Indian Journal of Science and Technology, *Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/70455, December 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Solving a Component Assignment Problem for a Stochastic Flow Network under Lead-Time Constraint

M. R. Hassan

Computer Science Branch, Department of Mathematics, Faculty of Science, Aswan University, Aswan, Egypt; m_r_hassan73@yahoo.com

Abstract

Evaluation of system reliability in stochastic flow networks under time constraints depends on the transmission time of minimal paths. The lead-time of a minimal path plays an important role in calculating the transmission time. Component assignments not only affect on the lead-time of a path but also the reliability value. Components assignment problem subject to total lead-time is never discussed. Thus, this paper focuses on solving this problem under total-lead time constraint, in which each component has an assignment lead-time. Subsequently an optimization method based on genetic algorithm is proposed to search the optimal components for a minimum total lead-time that maximizes the system reliability. The mathematical programming formulation for the assignment problem with optimal network reliability subject to total lead-time is formulated and solved by the presented genetic algorithm. The presented algorithm is applied on two given examples with different number of available components to assert its efficiency in solving the given assignment problem.

Keywords: Component Assignment Problem, Genetic Algorithm, Stochastic-Flow Network (SFN), System Reliability, Total Lead-Time

1. Introduction

Lead-time is generally defined as the period of time that customers have to wait before receiving completed products or services from enterprises. The lead-time of each component (arc) plays an important role when searching for the quickest path on a network. Evaluating transmission time of a path depends on three parts: the lead-time of that path, units of data to be sent and the capacity to send it[1–3].

The assignment problems discussed in the literature[4] have identified variations in the assignment problem. Their naming conventions have been developed to make it easier for researchers to develop variations in the assignment problem for a particular application and to find the relevant literature.

Component Assignment Problems (CAP) for SFN under a transmission budget has been studied and solved[5] and the optimal network reliability have been evaluated under component-assignments subject to a transmission budget, in which the transmission cost depends on each component's capacity. The CAP under assignment budget studied in[6] proposed a genetic algorithm to determine the optimal component assignments with maximal network reliability subject to the assignment budget. The multi-objective CAP for SFN has also been addressed[7]. This paper proposed a two-stage approach to solve the multi-objective problem of reliability maximization and cost minimization by finding the optimal component assignment for SFN.

The CAP under total lead-time constraint has never been discussed. Therefore, this paper studies CAP subject to the total lead-time of the assigned components such that the network reliability is maximized. The total lead-time of the assigned components depends on the component's lead-time. That is, both the time spent searching for the optimal components that maximize the system reliability as well as minimize the total lead-time.

**Author for correspondence*

Genetic Algorithms (GAs) are efficient and useful in solving many problems such as system reliability optimization and CAP[5–8]. GAs are also flexible in treating their operators such as crossover, mutation and selection. Thus, we propose a GA to solve the CAP under total lead-time constraint.

2. Notations

n Set of nodes.
m $\{a_e \mid 1 \le e \le m\}$: *set of arcs.*
MPs *Minimal paths.*
np *Number of minimal paths.*
mp_j *Is a minimal path no. j; j = 1, 2, …, np.*
cn *The number of available components.*
cp_k *The components number k, k=1,2,…,cn.*
$l(cp_k)$ *Lead-time of components* cp_k.
L_j *The lead-time of* mp_j.
$R_{d,T}$ *The system reliability to the demand d under time limit T.*
X *Capacity vector defined as* $X = (x_1, x_2, \dots x_e)$.
Popsize *Population size.*
Maxgen *Maximum number of generations.*
gn *Generation number.*
p_m *GA mutation rate.*
p_c *GA crossover rate.*

3. Problem Description and Formulation

The total lead-time of the network components (S_l) is defined by:

$$S_l = \sum_{i=1}^{m} l(a_i) \tag{1}$$

The total lead-time of a minimal path j (L_j) is given by:

$$L_j = \sum_{i=1}^{m} l(a_i)\big|_{a_{i \in mp_j}} \tag{2}$$

The problem is to find the set of optimal components that maximizes $R_{d,T}$ such that S_l does not exceed S_l^0, and L_j does not exceed the time limit T. The mathematical formulation of the CAP under S_l is as follows:

$$\text{Maximize } R_{d,T} \tag{3}$$

Such that

$$S_l \le S_l^0 \tag{4}$$

$$L_j \le T \tag{5}$$

Where j = 1,2, …, m and S_l^0 is the minimum total lead-time.

Constraint 4 means that the set of components with minimum S_l have been assigned to the network arcs. Constraint 5 states that the lead-time of the path mp_j (L_j) is less than the time limit T)

Lemma: Proof that the constraint (5), $L_j = \sum_{i=1}^{m} l(a_i)\big|_{a_{i \in mp_j}} < T$ is important and required to get feasible solutions.

It is known that the transmission time of a path mp_j is given by[9]:

$$\sum_{i=1}^{m} l(a_i)\big|_{a_{i \in mp_j}} + \left|\frac{d}{v}\right|,$$

Where v is the minimal capacity of mp_j.

To find the minimal capacity (v) of mp_j, we use the following inequality:

$$\sum_{i=1}^{m} l(a_i)\big|_{a_{i \in mp_j}} + \left|\frac{d}{v}\right| \le T.$$

By contradiction, assume that L_j is greater than or equal to T, then v will not exist or it will be infinity.

4. The Proposed GA

One of the important steps in GA development is coding, which explains how to represent a problem solution. A chromosome is used for this purpose and is typically encoded as a string to facilitate mutation and crossover operations. A fitness function is a mapping of the chromosomes in a population to their corresponding fitness (performance) values. A more detailed discussion on GAs can be found in the literature[10,11].

In the following subsections, we define the basic operations of our proposed GA:

4.1 Representation

The chromosome *ch* contains *m* fields, where *m* is the number of arcs (components) for the network. Each field in *ch* represents the components number assigned to an arc.

$$ch = (cp_i, cp_j, cp_k, \dots, cp_l)$$

Where cp_i, cp_j, cp_k and cp_l are random component numbers between cp_1 and , and *cn* is the number of components.

This means that the component cp_1 is assigned to arc a_1, the component cp_j is assigned to arc a_2, … and the component cp_l is assigned to arc a_m.

4.2 Crossover

In the proposed algorithm, we apply a uniform crossover[12] approach with simple modification to avoid duplicate genes. In particular, we randomly fill the new chromosome from the two parents by checking if the new gene from one parent exists in the offspring and then replace it with the alternate value from the other parent. The rate of the breeding operation is defined by the value of ***Pc***:

Figure 1 shows that the genes selected from parent 1 are genes 2, 4 and 1 and genes selected from parent 2 are 5, 7 and 4. It is clear that the value of the last gene in the candidate offspring (2, 5, 4, 7, 1, 4) has the same value of the third one.Therefore the last gene value is replaced with 8 selected from parent 2 and the final offspring becomes (2, 5, 4, 7, 1, 8).

4.3 Mutation

Swap mutation is used to avoid duplicate genes in the chromosome. Chromosomes undergo mutation according to mutation rate P_m, as follows:

Generate a random number $r_{m\in[0,1]}$
if $r_m \le p_m$ *then*
{Randomly generate two different numbers between 1 and m.
The two genes in the positions are then selected to exchange their values}.

The following figure explains the mutation process:

4.4 Fitness Function and Selection

The fitness function *E (ch)* is equal to the system reliability $R_{d,T}$ only if the generated chromosome satisfies all constraints. Otherwise, *E (ch)* is set to 0. The algorithm uses the FUSS mechanism to select chromosomes[13]. The fitness function has the following form:

$$E(ch)=\begin{cases} R_{d,T} & \text{if ch satisfies all constaints} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

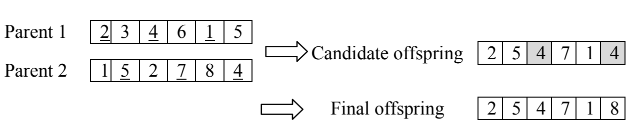

Figure 1. Modified uniform crossover.

5. The Pseudo-Code of the Proposed GA

Start the GA by setting parameters Pc, Pm, Popsize, Maxgen.
Input the network information (n, m,d, T)
Initialize gn = 1, gt = 1;
Generate initial Population;
While gn <= Maxgen, do
while gt <= Popsize, do
Use FUSS to select two chromosomes;
Apply crossover according to Pc;
Apply mutation according to Pm;
***If** the current chromosome satisfies the constraints 4 and 5. **Then** Evaluate it (i.e. Calculate $R_{d,T}$);*
***Else** generate a new one.*
If (E (F) > 0), then gt = gt + 1;
End do;
save the best solution found;
gn:=gn + 1;
replace parents;
End do
End GA

6. System Reliability Evaluation

The system reliability evaluation has been detailed extensively[13]. The same procedure found in[9] can be used to generate the set of lower boundary points $X^1, X^2, \ldots, X^m$ for (d, T). We can then use the inclusion-exclusion rule[15] or the Recursive Sum of Disjoint Products (RSDP)[16] to calculate the reliability. The system reliability $R_{d,T}$ is defined by:

$$R_{d,T} = \Pr\left\{\bigcup_{i=1}^{m}\{Y \mid Y \ge X^i\}\right\} \tag{7}$$

Where

$\Pr\{Y\} = \Pr\{y_1\}\cdot\Pr\{y_2\}\cdot\ldots\cdot\Pr\{y_n\}$. The inclusion-exclusion rule is as follows:

If $A_1 = \{Y \mid Y \ge X^1\}, A_2 = \{Y \mid Y \ge X^2\}, \ldots, A_m = \{Y \mid Y \ge X^m\}$, then

$$R_{d,T} = \sum_i \Pr\{A_i\} - \sum_{i\ne j}\Pr\{A_i \cap A_j\} + \sum_{i\ne j\ne k}\Pr\{A_i \cap A_j \cap A_k\} - \ldots + (-1)^{m-1}\Pr\{A_1 \cap A_2 \cap \ldots \cap A_m\} \tag{8}$$

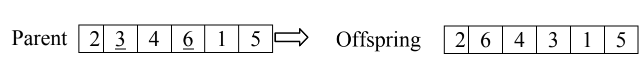

Figure 2. Mutation operation.

7. Illustrative Examples

7.1 Four Node Network Example

Here, we use the network in Figure 3. This network has four nodes and six arcs, which are numbered from a_1 to a_6. The capacity, probability and lead-time of each component (*cp*) is shown in Table 1[9].

There are six minimal paths: $mp_1 = \{a_1,a_2\}$, $mp_2 = \{a_1,a_5,a_8\}$, $mp_3 = \{a_1,a_2,a_6\}$, $mp_4 = \{a_1,a_2,a_7,a_8\}$, $mp_5 = \{a_3,a_6\}$, and $mp_6 = \{a_3,a_7,a_8\}$. Given d = 4 and T = 6, Table 2 lists the assigned components and the best value of system reliability for the first 10 generations when = 9.

The best value of system reliability is $R_{4,6} = 0.966285$ at generation number 5. Table 3 summarizes the best solutions found with the proposed GA for different values of (d, T) and $S_l^0 = 9$;

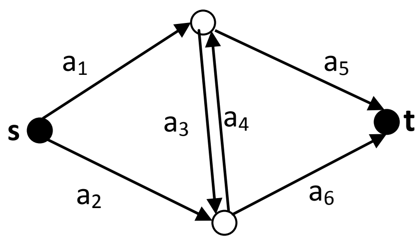

Figure 3. Computer network with 4 nodes and 6 arcs.

Table 1. Components capacities, probabilities and lead-time

cp	Capacity	Probability	Lead-time	*cp*	Capacity	Probability	Lead-time
cp_1	3	0.80	2	cp_6	4	0.60	2
	2	0.10			3	0.20	
	1	0.05			2	0.10	
	0	0.05			1	0.05	
					0	0.05	
cp_2	3	0.80	1	cp_7	5	0.55	2
	2	0.10			4	0.10	
	1	0.05			3	0.10	
	0	0.05			2	0.10	
cp_3	2	0.85	3		1	0.10	
	1	0.10			0	0.05	
	0	0.05					
cp_4	1	0.90	3	cp_8	3	0.80	1
	0	0.10			2	0.10	
cp_5	1	0.90	1		1	0.05	
	0	0.10			0	0.05	

Note: The genetic parameters used in the proposed GA are Popsize = 10, Maxgen = 100, $P_c = 0.95$, and $P_m = 0.05$. The algorithm is iterated 10 times.

7.2 Six Node Network Example

The network has six nodes and ten links (Figure 4). The MPs are as follows:

$MP_1 = \{a_1, a_4, a_9\}$, $MP_2 = \{a_1, a_4, a_7, a_8\}$, $MP_3 = \{a_1, a_5, a_8\}$, $MP_4 = \{a_1, a_5, a_7, a_9\}$, $MP_5 = \{a_1, a_3, a_6, a_8\}$, $MP_6 = \{a_1, a_3, a_6, a_7, a_9\}$, $MP_7 = \{a_2, a_6, a_8\}$, $MP_8 = \{a_2, a_6, a_7, a_9\}$, $MP_9 = \{a_2, a_3, a_4, a_9\}$, $MP_{10} = \{a_2, a_3, a_4, a_7, a_8\}$, $MP_{11} = \{a_2,a_3, a_5, a_8\}$, and $MP_{12} = \{a_2,a_3, a_5, a_7, a_9\}$. Table 4 shows the 20 components and associated information. Table 5 shows the results of applying the GA to the network given in Figure 2 for different values of (d, T) and minimum lead-time (S_l^0) equal to 14 and 15.

Table 2. The results of the first 10 generations when $S_l^0 = 9$

Gen. No.	Assigned Components	$R_{4,6}$	*Gen. No.*	Assigned Components	$R_{4,6}$
1	(8 6 5 1 2 7)	0.858000	6	(7 8 2 1 6 5)	0.965925
2	(8 5 7 1 2 6)	0.955350	7	(5 8 7 1 2 6)	0.765000
3	(8 7 5 1 2 6)	0.858000	8	(8 5 7 1 6 2)	0.955350
4	(8 6 7 1 2 5)	0.81000	9	(8 1 7 6 2 5)	0.810000
5	(7 8 2 1 6 5)	0.965925	10	(8 6 1 7 2 5)	0.765000

Table 3. The best solutions for different values of (d, T) and $S_l^0 = 9$

d,T	S_l^0	Assigned Components	$R_{d,T}$	S_l
4,6	9	(2 6 7 5 8 1)	0.983875	9
4,7		(2 7 5 1 6 8)	0.993725	9
4,8		-	-	-
4,9		(6 2 7 5 1 8)	0.994668	9

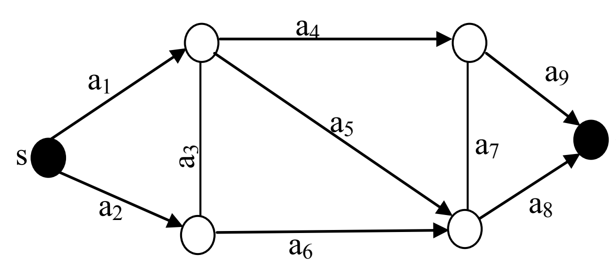

Figure 4. The six node network example.

Table 4. Arc Capacity, probability and lead-time for 20 available components

CPn	Capacities							lead-time
	0	1	2	3	4	5	6	
cp_1	0.01	0.00	0.01	0.00	0.01	0.00	0.97	2
cp_2	0.05	0.05	0.05	0.15	0.20	0.50	0	3
cp_3	0.07	0.08	0.00	0.85	0	0	0	2
cp_4	0.70	0.00	0.00	0.00	0.00	0.30	0	2
cp_5	0.01	0.00	0.00	0.05	0.00	0.00	0.94	1
cp_6	0.01	0.00	0.00	0.01	0.00	0.00	0.98	3
cp_7	0.50	0.50	0	0	0	0	0	3
cp_8	0.25	0.25	0.50	0	0	0	0	1
cp_9	0.15	0.25	0.10	0.10	0.10	0.10	0.20	2
cp_{10}	0.00	0.05	0.05	0.90	0	0	0	2
cp_{11}	0.01	0.99	0	0	0	0	0	1
cp_{12}	0.02	0.00	0.05	0.00	0.05	0.00	0.88	1
cp_{13}	0.07	0.00	0.28	0.00	0.00	0.65	0	3
cp_{14}	0.05	0.05	0.90	0	0	0	0	2
cp_{15}	0.60	0.40	0	0	0	0	0	2
cp_{16}	0.15	0.00	0.00	0.00	0.85	0	0	1
cp_{17}	0.10	0.10	0.10	0.70	0	0	0	1
cp_{18}	0.70	0.00	0.00	0.00	0.00	0.30	0	3
cp_{19}	0.07	0.18	0.75	0	0	0	0	2
cp_{20}	0.40	0.40	0.20	0	0	0	0	3

Table 5. The results of applying the GA to the Figure 4 network

d,T	S_l^0	Assigned Components	$R_{d,T}$	S_l
6,7	14	(16 5 11 15 10 19 17 8 12)	0.981486	12
6,8		(4 15 5 1 19 3 16 11 12)	0.989187	13
6,9	15	(8 5 12 2 10 19 11 14 13)	0.989316	15
8,9		(1 14 17 11 19 12 3 5 8)	0.979129	13

8. Conclusions and Future Work

We used a genetic algorithm to search the optimal components assigned to the network components with minimum total lead-time such that the system reliability is maximized. Examples from the literature are used to explain how to solve the problem and assert the efficiency of the proposed GA. Future work will use a multi-objective GA to solve the component assignment problem subject to total lead-time.

9. References

1. ChenYL, Chin YH. The quickest path problem. Computers and Operations Research. 1990; 17(2):153–61.
2. Chen GH, Hung YC. On the quickest path problem. Information Processing Letters. 1993 Jun; 46(3):125–8.
3. De Queiros Vieira Martins E, Esteves dos Santos JL. An algorithm for the quickest path problem. Operations Research Letters. 1997; 20(4):195–98
4. Pentico DW. Assignment problems: A golden anniversary survey. Discrete Optimization. European Journal of Operational Research. 2007 Jan; 176(2):774–93.
5. 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.
6. 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 Dec; 7(12):7033–50.
7. 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.
8. Lin YK, Yeh CT, Huang PS. A hybrid ant-tabu algorithm for solving a multistate flow network reliability maximization problem. Applied Soft Computing. 2013 Aug; 13(8):3529–43.
9. Lin YK. Extend the quickest path problem to the system reliability evaluation for a stochastic-flow network. Computers and Operation Research. 2003 Apr; 30(4):567–75.
10. Gen M, Cheng R. Genetic algorithms and engineering optimization. Wiley Series in Engineering, Design and Automation, 1999 Dec.
11. Bagheri S, Effatnejad R, Salami A. Transformer winding parameter identification based on frequency response analysis using hybrid Wavelet Transform (WT) and Simulated Annealing Algorithm (SA) and compare with Genetic Algorithm (GA). Indian journal of Science and Technology. 2014 May; 7(5):614–21.
12. Altiparmark F, Dengiz B, Smith AE. Reliability optimization of computer communication networks using genetic algorithms. Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics; San Diego, California, USA. 1998 Oct 11-14; 5:p. 4676–81.
13. Hutter M, Legg S. Fitness uniform optimization. IEEE Transactions on Evolutionary Computation. 2006 Oct; 10(5):568–9.
14. Peixin Z, Xin Z. A survey on reliability evaluation of stochastic-flow networks in terms of minimal paths. International Conference on Information Engineering and Computer Science, ICIECS'09; Wuhan. 2009 Dec 19-20. p. 1–4.
15. Janan X. On multistate system analysis. IEEE Transactions on Reliability. 1985 Oct; 34(4):329–37.
16. Zuo MJ, Tian Z, Huang HZ. An efficient method for reliability evaluation of multistate networks given all minimal path vectors. IIE Transactions. 2007; 39(8):811–7.