Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 14, No. 4, 2022 Article ID IJIM-1613, 11 pages
Research Article

Using a Hybrid NSGA-II to Solve the Redundancy Allocation Mmodel of Series-Parallel Systems

Mohammadreza Shahriari [*†]

Received Date: 2022-03-20 Revised Date: 2022-7-12 Accepted Date: 2022-09-19

Abstract

This article presents a proficient non-dominated sorting genetic algorithm (NSGA-II) for resolving the redundancy allocation problem (RAP) in series-parallel systems. The system comprises of subsystems arranged in series, where components are employed in parallel for each subsystem. The system and its subsystems can solely take two states of full working condition and complete failure. To accomplish the desired reliability, identical redundant components are incorporated. The subsystems' components, selected from a list available in the market, are distinguished by their cost, weight, and reliability. To achieve the optimal combination of the number of components for each subsystem, the mathematical formulation for the maximal reliability and minimal cost of the system configuration under the cost constraint is initially acquired. A modified NSGA-II is then proposed to solve the model, which integrates a heuristic method of generating a primary solution to obtain better solutions. Additionally, the algorithm uses the design of an experiment approach to calibrate its parameters. Finally, numerical examples are used to validate the solution and evaluate the proposed methodology's performance under different configurations, and to compare the performance with that of two other meta-heuristic algorithms. The experiment results generally favor the proposed solution algorithm.

Keywords : Reliability; Redundancy allocation problem; Series-parallel systems; Heuristic methods; Hybrid algorithm.

1 Introduction

The redundancy allocation problem (RAP) is a well-known optimization problem in engineering, operations research, and reliability engineering. It involves determining the optimal number and placement of redundant components in a system to maximize the system's reliability while minimizing the total cost of the components. The problem arises in situations where a system's failure may have severe consequences, such as in safety-critical systems or in mission-critical systems.

RAP has been studied extensively in the literature, and various methods have been proposed to solve it. One common approach is to use mathematical programming techniques, such as linear programming, mixed-integer programming, or nonlinear programming, to formulate the problem as an optimization problem. Meta-heuristic algorithms, such as genetic algorithms, simulated annealing, and tabu search, have also been proposed to solve the RAP.

RAP has numerous applications, including the design of safety-critical systems in aerospace, defense, and nuclear power plants, as well as in

[*]Corresponding author. shahriari.mr@gmail.com, Tel: +98(44)32759180
[†]Faculty of industrial management, South Tehran Branch, Islamic Azad University, Tehran, Iran.

the design of reliable computer networks, communication systems, and transportation systems. The redundancy allocation problem (RAP) is a complex combinational optimization problem, in which the goal is to determine the optimal combination of the components of a system to maximize the reliability under cost and weight constraints. This problem has many applications in industries. Electronic systems, power stations, manufacturing production systems, and the like are some examples of RAP applications.

In a binary system reliability framework, both the system and its subsystems can only take two possible states of "completely working" and "completely failed." In this research, a binary system is considered in which the subsystems are designed to work in series-parallel framework. In a series-parallel structure, the system consists of subsystems in series where for each subsystem multiple components are used in parallel. The configuration of such a system is depicted in Fig 1, where there are N subsystems in series (S_i ; $i = 1, 2, ..., N$), each having n_i ; $i = 1, 2, ..., N$components in parallel. Chern

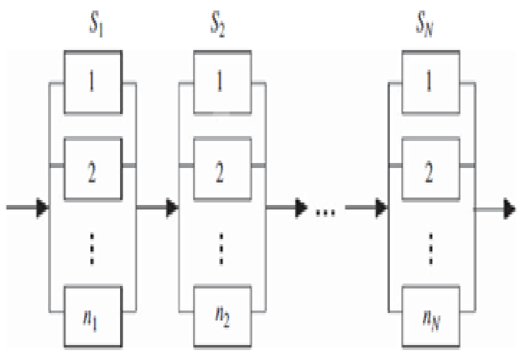


Figure 1: A series-parallel system

[1] showed the RAP is a strongly NP-hard problem. Moreover, many researchers using different approaches and techniques have studied RAP for binary–state series-parallel systems. Kulturel-Konak et al. [2] used a tabu search meta-heuristic algorithm to provide solutions to the system reliability optimization problem of redundancy allocation, Kuo and Prasad [3] provided an overview of the methods that have been developed since 1977 for solving various reliability optimization problems; applications of these methods to various types of design problems; and heuristics, meta-heuristic algorithms, exact methods, reliability-redundancy allocation, multi-objective optimization and assignment of interchangeable components in reliability systems, and Tillman et al. [4] presented a review of the current redundancy and reliability allocation literature.

While there are some exact algorithms available in literature to solve RAP (for example dynamic programming approach of Kuo et al. [5], branch and bond algorithm (B&B) algorithm of Kuo et al. [5] and Ha and Kuo [6]), the literature of meta-heuristic is very rich in this regard [7, 8, 9, 10, 11]. Ouzineb et al. [12] developed an efficient tabu search (TS) heuristic to solve the binary redundancy allocation problem for multi-state series–parallel systems, Levitin and et al. [13] used genetic algorithm to solve a redundancy allocation problem to multi-state systems where the system and its components have a range of performance levels-from perfect functioning to complete failure and Nahas et al. [14] applied a ant colony optimization and the degraded to solve ceiling algorithm for the redundancy allocation problem of series–parallel systems.

In this paper, an approach is proposed for the multi objective redundancy allocation problem of binary-state series-parallel systems that is more applicable in real-world environments. More specifically, the versions of the selected components along with their redundancy (i.e., number of components for different subsystems) are the two factors that usually affect the system reliability. To the best of authors' knowledge, these factors have not been simultaneously utilized in RAP of binary-state series-parallel system reliability. In this research, the aforementioned factors are used simultaneously to model the reliability of binary-state series-parallel systems. Furthermore, NSGA-II algorithm integrated with a primary-solution-generation heuristic is developed to solve the model. NSGA-II was first introduced by Deb et al. [15] where was employed in many fields in recent years (e.g., in facility location problem [16] and [17], in scheduling problem [18] and [19] etc).

In [20] a redundancy allocation problem for series-parallel systems is modeled in which components in each subsystem can be in different types. Also, a PSO is applied to solve the model where integrated with a heuristic.

Among the papers that have been done in recent years on the redundancy allocation prob-

lem (see [21, 22, 23]) we can mention the study of Dobani et al. [24]. They introduced a new MINLP model for the redundancy-reliability allocation problem and used a meta-heuristic algorithm called stochastic fractal search (SFS) to solve it. Ouyang et al. [25] also used an improved PSO algorithm to solve the redundancy-reliability allocation problem by considering heterogeneous components.
In our paper, we model a redundancy allocation problem for series-parallel systems such that components in each subsystem are identical.
In this research, a meta-heuristic multi-objective optimization algorithm has been combined with an innovative method of generating efficient initial solutions. In this regard, in order to achieve better convergence of the NSGA-II algorithm to a good solution, a local search heuristic procedure is proposed in this research.

The organization of the rest of the paper follows. In the next Section, the mathematical formulation of the problem is given. The modified NSGA-II algorithm comes in Section 3. Section 4 contains the parameter calibration of the algorithm along with validation and performance evaluation. Finally, we conclude the paper in Section 5.

2 Problem Formulation

The problem assumptions, notations, and the model development come in this Section as follow.

2.1 Assumptions

The considered assumptions in this research are:

1. Reliability, weight, and cost of all components are deterministic and known
2. There is no supply constraint for the components
3. Failed components are not repairable
4. The number of subsystems is fixed.

For a broader applicability of the model, the components are chosen from a list that is available in the market. Moreover, all components of a subsystem are selected from a similar type (brand). This is a common practice in real-world RAP. We further assume all redundancies are active and switching between components is performed perfectly.

2.2 Notations

The notations that are used in the rest of the paper are as follows:

S_i: The subsystems in series $(i = 1, 2, ..., N)$

n_i: Total number of component types available for subsystem i

x_{ij}: Number of j^{th} component type used for i^{th} subsystem $(j = 1, 2, ..., n_i)$

c_{ij}: Cost of j^{th} component type in i^{th} subsystem

C: Total system cost

w_{ij}: Weight of j^{th} component type in the i^{th} subsystem

W: Total system weight

r_{ij}: Reliability of j^{th} component type in the i^{th} subsystem

R: System reliability

L_i: Minimum number of components that can be used in parallel for i^{th} subsystem

U_i: Maximum number of components that can be used in parallel for i^{th} subsystem

2.3 The mathematical model

Based on the assumptions and notations, the mathematical formulation of the series-parallel system can be expressed as follows:

$$Max\ \ R = \prod_{\substack{i=1 \\ \forall j : x_{ij} \geq 1}}^{N} \left(1 - (1 - r_{ij})^{x_{ij}}\right)$$

s.t.

$$\sum_{i=0}^{N}\sum_{j=0}^{n_i} w_{ij}x_{ij} \leq \ W \tag{2.1}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{n_i} c_{ij}x_{ij} \leq C \tag{2.2}$$

$$L_i \leq \sum_{j=0}^{n_i} x_{ij} \leq U_i \tag{2.3}$$

The objective function is to maximize the system reliability, where constraint (2.1) implies that the total weight of the designed series-parallel system should be less than$W$. Constraint (2.2) expresses an upper bound on the total cost and the minimum and maximum redundancy limitations for each subsystem are shown in constraint (2.3).

Sine the mathematical model of the problem is strongly NP-hard [1], in the next Section a meta-heuristic algorithm is developed to solve it.

3 The proposed modified NSGA-II algorithm

In most RAP problems, due to complexity of model and number of numerous constraints, it cannot apply exact approaches to solve these problems.

In this work, a redundancy allocation problem is proposed where use NSGA-II and NRGA to solve it.

3.1 The proposed NSGA-II

NSGA-II is modified version which has a better sorting algorithm, incorporates elitism and no sharing parameter needs to be chosen a priori. NSGA-II is a MOEA method that first introduced by Deb et al. [10] with using pareto dominance solutions where is a computationally efficient algorithm implementing the idea of a selection method based on classes of dominance of all the solutions. The original NSGA-II algorithm is consisting of five operators: initialization, constrained non-dominated sorting, crossover, mutation and the elitist crowded-comparison operator. In this paper some of the details of these steps are outlined below.

*3.1.*1 Initialization

A particle refers to a point in the designed space that changes its position from one move (iteration) to another, based on exploration velocity updates. The type of particles is associated with the number of variables involved in a problem. In this research, there are n_i decision variables (x_{ij}) for subsystem i. As a result, each particle is represented by a $\sum_{i=1}^{N} n_i$dimensional array shown in Figure 2, where for example, the first n_1 components represent the number of the paralleled components of different types for the first subsystem.

x_{11}	x_{12}	...	x_{1n_1}	...	x_{N1}	x_{N2}	...	x_{Nn_N}

Figure 2: A particle representation

3.1.2 Fast non-dominated sorting

In this step, we compare R population that generated in previous step and sort them based on non-dominated fronts concept. In here, all chromosomes in the first non-dominated front are found. In order to find the chromosomes in the next non-dominated the solutions of the first front are discounted temporarily and the above procedure is repeated. In the worst case, the task of finding the second front also requires $O(MR^2)$computations, particularly when $O(R)$number of solutions belong to the second and higher non-dominated levels. M is the number of objectives and R is the population size [10].

3.1.3 Crowding distance

To get an estimate of the density of solutions surrounding a particular solution in the population, we calculate the average distance of two points on either side of this point along each of the objectives [10]. Consider a number of non-dominated solutions in F of the size Z and a number of objective functionsf_k, $k = 1, 2, ..., M$are given. Suppose $d_i$or $d_j$be the value of crowding distance on the solution i or j. The crowding distance can calculate as following steps:
Set $d_i = 0$ *for* $i = 1, 2, ..., Z$
For all objective functions f_k, $k = 1, 2, ..., M$ *ascending sort the set.*
For end solutions in each front, value of crowding distances i.e. d_1*and* d_Z*equal* $d_1 = d_Z = \infty$.
For $j = 2$ *to (Z-1), value of crowding distances calculates as* $d_j = d_j + (f_{k_{j+1}} - f_{k_{j-1}})$.

3.1.4 Crowding selection operator

In order to compare two solutions x and y, crowded tournament selection operator $\succ$ is defined in which select two solutions randomly every time and do this process for R population. Two solutions x and y compare in following condition. If occur, one of following conditions then solution

x is better another solution. (2.1) If the domination rank of solution x is smaller than the one of solution Y, or (2.2) their domination ranks are equal, and the crowding distance of solution x is larger than the one of solution y. So, the NSGA-II comparison criteria can be written as follows:
If $r_x< r_y$or $r_x= r_y$ and $d_x> d_y$then $x \succ y$

3.1.5 Genetic operators

Genetic operators for this research are as following descriptions.

In this paper, the crossover operator is as follows. Let $P_C$be crossover probability and $r_1$be a random number between 0 and 1. For R population if $r_1$be less of $P_C$then select the chromosome. Suppose two parents V_1 and V_2 is selected randomly for crossover operator. Therefore, the crossover operator to product offspring is as follows:

$$V_1^{'}= \lambda.V_1+(1-\lambda).V_2 \tag{3.4}$$

$$V_2^{'}= (1-\lambda).V_1+\lambda.V_2 \tag{3.5}$$

Where $V_1^{'}$ and $V_2^{'}$are offspring and λ is a random number between 0 and 1.
A polynomial mutation for k-th selected chromosome for mutation operator is formulated based on [26] as:

$$V_k^{'}= V_k+(V_k^u-V_k^l)\delta_k \tag{3.6}$$

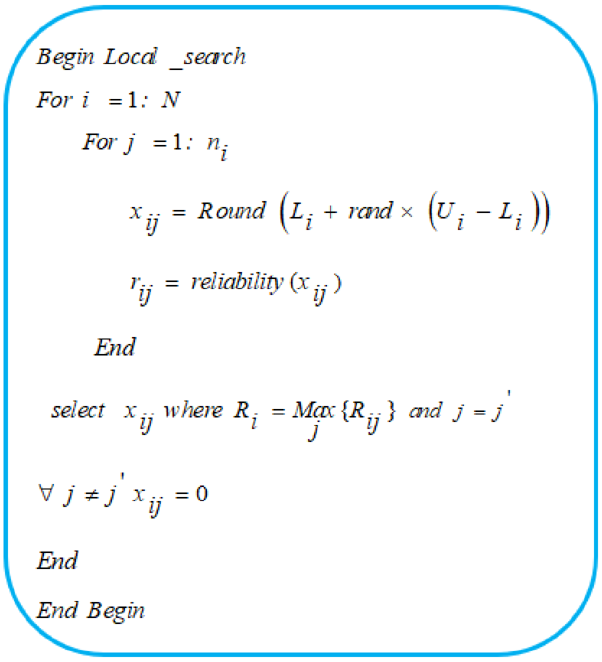
In equation (3.6) $V_k^{'}$is the child and $V_k$is the parent withV_k^u being the upper bound on the parent component, $V_k^l$is the lower bound and $\delta_k$is small variation which is calculated from a polynomial distribution by using:

$$\delta_k = \begin{cases} (2r_k)^{\left(\frac{1}{\eta_m+1}\right)} - 1 & r_k< 0.5 \\ 1-[2(1-r_k)]^{\left(\frac{1}{\eta_m+1}\right)} & r_k \geq 0.5 \end{cases} \tag{3.7}$$

$r_k$is a random number between (0, 1) and $\eta_m$is mutation distribution index?

3.1.6 Recombination and Selection

The offspring population is combined with the current generation population and selection is performed to set the individuals of the next generation. Since all the previous and current best individuals are added in the population, elitism is ensured. Population is now sorted based on non-domination [15].



```
Begin Local _search
For $i = 1: N$
    For $j = 1: n_i$
        $x_{ij} = Round\left(L_i + rand \times \left(U_i - L_i\right)\right)$
        $r_{ij} = reliability\,(x_{ij})$
    End
  select $x_{ij}$ where $R_i = \underset{j}{Max}\{R_{ij}\}$ and $j = j'$
$\forall\ j \neq j'\ x_{ij} = 0$
End
End Begin
```

Figure 3: pseudo code of the heuristic procedure

3.1.7 A heuristic to initialize chromosomes

In order to achieve better convergence of the NSGA-II algorithm to a good solution, a local search heuristic procedure is proposed in this research. In this procedure, for each type of component, we first generate an integer number between L_i and U_i, randomly. Then, based on the generated numbers, the reliabilities of the corresponding subsystems are calculated. Next, the j^{th} component type with the maximum reliability is selected. The pseudo code of the heuristic procedure is given in figure 3.

Since RAP is a constrained optimization problem and search can benefit from infeasible solutions [27], an adaptive penalty function $P(x)$is used to penalize infeasible solutions. This function employs the notion of a near-feasibility threshold for the constraints as proposed in [28]. In the RAP problem at hand, constraints are the total system weight and the total system cost. As a result, in this research the adaptive penalty functions are defined as [20]:

$$F(x) = P(x) \times R(x) \tag{3.8}$$

$$P\left(x\right) = Min\left(1, \frac{C}{C(x)}\right)^{\alpha} \times Min\left(1, \frac{W}{W(x)}\right)^{\beta} \tag{3.9}$$

where $F(x)$is the fitness of solution x, $R(x)$is the overall reliability of the solution x, $C(x)$and $W(x)$are, respectively, the cost and the weight of solution x, and α and β are the penalty factors.

4 Experimental results and discussion

In order test of the problem model, we generate an example for a system with 20 subsystems where each subsystem has 4 types. Table 1 show input data for the example in which second, third and fourth column are cost, weight, and reliability of each type for each subsystem respectively and fifth column is maximum number of components that can be used in parallel for each subsystem. The problem has a weight constraint that in this example we test the model for different amounts of total system weight W. Weight amounts change between 100 to 250.

Table 1: Input data

i	$c_{i1}, ..., c_{i4}$	$w_{i1}, ..., w_{i4}$	$r_{i1}, ..., r_{i4}$	U_i
1	3, 5, 6, 10	9, 7, 4, 1	.92, .90, .91, .92	7
2	2, 5, 7, 10	10, 6, 5, 3	.92, .90, .92, .91	8
3	1, 5, 7, 9	9, 6, 4, 1	.91, .93, .93, .92	6
4	1, 5, 7, 9	9, 7, 5, 3	.93, .93, .92, .90	5
5	1, 5, 6, 9	10, 7, 4, 2	.93, .91, .90, .91	6
6	2, 4, 7, 9	8, 7, 5, 1	.91, .90, .90, .92	6
7	1, 5, 6, 9	10, 7, 4, 3	.93, .90, .90, .90	5
8	2, 5, 7, 9	8, 7, 5, 3	.92, .92, .91, .92	6
9	1, 4, 6, 8	8, 7, 5, 2	.90, .91, .93, .90	8
10	1, 5, 7, 9	10, 7, 5, 1	.93, .91, .92, .91	7
11	1, 4, 6, 9	10, 6, 4, 2	.93, .93, .90, .91	5
12	1, 5, 7, 8	9, 6, 4, 1	.90, .93, .90, .90	8
13	1, 4, 6, 10	9, 6, 4, 1	.93, .90, .91, .92	7
14	2, 5, 6, 9	10, 6, 5, 1	.93, .92, .91, .91	5
15	2, 4, 6, 9	8, 7, 5, 3	.93, .93, .91, .91	6
16	2, 4, 6, 10	9, 6, 4, 3	.91, .90, .93, .91	5
17	2, 5, 7, 9	10, 6, 4, 3	.90,.90,.92,.92	7
18	1, 5, 6, 9	8, 7, 5, 2	.91,.92,.90,.91	5
19	3, 5, 7, 8	8, 6, 4, 3	.92,.90,.90,.93	6
20	1, 5, 6, 8	9, 7, 5, 2	.91,.91,.90,.93	5

The NSGA-II was coded in MATLAB software and run on a Pentium IV 2.5 GHz of Core two Due CPU and 3 GB of RAM in windows XP professional. To tuning parameters of NSGA-II and getting proper amounts, NSGA-II run frequently, and parameters adjust as manually. So, parameter values of NSGA-II are as: population size is 100, number of generations is 200, crossover probability (P_C) is 0.9 and mutation probability (P_m) is 0.1. To solving the example, NSGA-II runs for weight amounts 250, 220, 190, 160, 130 and 100. Figures 4 to 6 display optimal pareto solutions of two goals where to simplicity, In Figs 4 to 6, it clear that with increase W, more solutions are in left corner of pareto front. But with decrease W, aggregation of pareto solutions will be in right corner of pareto front where optimal cost value has decreased. Also, to more understand how production solutions, a goal of the problem i.e., maximization availability is considered. To solve the one objective problem, particle swarm optimization (PSO) algorithm that proposed by Beji et al. [20] is used where integrated with the mentioned heuristic algorithm to generating primary solutions and product a modified PSO algorithm. Table 2 show results of modified PSO algorithms in which to display accuracy it, results of hybrid PSO (HPSO) and TS are used. Results of Table 2 show that modified PSO algorithm is an efficient algorithm to solve the one objective problem. Table 3 give solutions of modified algorithms to different amounts W and C where first and second columns are amounts W and C respectively. In each row of the configuration part an array (a, b, c, d)corresponds to the number of the first, the second, the third, and the fourth type of a component used in a subsystem. For example, the configuration $(0, 0, 3, 0)$means that three components of the third type are used for a subsystem, while none of the other types used. Figures 4, 5 and 6 show the final non-dominant solution set for the maximum allowed weights of 250, 220 and 190 respectively.

Table 2: Comparison availability of the algorithms

i	W	C	PSO	GA	TS
1	100	100	0.22812	0.21432	0.21893
2	100	130	0.30931	0.30931	0.30931
3	100	160	0.41049	0.39807	0.40761
4	100	190	0.43454	0.43892	0.43501
5	100	220	0.65565	0.64976	0.64765
6	100	250	0.71826	0.7195	0.71204
7	130	100	0.51335	0.50908	0.51412
8	130	130	0.83834	0.81708	0.82476
9	130	160	0.89058	0.8876	0.88689
10	130	190	0.94034	0.94525	0.94252
11	130	220	0.98336	0.97861	0.98426
12	130	250	0.98336	0.98046	0.98426
13	160	100	0.98225	0.97654	0.97872
14	160	130	0.98225	0.97861	0.9834
15	160	160	0.98225	0.97861	0.9834
16	160	190	0.99531	0.98904	0.99128
17	160	220	0.99102	0.99356	0.99128
18	160	250	0.99202	0.99356	0.99128
19	190	100	0.97655	0.96703	0.97215
20	190	130	0.97655	0.97342	0.97531
21	190	160	0.98056	0.97672	0.97703
22	190	190	0.99543	0.98989	0.98655
23	190	220	0.99663	0.99321	0.98975
24	190	250	0.99819	0.99432	0.99875
25	220	100	0.97892	0.97915	0.97231
26	220	130	0.98232	0.98173	0.98341
27	220	160	0.98809	0.9867	0.98726
28	220	190	0.99589	0.98867	0.9912
29	220	220	0.99772	0.99793	0.99436
30	220	250	0.99796	0.99801	0.99785
31	250	100	0.98715	0.98788	0.98715
32	250	130	0.98715	0.98788	0.98715
33	250	160	0.99673	0.9943	0.99322
34	250	190	0.99918	0.99458	0.99487
35	250	220	0.99978	0.99815	0.99768
36	250	250	0.99978	0.99856	0.99997

Table 3: The PSO results

W	C	R	Configuration
100	100	0.22812	0010|0100|0100|1000|0100|1000|0010|0001|1000|0100|0010|0010|0010|0010|0010|0100|0010|0010|0010|1000|
100	130	0.30931	0001|0010|0010|0010|0100|0001|0002|0100|0010|0010|0100|0100|1000|0100|1000|1000|0001|0100|0001|0001|
100	160	0.41049	0001|0010|0010|0100|1000|0001|0010|0001|0002|0010|0100|0010|1000|0003|0100|0010|0010|0100|0002|0001|
100	190	0.43454	0001|0010|0020|0010|1000|0001|1000|0001|0010|0002|0002|0100|0003|0010|0010|0010|1000|0010|0001|0002|
100	220	0.65565	0100|0010|0010|0100|0010|0005|0010|0100|1000|0003|0100|0002|0100|0010|0010|0010|0100|0100|0010|0002|
100	250	0.71826	0000|0010|0100|0010|0010|0001|0010|0010|0002|0020|0100|0100|0002|0002|0002|0003|0001|1000|1000|0001|
130	100	0.51335	0010|0010|0100|0100|0010|2000|0010|0010|1000|0100|0100|0100|1000|0020|1000|1000|0010|0100|0010|0001|
130	130	0.83834	1000|0100|0003|0010|1000|0100|0010|0010|0010|1000|0100|0003|0100|1000|1000|0010|1000|0100|0010|1000|
130	160	0.89058	0010|0010|1000|0100|1000|0010|0010|0100|0010|0010|0100|1000|1000|0004|0010|0010|0010|0200|1000|0100|
130	190	0.94034	0010|1000|0020|0010|0020|0003|0010|0003|0100|0010|0010|0010|0100|1000|1000|0100|0100|0010|1000|0003|
130	220	0.98336	0100|0010|0005|0010|0010|0002|0010|0002|0010|1000|0010|0100|1000|0100|1000|0100|0010|0003|0003|2000|
130	250	0.98336	0100|0010|0005|0010|0010|0002|0010|0002|0010|1000|0010|0100|1000|0100|1000|0100|0010|0003|0003|2000|
160	100	0.98225	1000|0100|0100|0100|0010|0100|0010|1000|0100|1000|0010|0300|2000|0010|0100|0010|0100|0010|1000|1000|
160	130	0.98225	1000|0100|0100|0100|0010|0100|0010|1000|0100|1000|0010|0300|2000|0010|0100|0010|0100|0010|1000|1000|
160	160	0.98225	1000|0100|0100|0100|0010|0100|0010|1000|0100|1000|0010|0300|2000|0010|0100|0010|0100|0010|1000|1000|
160	190	0.99531	0010|0010|1000|0100|0100|0100|0100|0005|0010|0005|0200|1000|1000|1000|0010|1000|0100|1000|0010|1000|
160	220	0.99102	0004|0100|1000|0100|0100|0100|0100|0005|0010|0005|0200|1000|1000|1000|0010|1000|0100|1000|0010|1000|
160	250	0.99202	0003|1000|0003|1000|0010|0005|0100|0004|0100|0010|1000|0020|2000|0003|0100|0010|0030|1000|0200|1000|
190	100	0.97655	1000|0100|0100|1000|0100|0002|2000|1000|2000|0010|1000|1000|0010|1000|1000|0010|2000|1000|0100|0200|
190	160	0.98056	0003|0100|0010|0010|1000|0010|0010|3000|0100|0100|1000|0100|1000|2000|1000|0010|2000|0100|0010|0004|
190	190	0.99543	3000|0004|0004|0020|1000|1000|0010|0010|0010|0100|1000|0100|0100|1000|0010|1000|0010|0010|0003|3000|
190	220	0.99663	0004|1000|0010|0010|0004|0010|1000|0300|2000|0010|0010|0300|0100|0100|2000|0010|0003|1000|0100|0200|
190	250	0.99819	3000|0100|0100|0010|0020|0020|0100|0100|0004|0004|0100|0020|0100|0020|0300|1000|0004|0030|1000|0100|
220	100	0.97892	1000|1000|1000|0100|0010|0300|0010|0300|0100|0100|0100|0100|2000|1000|0100|0100|2000|1000|0100|2000|
220	130	0.98232	2000|2000|0200|0010|1000|0010|0200|0100|0100|0100|0100|0010|0010|0100|0100|0010|1000|1000|0200|0100|
220	160	0.98809	1000|0300|1000|4000|0100|3000|0010|0010|0100|0100|0100|0010|0030|0100|0004|0100|1000|1000|0010|0020|
220	190	0.99589	0100|0010|0100|1000|1000|1000|0010|0100|0010|0100|0005|1000|0100|3000|3000|0200|0200|1000|0004|0020|
220	220	0.99772	0010|0001|0100|0100|0100|0003|0030|0010|0010|0010|0003|1000|0100|1000|2000|3000|0010|0010|0005|0300|
220	250	0.99796	1000|0004|0100|0100|1000|1000|0010|0005|0100|1000|0030|0100|0100|1000|1000|0004|0003|1000|1000|0300|
250	100	0.98715	0100|1000|0200|0040|3000|1000|1000|0100|0010|1000|1000|0010|1000|2000|0100|1000|1000|3000|0100|0100|
250	130	0.98715	0100|1000|0200|0040|3000|1000|1000|0100|0010|1000|1000|0010|1000|2000|0100|1000|1000|3000|0100|0100|
250	160	0.99673	2000|1000|1000|0100|0010|1000|0010|1000|0010|0100|0010|0010|0010|2000|1000|1000|0040|3000|1000|0030|
250	190	0.99918	0030|2000|0020|0100|1000|1000|2000|2000|0200|0020|0200|0005|2000|0010|0200|0010|0010|1000|0004|1000|
250	220	0.99978	0010|2000|0005|0200|2000|3000|3000|1000|1000|0030|0100|0005|0300|0005|1000|3000|0020|2000|0030|0100|
250	250	0.99978	0010|2000|0005|0200|2000|3000|3000|1000|1000|0030|0100|0005|0300|0005|1000|3000|0020|2000|0030|0100|

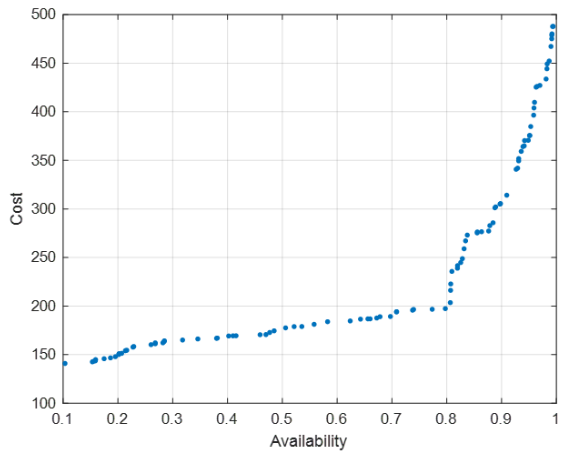


Figure 4: Pareto front $W = 250$

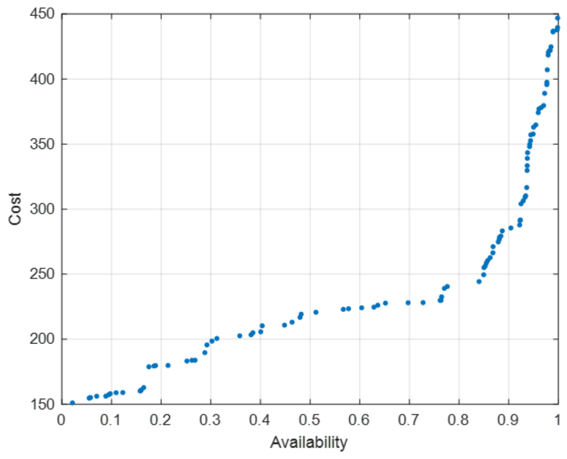


Figure 6: Pareto front $W = 190$

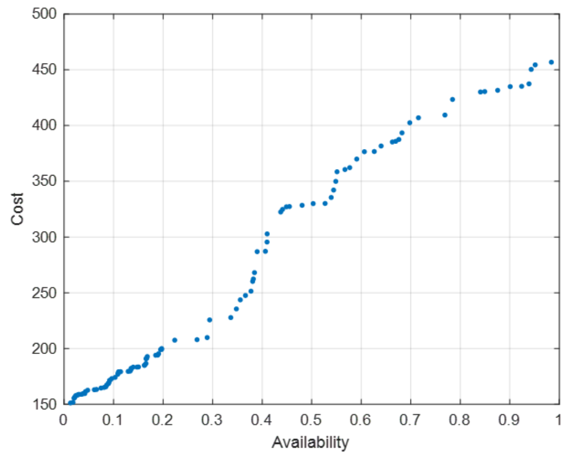


Figure 5: Pareto front $W = 220$

5 Conclusion

In this paper, a series-parallel redundancy allocation problem with the goal of maximizing the total system reliability was considered. The reliabilities of all components within all subsystems were known and the total system cost and weight were bounded. In this problem, while each component could be supplied in different brands, all components were assumed identical within a specific subsystem.

The mathematical formulation of the problem was first derived. Then, a parameter-tuned, in which a local search heuristic was integrated to find better solution, was employed manually to solve the model in different scenarios of weights and costs. A NSGA-II algorithm used to solve multi-objective RAP where the objective was maximization availability and minimization cost. Also, a modified PSO algorithm applied to solve the one objective problem in which goal was maximization availability. Finally, the performance of the proposed algorithm was evaluated using different problem instances and the solutions were compared to the ones obtained using two other meta-heuristic algorithms. The results were generally in favor of the proposed algorithm.

Although this study concentrated on series–parallel structures, the model can be easily extended for other system structures such as star, circular, etc. This investigation is one of the authors' plans for future work. The idea of solving the problem using other algorithms will be investigated in our future work as well. Also, for further studies, non-identical components can be allocated to each subsystem. Also adding more constraint make the problem more realistic. Using other Meta-heuristic algorithms is another suggestion.

References

[1] MS. Chern, On the computational complexity of reliability redundancy allocation in a series system, *Operations Research Letters* 12 (1992) 309-315.

[2] S. Kulturel-Konak, AE. Smith, Efficiently solving the redundancy allocation problem using tabu search, *IIE Transactions* 35 (2003) 515-526.

[3] W. Kuo, V. R. Prasad, An annotated overview of system-reliability optimization,

IEEE Transactions on Reliability 49 (20000 176-187.

[4] FA. Tillman, CL. Hwang, W. Kuo, Optimization techniques for system reliability with redundancy: a review, *IEEE Transactions on Reliability* 26 (1997) 148-155.

[5] W. Kuo, V. R. Prasad, C. L. Hwang, Optimal reliability Design: Fundamentals and Applications, *Cambridge university press*, 2001

[6] C. Ha, W. Kuo, Reliability redundancy allocation: an improved realization for non-convex nonlinear programming problems, *European Journal of Operational Research*, 171 (2006) 24-38.

[7] M. Sharifi, G. Cheragh, K. Dashti Maljaii, A. Zaretalab, M. Shahriari, Reliability and Cost Optimization of a System with k-out-of-n Configuration and Choice of Decreasing the Components Failure Rates, *Scientia Iranica* 28 (2021) 1-16.

[8] M. Sharifi, M. Shahriyari, A. Khajepour, SA. Mirtaheri, Reliability optimization of a k-out-of-n series-parallel system with Warm standby components, *Scientia Iranica* 2021.

[9] M. Sharifi, M. Saadvandi, MR. Shahriari, Presenting a series-parallel redundancy allocation problem with multi-state components using recursive algorithm and meta-heuristic, *Scientia Iranica* 27 (2020) 970-982.

[10] M. Sharifi, MR. Shahriari, A. Zaretalab, The effects of technical and organizational activities on redundancy allocation problem with choice of selecting redundancy strategies using the memetic algorithm, *International Journal of Industrial Mathematics* 11 (2019) 165-176.

[11] M. Sharifi, MR. Shahriari, S. Khoshniat, Optimization the availability of a system with short circuit and common cause failures, *International Journal of Industrial Mathematics* 11 (2019) 239-248.

[12] M. Ouzineb, M. Nourelfath, M. Gendreau, Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems, *Reliability Engineering & System Safety* 93 (2008) 1257-1272.

[13] G. Levitin, A. Lisnianski, H. Ben-Haim, D. Elmakis, Redundancy optimization for series–parallel multistate systems, *IEEE Transactions On Reliability* 47 (1998) 165-172.

[14] N. Nahas, M. Nourelfath, D. Ait-Kadi, Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series–parallel systems, *Reliability Engineering & System Safety* 92 (2007) 211-222.

[15] J. Kennedy, R. C. Eberhart, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks* (1995) 1942-1948.

[16] SH. Liao, CL. Hsieh, A capacitated inventory-location model: formulation, solution approach and preliminary computational results, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5579 (2009) 323-332.

[17] R. Bhattacharya, S. Bandyopadhyay, Solving conflicting bi-objective facility location problem by NSGA II evolutionary algorithm, *The International Journal of Advanced Manufacturing Technology* 51 (2010) 397-414.

[18] X. Yuan, L. Quanfeng, Bicriteria parallel machines scheduling problem with fuzzy due dates based on NSGA-II, *Intelligent Computing and Intelligent Systems* 3 (2010) 520-524.

[19] YH. Kang, Z. Zhang, W. Huang, NSGA-II Algorithms for Multi-Objective Short-Term Hydrothermal Scheduling, *Power and Energy Engineering Conference* (2009) 1-5.

[20] N. Beji, B. Jarboui, M. Eddaly, H. Chabchoub, A hybrid Particle Swarm Optimization Algorithm for the Redundancy Allocation Problem, *Journal of Computational Science* 1 (2010) 159-167.

[21] M. R. Kuiti, N. K. Hazra, M. Finkelstein, A note on the stochastic precedence order between component redundancy and system redundancy for k-out-of-n systems, *Communications in Statistics-Theory and Methods* (2020) 1-9.

[22] N. Mahdavi-Nasab, M. Abouei Ardakan, M. Mohammadi, Water cycle algorithm for solving the reliability-redundancy allocation problem with a choice of redundancy strategies, *Communications in Statistics-Theory and Methods* 49 (2020) 2728-2748.

[23] J. Chen, X. Lai, M. Wang, P. Zhao, Optimal allocation of two redundancies in an-component series system with proportional hazard rates lifetimes, *Communications in Statistics-Theory and Methods* (2020) 1-17.

[24] E. R. Dobani, M. A. Ardakan, H. Davari-Ardakani, M. N. Juybari, RRAP-CM: A new reliability-redundancy allocation problem with heterogeneous components, *Reliability Engineering & System Safety* 191 (2019) 106-119.

[25] Z. Ouyang, Y. Liu, S. J. Ruan, T. Jiang, An improved particle swarm optimization algorithm for reliability-redundancy allocation problem with mixed redundancy strategy and heterogeneous components, *Reliability Engineering & System Safety* 181 (2019) 62-74.

[26] S. Naka, T. Genji, T. Yura, Y. Fukuyama, Practical distribution state estimation using hybrid particle swarm optimization, *Proceedings of the IEEE Power Engineering Society Winter Meeting.*

[27] Y. Shi, RC. Eberhart, Empirical study of particle swarm optimization, *Proceedings of the 1999 Congress on Evolutionary Computation* (1999) 1945-1950.

[28] D. W. Coit, A. E. Smith, Reliability optimization of series-parallel systems using a genetic algorithm, *IEE Transportations On reliability* 45 (1996) 254-260.