	0.3548805	3	0.999999	1979667	2.71665193	3	0.9999999	1990225	7.9117455
4	0.999999	1997350	0.4374397	4	0.9998036	1979929	2.91274303	4	0.9999908	1989720	7.9200574
Deviation	0.0261832	51335.618	0.0777792	Deviation	0.0002833	4299.703	0.24256653	Deviation	5.575E-06	835.115	0.1535261
Run	Example4			Run	Example4			Run	Example4		
	R	G	CUPTime		R	G	CUPTime		R	G	CUPTime
1	0.9999275	1989179	0.7616357	1	0.9998649	1987536	3.5229845	1	0.9999967	1985963	8.5773889
2	0.999999	1996140	0.4814805	2	0.9973666	1982805	3.1176561	2	0.9248027	1826820	9.4720872
3	0.9999544	1989365	0.5310492	3	0.999999	1991918	3.036929	3	0.9248027	1836759	8.9525546
4	0.9992692	1990160	0.66602	4	0.999999	1986234	3.37431772	4	0.9999999	1985479	8.6937075
Deviation	0.0003468	3313.4263	0.1276481	Deviation	0.0012954	3768.202	0.22525283	Deviation	0.0434142	88965.2	0.3976551

6 Conclusion and future research

In this paper a bi-objective redundancy allocation model has been solved by a non-dominated sorting genetic algorithm (NSGAII) in which the first population was produced via a new approach. The new approach process was presented to reduce the infeasible chromosomes in the hope of gaining better solutions. Moreover, IDM method was introduced for filtering the final solutions to gain a unique answer. The results showed that the proposed algorithm is stable and workable for redundancy problems.

References

1. Fyffe, D. E., Hines, W. W., Lee, N. K., (1968). System Reliability Allocation and a Computational Algorithm. IEEE Transactions on Reliability, 17, 64 – 69.
2. You, P. S., Chen, T. C., (2005). Efficient heuristic for series – parallel redundant reliability problems. Computers & operations research, 32, 2117 – 2127.
3. Coit, D. W. (2001). Cold standby redundancy optimization for nonrepairable systems. IE transactions, 33, 471 – 478.
4. Coit, D. W., Liu, J., (2000). System reliability optimization with k – out – of – n subsystems. International Journal of reliability, quality & safety engineering, 7, 129 – 143.
5. Coit, D. W., (2003). Maximization of system reliability with a choice of redundancy strategies. IIE transactions, 35(6), 535 – 544.
6. Tavakkoli – Moghaddam, R., Safari, J., Sassani, F., (2008). Reliability optimization of series – parallel systems with a choice of redundancy strategies using a genetic algorithm. Reliability engineering and system safety, 93, 550 – 556.
7. Coit, D. W., Smith, A., (1996). Reliability optimization of series – parallel systems using a genetic algorithm. IEEE transactions of reliability, 45, 254 – 260.
8. Coit, D. W., Smith, A., (1995). Optimization approaches to the redundancy allocation to the redundancy allocation problem for series – parallel systems. Proceedings of the fourth industrial engineering research conference.
9. Coit, D. W., Smith, A., (1996). Solving the redundancy allocation problem using a combined neural network / genetic algorithm approach. Computers & operations research, 23, 515 – 526.
10. Liang, Y. C., Wu, C. C., (2005). A variable neighborhood descent algorithm for the redundancy allocation problem. lnd eng manage syst, 4(1), 109 – 116.
11. Liang, Y. C., Chen, Y. C., (2007). Redundancy allocation of series – parallel systems using a variable neighborhood search algorithm. Reliability engineering and system safety, 92, 323 – 331.
12. Onishi, J., Kimura, S., James, R. J. W., Nakagawa, Y., (2007). Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method. IEEE transactions on reliability, 56(1), 94 – 101.
13. Tavakkoli – Moghaddam, R., Safari, J., (2007). A new mathematical model for a redundancy allocation problem with mixing components redundant and choice of redundancy strategies. Applied mathematical sciences, 45(1), 2221 – 2230.
14. EbrahimNezhad Moghadam Rashti. M. , Maleki. Vishkaei., B., (2011). Increasing the Reliability and the Profit in a redundancy allocation problem. International Journal of applied operation research, 1(2), 57-64.
15. Coit, D. W., Konak, A., (2006). Multiple weighted objectives heuristic for the redundancy allocation problem. IEEE transactions on reliability 2006, 55(3), 551 – 558.
16. Coit, D. W., Smith, A., (1996). Penalty guided genetic search for reliability design optimization. Computers & industrial engineering, 30, 895 – 904. 15.