Vol. 1, No. 1, pp. 57-74, (December 2023)

ASWAN SCIENCE AND TECHNOLOGY BULLETIN (ASTB)

Online ISSN: 3009-7916, Print ISSN: 1110-0184

Journal homepage: https://astb.journals.ekb.eg/ E-mail: essamshaalan@sci.aswu.edu.eg

Original Article

Solving the Robust Design Problem for MMSFNs Considering Node Failure.

Noha H. Radwan[a], Hameda A. Senary[a] and Moatamad R. Hassan[a*]

[a]Department of Computer Science and Mathematics, Faculty of Science, Aswan University, Aswan, Egypt.

Received: 25/09/2023 Accepted: 12/11/2023

Abstract:

Robust design for a flow network is considered an NP-hard problem. It is defined as finding the best capacities for components so that the network survives even when nodes fail. MMSFNs (multi-source multi-sink flow networks are utilized across numerous real-world systems, including logistics, computing, transportation, and telecommunication systems. In this paper, the robust design for multi-commodity MMSFNs with node failures been investigated and solved using GA-based approach. Because despite many studies in this area, no study is given in this section. The solution approach to this problem is divided into two parts, outer GA and internal GA. The outer GA searches for the optimal capacity of nodes into the minimum-sum network. The internal GA searches for the best vector with maximum system reliability.

This study was applied to three networks to ensure their effectiveness and success in achieving the desired objective of the study.

Keywords: Robust Design, Flow Network, MMSFNs, Nodes Capacity, Genetic Algorithm**.**

Introduction

Network reliability is important because it largely determines the internet infrastructure complex's degree of security and reliability [Sarintip and Kailash, 2006]. System reliability can be considered as a performance signal to measure the capacity or accuracy level of a demand-supply system [Lin, 2009]. Network reliability is the durability by which a network with all its subnets and constituents is expected to successfully perform a task under conditions that occur for a specified period of time between the source and the target. Network reliability is the probability that a configured request will be successfully sent across the network according to the transmission budget [Lin and Yeh, 2011].

Corresponding author*: E-mail address: mr.hassan@sci.aswu.edu.eg

Network reliability as the probability that the accumulation of the sink is supplied by an accumulation source [Bobbio et al., 2011]. The reliability of a system is known as the probability of sending a certain amount of flow from a source node to a sink node [Forghani-elahabad and Mahdavi-Amiri, 2013;2014;2015;2016]. The assumption for a conventional power system's reliability assessment is that it will carry out its functions properly and without error for a set amount of time under typical operating circumstances [López-Prado et al., 2020]. Reliability is a useful tool that describes the operational efficiency of a transportation network [Niu et al., 2020].

Robust design is the process used to find the optimal representation of product design. The term "Best" is cautiously defined to mean that the design is the most affordable answer to complex product development based on the needs of the customer. [Fowlkes and Creveling, 1995]. Robust design is a statistical experimental design that is systematically used to improve the design of products and processes. Robust design is the process of improving quality through the design of better products or manufacturing methods [Yong, 1998]. In addition to understanding that an algorithm must provide "good" solutions for test problems with different sized and numerical characteristics, a strong person also understands that the algorithm must provide "good" solutions once restrictions are added or removed. We are aware that function [Chabrier, 2004] should Robust design is an effective method for achieving advantageous consistency quickly and inexpensively. [Yang, 2007]. The robust design challenge in capacitive existing networks is to calculate the exact capacity allotted to each arc so that the network can function even in the event of an arc failure. To address this challenge, an algorithm was developed [Chen, 2012]. In Research [Cho and Shin, 2012], a robust design model based on redacted sample selection is presented by developing a framework for redacted experimental design, a technique for evaluating robust design using censored maximum likelihood, and an optimization model for redacted robust design. The idea of manufacturing quality in products and processes serves as the foundation of this design, and it is regarded as among the most crucial ideas in system engineering design to enhance quality and streamline operations [Radwan et al., 2020] used GA to identify issues with robust design for solitary and dual stable product channels with node failures. In the study [Massoudi et al., 2021], strength is calculated as the sampled volume of required geometric and operational deviations. To handle the large number of additional evaluations required for the robustness calculation, an artificial neutral network is used to generate a fast and accurate proxy for the high-fidelity model Assessing a system's operational reliability is the procedure of figuring out whether it meets the certain standard. In the power system's making plans, design, and procedure, assessing the system's reliability is a critical and essential issue [Al-Shaalan].

MMSFN is important because it is implemented in numerous real-world networks, including telecommunications, computer, and transportation networks an algorithm to calculate the precise reliability coefficient for the MMSFN lowest path was presented in study [Elahabad and Amiri, 2016]. Study [Hassan and Abdou, 2019] provided a survey to analyze the MMSFN's dependability under time restraints. A diagram was presented in research [Hassan, 2021] to increase the accuracy of the volume vertex of the apportioned budget problem. The first section of this diagram explores the favored components that can be indicated at the lowest cost in the system. In the second section, we looked at the current arc with the highest reliability of the assigned components' capacitive arc. Research [Hassan, 2021] used applied GA to solve the reliability optimization model of the MMSFN system with transmission set budget. By obtaining a better set of lower bound points, these achieve the highest level of system reliability. Modeling and problem-solving of real-world technical systems, such as networking and telecommunication systems, power transmission systems, transport and logistics systems, and manufacturing systems, have frequently utilized multistate flow networks [Niu, 2021].

GA is a technique commonly used in computing to locate approximations of optimization solutions. Inheritance, mutation, choice, and fusion are just a few examples of the evolutionary algorithms used by the GA, a specific class of evolutionary algorithms [Melanie, 1999]. GA is a responsive algorithmic method based on the mechanisms and principles of natural selection and the idea of natural humankind's fittest [Katayama and Sakamoto, 2000]. GA is a global search and stochastic optimization technique that uses the metaphor of organic evolutionary biology. The goal is to use the preservation of the fitness values principle to sort through a population of potential solutions and come up with progressively better simplifications of the solution. Later, GA, which is based on biological evolutionary principles, gained popularity as an optimizer for resolving engineering problems [Lai, 2001]. GA is an example of evolutionary computation and a model for optimization type algorithms. Chromosomes, strands of DNA, provide an abstract model of living organisms. Subsections of chromosomes or genes are used to learn different characteristics of an individual. During reproduction, the genes of the parents cooperate to form the genes of the offspring [Lobo and Chavan, 2012]. GA is one of the best optimization algorithms. It is a general algorithm that works well in any search space and is suitable for generating quality solutions. The principles of selection and evolution are used to derive several solutions for a particular problem [Abed and Tang, 2013]. GA excels in continuous expansion and development. It is based on the principles of natural selection and the biological mechanisms of genetics. In nature, genetic and evolutionary biological processes simulated by GA, population searches and information exchange between different individuals are also performed to generate globally applicable optimization searches [Du, 2014]. A GA is proffered to enhance the MMSFN system's dependability when the flow allocation problem in [Hassan, 2016] is solved. GA is a graph based on the genetic mechanism of natural selection and reflective global stochastic search algorithms [Liu, 2016]. The authors in [Hamed et al., 2020] utilized GA to determine the optimum capacity with the least total capacity and most reliability, thus solving the capacity allocation problem in flow channels. Additionally, the authors in [Elden et al., 2020] were using GA to find the ideal elements that optimize reliability and reduce total allocation in order to solve the MMSFN system reliability optimization problem. Likewise, the MMSFN robust design problem was resolved by the authors in [Boubaker et al., 2023] using GA.

In this study, we used GA-based approach to solve the robust design problem in MMSFNs with node failure. We obtain the minimum sum of the node capacities and examine the conditions for adopting the flow vector that obtains the maximum reliability assessment. We illustrate three network examples to demonstrate the efficiency of the algorithm.

The paper is prepared as follows: we approach in section 2 notations and expectations. In section 3 we explain problem formulation. The proposed approach is presented in section 4. We depict the posited reach in section 5. We offer studied cases in section 6. Finally, we present the conclusion in section 7.

- Assumptions
- **The flow rate must accept the coefficient of low flow rate.**
- **Components have independent statistical capacities.**
- **The state capacity of a node is a random variable with an integer value that takes the values 0 < 1 < 2 < ...< Ɱe according to a certain distribution.**
- **The maximum capacity of a route is not exceeded by the flow along that route.**

2 Robust Design for MMSFN

For network capacity assignment and network structure analysis, the network structure is crucial. We define the minimal paths to network then determine coverage node and critical node to avoid the reliability drop to zero.

2.1 Capacity Assignment

The maximum capacity in a single-source single-sink system *Ɱi* (pointing to the maximum capacity of a node *i*). of $\mathfrak{n}_i$ varies from zero to the requirement value. Even so, in MMSFNs,

there are numerous requirements placed by the sinks. As an outcome, the following idiom is suggested here just to set $\mathcal{M}i$ values.
Let $\S_i = \sum\{ sd_{w,j} | n_i \in \mathcal{MP}_{i,j} \}$, then $0 \leq M_i \leq \S_i + \hbar$, $\hbar \in [0, U]$, U is a positive integer.

($sd_{w,j}$ pointing to demand for resource w at sink node t_j, $\mathcal{MP}_{i,j}$ sign to The k^{th} Minimal path from source i to sink j).
If n_i is a critical node, then M_i should not be less than $\S_i$.

2.2 Coverage Node

When $n_i, n_j \in N$. and no flow pass through n_j when n_i failed, we called this case n_j is covered by n_i and $n_j \subseteq n_i$

($\mathcal{N}$ sign to the set of nodes $\{n_b | 1 \leq b \leq n_a\}$, n_a sign to number of nodes),

2.3 Critical Node

The node n_i is called critical node if $\Re_S$ is zero when node failed. **(**$\Re_S$ pointing to the system reliability**)**

2.4 Probabilities States for each Node

At first we should be defining the probability r_i of n_i for the current capacity Ł (0≤ Ł ≤ M_i) for n_i, *Pr{Ł}* :

$$\Pr\{Ł\} = \binom{M_i}{Ł} r_i^{Ł} (1 - r_i)^{M_i - Ł} \qquad (1)$$

3 Problem Formulation

According to another subsection, this issue is split into two bifurcation issues. Finding the best ability to distribute among a group of network nodes is the first bifurcation issue. The second branching problem is to search for an optimal feasible solution for the current variable to assess the system's dependability.

3.1 The Problem of First Branch

Assign the capacity to the set of nodes $(n_1, n_2, \ldots, n_{n_a})$ as $M = (M_1, M_2, \ldots, M_{n_a})$. The mathematical formula of this problem is as follows:

$$Minimize\ Ş \qquad (2)$$

$$\text{S. t.}$$

$$Maximize\ \Re_S \qquad (3)$$

Where $Ş = \sum_i^{n_a} M_i$ pointing sum of the allocated capabilities and correlating system stability, $\Re_S$.

3.2 The Second Branch Problem

Inspecting flow vectors that succeed in realizing the equations (4)-(6) (debated in, [Hassan, 2016; Liu et al., 2007].

($\mathcal{F}$ pointing to the flow vector characterized by $\mathcal{F} = (f_{1,1,1,1}, f_{1,1,2,1}, \ldots, f_{i,j,k_{i,j},1}, \ldots, f_{i,j,k_{i,j},nr}, \ldots, f_{\sigma,\theta,k_{\sigma,\theta},nr})$, where $f_{i,j,k,w}$ represents the amount of resource flow w on $MP_{i,j,k}$, nr refers to number of resources,

$\sum_{i=1}^{\sigma}\sum_{k=1}^{k_{i,j}} f_{i,j,k,w} = sd_{w,j}, w = 1, \ldots, nr; j = 1, \ldots, \theta$ (4) (θ sign to number of sinks)

$\sum_{j=1}^{\theta}\sum_{k=1}^{k_{i,j}} f_{i,j,k,w} \leq sr_{w,i}, w = 1, \ldots, nr; i = 1, \ldots, \sigma$ (5) (σ sign to the number of sources**)**

X pointing to capacity vector, $X = (x_1, x_2, \ldots, x_{na})$

$$x_l \leq M_l \quad l = 1, \ldots, na \qquad (6)$$

Where,

$$x_l = \sum_{i=1}^{\sigma}\sum_{j=1}^{\theta}\sum_{k=1}^{k_{i,j}}\sum_{w=1}^{nr}\{f_{i,j,k,w} | n_b \in MP_{i,j,k}\} \qquad (7)$$

The capacity vector's dependability is explicated by this equation.

$$\Re_X = \prod_{l=1}^{na} Pr\,\{x \geq x_l\} \qquad (8)$$

4 The Proposed Approach

The proposed approach is divided into two GAs, where the final GA (internal) is used to fulfil conditions (4) to (6) according to equations (4) to (6) in order to determine the optimal interconnection to assign to the nodes, and the first GA (outer) is used to do so. The resulting current vector maximizes system reliability. The components of our proposed approach include chromosome representation, population at the beginning, fitness, the crossing procedure, and the mutagenesis process.

4.1 The Outer GA

4.1.1 Representation

The primary distinction among various GA models can be found in the chromosomal representation procedure. Using a collection of randomly generated solutions, a population of chromosomes is generated as the first phase of the GA procedure (chromosomes). Then, scores based on fitness are assigned to each member of the community to assess them. The chromosome Ɱ is demonstrating by a string of long *(ɳa)*, *(ɳa)* relates to the quantity of nodes as shown in this equation ($Ɱ_1, Ɱ_2, \ldots, Ɱ_{ɳa}$).

4.1.2 Initial Population

Initializing the population is the first step in an evolutionary algorithm. A portion of the current generation solution is the community. Chromosomes can also be used to define the term "population Ɱ." Typically, the first generation, the initial population Ɱ*(0)*, is randomly generated. In the iterative process, a population $Ɱ_i$ of generation i (i = 1, 2,…) is constructed.

4.1.3 Fitness Function

The fitness function allows the algorithm to identify the chromosome with the highest fitness among other chromosomes, which can guide the genetic algorithm to the best solution [Abed and Tang, 2013]. Choosing fitness is one of the most common ways of choosing parents. In this process, individuals can become parents with a probability symmetric to their fitness. Therefore, fit individuals are more likely to reproduce and propagate their traits to the next generation. This therefore exerts selection pressure on the fitter individuals in the population, leading to the evolution of better individuals over time. The fit indices calculation for each solution i in the population is demonstrated by the following algorithm.

$fit(i) = Ş$

$for\ i = 1 : ṆP1$ (*ṆP1* pointing to the number chromosomes)

$Normalize\_fitness = fit(i) / \sum_i fit(i)$

$End\ for$

4.1.4 Selection Process

The genetic algorithm's selection process is one of the most crucial phases. The purpose of the selection phase is to identify and preserve the genetic traits of the fittest individuals for the following generation. We choose two groups of people (parents) based on fitness ratings. Higher physical fitness levels increase the likelihood that a person will be chosen for reproduction.

Even though roulette wheel selection is used in this study, it is thought to be the most popular kind of selection. It is employed to choose possibly beneficial pass solutions and is also a superior standard technique of familial selection [Abed and Tang, 2013]. Selection in this method is symmetrical with individual fitness. The more fit an individual has (best chromosome), the more likely to be selected. In this research, the 2 different parents were chosen via a roulette wheel mechanism based on their cumulative total.

//Roulette wheel algorithm

Begin
$for\ j = 1\ to\ ŅP1$
$for\ i = j\ to\ ŅP1$
$CSum\ (j) = CSum\ (j) + Normalize\_fitness(i)$ //$CSum$ stands for commutative sum
$End\ for\ i$
$End\ for\ j$
$Generate\ number\ random$ ƙ ϵ $[0,1]$
$for\ j = 1\ to\ ŅP1$
if ƙ $> CSum\ (j)$
$parent\ 1 = j - 1$
$end\ if$
$end\ for\ j$
$Generate\ number\ random$ ƙ ϵ $[0,1]$
$for\ j = 1\ to\ ŅP1$
if ƙ $> CSm\ (j)$
$parent\ 2 = j - 1$
$end\ if$
$end\ for\ j$
End

4.1.5 Crossover

In a genetic algorithm, crossover is the most crucial step. A crossover point is chosen at random from the genes for each set of parents to mate. The process of producing offspring helps to exchange the We implement a one-point crossover operator that replicates the genes from the very first parent even before crossover point, followed by the genes from the second parent just after crossover point, which accepts a single crossover point at irregular intervals. [Malik, 2019].

$$N_{c1} = [N_{p1}(i)]_{i=1}^{ϱ} + [N_{p2}(i)]_{i=ϱ+1}^{ɳa}$$

$$N_{c2} = [N_{p2}(i)]_{i=1}^{ϱ} + [N_{p1}(i)]_{i=ϱ+1}^{ɳa}$$

Where $N_{c1}$and $N_{c2}$are the children and $N_{p1} and\ N_{p2}$are the parents and $ϱ$ is the random cut-point.

4.1.6 Mutation

The act of introducing new genetic material into a population is known as a mutation [Abed and Tang, 2013]. Through the application of this technique, we were able to sustain genetic diversity from one population generation to another. We also use it to avoid the genetic algorithm getting stuck in a local minimum. or the arrival at a less than ideal outcome.
Generate a random number ∂ ∈ [0,1]
If ∂ ≤ Ɱu1 then (*Ɱu1* pointing to mutation rate)
{
For i = 1 to ɳa do
$N_{c2}\ (i) =$ ɚ , ɚ ∈ $\{0,1, \ldots, Ɱ_i\}$
End for

}

4.2 The Internal GA

The ideal set of lower bound points that enhance the system's reliability are chosen using an internal GA. Maximizing system reliability requires a flexible formula of the optimal reduction control issue.

Identify the ideal set of X.

s.t.

$\Re_S$ is maximized.

Equations used to resolve searching problem for flow vectors given in section 4.2.

4.2.1 Representation

The chromosome $ƒ$ is symbolizing by a series of length *(Lƒ)*, *(Lƒ)* mentions to the number result by multiply number to minimal path *(ɳp)* and the number to resource *(ɳr)* ($ƒ_{1,1,1,1}$, $ƒ_{1,1,2,1}$, …, $ƒ_{i,j,k_{i,j},1}$, …, $ƒ_{i,j,k_{i,j},ɳr}$, …,$ƒ_{\sigma,\theta,k_{\sigma,\theta},ɳr}$).

4.2.2 Initial Population

A population of people is where the process begins. Every person is a part of the answer to the issue you're trying to solve. Genes are a set of parameters (variables) that represent an individual. A Chromosome is made up of a string of genes (solution).

4.2.3 Fitness Function

The fitness function is a function that requires an alternative solution to the problem as input and outputs how "fit" or "good" the alternative solution is in relation to the research problem under considering. The next algorithm shows how to determine the fitness for each solution i in the population, considering X^icorresponding to $ƒ^i$, each $ƒ^i$satisfies equations (4-6) presented in section 4.2.

$fit(i) = \Re_{X^i}$
$for\ i = 1: ƝP2$ **(** $ƝP2$ sign to number of chromosomes)
    $Normalize\_fitness = fit(i)/\sum_i fit(i)$
$End\ for$

4.2.4 Selection Process

We used roulette wheel mechanism to pick out two parents as shown in detail in section 5.1.4.

//Roulette wheel algorithm
Being
$for\ j = 1\ to\ ƝP2$
    $for\ i = j\ to\ ƝP2$
        $CSum\ (j) = CSum\ (j) +\ Normalize\_fitness(i)$
    $End\ for\ i$
$End\ for\ j$
$Generate\ number\ random\ ƙ \in [0,1]$
$for\ j = 1\ to\ ƝP2$

$if\ \Bbbk > CSum\ (j)$
$parent\ 1 = j - 1$
$end\ if$
$end\ for\ j$
$Generate\ number\ random\ \Bbbk \epsilon\ [0,1]$
$for\ j = 1\ to\ NP2$
$if\ \Bbbk > CSum\ (j)$
$parent\ 2 = j - 1$
$end\ if$
$end\ for\ j$
end

4.2.5 Crossover

Crossover is described in section 5.1.5 with some modification to be suitable for generating new offspring for F.

$$F_{c1} = [F_{p1}(i)]_{i=1}^{\varrho} + [F_{p2}(i)]_{i=\varrho+1}^{LF}$$

$$F_{c2} = [F_{p2}(i)]_{i=1}^{\varrho} + [F_{p1}(i)]_{i=\varrho+1}^{LF}$$

Where F_{c1} and F_{c2} $are\ the\ new\ vectors\ generaction\ and\ \varrho\ is\ the\ random\ cut - point.$

4.2.6 Mutation

Another genetic operator found in nature is mutation. The Explorative process utilizes mutation to preserve genetic variation in a population through one generation to another [Bourmistrova and Khantsis, 2010].

Produce a different value $\partial \in [\ 0,1]$

If $\partial \leq Mu2$ *then* ($Mu2$ pointing to mutation rate)
{
For i = 1 to LF *, do*
$F_{c1}\ (i) = \vartheta , \vartheta \in \{0,1, \dots , MaxF\}$, $MaxF$ the max value can be specific to f_i
End for
}

5 Evaluating Rs

If $X^1, X^2, \dots , X^l$ are each lower boundary points obtained from the generated vectors $X^1, X^2, \dots , X^{\mathrm{NP2}}$, [Lin, 2001], then system robustness $\Re_S$ is computed using equation (9)

$$\Re_S = pr \bigcup_{i=1}^{l} \{W | W \geq X^i\} \qquad (9)$$

$pr\{W\} = pr\{w_1\}. pr\{w_2\}. \dots . pr\{w_{n_{eq}}\}$. Afterward, we apply the iterative sum of disjoint products (RSDP) method introduced by [Zuo, 2007]

When $TM_1 = pr\{w \geq X^1\}$

And $TM_i = pr\{W \geq X^i\} - pr\{\cup_{j=1}^{i-1}\{W \geq X^{j,i}\}, \quad for\ i \geq 2$

$$\Re_S = pr \bigcup_{i=1}^{l} \{W | W \geq X^i\} = \sum_{i=1}^{l} TM_i \qquad (10)$$

If $W = (w_1, w_2, \dots, w_e, \dots, w_{ṇeq})$

$$pr(W) = \prod_{e=1}^{ṇeq} pr(w_e) \qquad (11)$$

6 The Proposed Approach

Start

Enter network data like the minimum path, requirement, and assets, *ɱG1, ƝP1, ƝG1, Ꞓr1, Ɱu1, ɱG2, ƝP2, ƝG2, Ꞓr2, Ɱu2.*

(*ɱG1 sign to limit number of generations, ƝP1 sign to number of chromosomes, ƝG1 pointing to number of genes equals to ŋa, Ꞓr1refers to* crossover rate*, Ɱu1 sign to mutation rate, ɱG2 sign to limit number of generations, ƝP2 sign to number of chromosomes, ƝG2 pointing to number of equals to Lƒ, Ꞓr2refers to* crossover rate, *, Ɱu2 pointed to mutation rate).*

Start outer GA

Randomly generate an initial population up to Ɱ.
Analyze the starting population.
While $ꞡ \leq ɱG1$, do
    While Ᵽ $\leq ƝP1$, do
        Choosing two chromosomes.
        Applying crossover and mutation to generate new childs.
        Ᵽ = Ᵽ + 1
    end do
        Evaluate the existing population.
        Save the appropriate answer to *Ɱ*.
        ꞡ = ꞡ + 1
end do
Report the best solution found.

Start internal GA

Randomly generate an initial population up to ƒ.
Evaluate the initial population.
While $ꞡ \leq ɱG2$, do
    While Ᵽ $\leq ƝP2$, do
        Select two chromosomes.
        Use GA operators to obtain new childs.
        Ᵽ = Ᵽ + 1
    end do
Asses the current population.
Save the best solution to ƒ.
ꞡ = ꞡ + 1
end do

Report the optimal lowers found.
End internal GA
Calculate the system reliability $\Re_S$.
End outer GA
End

7 Studied Cases

In the following subsections, three networks have been used to test the presented methodology. The appropriate GA parameter values were, *MG1* =100, *NP1* = 20, *Cr1* = 0.95, *Mu1* = 0.05, *MG2* =100, *NP2* = 10,15, and 20, *Cr2* = 0.95, and *Mu2* = 0.05.

7.1 Network with Three Source and Two Sinks

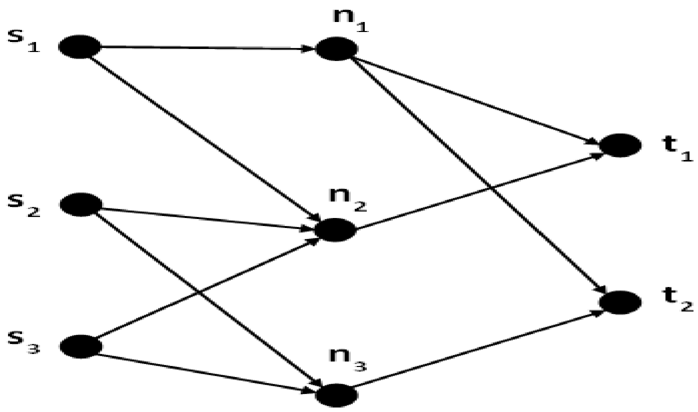

Figure 1: network with three source and two sinks

this route has 8 nodes as shown in figure 5, it has 7 *MPs*, $MP_{1,1,1} = \{s_1,n_1,t_1\}$, $MP_{1,1,2} = \{s_1,n_2,t_1\}$, $MP_{1,2,1} = \{s_1,n_1,t_2\}$, $MP_{2,1,1} = \{s_2,n_2,t_1\}$, $MP_{2,2,1} = \{s_2,n_3,t_2\}$, $MP_{3,1,1} = \{s_3,n_2,t_1\}$, $MP_{3,2,1} = \{s_3,n_3,t_2\}$. *neq* =*NG1*= 8, *LF* =*NG2*= 21, Resources: *RR* = $(sr_{1,1}, sr_{1,2}, sr_{1,3}, sr_{2,1}, sr_{2,2}, sr_{2,3}, sr_{3,1}, sr_{3,2}, sr_{3,3})$ = (5, 2, 3, 5, 3, 2, 2, 2, 3). Demand: *DD* = $(sd_{1,1}, sd_{1,2}, sd_{2,1}, sd_{2,2}, sd_{3,1}, sd_{3,2})$ = (3, 1, 2, 2, 1, 3). Table 1 shows the optimal minimal level Ş achieve maximum $\Re_S$ when *NP2* = 10, 15 and 20.

Table 1: The best solution found to Ş by change *NP2* and clarify $\Re_S$

NP2	*M*	Ş	$\Re_S$
10	[6 14 9 13 16 5 18 15]	96	0.999970
15	[9 10 5 12 14 7 18 15]	90	0.999997
20	[13 13 6 18 5 13 17 10]	95	0.999923

7.2 Network with Two Source and Two Sinks

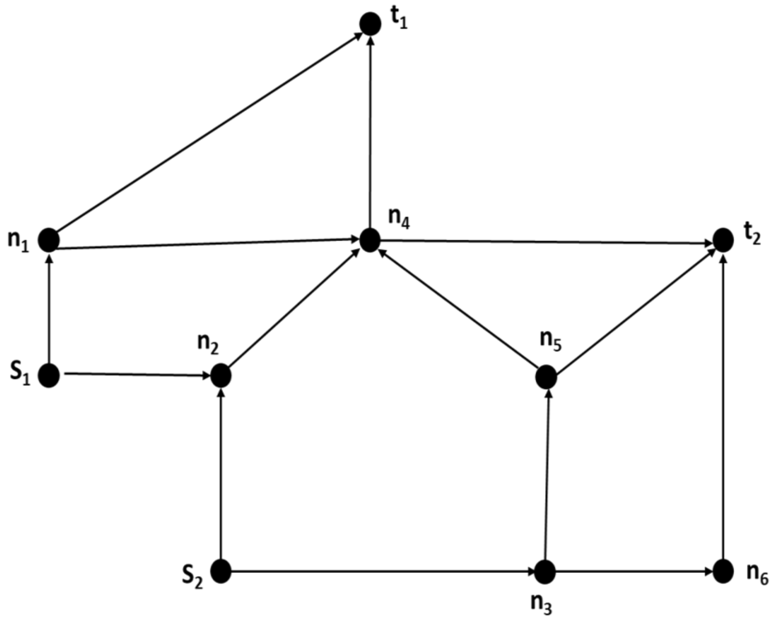

Figure 2: network with two source and two sinks

this route has 10 nodes as shown in figure 6, it has 11 *ⱮPs*, *ⱮP*$_{1,1,1}$= {s_1, n_1, t_1}, *ⱮP* $_{1,1,2}$= {s_1, n_1, n_4, t_1}, *ⱮP* $_{1,1,3}$= {s_1, n_2, n_4, t_1}, *ⱮP* $_{1,2,1}$= {s_1, n_1, n_4, t_2}, *ⱮP* $_{1,2,2}$= {s_1, n_2, n_4, t_2}, *ⱮP* $_{2,1,1}$= {s_2, n_2, n_4, t_1}, *ⱮP* $_{2,1,2}$= {s_2, n_3, n_5, n_4, t_1}, *ⱮP* $_{2,2,1}$= {s_2, n_2, n_4, t_2}, *ⱮP* $_{2,2,2}$= {s_2, n_3, n_5, n_4, t_2}, *ⱮP* $_{2,2,3}$= {s_2, n_3, n_5, t_2}, and *ⱮP* $_{2,2,4}$= {s_2, n_3, n_6, t_2} *ɳeq* =*ƝG1*=10, *LƑ*=*ƝG2*=22, Resources: *ƦƦ* = ($sr_{1,1}$, $sr_{1,2}$, $sr_{2,1}$, $sr_{2,2}$) = (15, 17, 10, 13). Demand: *DD* = ($sd_{1,1}$, $sd_{1,2}$, $sd_{2,1}$, $sd_{2,2}$) = (4, 6, 4, 5).

Table 2 shows the best minimum Ş achieve maximum$\Re_S$ when *ƝP2* = 10, 15 and 20.

Table 2: The best solution found to Ş by change *ƝP2* and clarify $\Re_S$

ƝP2	*Ɱ*	Ş	$\Re_S$
10	[8 15 17 10 14 22 12 24 24 22]	168	0.997224
15	[15 13 12 15 13 20 19 20 11 17]	155	0.999897
20	[18 10 7 24 13 21 13 10 23 24]	163	0.980268

7.3 Network with Two Source and Three Sinks

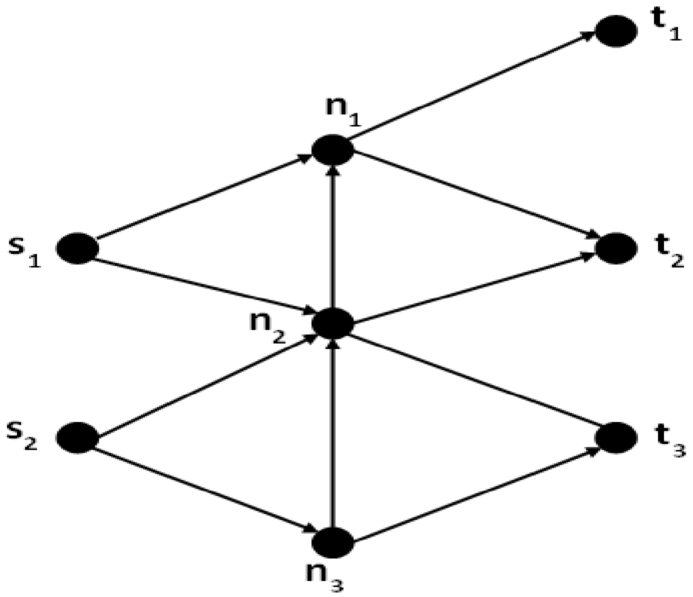

Figure 3: network with two source and three sinks

This network has 8 nodes as shown in figure 7, it has 13 MPs, $MP_{1,1,1}$= { s_1,n_1,t_1}, $MP_{1,1,2}$= { s_1,n_2,n_1,t_1}, $MP_{1,2,1}$= { s_1,n_1,t_2}, $MP_{1,2,2}$= { s_1,n_2,t_2}, $MP_{1,2,3}$= { s_1,n_2,n_1,t_2}, $MP_{1,3,1}$= { s_1,n_2,t_3}, $MP_{2,1,1}$= { s_2,n_2,n_1,t_1}, $MP_{2,1,2}$= { s_2,n_3,n_2,n_1,t_1}, $MP_{2,2,1}$= { s_2,n_2,t_2}, $MP_{2,2,2}$= { s_2,n_3,n_2,t_2}, $MP_{2,3,1}$= { s_2,n_2,t_3}, $MP_{2,3,2}$= { s_2,n_3,n_2,t_3}, *and* $MP_{2,3,3}$={ s_2,n_3,t_3}. $\eta eq = NG1$=8, $LF=NG2$=26, Resources: $RR = (sr_{1,1}, sr_{1,2}, sr_{2,1}, sr_{2,2})$ = (10, 19, 14, 19). Demand: $DD = (sd_{1,1}, sd_{1,2}, sd_{1,3}, sd_{2,1}, sd_{2,2}, sd_{2,3})$ = (3, 2, 2, 3, 3).

Table 3 shows the best minimum Ş achieve maximum$\Re_S$ when $NP2$ = 10, 15 and 20.

Table 3: The best solution found to Ş by change $NP2$ and clarify $\Re_S$

$NP2$	M	Ş	$\Re_S$
10	[13 18 7 10 15 9 9 16]	97	0.916984
15	[6 13 12 10 13 17 18 8]	97	0.974026
20	[9 14 11 13 8 13 16 12]	96	0.999993

8 Conclusions

This study succeeded in solving the MMSFNs robust design issue. In this research, the robust design problem was solved using a GA-based approach. In order to achieve the highest system reliability value, we divided the problem into two branches. The first branch investigated the best nodes' capacity when assigned to network elements with the lowest sum. The second branch searched for the best lower vectors. We applied the suggested GA-based approach to a collection of networks, and we got good results that helped us solve the problem at hand. Due to the lack of prior disagreements regarding the robust design problem for MMSFNs with nodes failure study, we are unable to recall any comparative studies.

References

Abed M. H. and Tang A. Y. C., " Hybridizing Genetic Algorithm and Record-to-Record Travel Algorithm for Solving Uncapacitated Examination Timetabling Problem", electronic Journal of Computer Science and Information Technology (eJCSIT), Vol. 4, No. 1, 2013.

Al-Shaalan A. M., Chapter, "Reliability Evaluation of Power Systems", DOI: http: //dx.doi.org /10.5772/ intechopen.85571.

Bobbio A., Terruggia R., Ciancamerla E., and Minichino M., "Reliability Analysis of Multi-source Multi-sink Critical Interacting Systems", IEEE, PP. 127-132, 2011.

Boubaker S., Radwan N. H., Hassan M. R., Alsubaei F. S., Younes A. and Sennary H. A., "Robust Design Problem for MultiSource Multi-Sink Flow Networks Based on Genetic Algorithm Approach", Mathematics 2023, 11, 3902.

Bourmistrova A. and Khantsis S., " Genetic Programming in Application to Flight Control System Design Optimisation", Published: February 1st, 2010, DOI: 10.5772/8055

Chabrier A., Danna E., LE Pape C. and Perron L., "Solving a Network Design Problem", Annals of Operations Research, Vol. 130, PP. 217–239, 2004.

Chen S. G., "An optimal capacity assignment for the robust design problem in capacitated flow networks", Applied Mathematical Modelling, Vol. 36, PP. 5272-5282, 2012.

Cho B. R. and Shin S., "Quality Improvement and Robust Design Methods to a Pharmaceutical Research and Development" Mathematical Problems in Engineering Volume 2012, Article ID 193246, 14 pages doi:10.1155/2012/193246

Du W., "The genetic algorithm optimize computing applications in computer network reliability analysis", BTAIJ, Vol. 10, Iss. 9, PP. 3670-3678, 2014.

Elahabad M. F. and Amiri N. M., "Reliability assessment of a multi-source multi-sink stochastic-flow network in terms of minimal paths Majid", International Iranian Operation Research Conference, PP. 1065-10969, 2016.

Elden N. N., Hassan M. R. and Abd El-Aziz M., " Components Assignment Problem for Multi-Source Multi-Sink Flow Networks with Reliability and Budget Constraints", Journal of Computer and Communication, Vol. 10, PP. 99-111, 2022.

Forghani-elahabad M. and Mahdavi-Amiri N., “A new efficient approach to search for all multi-state minimal cuts,” IEEE Trans. Reliab. vol. 63(1), pp. 154-66, 2014.

Forghani-elahabad M. and Mahdavi-Amiri N., “An efficient algorithm for the multi-state two separate minimal paths reliability problem with budget constraint,” Reliab. Eng. Syst. Saf. vol. 142, pp. 472-481, 2015.

Forghani-elahabad M. and Mahdavi-Amiri N., “An efficient method for generating all minimal vectors for the q SMPs reliability problem with time and budget constraints,” IEEE Trans. Reliab. vol. 65 (2), pp. 828-842, 2016.

Forghani-elahabad M. and Mahdavi-Amiri N., “An improved algorithm for finding all upper boundary points in a stochastic-flow network,” Appl. Math. Model., vol. 40 (4), pp. 3221-3229, 2016.

Forghani-elahabad M. and Mahdavi-Amiri N., “On search for all d-MCs in a network flow,” Iranian Journal of Operations Research, vol. 4(2), pp. 108-126, 2013.

Fowlkes and Creveling, "Engineering Methods for Robust Product Design". Addison-Wesley Publishing Company,1995.

Hamed A. Y., Alkinani M. H. and Hassan M. R., "A Genetic Algorithm to Solve Capacity Assignment Problem in a Flow Network" Computers, Materials & Continua CMC, vol.64, no.3, pp.1579-1586, 2020.

Hassan M. R. and Abdou H., "Multi-Source Multi-Sink Stochastic-Flow Networks Reliability under Time Constraints", Indian Journal of Science and Technology, Vol 12(22), 2019.

Hassan M. R., "Maximizing Reliability of the Capacity Vector for Multi-Source Multi-Sink Stochastic-Flow Networks Subject to An Assignment Budget", Journal of Industrial and Management Optimization, Vol. 17, Num. 3, PP. 1253-1267, 2021.

Hassan M. R., "System Reliability Optimization of Multi-Source Multi-Sink Stochastic Flow Networks with Budget Constraint", International Journal of Reliability, Quality and Safety Engineering, Vol. 28, No. 4, PP. 2150025-1-2150025-11, 2021.

Hassan M. R., Solving Flow Allocation Problems and Optimizing System Reliability of Multi-Source Multi-Sink Stochastic Flow Network, The International Arab Journal of Information Technology (IAJIT), Vol. 13, PP. 477-483, 2016.

Katayama K. And Sakamoto H., " The Efficiency of Hybrid Mutation Genetic Algorithm for the Travelling Salesman Problem", Mathematical and Computer Modelling, Vol. 31, PP. 197-203, 2000.

Lai H. Y. , "Genetic Algorithm Based Optimization in Computer Network Applications", 2001.

Lin Y. K. and Yeh C. T., "Computer network reliability optimization under double-resource assignments subject to a transmission budget", Information Sciences Vol. 18, PP. 582–599, 2011.

Lin Y. K., " A MP-BASED ALGORITHM FOR A MULTICOMMODITY STOCHASTIC-FLOW NETWORK WITH CAPACITY WEIGHTS", International Journal of Industrial Engineering, Vol. 16, No. 4, PP. 282-292, 2009.

Lin Y. K., "A simple algorithm for reliability evaluation of a stochastic-flow network with node failure". Computer and Operations Research, 28, 1277–1285., 2001.

Liu L., Optimization Design of Computer Network Reliability Based on Genetic Algorithms, CHEMICAL ENGINEERING TRANSACTIONS, VOL. 51, PP. 775 : 780, 2016.

Liu Q., Zhang H., Ma X. and Zhao Q., "Genetic Algorithm-based Study on Flow Allocation in a Multicommodity Stochastic-flow Network with Unreliable Nodes", Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, PP. 576-581, 2007.

Lobo L.M.R.J and Chavan S. B., "Genetic Algorithm in Network Security", International Journal of Computer Applications, (0975 – 8887) Vol. 53, No.8, September 2012.

López-Prado J. L., Vélez J. I., and Garcia-Llinás G. A., "Reliability Evaluation in Distribution Networks with Microgrids" Review and Classification of the Literature, MDPI Journal, Vol. 13, Issue 23, 2020.

Malik A., "A Study of Genetic Algorithm and Crossover Techniques, International Journal of Computer Science and Mobile Computing", Vol.8, Issue.3, PP. 335-344, 2019.

Massoudi S., Picard C. and Schiffmann J., "Robust design using multiobjective optimisation and artificial neural networks with application to a heat pump radial compressor" Published online by Cambridge University Press, Vol. 8, PP. 1-31, 2021.

Melanie M., "An Introduction to Genetic Algorithms", 1999 First MIT Press paperback edition, 1998 Copyright © 1996 Massachusetts Institute of Technology

Niu Y. F., " Performance measure of a multi-state flow network under reliability and maintenance cost considerations", Reliability Engineering and System Safety, 215, 107822, 2021.

Niu Y. F., Xu Z., He C., Ding D., and Liu Z. Z., "Capacity Reliability Calculation and Sensitivity Analysis for a Stochastic Transport Network". IEEE Access, 8, 133161-133169, doi: 10.1109/ACCESS.2020.3010525, 2020.

Radwan N. H., Hassan M. R., and Hussein M. E., "A genetic algorithm to solve the robust design problem for a flow network with node failure", Transaction on networks and communication, Vol. 8, No. 4, PP. 1-10, 2020.

Radwan N. H., Hassan M. R., and Hussein M. E., "Solving the robust design problem for a two-commodity flow network with node failure", American Journal of engineering and applied sciences, Vol. 13, No. 4, PP. 837-845, 2020.

Sarintip S., and Kailash K. C., "An algorithm for lower reliability bounds of multistate two-terminal networks". IEEE Transactions on Reliability, 55(2), 199-206, 2006.

Yang G., "Life Cycle Reliability Engineering". First published:24 January 2007, Print ISBN:9780471715290, Online ISBN:9780470117880, DOI:10.1002/9780470117880, Inc. John Wiley & Sons, 2007.

Yong K., " Robust Design and Reliability", Article in Transactions of Nanjing University of Aeronautics and Astronautics,1998.

Zuo M. J., Tian Z., and Huang H. Z., "An efficient method for reliability evaluation of multistate networks given all minimal path vectors". IIE Transactions, 39, 811-817, 2007.