Solving Flow Allocation Problems and Optimizing System Reliability of Multisource Multisink Stochastic Flow Network

Moatamad Hassan
Department of Mathematics, Aswan University, Egypt

Abstract: *Flow allocation problem is one of the important steps in reliability evaluation or optimization of a stochastic flow network. In a single source single sink networks it is easy to determine the flow on each path by using one of best known methods. While, in the case of multisource multisink flow network the flow allocation problem becomes more complicated and few studies have dealt with it. This paper investigates the flow allocation problem of multisource multisink stochastic-flow network with assuming that there are several sorts of resource flows transmitting through that network with unreliable nodes. The mathematical formulation of the problem is modified to increase the efficiency of obtaining optimal solutions that satisfy all constraints. A Genetic Algorithm (GA) is proposed to solve the flow allocation problem in existing multisource multisink networks such that the reliability of the system capacity vector is maximized. The results obtained for test cases are compared with other proposed methods to show that the proposed algorithm is efficient in obtaining optimal solutions that satisfy all constraints, and it achieves a maximum value of reliability of the system capacity vector. Finally, the proposed GA has employed to optimize the system reliability of multisource multisink stochastic flow networks.*

Keywords: *Flow allocation problem, stochastic-flow network, GA.*

Received August 26, 2013; accepted February 4, 2015; published online October 29, 2015

1. Introduction

The paper addresses the capacitated-flow networks, whose capacities of arcs are independent, limited, and integer-valued random variables. For these types of flow networks, several different approaches have been proposed to compute the probability that the system capacity level (i.e., the maximal flow of the system) is not less than a single commodity [8, 9, 10, 16, 18, 26]. In particular, Lin *et al*. [9, 18, 26] evaluated the reliability of flow networks with a single commodity and budget constraints, while in [16], the reliability has been evaluated under cost constraint. The reliability of flow networks with multiple commodities has been studied in [20, 21]. Stochastic-flow networks has been studied in [8, 9, 10, 16, 18, 19, 20, 21, 26] are single-source single-sink networks.

System reliability evaluation in the case of existing bi-directed arcs has been studied in [17], where an algorithm presented to evaluate the reliability of an overall-terminal multistate flow network whose arcs are all bi-directed in terms of Minimal Paths (MPs). The system reliability under time constraints has been studied in [12, 15, 19].

Recently, Genetic Algorithms (GAs) have been used to solve many problems related to the reliability of flow networks. Lin and Yeh [14] proposed a GA based algorithm to determine the optimal components assignment with maximal network reliability subject to the assignment budget. In [27], a GA is used to calculate the reliability of a flow network with unreliable nodes. Lin and Yeh [11] introduced a GA to evaluate optimal network reliability under components assignments subject to a transmission budget, where the transmission cost depends on the capacity of each component. The algorithm in [11] depends on searching the best set of components to maximize the reliability. Lin and Yah [13] proposed a two-stage Non-dominated Sorting Genetic algorithm II (NSGA-II) and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) based approach to solve the problem of finding the optimal transmission line assignment with maximal network reliability and minimal cost for an stochastic computer network in which each transmission line has multiple states with a probability distribution. It should be noted that the stochastic-flow networks studied in [11, 13, 14, 27] are single-source single-sink networks.

Flow allocation problem in the case of multisource multisink stochastic-flow networks [22] where a GA is applied to solve the flow allocation problem. The algorithm in [22] succeeds in finding optimal solutions, but these solutions do not satisfy all constraints, such as: Reliability of the capacity vector.

The Flow allocation problem in the case of multisource multisink stochastic-flow networks with cost constraint has been studied in [23] proposed a multi-objective GA to solve the flow allocation problem with cost constraint. Also, the algorithm in

[23] succeeds in finding optimal solutions, but these solutions do not satisfy the demand constraints.

In this paper, the mathematical formulation of the flow allocation problems will be modified to increase the efficiency of searching the optimal solutions that satisfy all constraints. Also, presents a GA to solve that modified formulation of the flow allocation problems. In addition, the presented GA has employed to optimize the reliability of multisource multisink flow network by searching the optimal capacity vectors.

This paper is organized as follows. Section 2 presents notations and assumptions. Section 3 investigates the mathematical formulation of the flow problem. Section 4 presents reliability calculation of the capacity vector. Section 5 describes the different components of the proposed GA. Section 6 provide the pseudo code of the GA. Section 7 shows the illustrative examples used. Section 8 investigates the system reliability optimization problem. Finally, section 9 presents conclusions.

2. Notations and Assumptions

2.1. Notations

- $G(A, N, M, S, T)$: Denotes a multisource multisink stochastic-flow network containing a set of n arcs $A=\{a_e|\ 1\le e\le n\}$ and a set of q nodes $N=\{a_e|\ n+1\le e\le n+q\}$. The maximum capacity of each component a_e (arc or node) is denoted by $M=\{M_1, M_2, \ldots, M_{n+q}\}$, where M_e is an integer.
- $S\{s_1,\ldots, s_\sigma\}$: set of source nodes, σ is number of source nodes.
- $T\{t_1, \ldots, t_\theta\}$: Set of sink nodes, θ is the number of sink nodes.
- R: $\{r_{w,\ i}|\ 1\le w\le m,\ 1\le i\le\sigma\}$, where $r_{w,\ i}$ is the maximum quantity of resource w that source node s_i can supply.
- D: $\{d_{w,\ j}|\ 1\le w\le m,\ 1\le i\le \theta\}$, where $d_{w,\ j}$ is the demand for resource w at sink node t_j.
- $MP_{i,j,k}$: The k^{th} MP from s_i to t_j.
- $MPS\ \{MP_{i,\ j,\ k}|\ 1\le i\le\sigma,\ 1\le j\le\theta,\ 1\le k\le k_{i,\ j}\}$: A set of all MPs, where $k_{i,\ j}$ represents the number of MPs from s_i to t_j.
- Np: Total number of MPs contained in MPS.
- $f_{i,j,k,w}$: Quantity of resource w flowing through $MP_{i,j,k}$.
- F: Flow vector defined as $F=(f_{1,\ 1,\ 1,\ 1}, f_{1,1,2,1}, \ldots, f_{i,j,k_{i,j},1}, \ldots, f_{i,j,k_{i,j},m}, \ldots, f_{\sigma,\theta,k_{\sigma,\theta},m})$.
- X: Capacity vector defined as $X=(x_1, x_2, \ldots, x_e, \ldots, x_{n+q})$, where $x_e=\sum_{i=1}^{\sigma}\sum_{j=1}^{\theta}\sum_{k=1}^{k_{i,j}}\sum_{w=1}^{m}\{f_{i,j,k,w}\,|a_e\in MP_{i,j,k}\}$.
- $R(X)$: Probability of X.
- $Popsize$: Population size.
- $Maxgen$: Maximum number of generations.
- gn: Generation number.
- p_m: GA mutation rate.
- p_c: GA crossover rate.

2.2. Assumptions

1. The capacity of each component a_e is an integer-valued random variable, which takes values 0< 1< 2< …< M_e according to a given distribution.
2. The capacities of the components are statistically independent.

3. Formulation of Flow Allocation Problem

In this section, we analyze the formulation of the flow allocation problem introduced in [22] and then investigate how to modify it. Every F is considered as the solution of problem model and each element of stands for the flow quantity in a certain MPs. For each flow allocation pattern F, one can construct a vector $X_F=(x_{F1}, x_{F2}, \ldots, x_{Fe}, \ldots, x_{F(n+q)})$:

$$x_{Fe}=\sum_{i=1}^{\sigma}\sum_{j=1}^{\theta}\sum_{k=1}^{k_{i,j}}\sum_{w=1}^{m}\{f_{i,j,k,w}\,|a_e\in MP_{i,j,k}\} \tag{1}$$

Let Ω Set of all generated X_F. The objective function is the probability of successfully transmitting resources according to F denoted by $R(X_F)$. The problem formulation presented in [22] is as follows:

$$Max\ R(X_F)\ \ \forall X_F\in\Omega\ \ s.t.$$

$$\sum_{i=1}^{\sigma}\sum_{k=1}^{k_{i,j}} f_{i,j,k,w}=d_{w,j},\quad w=1,\ldots,m\,;\ j=1,\ldots,\theta \tag{2}$$

$$\sum_{j=1}^{\theta}\sum_{k=1}^{k_{i,j}} f_{i,j,k,w}\le r_{w,j},\quad w=1,\ldots,m\,;\ i=1,\ldots,\sigma \tag{3}$$

$$x_{Fe}\le M_e\qquad e=1,\ldots,n+q \tag{4}$$

$$f_{i,j,k,w}\in N^*\ \ i=1,\ldots,\sigma,\ \ j=1,\ldots,\theta\ \ \ w=1,\ldots,m,\ k=1,\ldots,k_{i,j} \tag{5}$$

Where N^* is the set of nonnegative integers.

Constraint 5 implies that the flow on every MP belongs to N^*. In the absence of any restrictions, the abovementioned result leads to the following realizations:

1. The selected flow values on any MP may exceed the capacity of that path.
2. A greater number of generations will be required to obtain feasible solutions.

To alleviate the disadvantages mentioned above, the formulation of the flow allocation problem is modified as follows:

If $MP^1, MP^2, \ldots, MP^{np}$ are MPs contained in MPS, then the maximum capacity L_j of MP^j is defined as:

$$L_j=min\{M_e\,|a_e\in MP^j\} \tag{6}$$

According to Equation 6 the lowest capacity of any component (arc or node) contained in a MP MP^j is the maximum capacity of that path. For example, consider the path $MP=\{a_1, a_2, a_5\}$ and $M_1=3$, $M_2=2$ and $M_5=6$ then the maximum capacity of that path equals to 2. Additional details about the maximum path capacity and its applications can be found in [1, 16].

Flow f^j on MP^j must satisfy $f^j \leq L_j$ for each j= 1, 2, …, np i.e., the flow carried by the path doesn't exceed the maximum capacity of that path.

We define $Min\ C=min\ \{L_j | j= 1, 2, \ldots, np\}$, where $Min\ C$ is the maximum value of the flows generated on the MPs contained in *MPs*. Consequently, constraint 5 becomes:

$$f_{i,j,k,w} \leq MinC \quad i=1,\ldots,\sigma,\ j=1,\ldots,\theta \\ w=1,\ldots,m,\ k=1,\ldots,k_{i,j} \tag{7}$$

Where $f_{i,j,k,w}$ is a nonnegative integer.

To obtain optimal solutions, constraint 2 is modified as:

$$\sum_{i=1}^{\sigma}\sum_{k=1}^{k_{i,j}} f_{i,j,k,w} \leq d_{w,j},\quad w=1,\ldots,m\,;\ j=1,\ldots,\theta \tag{8}$$

By modifying constraints 2 and 5, we guarantee that our algorithm solves the flow allocation problem and obtains the optimal solution with fewer generations.

4. Calculating the Reliability of the Capacity Vector

Let X= (x_1, x_2, x_e, …, x_{n+q}) represent the capacity vector of the network, where x_e is an integer-valued random variable representing the current capacity of component a_e. We define B=(b_1, b_2, …,b_e, …, b_{n+q}), where b_e represents the probability that the transmission is successful, i.e., $b_e=Pr\{b_e \geq x_e\}$. Then:

$$R(x)=Pr(b_1 \cap b_2 \cap \ldots \cap b_{n+q}) = \prod_{e=1}^{n+q} Pr(b_e) \tag{9}$$

For more details about computing the state probability of a multistate system, [6].

5. Proposed GA

A GA is an evolutionary algorithm that provides near optimal solutions to combinatorial optimization problems [7]. A chromosome is a data structure containing a "string" of task parameters, or genes. A chromosome represents a potential problem solution and is typically encoded as a string to facilitate mutation and crossover operations. The fitness of an individual solution is a value that reflects its performance (i.e., how well it solves a certain task). A fitness function is a mapping of the chromosomes in a population to their corresponding fitness values. A more detailed discussion on GAs [2]. For more details about GA operations (coding, mutation, crossover, etc.,) refer to [3].

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

5.1. Coding

If the network has *np MPs*, then the chromosome F contains $m \times np$ fields, where m is the number of resources. Each field in F represents the (current) flow of a path, F= ($f_{1,1,1,1}$, $f_{1,1,1,2}$, …, $f_{i,j,k_{i,j},1}$, …, $f_{i,j,k_{i,j},m}$, …, $f_{\sigma,\theta,k_{\sigma,\theta},m}$).

5.2. Crossover

In our algorithm, we apply a one-cut-point crossover approach, [2]. In particular, an integer value (rn) is randomly generated in the range (1, $m \times np-1$) where $m \times np$ is the length of the chromosome. This value is used to breed two offsprings (two new chromosomes f_C and f_D) from two parents (f_A and f_B) selected randomly. The rate of the breeding operation is defined by the value of Pc:

$$f_C^{t+1} = [f_A^t(j)]_{j=1}^{m} + [f_B^t(j)]_{j=m+1}^{m*np}$$

$$f_D^{t+1} = [f_B^t(j)]_{j=1}^{m} + [f_A^t(j)]_{j=m+1}^{m*np}$$

5.3. Mutation

Chromosomes undergo mutation according to mutation rate P_m, [22, 23] as follows:

Generate a random number $r_m \in [0,1]$
if $r_m \leq P_m$ *then*
{
*for j = 1 to m*np, do*
If $(f_A^t(j) = Minc)$ *then* $f_A^t(j) = 0$
Else $f_A^t(j) = v, v \in \{0,1,\ldots,Min\,C\}$
$f_A^{t+1}(j) = f_A^t(j)$
End for
}

5.4. Fitness Function

The fitness function $E(F)$ equals to the probability of the capacity vector $R(X_F)$ only if the generated chromosome satisfies all constraints. Otherwise, $E(F)$ is set to 0. Hence, the fitness function has the following form:

$$E(F) = \begin{cases} R(X_F) & \textit{if F satisfies all constraints} \\ 0 & \textit{otherwise} \end{cases} \tag{10}$$

5.5. Selection Mechanism

The Fitness Uniform Selection Scheme (FUSS), [5], is used in the proposed GA to select new chromosomes. FUSS is defined as: if the lowest/highest fitness values in the current population are $E(F)_{min/max}$, we select a fitness value $E(F)$ uniformly in the interval [$E(F)_{min}$, $E(F)_{max}$]. Then, the individual *i ™p* with fitness nearest to $E(F)$ is selected.

6. Pseudo-Code of the Proposed Algorithm

Algorithm 1: Overall GA to maximize $R(X_F)$.
Begin GA.
Input parameters: M, R, D, Pc, Pm, Popsize, Maxgen;
gn=0, *gt*=0;
Initialize P(*gn*);

```
While gn < Maxgen, do
    While gt < Popsize, do
        Use FUSS to select two chromosomes;
        Apply crossover according to Pc;
        Apply mutation according to Pm;
        Evaluate the current chromosome;
        If (E(F) > 0), then gt= gt+1;
    End do;
    Save best p(gn);
    gn:=gn+1;
    Replace worst p(gn) with best p(gn−1);
End do
End GA
```

7. Illustrative Examples

7.1. Network with Two Source and Two Sink Nodes

To illustrate the proposed GA, we use the network presented in Figure 1, taken from [22]. This network contains two source and two sink nodes. The probability distribution of the capacity of each arc was included in [22]. The genetic parameters used in the proposed algorithm are *Popsize*=10, *Maxgen*=1000, P_c=0.95 and P_m=0.05. The algorithm is iterated 10 times. The *MPs* are as follows:

$$MP_{1,1,1}=\{a_1, a_{15}, a_5\}$$
$$MP_{1,1,2}=\{a_1, a_{15}, a_6, a_{18}, a_9\}$$
$$MP_{1,1,3}=\{a_2, a_{16}, a_7, a_{18}, a_9\}$$
$$MP_{1,2,1}=\{a_1, a_{15}, a_6, a_{18}, a_{14}\}$$
$$MP_{1,2,2}=\{a_2, a_{16}, a_7, a_{18}, a_{14}\}$$
$$MP_{2,1,1}=\{a_3, a_{16}, a_7, a_{18}, a_9\}$$
$$MP_{2,1,2}=\{a_4, a_{17}, a_8, a_{19}, a_{13}, a_{18}, a_9\}$$
$$MP_{2,2,1}=\{a_3, a_{16}, a_7, a_{18}, a_{14}\}$$
$$MP_{2,2,2}=\{a_4, a_{17}, a_8, a_{19}, a_{13}, a_{18}, a_{14}\}$$
$$MP_{2,2,3}=\{a_4, a_{17}, a_8, a_{19}, a_{10}\}$$
$$MP_{2,2,4}=\{a_4, a_{17}, a_{11}, a_{20}, a_{12}\}$$

Note that, M=(12,10, 10, 14, 8, 10, 14, 12, 16, 8, 10, 10, 12, 16, 14, 16, 16, 24, 14, 10), R= ($r_{1,1}$, $r_{1,2}$, $r_{2,1}$, $r_{2,2}$)= (15,17,10,13), D=($d_{1,1}$, $d_{1,2}$, $d_{2,1}$, $d_{2,2}$)= (11, 12, 7, 10).

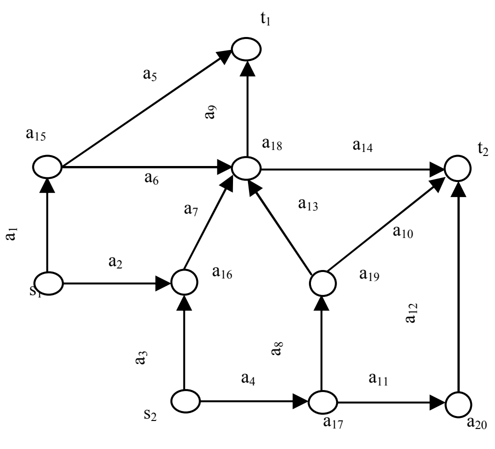

Figure 1. Network with two source and two sink nodes.

In Table 1, we list the best value of $E(F)$ and the corresponding F vector for the 10 best generations. In the table, “No.” denotes the generation number.

Table 1. Best E(F) values at each generation and the corresponding F vector.

No.	Best E(F)	Corresponding F
419	0.764049	1 2 0 2 1 1 1 0 1 0 2 0 0 0 0 0 0 3 3 0 0 1
421	0.752860	3 2 0 2 1 1 1 0 1 0 0 0 0 0 0 0 0 3 3 0 0 1
405	0.752518	1 2 0 2 1 1 1 0 1 0 0 2 0 0 0 0 0 3 4 0 0 1
434	0.749821	3 2 0 2 1 1 0 0 1 0 0 2 0 0 0 0 0 3 3 0 0 1
436	0.733635	3 2 0 2 1 1 0 0 1 0 0 1 0 1 0 0 1 4 0 0 0 1
414	0.718476	1 2 0 2 1 1 1 0 1 0 2 2 0 0 0 0 0 3 3 0 0 1
415	0.716303	3 2 0 2 1 1 1 0 1 0 0 2 0 0 0 0 0 3 3 0 0 1
371	0.716053	3 2 0 2 1 1 1 0 1 0 2 2 0 1 0 0 0 2 0 0 0 1
387	0.716053	3 2 0 2 1 1 1 0 1 0 2 2 0 1 0 0 0 2 0 0 0 1
671	0.706817	4 0 1 1 1 1 0 0 1 0 0 1 1 1 0 1 0 4 1 1 0 1

The best value of $E(F)$ is 0.764049, which was obtained in generation No. 419. In comparison with the results obtained by [22], the best value of $E(F)$ was 0.16969. Also, $R(X)$ values have been calculated to the six optimal solutions obtained by the algorithm presented in [22] and it has discovered that all $R(X)$ values equal to 0.

In addition, to compare our algorithm with that one presented in [23], the cost constrain has been added to the problem formulation. Table 2 compares the results obtained by the GA presented in [23] and the results obtained by the proposed algorithm presented in this paper. The values of $E(F)$ (named in [23] as $R(F)$) obtained by the presented algorithm are better than that obtained by [23] with less costs.

Table 2. Comparison between the proposed algorithm and the algorithm presented in [23].

Results Obtained by the Proposed Algorithm		Results Obtained by [18]	
E(F)	CPe(F)	R(F)	CPe(F)
0.76405	2.9958	0.6534176	3.8822
0.75286	2.9466	0.6287173	3.7453
0.75252	3.3021	0.5890704	3.6284
0.74982	3.253	0.5316544	3.5326
0.73364	3.1726	0.5509963	3.5605
0.71848	3.4592	0.5700666	3.5873
0.71630	3.4100	0.6366206	3.7587
0.71605	3.2820	0.5175207	3.5159
0.70682	3.5593	0.5305662	3.5315
0.70596	3.5358	0.5272422	3.5293

7.2. Network with Two Source and Three Sink Nodes

The algorithm has been applied to another network given in Figure 2.

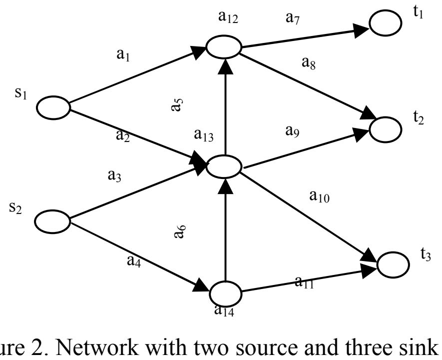

Figure 2. Network with two source and three sink nodes.

Let, M= (12, 10, 10, 14, 8, 10, 14, 12, 16, 8, 10, 10, 12, 16)

R= ($r_{1,1}$, $r_{1,2}$, $r_{2,1}$, $r_{2,2}$)= (15, 17, 17, 18)

D=($d_{1,1}$, $d_{1,2}$, $d_{1,3}$, $d_{2,1}$, $d_{2,2}$, $d_{1,3}$)= (16, 18, 17, 15, 20, 18)

The information of each component is given in Table 3. The *MPs* are as follows:

$$MP_{1,1,1}=\{a_1, a_{12}, a_7\}$$
$$MP_{1,1,2}=\{a_2, a_{13}, a_5, a_{12}, a_7\}$$
$$MP_{1,2,1}=\{a_1, a_{12}, a_8\}$$
$$MP_{1,2,2}=\{a_2, a_{13}, a_9\}$$
$$MP_{1,2,3}=\{a_2, a_{13}, a_5, a_{12}, a_8\}$$
$$MP_{1,3,1}=\{a_2, a_{13}, a_{10}\}$$
$$MP_{2,1,1}=\{a_3, a_{13}, a_5, a_{12}, a_7\}$$
$$MP_{2,1,2}=\{a_4, a_{14}, a_6, a_{13}, a_5, a_{12}, a_7\}$$
$$MP_{2,2,1}=\{a_3, a_{13}, a_9\}$$
$$MP_{2,2,2}=\{a_4, a_{14}, a_6, a_9\}$$
$$MP_{2,3,1}=\{a_3, a_{13}, a_{10}\}$$
$$MP_{2,3,2}=\{a_4, a_{14}, a_6, a_{10}\}$$
$$MP_{2,3,3}=\{a_4, a_{14}, a_{11}\}$$

Table 3. Probability distribution of components' capacities.

a_n	0	1	2	3	4	5	6	7
a_1	0.001	0.001	0.003	0.004	0.005	0.005	0.006	0.007
a_2	0.001	0.003	0.003	0.004	0.005	0.007	0.007	0.008
a_3	0.002	0.002	0.003	0.006	0.007	0.007	0.010	0.012
a_4	0.001	0.001	0.002	0.003	0.005	0.008	0.010	0.011
a_5	0.001	0.002	0.009	0.012	0.020	0.040	0.050	0.060
a_6	0.001	0.002	0.002	0.005	0.010	0.012	0.015	0.017
a_7	0.001	0.001	0.002	0.005	0.008	0.010	0.012	0.015
a_8	0.001	0.002	0.005	0.005	0.007	0.008	0.010	0.012
a_9	0.001	0.001	0.002	0.002	0.003	0.004	0.005	0.008
a_{10}	0.002	0.003	0.005	0.006	0.007	0.009	0.012	0.015
a_{11}	0.002	0.002	0.003	0.005	0.007	0.008	0.010	0.011
a_{12}	0.001	0.002	0.003	0.005	0.008	0.009	0.010	0.012
a_{13}	0.001	0.001	0.003	0.005	0.005	0.010	0.011	0.017
a_{14}	0.001	0.001	0.002	0.002	0.003	0.005	0.007	0.009
8	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
0.010	0.015	0.060	0.150	0.733	0.000	0.000	0.000	0.000
0.009	0.010	0.943	0.000	0.000	0.000	0.000	0.000	0.000
0.015	0.017	0.919	0.000	0.000	0.000	0.000	0.000	0.000
0.012	0.015	0.015	0.016	0.020	0.025	0.856	0.000	0.000
0.806	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.020	0.025	0.891	0.000	0.000	0.000	0.000	0.000	0.000
0.015	0.017	0.020	0.022	0.025	0.030	0.817	0.000	0.000
0.015	0.015	0.016	0.020	0.884	0.000	0.000	0.000	0.000
0.009	0.010	0.011	0.015	0.016	0.017	0.019	0.020	0.857
0.941	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.020	0.030	0.902	0.000	0.000	0.000	0.000	0.000	0.000
0.015	0.040	0.895	0.000	0.000	0.000	0.000	0.000	0.000
0.018	0.020	0.025	0.031	0.853	0.000	0.000	0.000	0.000
0.016	0.021	0.024	0.025	0.030	0.035	0.040	0.060	0.719

Table 4 summarizes the results of applying the proposed GA on the network given in Figure 2.

Table 4. Best E(F) values and the corresponding F vector of the network given in Figure 2.

Best F(F)	Corresponding F
0.616239	0 0 2 0 0 2 2 0 0 2 0 1 0 1 1 1 0 2 1 0 1 2 2 0 0 1
0.551932	0 1 2 1 0 1 2 0 0 2 1 0 0 1 0 1 2 2 2 0 1 0 0 0 1 0
0.723281	0 1 2 1 0 1 2 0 0 2 1 0 0 0 0 0 1 2 0 1 1 2 1 0 0 1
0.676186	2 0 2 0 2 2 0 0 0 2 0 0 1 1 0 0 2 1 1 1 2 0 0 0 0 2
0.510357	0 1 2 1 0 1 2 0 0 2 1 0 0 1 0 1 1 2 1 1 2 2 0 0 1 0
0.617383	1 1 0 2 1 2 1 1 0 0 0 1 2 0 0 0 1 2 0 1 1 2 1 0 0 1
0.646794	0 0 2 0 0 2 2 0 0 2 0 1 0 1 0 1 1 2 1 1 2 2 0 0 1 0
0.576957	1 0 0 1 0 1 0 1 0 2 0 0 1 1 1 1 0 2 1 0 1 2 2 0 0 1
0.766645	2 1 1 0 1 1 2 1 1 0 0 0 0 0 1 0 0 0 0 0 2 2 0 0 1 1
0.486069	0 1 2 1 0 1 2 0 0 2 1 0 0 1 1 0 2 2 0 2 0 2 2 0 1 0

8. Optimizing System Reliability RS of Multisource Multisink Stochastic-Flow Network

This section investigates how to modify the GA presented in section *V* to optimize the system reliability of multisource multisink network. It is known that, if the flows have been determined then the capacity vector X_F can be evaluated. So, the proposed GA can be used to determine the optimal set of lower boundary points that maximize the system reliability. The following subsections investigate the formulation of the R_S optimization problem, R_S Evaluation, solving R_S optimization problem, and presenting an illustrative example to show how the presented GA solve R_S optimization problem.

8.1. Formulation of R_S Optimization Problem

The formulation of the system reliability optimization problem in the case of the multisource multisink networks is as follows:

Find the optimal set of X_F s.t.
R_S is maximized

The set of optimal capacity vectors X_F are obtained by using the presented GA to solve the flow allocation problem presented in section 2.

B. Evaluating R_S

If X^1, X^2, …, $X^{Popsize}$ are the generated set of the capacity vectors by using proposed GA presented in *V*. Then, all lower boundary points can be obtained by removing non-minimal ones in $X^1, X^2, \ldots, X^{Popsize}$ [15].

If X^1, X^2, …, X^q are all lower boundary points, then the system reliability R_S is evaluated by:

$$R_S = Pr\{\textstyle\bigcup_{i=1}^{q}\{Y \mid Y \geq X^i\}\} \tag{11}$$

Where $Pr\{Y\}= Pr\{y_1\}. Pr\{y_2\}, \ldots, Pr\{y_n\}$. The inclusion-exclusion rule presented in [6] can be used to calculate R_S as follows:

If $A_1= \{Y|Y\geq X^1\}$, $A_2= \{Y|Y\geq X^2\}$, …, $A_q= \{Y|Y\geq X^q\}$, then apply inclusion-exclusion rule to calculate R_S by using the relation:

$$R_S = \sum_i Pr\{A_i\} - \sum_{i\neq j} Pr\{A_i \cap A_j\} + \sum_{i\neq j\neq k} Pr\{A_i \cap A_j \cap A_k\} - \\ - \ldots + (-1)^{q-1} Pr\{A_1 \cap A_2 \cap \ldots \cap A_q\} \tag{12}$$

Note that, if $A= (a_1, a_2, \ldots, a_e, \ldots, a_{n+q})$ then:

$$Pr(A) = \prod_{e=1}^{n+q} Pr(a_e) \tag{13}$$

8.2. Solving RS Optimization Problem

The pseudo code of the proposed GA Algorithm 1 given in section *V* will be extended to solve the *Rs* optimization problem as the following steps:

Algorithm 2: Overall GA to maximize R_S.

Begin GA

Input parameters: *M, R, D, Pc, Pm, Popsize, Maxgen;*
    gn= 0, *gt*= 0*;*
    Initialize P(*gn*)*;*
    While gn < Maxgen, do
        While gt < Popsize, do
            Use FUSS to select two chromosomes;
            Apply crossover according to Pc;
            Apply mutation according to Pm;
            Evaluate the current chromosome;
            If (*E*(*f*) > 0), *then gt*= *gt*+1*;*
        End do;

Evaluate the set of lower boundary points X_F for the obtained F;
Evaluate R_S for the generated Xs as described in B.
save best p(gn);
gn:=gn+1;
Replace worst p(gn) with best p(gn−1);
End do
End GA

8.3. Illustrative Example

The proposed GA presented in *B* has been applied to the network example given in section 7.1. Table 5 shows the set of lower boundary points of the first generation, R_S equals to 0.764049. The genetic parameters used in the proposed algorithm are *Popsize*=10, *Maxgen*=1000, P_c=0.95 and P_m=0.05. The algorithm is iterated 10 times.

Table 5. Results of first generation.

The Optimal Lower Boundary Points
9 1 4 13 5 4 5 8 7 3 5 5 5 5 9 4 13 14 8 5
6 5 4 13 2 4 9 8 9 3 5 5 5 6 6 7 13 18 8 5
5 5 4 14 1 4 9 8 11 4 6 6 4 5 5 8 14 17 8 6
8 8 5 12 3 5 13 8 9 4 4 4 4 12 8 8 12 22 8 4
9 1 8 12 5 4 9 9 10 2 3 3 7 8 9 8 12 20 9 3
8 8 1 12 3 5 9 10 12 6 2 2 4 5 8 4 12 18 10 2
6 5 8 7 2 4 13 5 9 2 2 2 3 8 6 11 7 20 5 2
8 5 3 10 1 7 8 9 12 4 1 1 5 5 8 6 10 20 9 1
12 4 6 10 3 9 10 9 14 4 1 1 5 6 12 7 10 24 9 1

In Table 6, we list the best value of R_S for the 10 best generations.

The best value of R_S is 0.830087, which was obtained in generation No. 421.

Table 6. Best 10 RS values and the corresponding generation number.

No.	R_S	No.	R_S
421	0.830087	405	0.801639
420	0.815528	414	0.801074
419	0.814427	426	0.799554
406	0.804929	413	0.801442
416	0.802693	412	0.801401

9. Conclusions and Discussion

We modified the formulation of the flow allocation problem to guarantee that the optimal solution is obtained. Then, we proposed a GA to solve this problem while considering that the probability of the capacity vector is maximized.

The proposed algorithm obtained solutions with values of $E(F)$ greater than 0, i.e., $R(X_F)$ was maximized. For comparison, we used the algorithm presented in [22] to obtain solutions for the same problem and their $R(X_F)$ values have been calculated. We discovered that all $R(X_F)$ values were equal to 0, which implies that the solutions obtained using the algorithm in [22] fail to satisfy all constraints.

Our algorithm, which is based on the proposed formulation, can efficiently solve the flow allocation problem while maximizing the reliability of the capacity vector with least costs in comparison with [23].

Finally, the presented GA has been employed to solve the system reliability optimization problem. The presented GA is based on generating the optimal set of lower boundary points and then the system reliability can be evaluated.

For the future work, the flow problem presented in this paper can be solved by IGA [24] or by other famous algorithms such as DE [4] and PSO [25], May improve the results.

References

[1] Aggarwal K., Chopra Y., and Bajwa J., "Capacity Consideration in Reliability Analysis of Communication Systems," *IEEE Transactions on Reliability*, vol. 31, no. 2, pp. 177-181, 1982.

[2] Coello C. and Christiansen A., "Two New GA-based Methods for Multi-Objective Optimization," *Civil Engineering Systems*, vol. 15, no. 3, pp. 207-243, 1998.

[3] Gen M. and Cheng R., *Genetic Algorithms and Engineering Optimization*, Wiley series in Engineering, Design and Automation, 2000.

[4] Gu F. and Liu H., "An Adaptive Multiobjective Differential Evolution Algorithm," *Journal of Computers*, vol. 8, no. 2, pp.294-301, 2013.

[5] Hutter M. and Legg C., "Fitness Uniform Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 568-589, 2006.

[6] Janan X., "On Multistate System Analysis," *IEEE Transactions on Reliability*, vol. 34, no. 4, pp. 329-337, 1985.

[7] Liu L ., Mu H., Yang X., He R., and Li Y., "An Oriented Spanning Tree based Genetic Algorithm for Multi-Criteria Shortest Path Problems," available at: http://www.sciencedirect.com/science/article/pii/S1568494611002985, last visited 2012.

[8] Lin J., Jane C. and Yuan J., "On Reliability Evaluation of a Capacitated-Flow Network in Terms of Minimal Path Sets," *Networks*, vol. 25, no. 3, pp. 131-138, 1995.

[9] Lin J., "Reliability Evaluation of Capacitated-Flow Networks with Budget Constraints," *IIE Transactions*, vol. 30, no. 12, pp. 1175-1 180, 1998.

[10] Lin Y., "A Simple Algorithm for Reliability Evaluation of a Stochastic-Flow Network with Node Failure," *Computers and Operations Research*, vol. 28, no. 13, pp. 1277-1285, 2001.

[11] Lin Y. and Yeh C., "Evaluation of optimal network reliability under components-assignments subject to a transmission budget,"

IEEE Transactions on Reliability, vol. 59, no. 3, pp. 539-550, 2010.

[12] Lin Y., "Extend the Quickest Path Problem to the System Reliability Evaluation for a Stochastic-Flow Network," *Computers and Operations Research*, vol. 30, no. 4, pp. 567-575, 2003.

[13] Lin Y. and Yeh C., "Multi-Objective Optimization for Stochastic Computer Networks using NSGA-II and TOPSIS," *European Journal of Operational Research*, vol. 218, no. 3, pp. 735-746, 2012.

[14] Lin Y. and Yeh C., "Multistate Components Assignment Problem with Optimal Network Reliability Subject to Assignment Budget," *Applied Mathematics and Computation*, vol. 217, no. 24, pp. 10074-10086, 2011.

[15] Lin Y., "Network Reliability of a Time-Based Multistate Network Under Spare Routing With *p* Minimal Paths," *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 61-69, 2011.

[16] Lin Y., "Reliability Evaluation for an Information Network with Node Failure under cost Constraint," *IEEE Transactions on Systems, Man. and Cybernetics*, Part A: System and Humans, vol. 37, no. 2, pp. 180-188, 2007.

[17] Lin Y., "Reliability Evaluation for Overall-Terminal Multistate Flow Networks with Bi-Directed Arcs," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6669-6674, 2010.

[18] Lin Y., "Reliability of a Stochastic-Flow Network with Unreliable Branches and Nodes under Budget Constraints," *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 381-387, 2004.

[19] Lin Y., Kuo P., and Chang C., "System Reliability for a Multistate Flow Network with Multiple Joint Minimal Paths under Time Constraint," *Simulation Modeling Practice and Theory*, vol. 29, pp. 78-92, 2012.

[20] Lin Y., "System Reliability of a Limited-Flow Network in Multicommodity Case," *IEEE Transactions on Reliability*, vol. 56, no. 1, pp. 17-25, 2007.

[21] Lin Y., "Two-Commodity Reliability Evaluation for a Stochastic-Flow Network with Node Failure," *Computers and Operations Research*, vol. 29, no. 13, pp.1927-1939, 2002.

[22] Liu Q., Xiaoxian H. and Zhao Q., "Genetic Algorithm-based Study on Flow Allocation in a Multicommodity Stochastic-Flow Network with Unreliable Nodes," *in Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, pp. 576-581, August 2007.

[23] Liu Q., Zhao Q. and Zang W., "Study on Multi-Objective Optimization of Flow Allocation in a Multi-Commodity Stochastic-Flow Network with Unreliable Nodes," *Journal of Applied Mathematics Computing*, vol. 28, no. 1, pp. 185-198, 2008.

[24] Mansour N., Awad M., and El-Fakih K., "Incremental Genetic Algorithm," *The International Arab Journal of Information Technology*, vol. 3, no. 1, pp. 42-47, 2006.

[25] Niknam T. and Amiri B., "An Efficient Hybrid Approach based on PSO, ACO and K-Means for Cluster Analysis," *Applied Soft Computing*, vol. 10, no. 1, pp.183-197, 2010.

[26] Yeh W., "An Improved Method for Multistate Flow Network Reliability With Unreliable Nodes and a Budget Constraint Based on Path Set," *IEEE Transactions on Systems, Man and Cybernetics*, Part A: Systems and Humans, vol. 41, no. 2, pp. 350-355, 2011.

[27] Younes A. and Hassan M., "A Genetic Algorithm for Reliability Evaluation of a Stochastic-Flow Network with Node Failure," *International Journal of Computer Science and Security*, vol. 4, no. 6, pp. 528-537, Feb. 2011.

Moatamad Hassan received a PhD degree of Computer Science from Aswan University, Faculty of Science, Aswan, Egypt. Currently, he is an assistant professor at the Department of Mathematics, Computer Science Branch, Faculty of Science, Aswan University, Aswan, Egypt. His work deals with QoS, reliability and computer network design problems.