# Multi-objective Components Assignment Problem Subject to Lead-time Constraint

#### M. R. Hassan<sup>\*</sup> and H. Abdou

Computer Science Branch, Department of Mathematics, Faculty of Science, Aswan University, Aswan, Egypt; m\_r\_hassan73@yahoo.com, haniabdou2000@yahoo.com

### Abstract

**Objectives**: The study aims to present a multi objective genetic algorithm in order to solve multi-objective components assignment problem subject to lead-time constraints. **Methods/Statistical Analysis**: The study has used non-dominated sorting genetic algorithm II to solve component assignment problems under total lead-time constraints and determine the most optimal solution characterized by a maximum reliability and minimum total lead-time. The proposed method is tested on different examples from the literature to illustrate its efficiency in comparison with a single genetic algorithm. **Findings**: The proposed algorithm succeeded in identifying the optimal solution to the presented problem in comparison with the single genetic algorithm without guessing or determining the initial value for the total lead-time. Moreover, similar observation was identified for the six-node network example. However, no comparison for TANET example was present because there is no literature dealt it for the presented problem. The proposed approach succeeded by obtaining the most optimal solution to the presented problem. **Application/Improvements**: With the help of proposed approach, the system reliability is maximized and total lead-time is minimized. Future researches may focus on other algorithms to improve the reliability and lead-time.

**Keywords:** Disjoint Minimal Paths, Multi-Objective Components Assignment Problem, Stochastic-Flow Network, System Reliability, Total Lead-time

## 1. Introduction

The reliability of a Stochastic Flow Network (SFN) under a time constraint is defined as the probability that the SFN can send the required amount of data from the source to the sink within a specified amount of time<sup>1</sup>. In this case, each link has two attributes: the lead-time and the capacity<sup>2-4</sup>. In<sup>5</sup>, the quickest path problem was modified to evaluate the system reliability such that data could be sent from the source to the sink through two disjoint Minimal Paths (MPs) given a time constraint. On the other hand, in<sup>6</sup>, the reliability was evaluated using spare routing, which is defined as the transmission rule that indicates the first and the second priority pairs of MPs.

The problem of assigning multistate components to a network such that the network reliability is maximal under the transmission budget is addressed in<sup>2</sup>. Component

\*Author for correspondence

Assignment Problems (CAP) given assignment budgets were studied<sup>8</sup>. CAP under total lead-time constraints have been addressed by<sup>2</sup>, and a genetic algorithm has been proposed to search the optimal components that maximize the reliability such that the total lead-time cannot exceed a specified amount.

Multiple objectives are typically conflicting and non-commensurable and must be satisfied simultaneously<sup>10</sup>. These authors proposed a Genetic Algorithms (GA)-based multi-objective optimization technique and applied it on two design problems. A reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. In <sup>11</sup> genetic algorithms presented specifically for problems with multiple objectives. The multi-objective CAP for SFNs has also been addressed by<sup>12</sup>. They proposed a twostage approach to solve the multi-objective problem of reliability maximization and cost minimization by finding the optimal component assignment for SFNs.

In this paper, we use a multi-objective GA to determine the optimal solution to the multi-objective components assignments problem subject to total lead-time and reliability constraints. The proposed GA is based on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) presented by<sup>13</sup>. The NSGA-II is more efficient at solving multi-objective optimization problems<sup>14-16</sup>.

The rest of the paper is organized as follows: Section 2 presents our notations. The problem formulation is presented in section 3. The reliability evaluation under a time constraint is presented in section 4. The proposed multi-objective GA-based NSGA-II appears in section 5. Section 6 includes illustrative examples. Finally, section 7 presents our conclusions.

## 2. Notations

n	Set of nodes.
m	$\{a_e   1 \le e \le m\}$ : set of arcs.
MPs	Minimal paths.
np	Number of minimal paths.
mp <sub>i</sub>	Minimal path no. j; $j = 1, 2,, np$ .
cn	The number of available components.
cp <sub>k</sub>	The components number k, k=1,2,,cn.
l(cp <sub>k</sub> )	Leadtime of components cp <sub>k</sub> .
L	The leadtime of mp <sub>i</sub>
Ŕ <sub>d,T</sub>	The system reliability to the demand d under
time lim	it T.
Х	Capacity vector defined as $X = (x_1, x_2,, x_e)$ .
ps	Population size.
Maxgen	Maximum number of generations

- Generation number. gn
- GA mutation rate. P<sub>m</sub>
- GA crossover rate. p<sub>c</sub>

# 3. Problem Formulation

This section presents the mathematical formulation of the multi-objective optimization problem to maximize system the reliability of a flow network and minimize the total lead-time. Let be a set of available components. Let  $B = \{b_1, b_2, \dots, b_m\}$  be the components assignment in which the component  $cp_{k}$  is assigned to the arc  $a_{k}$  if  $b_{k} = k$ . Let be the total leadtime associated with a specified assignment B. Let be the total lead-time of a MP j. Thus, the mathematical programming formulation of the addressed problem is:

Maximize 
$$R_{d^{n}T}(B)$$
 ...(1)

Minimize 
$$S_l(B)$$
 ...(2)

Subject to be=k,  $k \in \{1, 2, \dots, cn\}$  for  $e=1, 2, \dots, m$ . ... (3)

 $b_{a}b_{u}$ forev ...(4)

Constraints (3) and (4) assert that each link should be given one component and that each component can be assigned to at most one link. All feasible component assignments are generated using constraints (3) and (4). Constraint (5) states that the lead-time of the path  $mp_i$  $(L_i)$  is less than the time limit (T); see<sup>9</sup> for more details about the importance and proof of this constraint. The multi-objective components assignments problem in the case of a maximal and minimal objective is transformed into either a multi-objective minimization problem or a multi-objective maximization problem. Therefore, the original problem formulation is modified to be of the minimal type:

Minimize	$S_{I} = 1 - R_{d^{P}T}(B)$	(7)
Minimize	$S_2 = S_1(B)$	(8)
Subject to		
be=k,	<i>k</i> {1, 2,, <i>cn</i> } for $e = 1, 2,$	, m(9)
$b_{e} b_{v}$	for <i>ev</i>	(10)
, j=1,2,, 1	ıp.	(11)

## 4. Reliability Evaluation given a Lead-time Constraint

Each component  $cp_{k}$  has a maximum capacity  $M^{k}$ . The capacity values of  $cp_k$  vary from  $0-M^k$ . The lead-time of component  $cp_k$  is  $l(cp_k)$ . The system reliability of the candidate chromosome can be evaluated as follows:

- Step 1. Check if the candidate chromosome satisfies constraint (11).
- Step 2. Use the procedure described by<sup>1</sup> to generate the capacity vector X<sup>i</sup> corresponding to path mp<sub>i</sub>.
- Step 3. Calculate the network reliability of the chromosome:  $R_{d,T} = Pr\{\bigcup_{i=1}^{q} \{X | X \ge X^i\}$  using the inclusion-exclusion rule given by<sup>17</sup>.

# 5. The Proposed Approach based on NSGA-II

In this section, we develop an approach to solve the multiobjective optimization problem based on NSGA-II. The NSGA-II is used to determine the highest ranking solution (non-dominated solution) to the problem. The initial inputs include data related to the components such as the total lead-time and reliability, the parameters of NSGA-II, and the network topology.

### 5.1 Non-dominated Sorting Algorithm

This section explains how to determine a set of solutions that dominates other solutions. For example, solution  $S_3$ dominates the solutions  $\{S_1, S_2\}$  if the fitness value of  $S_3$  is smaller than the fitness values of  $S_1$  and  $S_2$ . Likewise, this same procedure can be used to determine if  $S_4$  dominates the solutions  $\{S_1, S_5\}$  and so on. The individual  $B_3$  has a rank of 1 because it is not dominated by any solution. Furthermore,  $B_4$  has a rank 2 and so on. The final sorted ranking of the individuals is  $B_3$ ,  $B_4$ , ..., $B_{ps}$ .

### 5.2 Crowding-distance Assignment

First, sort the solutions for the objective function. Next, set the distance for the first and last solutions (boundary solutions) to infinity. We can calculate the crowding distance (*cdist*) for other solutions by using the following relation:

 $cdist_i = (S_{i+1} - S_{i-1})/(S_{max} - S_{min}), 1 < i < ps.$ 

# 5.3 Crossover, Mutation and Selection Operations

We use the modified uniform crossover and mutation presented in<sup>9</sup> to generate new offspring. The crossover operation is described as follows: Let  $B_1 = (5, 2, 1, 3, 4, 6)$ and  $B_2 = (3, 1, 4, 2, 6, 5)$ be two parents. The new offspring are filled randomly by selecting genes from  $B_1 = (\underline{5}, 2, \underline{1}, 3, \underline{4}, 6)$  and  $B_2 = (3, \underline{1}, 4, \underline{2}, 6, \underline{5})$ . The new offspring will be ( $\underline{5}, \underline{1}, \underline{1}, 2, 4, \underline{5}$ ), the second and the third genes will be equal, and two different genes (3 and 6) are free. We can use these free genes (3 and 6) to replace the second and the sixth genes in the offspring. The offspring will be (5, 3, 1, 2, 4, and 6). Figure 1 shows this crossover process. Swap mutation is used to avoid duplicated genes in a chromosome. Let  $B_1 = (5, 3, 1, 2, 4, 6)$ . Next, the first and fourth genes are selected to swap their values. Then, the new offspring will be  $B_i = (2, 3, 1, 5, 4, 6)$ . Figure 2 shows this mutation process. The selection operation is based on the *crowded-comparison operator*<sup>13</sup>. Each individual  $B_i$  in the population has two attributes: a non-dominated rank  $(i_{rank})$  and a crowding distance  $(i_{distance})$ . Considering two individuals  $B_i$  and  $B_j$ , the crowded-comparison operator is defined as follows:

$$B_i$$
 is better than  $B_j$  if  $(i_{rank} < j_{rank})$   
or  $(i_{rank} = j_{rank})$  and  $(i_{distance} > j_{distance})$ .

That is, NSGA-II prefers the individual with lower rank; otherwise, it prefers the individual that is in a less crowded region.

Parents Offspring The final offspring  

$$(\underline{5}, 2, \underline{1}, 3, \underline{4}, 6) \longrightarrow (\underline{5}, \underline{1} \rightarrow 3, \underline{1}, 2, 4, \underline{5} \rightarrow 6) \longrightarrow (5, 3, 1, 2, 4, 6)$$
  
 $(3, 1, 4, 2, 6, 5)$ 

Figure 1. Modified uniform crossover.

Parents	Offspring	The final offspring
(5, 3, 1, 2, 4, 6) 📥	(5→2, 3, 1, <u>2</u> →5, 4,6)	∟) (2, 3, 1, 5, 4, 6)

Figure 2. Mutation operation.

### 5.4 Determine the Optimal Solution

The final step of the proposed approach is to determine the optimal solutions and, in fact, the most optimal solution. By searching the Pareto solution at each generation, the optimal solution can be determined for each pair of weights and finally the best optimal solution can be determined using simple comparison. The optimal solution determination depending on weights has been well investigated and was used by<sup>12</sup>.

Input  $w_1$  and  $w_2$  for the reliability  $R_{d,T}$  and the total lead-time  $S_p$  respectively. Let  $R_{d,T}(i)$  and  $S_l(i)$  be the corresponding values for the solution i, i=1,2,...,ps. Step 1. Find the normalized values of  $R_{d,T}$  and  $S_l$  as follows: Step 1.1. Normalized value for  $R_{d,T}(i)$ :

 $\begin{aligned} & \text{NR}_{d,T}(i) = \text{R}_{d,T}(i) \ / \text{Max}(\text{R}_{d,T}(1), \text{R}_{d,T}(2), \ \dots, \ \text{R}_{d,T}(ps) \ ) \\ & \text{Step 1.2. Normalized value for } S_{l}(i): \end{aligned}$ 

 $NS_{1}(i) = Min(S_{1}(1), S_{1}(2), ..., S_{1}(ps)) / S_{1}(i)$ 

- Step 2. Calculate the weighted score  $H(i) = w_1 * NR_{d,T}(i) + w_2 * NS_l(i)$
- Step 3. The optimal solution that has the largest *H* (*i*) is selected.

#### 5.5 The Entire Algorithm based on NSGA-II

- Step 1. Set the population size (*ps*), the crossover rate (*pc*), the mutation rate (*pm*) and the number of generations (*gn*).
- Step 2. Generate the initial population including successful individuals  $B_1, B_2, \ldots, B_{ps}$ .
- Step 3. For each individual, evaluate the network unreliability  $S_1 = 1 R_{d^2T}(B)$  and total lead time  $S_2 = S_1(B)$ .
- Step 4. Rank all individuals using a non-dominated, sorting-based method (presented in section 5.1).
- Step 5. Utilize a crowding-distance assignment algorithm (presented in section 5.2) to calculate the crowding distances for all individuals in the population.
- Step 6. Apply the GA operations (described in section 5.3) to generate new populations.
- Step 7. Combine the new population with the original population to determine the intermediate population.
- Step 8. Rank all individuals in the intermediate population using a non-dominated, sorting-based method.
- Step 9. Determine the Pareto set in the current generation.
- Step10. Calculate the crowding distances for all individuals in the intermediate population utilizing the crowding-distance assignment algorithm.
- Step 11. Select *ps* individuals from the intermediate population to be the new population(presented in section 5.3).
- Step 12. If gn<Maxgen go to step 6. Otherwise, the algorithm stops and outputs the best optimal solution for each pair of weights (as shown in section 5.4).

After obtaining a Pareto set, change the network unreliability to be the network reliability for each Pareto solution.

## 6. Illustrative Examples

This section shows the results of applying the proposed approach on two networks. Six nodes and TANET (Taiwan Academic Network). The genetic parameters used in the proposed multi objective GA are ps = 10, Maxgen = 100,  $P_c = 0.95$ , and  $P_m = 0.05$ . The algorithm was iterated 10 times.

### 6.1 Six-node Network Example

The network has six nodes and 10 links (Figure 3), as studied by<sup>2</sup>. The MPs are as follows:

$cp_k$	Capacity				$l(cp_k)$			
	0	1	2	3	4	5	6	
1	0.01	0.00	0.01	0.00	0.01	0.00	0.97	2
2	0.05	0.05	0.05	0.15	0.20	0.50	0	3
3	0.07	0.08	0.00	0.85	0	0	0	2
4	0.70	0.00	0.00	0.00	0.00	0.30	0	2
5	0.01	0.00	0.00	0.05	0.00	0.00	0.94	1
6	0.01	0.00	0.00	0.01	0.00	0.00	0.98	3
7	0.50	0.50	0	0	0	0	0	3
8	0.25	0.25	0.50	0	0	0	0	1
9	0.15	0.25	0.10	0.10	0.10	0.10	0.20	2
10	0.00	0.05	0.05	0.90	0	0	0	2
11	0.01	0.99	0	0	0	0	0	1
12	0.02	0.00	0.05	0.00	0.05	0.00	0.88	1
13	0.07	0.00	0.28	0.00	0.00	0.65	0	3
14	0.05	0.05	0.90	0	0	0	0	2
15	0.60	0.40	0	0	0	0	0	2
16	0.15	0.00	0.00	0.00	0.85	0	0	1
17	0.10	0.10	0.10	0.70	0	0	0	1
18	0.70	0.00	0.00	0.00	0.00	0.30	0	3
19	0.07	0.18	0.75	0	0	0	0	2
20	0.40	0.40	0.20	0	0	0	0	3

Table 1. Arc capacity,	probability,	and lead-time	e for
the 20 available comp	onents		

d,T	Weight	$R_{d,T}(B)$	$S_{l}(B)$		Assigned Components
6,7	(0.9,0.1)	0.981486	12		16 5 11 15 10 19 17 8 12
	(0.5,0.5)	0.828303			19 8 16 12 10 5 17 15 11
	(0.1,0.9)	0.812834			9 8 5 16 14 17 12 3 11
6,9	(0.9,0.1)	0.959305		16	1 11 17 16 2 7 12 10 8
	(0.5,0.5)	0.943836		15	4 16 5 18 19 7 12 15 8
	(0.1,0.9)	0.916763		13	14 19 17 5 12 4 3 11 8

 Table 2. The best optimal solutions to six nodes network

$$\begin{split} & MP_1 = \{a_1, a_4, a_9\}, MP_2 = \{a_1, a_4, a_7, a_8\}, MP_3 = \{a_1, a_5, a_8\}, \\ & MP_4 = \{a_1, a_5, a_7, a_9\}, MP_5 = \{a_1, a_3, a_6, a_8\}, MP_6 = \{a_1, a_3, a_6, a_7, a_9\}, MP_7 = \{a_2, a_6, a_8\}, MP_8 = \{a_2, a_6, a_7, a_9\}, MP_9 = \{a_2, a_3, a_4, a_7, a_8\}, MP_{11} = \{a_2, a_3, a_5, a_7, a_9\}, MP_9 = \{a_2, a_3, a_6, a_8\}, MP_{12} = \{a_2, a_3, a_5, a_7, a_9\}. \\ & MP_{12} = \{a_2, a_3, a_5, a_7, a_9\}. \\ & Table 1 lists the 20 components and associated information. Table 2 lists the best optimal solution found by the proposed approach for this network given different values of$$
*d*,*T* $. \end{split}$ 



Figure 3. The six-node network example.

#### 6.2 The TANET Example

The Taiwan Academic Network (TANET) with two sources and two sinks shown in Figure 4, it has 14 MPs found by <sup>18</sup>. The 14 paths are as follows:

$$\begin{split} MP_1 = & \{a_{10}, a_4, a_{32}, a_{17}, a_{18}, a_{19}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}\}, MP_2 = & \{a_8, a_4, a_{32}, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{22}\}, MP_3 = & \{a_9, a_5, a_{17}, a_{18}, a_{19}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}\}, MP_4 = & \{a_7, a_5, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{22}\}, MP_5 = & \{a_{11}, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}\}, MP_6 = & \{a_9, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{20}, a_{21}, a_{22}\}, MP_7 = & \{a_{10}, a_1, a_{32}, a_{17}, a_{18}, a_{19}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}\}, MP_8 = & \{a_8, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{20}, a_{21}, a_{22}\}, MP_7 = & \{a_{10}, a_1, a_{32}, a_{17}, a_{18}, a_{19}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}\}, MP_8 = & \{a_8, a_1, a_{32}, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{22}, a_{17}, a_{18}, a_{19}, a_{22}, a_{17}, a_{18}, a_{19}, a_{23}, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{22}\}, MP_1 = & \{a_7, a_3, a_{23}, a_{24}, a_{16}\}, MP_{10} = & \{a_7, a_2, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{22}\}, MP_{11} = & \{a_7, a_3, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_{23}, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_{23}, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}\}, MP_{13} = & \{a_8, a_9, a_{23$$

 $a_{3,}a_{23,}a_{24,}a_{25,}a_{26,}a_{27,}a_{33,}a_{30}$ }, and  $MP_{14} = \{a_{8,}a_{3,}a_{23,}a_{24,}a_{25,}a_{26,}a_{27,}a_{33,}a_{31}\}$ .



Figure 4. TANET example.

Table 3 lists 80 available components, with capacities and probabilities taken from<sup>12</sup>; the corresponding lead-time for each component is randomly assigned in this article. Table 4 lists the most optimal solution found by the proposed multi-object GA-based NSGA-II approach.

cp <sub>k</sub>	Capacity					$l(cp_k)$
	0	1	2	3	4	
1	0.0004	0.0392	0.9604	0	0	1
2	0.000512	0.017664	0.203136	0.778688	0	1
3	0.000343	0.013671	0.181629	0.804357	0	1
4	0.015	0.985	0	0	0	2
5	0.0016	0.0768	0.9216	0	0	2
6	0.005929	0	0.142142	0	0.851929	1
7	0.003	0	0.997	0	0	2
8	0.007225	0	0.15555	0	0.837225	1
9	0.005929	0	0.142142	0	0.851929	1
10	0.003	0.997	0	0	0	2
11	0.034	0.966	0	0	0	2
12	0.0036	0.1128	0.8836	0	0	3
13	0.000001	0.000297	0.029403	0.970299	0	2
14	0.000784	0.054432	0.944784	0	0	1
15	0.000225	0.02955	0.970225	0	0	1
16	0.095	0.905	0	0	0	3
17	0.005776	0.140448	0.853776	0	0	3
18	0.000625	0.04875	0.950625	0	0	2
19	0.000729	0.022113	0.223587	0.753571	0	1
20	0.001	0.027	0.243	0.729	0	2
21	0.000512	0.017664	0.203136	0.778688	0	1
22	0.004225	0.12155	0.874225	0	0	3
23	0.005929	0	0.142142	0	0.851929	1
24	0.003	0	0.997	0	0	2
25	0.000216	0.010152	0.159048	0.830584	0	3
26	0.034	0.966	0	0	0	2
27	0.000512	0.017664	0.203136	0.778688	0	2
28	0.000343	0.013671	0.181629	0.804357	0	1
29	0.001	0.027	0.243	0.729	0	3
30	0.0009	0.0582	0.9409	0	0	2
31	0.002809	0.100382	0.896809	0	0	1
32	0.000166375	0.008575875	0.147349125	0.843908625	0	2
33	0.000125	0.007125	0.135375	0.857375	0	2
34	0.0001	0.0198	0.9801	0	0	1
35	0.025	0.975	0	0	0	3
36	0.024	0.976	0	0	0	3
37	0.000125	0.007125	0.135375	0.857375	0	2
38	0.000110592	0.006580224	0.130507776	0.862801408	0	1
39	0.0001	0	0.0198	0	0.9801	1

Table 3. Component information

40	0.001849	0	0.082302	0	0.915849	3
41	0.001024	0.061952	0.937024	0	0	2
42	0.000676	0.050648	0.948676	0	0	2
43	0.007921	0.162158	0.829921	0	0	4
44	0.000512	0.017664	0.203136	0.778688	0	2
45	0.001	0.027	0.243	0.729	0	5
46	0.097	0	0.903	0	0	4
47	0.000001	0.000297	0.029403	0.970299	0	3
48	0.022	0.978	0	0	0	2
49	0.000256	0	0.031488	0	0.968256	1
50	0.001225	0	0.06755	0	0.931225	1
51	0.025	0.975	0	0	0	3
52	0.000274625	0.011851125	0.170473875	0.817400375	0	2
53	0.000529	0	0.044942	0	0.954529	3
54	0.000144	0	0.023712	0	0.976144	1
55	0.000216	0.010152	0.159048	0.830584	0	2
56	0.000117649	0.006850053	0.132946947	0.860085351	0	1
57	0.046	0	0.954	0	0	2
58	0.083	0	0.917	0	0	3
59	0.000015625	0.001828125	0.071296875	0.926859375	0	3
60	0.000274625	0.011851125	0.170473875	0.817400375	0	2
61	0.001369	0.071262	0.927369	0	0	2
62	0.000001	0.000297	0.029403	0.970299	0	2
63	0.000512	0.017664	0.203136	0.778688	0	3
64	0.006084	0.143832	0.850084	0	0	2
65	0.004096	0.119808	0.876096	0	0	5
66	0.003481	0.111038	0.885481	0	0	4
67	0.035	0.965	0	0	0	2
68	0.022	0	0.978	0	0	3
69	0.000166375	0.008575875	0.147349125	0.843908625	0	3
70	0.000042875	0.003546375	0.097778625	0.898632125	0	3
71	0.000024389	0.002449833	0.082027167	0.915498611	0	2
72	0.000324	0	0.035352	0	0.964324	1
73	0.00000343	0.000145971	0.020707029	0.979146657	0	2
74	0.004356	0.123288	0.872356	0	0	3
75	0.055	0.945	0	0	0	2
76	0.001936	0.084128	0.913936	0	0	5
77	0.000035937	0.003159189	0.092573811	0.904231063	0	4
78	0.000484	0	0.043032	0	0.956484	2
79	0.000121	0	0.021758	0	0.978121	1
80	0.001	0.999	0	0	0	2

d,T	Weight	$R_{d,T}(B)$	$S_l(B)$	Assigned Components											
	(0.9,0.1)		55	55	999992 55 32 2 14 80 5 27 54 62 52 7 20 9 79 71 50 60 31 13										
4,16	(0.5,0.5)	0.999992				55	55	<i>32</i> 2 14 80 5 27 54 62 52 78 <i>3</i> 28 38 15 8 19 23 7 20 9 79 71 50 60 31 13 69 11 58 68 75 67 44							
	(0.1,0.9)				7 20 7 77 71 50 00 51 15 07 11 50 00 75 07 44										
	(0.9,0.1)		4612 53												
6,16	(0.5,0.5)	0.994612		53	53	53	53	53	53	53	53	53	53	53	8 51 13 17 45 9 67 15 57 75 38 34 78 50 1 3 19 24 2 61 37 73 14 40 23 55 21 32 39 22 41 7 71
	(0.1,0.9)														

Table 4. The results of applying the proposed approach to the Figure 4 network.

## 7. Discussion

The results obtained in table 2 are compared with that obtained by<sup>9</sup> for the same problem. we found that the results is closed to [9] for d=6 and T=7 at the weights (0.9, 0.1). So, the proposed algorithm succeed in finding the optimal solution to the presented problem in comparison with the single genetic algorithm presented by [9] without guessing or determining the initial value for  $S_p$ . Also, the same observation for d =6 and T=9. But, no comparison for TANET Example because there is no literature dealt it for the presented problem.

## 8. Conclusion

A multi-objective components assignments problem subject to lead-time is presented and formulated as a multi-objective minimization problem. Furthermore, a multi-objective GA-based NSGA-II approach is proposed to solve the presented problem. The proposed approach succeeds by obtaining the most optimal solution to the presented problem, i.e., the system reliability is maximized and the total lead-time is minimized. Examples from the literature are used to explain how to solve the problem and prove the efficiency of the proposed approach. Future researches may focus on other algorithms to improve the reliability and lead-time.

# 9. References

- 1. Lin Y. Extend the quickest path problem to the system reliability evaluation for a stochastic-flow network. Computers and Operation Research. 2003; 30:567–75. Crossref
- 2. Chen YL, Chin YH. The quickest path problem. Computers and Operations Research. 1990; 17(2):153–61. Crossref
- 3. Chen GH, Hung YC. On the quickest path problem. Information Processing Letters. 1993; 46:125–8. Crossref

- Martins VQE, Santos JLE. An algorithm for the quickest path problem. Operations Research Letters. 1997 May; 20(4):195–8. Crossref
- Lin YK. Time version of the shortest path problem in a stochastic-flow network. Journal of Computational and Applied Mathematics. 2009; 228:150–7. Crossref
- Lin YK. On performance evaluation for a multistate network under spare routing. Information Sciences. 2012; 203:73–82. Crossref
- Lin YK, Yeh CT. Evaluation of optimal network reliability under Components-Assignments subject to a transmission budget. Transactions on Reliability. 2010 Sep; 59(3):539–50.
- Lin YK, Yeh CT. Maximizing network reliability for stochastic transportation networks under a budget constraint by using a genetic algorithm. International Journal of Innovative Computing Information and Control ICIC. 2011; 7(12):7033–50.
- 9. Hassan MR. Solving a component assignment problem for a stochastic flow network under lead-time constraint. *Indian Journal of Science and Technology* 2015; 8(35):1-5.
- Coello CA, Christiansen AD. Multi objective optimization of trusses using genetic algorithms. Computers and Structures. 2000; 75:647–60. Crossref
- 11. Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety. 2006; 91:992–1007. Crossref
- 12. Lin YK, Yeh CT. A two-stage approach for a multi-objective component assignment problem for a stochastic-flow network. Engineering Optimization. 2013; 45(3):265–85. Crossref
- Deb K, Pratap K, Agarwal S, Meyarivan T. A fast and elitist multi objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002 Apr; 6(2):182–98. Crossref
- 14. Brownlee AEI, Wright JA. Constrained, mixed-integer and multi-objective optimization of building designs by NSGA-II with fitness approximation. Applied Soft Computing Journal. 2015; 35:114–26. Crossref
- 15. Chatterjee S, Abhishek K, Mahapatra SS, Datta S, Yadav RK. NSGA-II approach of optimization to study the effects

of drilling parameters in AISI-304 stainless steel. Procedia Engineering. 2014; 97:78–84. Crossref

- Pasandideh SHR, Niaki STA, Asadi K. Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: NSGA-II and NRGA. Information Sciences. 2015 Jan; 292:57–74. Crossref
- 17. Xue J. On multistate system analysis. IEEE Transactions on Reliability. 1985; 34(4):329–37.
- Chen SG, Lin YK. Search for all minimal paths in a general large flow network. IEEE Transactions on Reliability. 2012; 61(4):949–56. Crossref.