ISSN 2277-3061

Genetic Algorithm for solving flow problems in a Stochastic-flow Network under Budget Constraints

M. R. Hassan
Computer Science Branch, Mathematics Department,
Faculty of Science, Aswan University,
Aswan, Egypt
E-mail: m_r_hassan73@yahoo.com

ABSTRACT

The system reliability ($R_{d,c}$) calculation is based on the generation of all the lower boundary points for the given demand, d, such that the total transmission cost is less than or equal to c (budget). This generation of the lower boundary points is based on finding all feasible solutions to the flow vector, F. The number of feasible solutions cannot be anticipated. Thus, a genetic algorithm is useful for solving this problem. Therefore, this paper presents a genetic algorithm to generate all feasible solutions to the flow vector, F, and then calculate $R_{d,c}$. The algorithm is applied to two problem models. In the first, each arc of a flow network has several capacities and may fail**.** In the second, both the arc and node have several capacities and may fail**.** The results show that the algorithm is efficient at solving a flow problem and is useful for calculating $R_{d,c}$.

Indexing terms/Keywords

Genetic algorithms; budget constraint; minimal path; stochastic-flow network; system reliability.

Academic Discipline And Sub-Disciplines

Computer Science and Engineering.

SUBJECT CLASSIFICATION

Flow Networks; System Reliability;

TYPE (METHOD/APPROACH)

Theory.

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 12, No. 8
editor@cirworld.com
www.cirworld.com, www.ijctonline.com

ISSN 2277-3061

INTRODUCTION

The system reliability of a flow network, R_d, is the probability that the maximum network flow is not less than a single commodity d. In a case where each arc has several capacities and may fail, Lin et al. [1] presented an algorithm to evaluate R_d in terms of minimal path sets. In [2], Lin presented an algorithm to evaluate R_d in a case where both the arc and node have several capacities and may fail. A flow network with two commodities was studied in [3]. In [4], an algorithm based on minimal cuts presented to evaluate the reliability of a multiple commodity capacitated-flow network under budget constraints.

The system reliability of a flow network, $R_{d,c}$, is defined as the probability that a single commodity d can be transmitted from the source node to the sink node such that the total transmission cost is less than or equal to c. This can be computed in terms of the minimal path vectors to level (d,c) (called (d,c)-MPs). In [5], Lin presented an algorithm to generate all of the (d,c)-MPs of such a system for each level (d,c) in terms of the minimal path sets, while considering that each arc has several capacities and may fail. In a case where each node and arc with a designated capacity have different lower levels caused by various partial and complete failures, Lin [6], based on minimal paths, proposed an efficient algorithm to generate all the lower boundary points for (d,c). The system reliability can then be calculated in terms of all of these lower boundary points for (d,c) by applying the inclusion-exclusion rule. The system reliability, $R_{d,c}$, of a multicommodity flow network was studied in [7] and [8].

Genetic algorithms (GAs) have been applied to various problems related to networks [9–10]. The network reliability under a component assignment can be computed in terms of the minimal paths and state-space decomposition. Lin and Yeh [11] proposed an optimization method based on a genetic algorithm to maximize network reliability subject to the transmission budget. Hassan and Younes [12] presented a genetic algorithm to calculate the system reliability of a stochastic-flow network for a given demand, d. Lin and Yeh [13] proposed a GA based algorithm to determine the optimal components assignment with maximal network reliability subject to the assignment budget.

In [1–8] and [11], the authors used the implicit enumeration method to find all of the feasible solutions for a flow vector, which is not efficient for a large network (for more details about this method, see [14]). The authors in [12] used a genetic algorithm to calculate R_d. In this paper, a genetic algorithm (GA) is proposed to find the set of all of the feasible solutions for a flow vector under budget constraints. The proposed GA is based on minimal paths (MPs) to find all the lower boundary points for (d,c) and then calculate the system reliability, $R_{d,c}$.

NOTATIONS AND ASSUMPTIONS

Notations

G(A, N, M) Stochastic-flow network with a set of arcs, **A =** $\{a_i \mid 1 \le i \le n\}$, a set of nodes, $N = \{a_i \mid n+1 \le i \le n+p\}$ and $M = \{M^1, M^2, ..., M^{n+p}\}$, with M^i (an integer) being the maximum capacity of each component, a_i (arc or node)

X Capacity vector, X = (x_1, x_2, …, x_{n+p})

F Flow vector, F = (f_1, f_2, …, f_m)

MPs Minimal paths

mp_j Minimal path no. j, j = 1, 2, …, m

$\mathbf{L_j}$ Maximum capacity of mp_j, $L_j = \min\{M^i \mid a_i \in mp_j\}$

$R_{d,c}$ System reliability for a given demand, d

Popsize Population size

Maxgen Maximum number of generations

p_m GA mutation rate

p_c GA crossover rate

Assumptions

The capacity of each component, a_i **, is an integer-valued random variable that** $0 < a_i < M^i$.

1- The flow-conservation law must be satisfied.

2- The capacity of each component is statistically independent.

PROBLEM DESCRIPTION AND FORMULATION

The main purpose of the GA proposed in this paper is to find the set of all feasible solutions of F that satisfies the flowing constraints:

$$\sum_{j=1}^{m} f_j = d \qquad (1)$$

$$f_j \le L_j \text{ for each j = 1, 2, …, m} \qquad (2)$$

$$\sum_{\mathbf{j}=1}^{\mathbf{m}} \{\mathbf{f_j} \mid \mathbf{a_i} \in \mathbf{mp_j}\} \le \mathbf{M^i} \text{ for each i = 1, 2, …, n+p} \qquad (3)$$

$$\sum_{j=1}^{m} W_j f_j \le C \qquad (4)$$

where $W_j = \sum_i c_i \mid a_i \in mp_j$ for each j = 1, 2, …, m, and C is the total transmission cost (budget).

X (X is a (d,c)-mp) can be deduced from F = (f_1, f_2, …, f_m) by using the following equation:

$$\mathbf{x_i} = \sum_{\mathbf{j}=1}^{\mathbf{m}} \{\mathbf{f_j} \mid \mathbf{a_i} \in \mathbf{mp_j}\} \text{ for each i = 1, 2, …, n.} \qquad (5)$$

If X^1, X^2,…, X^q are the collection of all (d,c)-mps, then the system reliability, $R_{d,c}$, is defined as:

$$R_{d,C} = \Pr\{\bigcup_{i=1}^{q} \{Y \mid Y \ge X^i\}\} \qquad (6)$$

where $\Pr\{Y\} = \Pr\{y_1\} \cdot \Pr\{y_2\} \cdot \ldots \cdot \Pr\{y_n\}$. We will use the inclusion-exclusion rule presented in [14] to calculate $R_{d,c}$, as follows.

If $A_1 = \{Y \mid Y \ge X^1\}, A_2 = \{Y \mid Y \ge X^2\}, \ldots, A_q = \{Y \mid Y \ge X^q\}$, then apply the inclusion-exclusion rule to calculate $R_{d,c}$ by using the relation**:**

$$R_{d,C} = \sum_{i} \Pr\{A_i\} - \sum_{i \ne j} \Pr\{A_i \cap A_j\} + \sum_{i \ne j \ne k} \Pr\{A_i \cap A_j \cap A_k\} - \ldots + \\ + (-1)^{q-1} \Pr\{A_1 \cap A_2 \cap \ldots \cap A_q\}$$

PROPOSED GENETIC ALGORITHM

This section describes the different components of the proposed GA.

Representation

If the network has m minimal paths, then chromosome F has m fields, where each field represents the (current) flow on a given path, i.e.,

F **=** (f_1, f_2, …, f_m) and f_j is the current flow on mp_j,

where f_j is a nonnegative integer and $f_j \le L_j$, j=1,2,…m.

Initial population

The initial population is generated using the following algorithm:

While (i <=Popsize)

Do

For j = 1 to m do

f_j = random number between 0 and L_j

End for

If the generated F satisfy constraint 1 then increase i; otherwise generate another.

End Do

Crossover operator

In the proposed GA, one-cut point crossover is used to breed two offspring (two new chromosomes, F_{N1} and F_{N2}) from two parents (F_{P1} and F_{P2}) selected randomly according to the Pc value, as follows:

$$F_{N1} = [F_{P1}(\mathrm{j})]_{\mathrm{j}=1}^{\mathrm{rn}} + [F_{P2}(\mathrm{j})]_{\mathrm{j}=\mathrm{rn}+1}^{\mathrm{m}}$$

$$F_{N2} = [F_{P2}(\mathrm{j})]_{\mathrm{j}=1}^{\mathrm{rn}} + [F_{P1}(\mathrm{j})]_{\mathrm{j}=\mathrm{rn}+1}^{\mathrm{m}}$$

where **rn** is an integer value randomly generated in the range (0, m-1) and m is the length of the chromosome.

Mutation operator

A chromosome, F, undergoes mutation according to the mutation probability, P_m, as follows:

for j = 1 to m do

Begin;

randomly choose one component f_i

if (f_j == v), then set f_j = k; k $\in$ {0,1,…,L_j} and k $\neq$ v

End do.

Objective function

The problem can be formulated as:

Find the set of all feasible solutions, F, such that equations 1, 3, and 4 have been satisfied.

PSEUDO CODE OF ALGORITHM

This section presents the pseudo code of the proposed GA for computing the system reliability ($R_{d,c}$) of a stochastic-flow network with budget constraints. We call the algorithm RDC-GA:

```
Begin RDC-GA
    Input parameters: Pc, Pm, Popsize, Maxgen;
        gn = 0, gt = 0;
        Initialize P(gn);
        while gn < Maxgen, do
            while gt < Popsize, do
             Randomly select two chromosomes;
             Apply crossover according to Pc;
             Apply mutation according to Pm;
             Evaluate the current chromosome;
             If the current chromosome satisfies constraints 1, 2, and 4, then gt = gt + 1;
            End do;
         save p(gn);
```

gn:=gn + 1;

Keep current population p(gn) to the next generation;

End do

Output all obtained solutions (F) and then calculate $R_{d,c.}$

End RDC-GA

EXPERIMENTAL RESULTS

This section shows how to use RDC-GA algorithm to calculate the system reliability of a stochastic-flow network under budget constraints. The RDC-GA has been applied to two different models.

Model 1 example

This model was studied by J-S Lin in [5]. He supposed that the network arc has several capacities and may fail. The network shown in Fig. 1 has 4 nodes and 6 arcs. The arcs are numbered from a_1 to a_6. The probability distributions of the arc capacities are listed in Table 1.

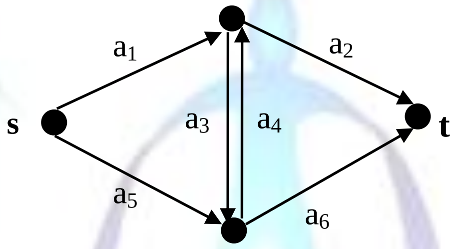


Fig1: Computer network of model 1 example.

Table 1: Arc capacities and probabilities.

Arc	Capacity	Probability
a_1	3	0.60
	2	0.25
	1	0.10
	0	0.05
a_2	2	0.70
	1	0.20
	0	0.10
a_3-a_4	1	0.90
	0	0.10
a_5	1	0.80
	0	0.20
a_6	2	0.70
	1	0.20
	0	0.10

Table 2 summarizes the $R_{d,c}$ values obtained by the RDC-GA algorithm using different values for d and c. The obtained results are compared with those obtained in [5] using a heuristic method.

ISSN 2277-3061

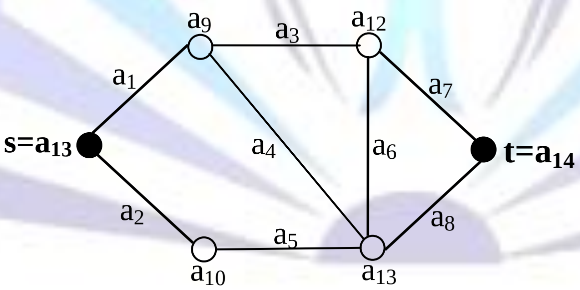
Table 2: $R_{d,c}$ values obtained by RDC-GA in comparison with [5].

$R_{d,c}$	Algorithm used	c = 6	c =10	c = 14	c = 18	c = 22
d = 1	J.-S. Lin	0.97479	0.97803	0.97803	0.97803	0.97803
	RDC-GA	**0.97803**	0.97803	0.97803	0.97803	0.97803
d = 2	J.-S. Lin	0.00000	0.59500	0.86266	0.86266	0.86266
	RDC-GA	0.00000	**0.71136**	**0.83529**	**0.83529**	**0.83529**
d = 3	J.-S. Lin	0.00000	0.00000	0.00000	0.58212	0.58212
	RDC-GA	0.00000	0.00000	**0.514080**	0.58212	0.58212
d = 4	J.-S. Lin	0.00000	0.00000	0.00000	0.00000	0.21168
	RDC-GA	0.00000	0.00000	0.00000	0.00000	0.21168

The results obtained by the RDC-GA algorithm that are shown in bold are different from those obtained by J-S Lin [4]. These refer to the values of W_j that were calculated incorrectly in [4], according to the information given in Table 1. It is easy to verify that the correct values of W_j are as follows: $W_1 = 5$, $W_2 = 6$, $W_3 = 4$, and $W_4 = 3$. Thus, we are sure that the results obtained by the RDC-GEN algorithm are correct. If we use the same values of W_j ($W_1 = 5$, $W_2 = 6$, $W_3 = 7$, and $W_4 = 6$) that are listed in [4], we will get the same results.

Model 2 example

This model was studied by Y-K Lin in [6]. The example network shown in Fig. 2, this network has 6 nodes and 8 arcs. The arcs are numbered from a_1 to a_8, whereas the nodes are numbered from a_9 to a_{14}. For more details about this example, please refer to [6]. Table 3 shows the results of $R_{d,c}$ obtained by RDC-GA in comparison with [6].



ig 2: Computer network of model 2 example.

Table 3: $R_{d,c}$ values obtained by RDC-GA in comparison with [6].

$R_{d,c}$	Algorithm used	c = 900	c = 920
d = 9	Y-K Lin	0.602561	0.634667
	RDC-GA	0.602561	0.634038

The genetic parameters used in the proposed algorithm are Popsize = 20, Maxgen = 500, $P_c = 0.95$, and $P_m = 0.05$.

Discussion and Time Analysis

The results obtained by the RDC-GA algorithm were found at the first generation, which showed that the number of feasible solutions did not exceed the Popsize value for all studied cases. However, if the $R_{d,c}$ value is equal to 0, the algorithm reaches Maxgen, which shows that there is no solution left. Tables 4 and 5 show the values of F and X of models 1 and 2, respectively. It is clear from Tables 4 and 5 that the number of solutions varied from 2 to 7, and were all less than the Popsize value.

Table 4: Values of F and X for model 1 example.

(d,c) values	Flow vector, F	Set of lower boundary points, X
(1,6), (1,10), …, (1, 22)	0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0	1 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0
(2,10)	0 1 1 0 1 0 1 0 0 1 0 1 1 0 0 1	1 2 0 1 1 0 1 0 1 0 1 2 1 1 0 0 1 1
(2,14),(2,18), and (2,22)	0 1 1 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 1	2 1 1 0 0 1 1 2 0 1 1 0 1 0 1 0 1 2 1 1 0 0 1 1

Table 5: Values of F and X for model 2 example.

(d,c) values	Flow vector, F	Set of lower boundary points, X
(9,900)	4 0 0 0 5 0 0 4 0 1 0 4 0 0	4 5 4 0 5 0 4 5 4 5 4 5 5 4 4 1 4 0 4 5 5 4 4 5
(9,920)	3 0 1 0 4 1 0 3 0 0 1 5 0 0 3 0 2 0 3 1 0 4 0 0 0 5 0 0 4 0 1 0 4 0 0 3 0 1 1 4 0 0 5 0 0 0 4 0 0	4 5 3 1 5 1 4 5 4 5 4 6 5 4 3 2 4 1 4 5 5 4 4 6 4 5 4 0 5 0 4 5 4 5 4 5 5 4 4 1 4 0 4 5 5 4 4 5 5 4 5 0 4 0 5 4 5 4 5 4

The following table shows the time taken by RDC-GA algorithm for each $R_{d,c}$ value, the time has calculated in Seconds.

Table 6: The time in Seconds taken by RDC-GA for each $R_{d,c}$ value.

$R_{d,c}$	CPU time in Sec	$R_{d,c}$	CPU time in Sec	$R_{d,c}$	CPU time in Sec	$R_{d,c}$	CPU time in Sec
$R_{1,6}$	0.56	$R_{2,6}$	0.50	$R_{3,6}$	0.56	$R_{4,6}$	0.23
$R_{1,10}$	0.46	$R_{2,10}$	0.67	$R_{3,10}$	0.41	$R_{4,10}$	0.73
$R_{1,14}$	0.44	$R_{2,14}$	0.48	$R_{3,14}$	1.0	$R_{4,14}$	0.63
$R_{1,18}$	0.22	$R_{2,18}$	0.62	$R_{3,18}$	1.2	$R_{4,18}$	0.83
$R_{1,22}$	0.63	$R_{2,22}$	0.21	$R_{3,22}$	0.60	$R_{4,22}$	0.76

CONCLUSIONS AND FUTURE WORK

This paper proposed a genetic algorithm to calculate the system reliability $R_{d,c}$ of a flow network. The algorithm was based on determining the set of all feasible solutions for a flow vector that satisfies the budget constraint; generating the set of all lower boundary points for the given d and budget, c; and then calculating $R_{d,c}$.

Finally, we illustrated the use of the proposed algorithm by calculating the reliability of a flow network for two model networks taken from the literature. Using this algorithm can save much time compared to heuristic methods that use an enumeration method to find the set of all of the feasible solutions for a flow vector when the network's size is large.

Moreover, the algorithm is efficient and may be extended to compute the system reliability, $R_{d,c}$, of a flow network in a multicommodity case.

REFERENCES

[1] Lin, JS., Jane, CC., and Yuan J.(1995). "On Reliability Evaluation of a Capacitated-Flow Network in Terms of Minimal Path sets", NETWORKS, 25, 131-138.

[2] Lin, YK. (2001). "A simple algorithm for reliability evaluation of a stochastic-flow network with node failure", Computers and Operations research, 28(13), 1277-1285.

[3] Lin, YK. (2002). "Two-commodity reliability evaluation for a stochastic-flow network with node failure", Computers & Operations Research, 29(13), 1927-1939.

[4] Lin, YK. (2008). "Multiple Commodity Flow Problem with Unreliable Nodes subject to Budget Constraints", International Journal of Performability Engineering, 4(1), 31- 42.

[5] Lin, JS. (1998), "Reliability evaluation of capacitated-flow networks with budget constraints," IIE Transactions, 30(12), 1175-1180.

[6] Lin, YK. (2004). "Reliability of a Stochastic-Flow Network With Unreliable Branches & Nodes Under Budget Constraints", IEEE Transactions on Reliability, 53(3), 381-387.

[7] Lin, YK. (2007). "Reliability evaluation for an information network with node failure under cost constraint," IEEE Transactions on Systems, Man, and Cybernetics, Part A: System and Humans, 37(2), 180-188.

[8] Lin, YK. (2007). "System Reliability of a Limited-Flow Network in Multicommodity Case," IEEE Transactions on Reliability, 56(1), 17-25.

[9] Mitsuo, G. and Runwei, C.(2000). "Genetic Algorithms and Engineering Optimization," Wiley Series In Engineering Design and Automation

[10] Lo, CC. and Chang WH. (2000) "A multiobjective hybrid genetic algorithm for the capacitated multipoint network design problem," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 30(3), 461-469.

[11] Lin, YK. and Yeh, CT. (2010). "Evaluation of Optimal Network Reliability Under Components-Assignments Subject to a Transmission Budget," IEEE Transactions on Reliability, 59(3), 539-550.

[12] Younes, A. and Hassan, M. R. (2011)."A Genetic Algorithm for Reliability Evaluation of a Stochastic-Flow Network With Node Failure," International Journal of Computer Science and Security (IJCSS), 4(6), 528-537.

[13] Lin, YK. and Yeh, CT. (2011). "Multistate components assignment problem with optimal networkreliability subject to assignment budget", Applied Mathematics and Computation, 217, 10074–10086.

[14] Horowitz, E. and Sahni, S. (1978). "Fundamentals of Computer Algorithms," Computer Science Press.

[15] Janan, X. (1985). "On Multistate System Analysis," IEEE Transactions on Reliability, 34(4), 329-337.