A Multi-Objective Optimization Model for a Reliable Generalized Flow Network Design

Mina Dehghani[a], Vahab Vahdat[b], Maghsoud Amiri[a], Elaheh Rabiei[c], Seyedmohammad Salehi[d]

[a]*Department of Industrial Management, Allameh Tabatabai University, Tehran, Iran*
[b]*Department of Mechanical and Industrial Engineering, Northeastern University, Boston, USA*
[c]*Univeristy of California Los Angeles, Los Angeles, USA*
[d]*Department of Computer and Information Sciences, University of Delaware, USA*

Abstract

Design of a reliable network in presence of flow loss has become the primary objective of today's network designers. However, there are other important conflicting objectives that hinder the process of efficient network design. This study proposes a multi-objective optimization model for reliable communication flow networks, including maximizing the network reliability, minimizing total cost, and maximizing network flow, simultaneously. The total cost comprises the cost of construction of network arcs and the cost of flow, while arcs may fail to operate in full-capacity and may only function to a fraction of their capacity. The reliability-based network-design is modeled as a mixed-integer linear programming and solved by three metaheuristic multi-objective methods namely multi-objective particle swarm optimization (MOPSO) and two versions of non-dominated sorting genetic algorithm (i.e., NSGA-II and NSGA-III). In order to select the best compromise solution from the Pareto front members, a fuzzy-based mechanism is utilized. Finally, in order to measure the performance of the three algorithms, several numerical examples in small and large-scale are solved. The computational results indicate that NSGA-III is superior to MOPSO and NSGA-II in terms of convergence rate and running time especially for large-scale problems.

Keywords: Reliability, reliable network design, multi-objective optimization, generalized flow network, NSGA-III, MOPSO.

ACRONYMS

NSGA-II	Non-dominated Sorting Genetic Algorithm-II
NSGA-III	Non-dominated Sorting Genetic Algorithm-III
PSO	Particle Swarm Optimization
MOPSO	Multi-Objective Particle Swarm Optimization

*Corresponding author
Email address: salehi@udel.edu (Seyedmohammad Salehi)

NP	Non-deterministic Polynomial Time
NOM	Number of Pareto Members
MID	Mean Ideal Distance
$\mathrm{HV_I}$	Hyper-Volume Indicator

MATHEMATICAL MODEL NOTATIONS

s	Source node
t	Sink node
q_{ij}	Probability of failure for arc (i, j)
$\mathrm{CB_{ij}}$	Cost of building arc (i, j)
$\mathrm{CF_{ij}}$	Per-unit cost of flow on arc (i, j)
l_{ij}	Lower bound of flow on arc (i, j)
u_{ij}	Upper bound of flow on arc (i, j)
a'_{ij}	Loss factor on arc (i, j), $0 \leq a'_{ij} \leq 1$
a_{ij}	Arc multiplier, calculated as $a_{ij} = 1 - a'_{ij}$
F_{in}	Input flow
F_{out}	Output flow
x_{ij}	Binary decision variable for Network topology design which is 1 when there is connection between i and j, and is 0 otherwise
f_{ij}	Continuous decision variable indicating the amount of flow on arc (i, j)
$R(x)$	Network reliability
$H(d)$	Jans upper bound
d_i	Degree of node i

SOLUTION METHODS NOTATIONS

λ	Lebesgue measure
H	Number of reference points
$\mathrm{n_O}$	Number of objective functions
$\mathrm{n_D}$	Number of divisions on every objective axis
$\mathrm{n_P}$	Population size
P_t	Population at generation t
Q_t	Offspring population at generation t
R_t	Combined Population and offspring at generation t
M_{it}	Maximum number of iterations for each metaheuristic algorithm
F_i^{min}, F_i^{max}	Minimum and the maximum value of the i^{th} objective function
l_i	Distance of i^{th} Pareto solution from the ideal point
D_l	Individuals of a non-dominated solutions set
μ	Compatibility level
P_m	Probability of mutation

P_c	Probability of crossover
Δ	Diversity metric
n	Number of Pareto solutions
A	Reference point
x_i^k	Position of particle i^{th} in iteration k^{th}
v_i^k	Velocity of particle i^{th} in iteration k^{th}
c_l	Local learning coefficient in MOPSO
c_g	Global learning coefficient in MOPSO
μ_{rate}	Mutation rate in MOPSO
n_r	Number of repository in MOPSO
w	Inertia weight in MOPSO
w_{damp}	Inertia weight damping rate,

1. Introduction

In graph theory, a network is defined as a series of nodes that are connected to each other through arcs. Networks have been widely used to design and solve problems in different fields such as transportation, supply chain, and telecommunications with the intention to transport commodities/information between the nodes to meet a distributed demand-supply pattern. The topology or architecture of a network describes how the nodes are connected by arcs. For instance, a complete graph is a fully connected network where all nodes are interconnected by arcs.

The telecommunication industry has seen several advances in recent decades, continuing to improve both in hardware and the efficiency of underlying algorithms (Mousavi et al., 2019). Modern wireless networks have been developed based on high-speed cellular/Wi-Fi technologies and very efficient routing/scheduling algorithms, with capabilities of transporting sophisticated services including voice, video, and data with different quality of service (QoS) requirements (Esmailpour et al., 2013; Salehi et al., 2018). With increasing data demand, new technological networks need to provide their services with higher reliability. Thus, it is crucial to create efficient computational methods to estimate the reliability of the technological networks. However, even for simplistic networks, it can be shown that most optimal reliable design problems are NP-complete (Provan & Ball, 1983). Consequently, finding an optimal solution for realistic network size in a reasonable time frame has not been achieved yet (Larsson, 2014). To estimate the optimal solution, approaches such as graph theory, optimization, simulation methods, and probability theory concepts are used to design the novel communication networks.

While classical network problems assume flow preservation over the network, in many real applications, flow is not conserved. Instead, generalized networks can incorporate some levels of gains or losses as the flow goes along an arc. Each arc has a multiplier to represent the amount of gains and losses. As a result, generalized network can be applied to model flow distribution systems in which evaporation, erosion or sediment would waste the flow in some pipes. For instance, in supply chain networks, the damage to cargo in transit can detriment the flow. Another example can be in power generation systems where some energy loss is unavoidable to transfer electricity from one location to the other. In fact, the generalized network problems can be discerned as the classic maximum flow problem with capability to model the leaking flow.

Designing an optimal network that is both cost-efficient and reliable is emerging as an example of well-performing network. In this study, we formulate and analyze a multi-objective optimization model for reliable generalized flow network with respect to the conflicting objectives, including simultaneously maximizing the network reliability, maximizing network flow and minimizing the total cost. Since computational complexity of the proposed model makes it impractical to be solved by exact methods, three forefront metaheuristic algorithms namely multi-objective particle swarm optimization (MOPSO) algorithm and non-dominated sorting genetic algorithm-II and III (NSGA-II and NSGA-III) are utilized. The results indicate the effectiveness of the solution procedure for solving many objective optimization problems.

The main contributions of this paper are summarized as follows:

- In order to design a realistic communication network, three conflicting objectives are considered in the mathematical model. While maximizing the reliability of a network is the premier objective of this research, most decisions are not made by network reliability in solitary. In addition, minimizing the network cost and maximizing network flow need to be considered to provide a comprehensive yet robust solution. Unlike previous studies that utilize at most two objectives, this research recognizes the importance of flow, cost and reliability and develops a model to simultaneously optimize three goals in one platform. The total cost comprises the cost of construction of network arcs and the cost of flow. The network reliability is estimated by the Jans upper bound (Rong-Hong, 1993) which can be used in large-scale networks where the exact methods are not feasible. Finally the flow of the network from source node to sink node is favored to be maximized while the arcs have different reliabilities.

- In order to decide on selecting the prior or best compromise solution from the Pareto front, a fuzzy-based mechanism is applied. Therefore, the proposed model leads to a final reliable generalized flow network design, which represents the desired compromise between the different objectives from the decision-makers perspective.

The rest of the paper is organized as follows. In the following Section, we review the state-of-the-art literature relevant to maximum flow/minimum cost problem, generalized network flow problems, and reliable network design. Also, Section 2 discusses the application of metaheuristic approaches in solving the network problems such as multi-objective optimization using MOPSO, NSGA-II, and NSGA-III. In Section 3, we provide a summary of problem statement with underlying assumptions and the mathematical model for reliable communication networks. In Section 4, we present a holistic description of solution procedure for NSGA-II, NSGA-III, and MOPSO and tuning their principal parameters. Performance of proposed solution methods are further evaluated by numerical examples in Section 5. Strengths and limitations of the proposed model and methods are further investigated in Section 6. Finally, Section 7 concludes the paper and suggests future research directions.

2. Literature Review

Maximum flow problems in which the objective is transferring the maximum flow by the means of the arcs capacity have been studied by many scholars. However, investigating the flow deterioration-effect in arcs, rather than flow preservation through the

network, makes these models a good representation of many real-case problems, especially in electrical power networks or supply chains. Yeh (2013) presented a deterministic multi-state flow network, considering deterioration effect. Authors evaluated the reliability of the model by a path-based search algorithm. Kuipers et al. (2014) studied a maximum-flow problem in stochastic networks and used a convex optimization-based algorithm to solve the proposed problem. Authors utilized a log-concave probability distribution for delay and bandwidth of arcs. It is shown in Kuipers et al. (2014) that by applying an additional delay constraint, the problem becomes NP-hard. Jane & Laih (2017) studied a multi-state networks in which not only multi-valued capacities and associated operation probabilities were considered, but also the transit period was used for weighting the arcs. They presented an algorithm to compute the probability distribution of dynamic max-flow value.

A variation of maximum flow problem is to minimize the minimum cost of the flow. The objective in minimum cost flow problem is to send a certain flow from supply to demand nodes with smallest cost, subject to arc capacity constraints. Minimum cost flow problems have been also studied excessively, both in application and theory. Hochbaum (2016) explored links between the minimum cost flow problem and the time cost trade-off (TCF) problem in project management. Their proposed approach resulted in faster polynomial algorithms for both problems. Sherman (2017) considered an unlimited flow with minimal cost in undirected graphs and presented a new framework for optimizing it. In this framework, to achieve highly accurate solvers, they combined available imprecise solvers with non-Euclidean preconditioning. Cohen & Megiddo (1994) proposed the uncapacitated generalized transshipment problem and the generalized circulation problem, and solved them by an iterative approximation algorithm. They also considered these problems on bi-directed generalized networks.

Another important network design property is the reliability of networks that investigate the failure probability of networks' components. Malinowski (2016) provides a survey of reliability in computer networks, water distribution network, pipeline systems, and electricity distribution networks. Zuev et al. (2015) presented a quantitative evaluation for the reliability of network service in a stochastic environment. They presented a model for a general network reliability problem and used subset simulation models for calculating the service reliability. Then, small failure probabilities of complex dynamic systems were computed using Markov Chain Monte Carlo technique. Lin & Chang (2013) evaluated the performance of a stochastic-flow manufacturing network considering the failure rates of stations. Due to the station failure, not all demands were met, and the probability of demand realization was considered as a performance indicator. Therefore, authors constructed a manufacturing system considering the uncertain capacities with probabilistic distributions and the multi-fold production lines. Elshqeirat et al. (2015) form a communication network with the goal of minimizing the cost satisfying a preconditioned reliability constraint. For solving the problem, authors applied a dynamic programming approach. Since the problem was demonstrated to be NP-hard, they also proposed three greedy heuristics. A flow network with inflow points, transit-only nodes, and outflow points was considered by Malinowski (2016). In their network, arcs were directed, and components were repairable with constant failure and repair rates. The performance indicator was defined as the ratio of the total satisfied demand to the total desired demand at all outflow points.

The problem of designing a network that takes reliability into account is NP-hard

(Provan & Ball, 1983), and the current literature advocates that exact methods were only able to solve small network design problems with 15 or less arcs. In turn, approximate methods and heuristics have been generally utilized for solving reliable network design problems. Yeh (2015) developed an approximate algorithm based on depth first search (DFS) to solve a multi-state flow network such that the net flow into and out of a node was set to zero. This algorithm found all minimal paths and removed all infeasible candidates of the minimal path reliability problem. Srivaree-ratana et al. (2002) estimated all-terminal network reliability using artificial neural network. They developed and trained neural networks based on the all-terminal reliability of network topologies and arc reliabilities.

For larger networks, metaheuristic techniques have been utilized. Metaheuristics are high-level searching procedures that provide sufficiently good solution approximations to large-scale optimization problems. Examples of metaheuristics applied to reliability in network designs are genetic algorithms (Deeter & Smith, 1997), simulated annealing (Suman, 2003), tabu search (Beltrán & Skorin-Kapov, 1994), and ant colony algorithms (Watcharasitthiwat & Wardkein, 2009). The design of reliable communication networks with the goal of minimizing the cost with bidirectional arcs was first considered by Dengiz & Smith (2000). They used genetic algorithm focusing on the costs and reliability of the arcs. To form a reliable telecommunication network with two-node connectivity Konak & Smith (2011) developed a bi-objective genetic algorithm, which encompassed three methods of an exact method for calculating reliability, a Monte Carlo estimation technique, and an upper bound for the all-terminal reliability of networks with optional arc reliabilities. Deb et al. (2000) developed the non-dominated sorting genetic algorithm-II (NSGA-II) and later, they utilized a fast non-dominated sorting genetic algorithm in Deb et al. (2002) that prevents losing good solutions while being efficient in terms of convergence rate and diversity. In 2014, an extended version of NSGA-II algorithm (called NSGA-III) is proposed by Deb & Jain (2014), capable of solving problems with many objectives efficiently. Some examples of this method in the field of reliability application can be found in (Li et al., 2016; Mobin et al., 2017).

Multi-objective bi-level particle swarm optimization was used by Xu et al. (2012) to solve a minimum cost network flow problem using a nonlinear multi-objective bi-level model. The upper level minimizes both direct and transportation time costs and the lower level minimizes transportation costs. Pant et al. (2015) proposed a reliability optimization problem with two conflicting objectives of maximizing the reliability of system while also minimizing the associated costs. To solve the model, they developed a multi-objective evolutionary algorithm, namely a particle swarm optimization that considers crowding distance for selecting frontier solutions.

In order to better classify the most recent literature pertaining to network reliability optimization problems and demonstrate the gap filled by this research, Table 1 is provided. Previous studies are classified based on reliability assessment methods and types, modeling of network flow, cost considerations, network types, objective functions, and solution methodologies. Since arcs are more vulnerable to failure than the nodes, the majority of the reviewed literature, similar to this research, considered source-sink reliability and the failure to be occurred on the arcs. Popular network reliability assessment methods are: approximation methods, quickest-path reliability, and upper-lower bounds. In terms of flow types, both deterministic and stochastic flows have been studied in previous research, however, this research utilizes transmission in the deterministic flow

model that allows a fraction of arcs capacity to transmit the flow.

In terms of the cost, few studies simultaneously considered both cost of flow (variable costs) and cost of network construction (fixed costs), while both costs are important in strategic and tactical decisions of any network. The above literature review indicates that no studies have been carried out to design communication networks that consider cost, flow, and reliability simultaneously, and subsequently, an effective optimization algorithm, such as NSGA-III has not been customized for this purpose. The objectives and other features of the model has resulted a generic model which can be used for different classes of networks such as telecommunication networks, transportation networks, mechanical networks (Lin & Yeh, 2015; Guo et al., 2018; Jian et al., 2018), and electrical power networks (power grids).

3. Problem Description

Three important criteria in designing sustainable networks are reliability, cost, and the amount of flow to transport. The importance of designing reliable networks is more explicit for infrastructure networks such as computer networks, traffic, and energy networks (Namin et al., 2019). Similarly, disruptions to supply chain networks such as pharmaceutical chains may threaten lives of many patients in need (Azghandi et al., 2018). Any disruption or failure in the infrastructure networks can lead to unpredicted catastrophic events, both during the disruption and after the network recovery. In many of these networks, stability and sustainability of the network are consistently assessed and with the assist of preventive maintenance, the reliability of network is sustained in a high level.

Depending on the network reliability importance, in general two main criteria for reliable networks are considered, namely *all-terminal* and *source-sink*. All-terminal reliability focuses on the ability of communications among all nodes of the network via some unidentified paths. Source-sink reliability is focused on the ability of a predefined source node to communicate with a predefined sink node via some unidentified paths. This paper considers only source-sink reliability, since it is more generalizable to large scale networks. One application of such networks is computer networks, where a user (source node) is interested to connect to a server (sink node). While there is no need for all internal nodes to communicate with other nodes, while an undefined path between source and sink nodes should be available.

A significant area of research in network design problem is estimating the network reliability. In general, four widely-used methods for assessing network reliability are: (1) Exact assessment using analytic methods (single and parallel process) (Nguyen et al., 2016), (2) Estimation using approximation methods such as Monte Carlo simulation (Gertsbakh & Shpungin, 2016; Praks et al., 2017; Chen, 2017), (3) Finding upper or lower bounds (Jan et al., 1993; Srivaree-ratana et al., 2002), and (4) Finding a crude substitute(Konak & Smith, 2006; Dubourg et al., 2013). In this research, Jans upper bound method (Rong-Hong, 1993) is used to estimate the network reliability.

3.1. Mathematical Model Formulation

In this paper, reliable generalized flow network design problem is modeled as a multi-objective mixed-integer linear mathematical model. The proposed model determines

Table 1: A review of literature related to network reliability models

Features			Research Studies	Srivaree-ratana et al. (2002)	Konak & Smith (2011)	Lin & Chang (2013)	Yeh (2013)	Kuipers et al. (2014)	Elshqeirat et al. (2015)	Yeh (2015)	Zuev et al. (2015)	Pant et al. (2015)	El Khadiri & Yeh (2016)	Malinowski (2016)	Hochbaum (2016)	Jane & Laih (2017)	Sherman (2017)	Ruiwen et al. (2017)	Yeh & Chu (2018)	Cancela et al. (2019)	Proposed model
Reliability	All Terminals			✓	✓				✓												
	Source-Sink					✓	✓	✓		✓	✓		✓			✓		✓	✓	✓	✓
	Failure	On Node										✓		✓				✓			
		On Arc		✓	✓	✓	✓			✓	✓		✓			✓			✓	✓	✓
	Assessing Reliability	Series-parallel reducible										✓		✓							
		Approximation methods		✓	✓				✓		✓		✓							✓	
		Upper/Lower bounds		✓	✓			✓	✓									✓			✓
		Inclusion Exclusion					✓			✓			✓			✓					
		Recursive sum of disjoint				✓													✓		
		Quickest-path reliability					✓			✓			✓			✓			✓		
		Dynamic Programming							✓												
Flow	Capacity	Deterministic		✓		✓	✓	✓			✓	✓			✓	✓			✓		✓
		Stochastic				✓	✓	✓					✓					✓		✓	
		Multi-valued														✓				✓	
	Transmission Factor						✓														✓
Cost	Cost of Flow							✓							✓	✓	✓				✓
	Fixed Cost			✓	✓				✓												✓
Network	Dynamic									✓	✓			✓		✓		✓			
	Multi-state					✓	✓			✓	✓		✓			✓		✓	✓	✓	
Objectives	Reliability				✓	✓	✓					✓	✓					✓	✓	✓	✓
	Flow					✓		✓			✓		✓			✓	✓				✓
	Cost			✓	✓				✓			✓			✓		✓				✓
Solution	Heuristic							✓		✓											
	Metaheuristic			✓	✓				✓			✓							✓		✓
	Hybrid																		✓		
	Simulation			✓	✓			✓	✓		✓		✓					✓		✓	
	Exact					✓	✓			✓			✓	✓	✓	✓	✓				

an optimal network design considering gains and losses of the flow through the given network with respect to the conflicting objectives i.e., minimizing the total system cost, and maximizing flow and reliability.

The following assumptions are made in the proposed mathematical model: (1) The probability of each arc's failure is independent of other arcs failures. (2) The costs and probability of failure of arcs are known and deterministic. (3) Arcs are active and operational if they are not failed, and the repairing option is not considered in the model. (4) In an optimal network, the flow will only pass through optimal arcs, therefore, there might be arcs with no flow. (5) The loss and transmission factors of different arcs are statistically independent random variables and are real positive numbers.

Given a directed graph $G(N, A)$ where $N = \{1, \ldots, n\}$ denotes the set of nodes including source (s) and sink node (t), and A denotes the set of possible arcs. Each arc $(i, j) \in A$ has a probability of failure $0 \leq q_{ij} \leq 1$. We assumed that the probability of failure for all arcs are similar during each scenario, hence q is used instead of q_{ij} for easier interpretation. Let $\mathrm{CB_{ij}} > 0$ denotes the cost of building arc (i, j). Examples of such cost are material costs, connection and terminal costs, installation costs such as peeling and embankment and leveling, land or right of way costs inherent with the cabling. Since most of these items are unit costs, the total cost of an arc can be estimated by the length of the arc. However, for the sake of simplicity, $\mathrm{CB_{ij}}$ is assumed fixed per arc in this model. $\mathrm{CF_{ij}} > 0$ indicates the per-unit cost of flow through the arc (i, j). The allowable amount of flow along (i, j) must be at least equal to the flow lower bound (l_{ij}) and at most equal to upper bound (u_{ij}). Here, the lower bounds on all arc flows are zero. Additionally, let a'_{ij} denotes loss factor on $(i, j) \in A$. Then, the arc multiplier, represented by $a_{ij} = 1 - a'_{ij}$ is the fraction of arcs capacity that can transmit the flow, which we call *transmission factor*.

Note that if a_{ij} is equal to 1, a pure or conventional network formulation exists; if $a_{ij} > 1$, the flow is augmented (gain); and if $a_{ij} < 1$, the flow is decreased (loss). In addition, in this problem, F_{in} and F_{out} denote the input flow and the output flow, respectively. Note that flow conservation may not achieve in this model, since F_{in} is not necessarily equal to F_{out}. This is due to flow adjustments caused by the transmission factor. Finally, x_{ij} and f_{ij} are binary and continuous decision variables, respectively. If arc (i, j) is included in the optimal design, x_{ij} is equal to 1, and 0 otherwise. The amount of flow along arc (i, j) is indicated by $f_{ij} \in \mathbb{R}^+$. X is the arc topology of $x_{12}, \cdots, x_{ij}, \cdots, x_{N-1,N}$ and $R(x)$ is the reliability of X. The multi-objective mixed-integer linear programming model can be formulated as follows:

$$\max \quad R(X) \tag{1a}$$

$$\max \quad F_{out} = \sum_i \sum_j f_{ij} \tag{1b}$$

$$\min \quad \sum_i \sum_j \mathrm{CB}_{ij} x_{ij} + \sum_i \sum_j \mathrm{CF}_{ij} f_{ij} \tag{1c}$$

subject to

$$\sum_j f_{sj} - \sum_j a_{js} f_{js} \leq F_{in} \tag{1d}$$

$$\sum_j f_{ij} - \sum_j a_{ji} f_{ji} = 0, \quad \forall i, i \neq \{s, t\} \tag{1e}$$

$$\sum_j f_{tj} - \sum_j a_{jt} f_{jt} = -F_{out} \tag{1f}$$

$$L_{ij} \leq f_{ij} \leq U_{ij} x_{ij}, \qquad \forall (i,j) \tag{1g}$$

$$\sum_j x_{ij} + \sum_j x_{ji} = d_i \qquad \forall i = \{1, \cdots, n\},\ i \neq j \tag{1h}$$

$$R(X) \leq \Big[H(d) \cong 1 - \Big[\sum_i q^{d_i} \times \prod_{k=1}^{m_i} \left(1 - q^{d_k - 1}\right) \times \prod_{k=m_{i+1}}^{i-1} \left(1 - q^{d_k}\right) \Big] \Big] \tag{1i}$$

$$f_{ij} \in \mathbb{R}^+$$

$$x_{ij} \in \{0, 1\}$$

$$d_i, m_i \in \mathbb{Z}^+$$

To design a network by choosing a subset of the potential arcs, the network reliability, denoted in (1a), and the output flow (the real amount of flow from s to t), denoted in eq. (1b), need to be maximized. At the same time, the total cost including total building costs and total costs of flow, denoted in (1a), should be minimized. However, these objectives are in conflict with each other; cost and reliability are the two conflicting objectives and the trade-off between them is not linear. Here, we assume that reliability comes at the price of higher cost with a range from 0 to 1 representing the spectrum between a non-operational state and a fully reliable state. Furthermore, assuming cost for a flow, maximum flow is more costly to transport through the network. In this research, the Pareto optimal solutions are obtained from the set of optimal trade-offs between the conflicting objectives.Pareto optimal solutions are the set of solutions in which none of the objectives can be improved without degrading at least one of the other objectives.

To show possible imbalances between the amount of flows entering and exiting an arc, the flows of a generalized network are multiplied by gain/loss factors in the constraint set (1d, 1e and 1f). The flow on each arc cannot exceed the specified lower and upper bounds implied by the scenario as defined in constraint (1g). It also ensures that for non-selected arcs, the flow remains zero. Eq. (1h) computes the degree of each node i, denoted as d_i,

which is the number of incoming and outgoing arcs connected to each node. Utilizing degree of nodes, constraint (1i) shows the Jans upper bound that is used to estimate the network reliability where q is the unreliability of links, and $m_i = min(d_i, i-1)$. Finding network reliability upper bound can be transformed to find a network with degree sequence $(d_1, d_2, \ldots, d_n)$ such that $H(d)$ is maximal and $\sum_{i=1}^{n} di = 21$ holds where l denotes total number of arcs in the network. In other words, each network can also be associated with a unique sequence of degree called its degree sequence. For example, Fig. 1 shows the degree sequence for a sample graph. To use the Jans upper bound method, two conditions are required for the network: (1) The graph must be connected i.e., the network is not divided into several parts. (2) Each node must be associated with at least two other nodes (two-connectivity condition) so that in case of failure of an edge, the node is not isolated and can maintain its connection to the network. These two conditions are considered at the reliability objective function as a penalty function. The pseudo-code related to this requirement are included in Appendix A.

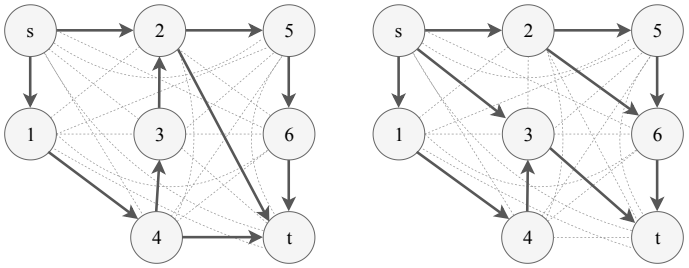


Figure 1: Sample graph with 8 nodes representing the studied problem. Note that s represents the source node and t denotes the sink node. The solid lines represents selected arcs ($x_{ij} = 1$). Degree sequence for each network is different. The degree sequence is $\{2, 2, 4, 2, 3, 2, 2, 3\}$ and $\{3, 2, 3, 3, 2, 2, 3, 2\}$ for the left and right graph, respectively.

4. Solution Procedure

Computational complexity has been a barrier to many large scale problems. Many real-world optimization problems belong to the class of NP-hard and when the size of problem increases, there is no exact analytical method to solve them in efficient time. Therefore, metaheuristic algorithms are applied as efficient tools to optimize this class of problems. The network design reliability problem is an NP-hard combinatorial optimization problem (Garey, 1979; Provan & Ball, 1983). Therefore, the metaheuristic algorithms are good candidates to tackle such problems. While there are several metaheuristic methods that can be applied to multi-objective problems, we have utilized and compared three algorithms namely *MOPSO, NSGA-II, NSGA-III*. The differences between these methods and their strength in dealing with multi-objective problems led to selection of these methods. While both NSGA and MOPSO methods are population-based metaheuristics, the difference is on selecting and ranking Pareto front solutions procedures. More specifically, NSGA-III utilizes reference points for ranking the solutions, while MOPSO use probabilistic mechanisms such as roulette wheel selection. Thus, the two metaheuristic procedures are distinct and can provide further insights in computational results. Meanwhile, based on the existing literature, MOPSO has been shown to

be an efficient and fast method for solving large-scale mixed-integer programming problems(Bagherinejad & Dehghani, 2016; Pasandideh et al., 2013). PSO embodies swarm intelligence that leverages the accuracy and speed of converging to best solutions. In swarm intelligence, not only the experience of each particle involves in finding next solution, but also social behavior of all other particles simulates in this procedure. Also, PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

4.1. NSGA Description

Genetic Algorithm (GA) is a population-based search algorithm that has been widely used to solve large-scale optimization problems in different domains. However, it can only be utilized for single-objective optimization problems. An extension to GA algorithm for multi-objective optimization problems is called NSGA-II, or The Non-dominated Sorting Genetic Algorithm-II. It has been first proposed by Deb et al. (2000, 2002), where set of Pareto front solutions are efficiently selected, constrained by multiple objective functions. For this purpose, NSGA-II, similar to GA, employs set of bio-inspired operators with surrogates including selection, crossover, and mutation. In this paper, each member of the population in NSGA-II is a structure array that carry *the members' Position, solution, Cost, Ranking, Domination Set, Dominated Count, and Crowding Distance.* The *Position* is a zero/one $N \times N$ adjacency matrix, which demonstrates the connection between each pair of nodes, where N is the number of nodes. The *Cost* includes three objective function values, i.e., reliability(1a), total cost(1b), and maximum flow(1c). The *Solution* contains the flow matrix, the violations, and whether the solution is feasible or not. The *Ranking* indicates the rating of the solution. The *Domination Set* includes the set of population members, dominated by the solution. The *Dominated Count* retains the frequency of times that the solution has dominated by other solution during the search process. As a result, for the Pareto optimal solution, the *Dominated Count* should be zero. The last component presents the *crowding distance* that measures population density around a solution by the average distance of the solution with two neighboring solutions (Konak et al., 2006). In NSGA-II, the *crowding distance* is used in a selection technique called the crowded tournament, where selection operator use dominance and crowding distance as primary and secondary selection criteria.

Simultaneous encounter with many objectives (more than two) is one of the prominent features of the NSGA-III. This algorithm was proposed by Deb & Jain (2014) by changing selection operators of NSGA-II, in which one or a set of pre-specified points are considered as reference points. The steps in the algorithm are as follows (Bhesdadiya et al., 2016):

4.1.1. Designation of Reference Points on a Hyper-plane for NSGA-III

An approach to ensure the diversity (Δ) of the generated solutions is to determine a set of reference points. In NSGA-III, several points are placed on a normalized hyper-plane where they have identical direction in all of axes. The number of reference points (H) is calculated by:

$$H = \binom{\mathrm{n_O} + \mathrm{n_D} - 1}{\mathrm{n_D}}, \tag{2}$$

where $\mathrm{n_O}$ is the number of objective functions, and $\mathrm{n_D}$ is the number of divisions to consider on every objective axis (e.g., for a problem with three-objective functions and

five divisions, 21 reference points are required). The Pareto members will be associated with the generated reference points on the hyper-plane.

4.1.2. Genetic Operators

Using cross-over and mutation, new solutions (i.e., children) are generated in each iteration of NSGA-III. For making equal significance among all of the population members, population size (n_P) is considered close to the number of reference points (H). Thus, at a generation t, all population members (P_t) convert in non-dominated solutions using the same sorting method of NSGA-II. After that, each P_t generates new children population (Q_t) by mutations and recombination operators in which every population member associated with each reference point and any selection operator will authorize a competition to be set among various reference points. Then, a combined population ($R_t = P_t \cup Q_t$) is organized and the non-dominated sorting mechanism is performed to select new generation.

4.1.3. Normalization of the Population Members

For determining an ideal point of the current population, the minimum of each objective function ($F_i^{min}, i = \{1, 2, \cdots, n_O\}$) must be identified. Then, each objective function will be moved by subtracting $z_i^{min} = (F_1^{min}, F_2^{min}, \cdots, F_{n_O}^{min})$ to the objective f_i. Then, a hyper-plane can be created by the steps presented in Xu et al. (2012). Since the objective functions have different scales in this problem, they are required to be normalized.

4.1.4. Association Between Reference Points and Population Members

At this stage, there must be an association between the members of the population and the reference points. Here, a reference line is drawn by connecting the reference point and the origin point. Next, the perpendicular distance between each solution and each reference line is computed. Finally, the reference point which has the shortest distance to the reference line from a population member is associated with this solution.

4.1.5. Niche-preserving Process

Using the perpendicular distance from the reference line, the solution that has least distance from reference point, must be kept. Fig. 2 shows a graphical representation of NSGA-III steps. To utilize NSGA-III in this problem, each member of the population is an array structure with these associated information: *Position, Cost, Solution, Rank, Domination Set, Dominated Count, Normalized Cost, Associated Ref, Distance To Associated Ref.* Definition of *Position, Cost, Solution, Domination Set, and Domination count* is similar to description provided for NSGA-II. A *Rank* is defined for each front level based on its non-domination level. The solution that is not dominated by any of the chromosomes, is assigned to level 1. Level 2, is the secondary level in which the chromosome is dominated by some chromosomes only in level 1, and so on. Note that while both NSGA-II and NSGA-III utilize ranking, the function and underlying outcome is different. The *Normalized Cost* includes normalized objective function values (see (Deb & Jain, 2014)). After normalizing each objective, each population member is associated with a reference point. For this purpose, a reference line is defined by connecting each reference point on the hyper-plane with the origin. Then, the perpendicular distance (*Distance To Associated Ref*) of each population member from each of the reference lines is calculated.

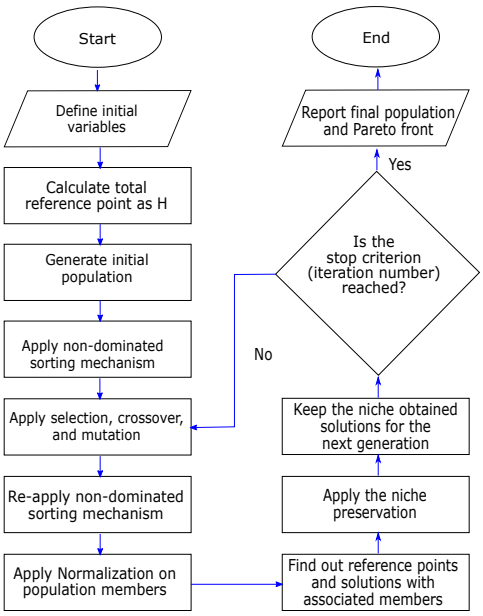


Figure 2: Graphical representation of NSGA-III.

The reference point with shortest distance of its reference line to a solution is associated with the solution (*Associated Ref*).

4.2. Multi-objective Particle Swarm Optimization (MOPSO)

Similar to GA, Particle Swarm Optimization (PSO) algorithm is another population-based single-objective metaheuristic algorithm that had been widely used for problems with continuous-space solution domain (Kennedy & Eberhart, 1995). In PSO, a population of candidate solutions is called "*particle*" where each particle flies over the solution space by carrying five individual properties: (1) current position, (2) current velocity, (3) objective function value corresponding to the position, (4) the best position that has been reached by the particle (best local solution), and (5) the objective function value corresponding to the best position of the particle. The velocity and the position for each particle will be updated in each iteration, influenced by the best local solution for each particle and the best global solution among all particles.

An extension to PSO to comply with multi-objectives problems is called MOPSO (Multi-objective PSO). MOSPO has a repository (Coello et al., 2004) as an external archive of solutions including the non-dominated solutions. The repository members provide an approximation of real Pareto front of the optimization problem. The pseudocode of MOPSO is illustrated in Algorithm 1.

4.2.1. MOPSO Refinement

Most particle swarm optimization algorithms are designed for continuous search space domain. In this research, since the assignment variables $x_{ij} \in \{0, 1\}$ are discrete, particle velocity and position limits converts the set of solutions from the continuous domain to

Algorithm 1 Pseudocode of MOPSO Steps

```
Initialize the swarm containing n_P particles
Evaluate particles
Specify non-dominated solutions and store them in repository
while stop conditions are not satisfied do
    for every particle in the swarm do
        Select the leader for the selected particles
        Update the velocity according to the velocity update rule
        Update the position according to the position update rule
        Evaluate the objective functions for the particle
        Apply mutation
        Update best position based on new position:
        if new position dominates previous best then
            replace the best
        else if new position is dominated by previous best then
            keep the best
        else
            randomly choose one of them as the best
        end if
    end for
    Add the non-dominated particles to repository
    Remove dominated members of repository
end while
```

discrete values. This conversion applies in each iteration, before these values are used for updating the location of next particle's solution.

Each member of the population is modeled as a structure array with fields: *Position, Solution, Velocity, Cost, Best solution, Dominated(T/F), Grid Index, and Grid Sub Index.* The *Position*, *Cost*, and *Solution* definitions are similar to description provided for NSGA-II. The *Best solution* keeps the best position and its objective value explored so far by the particle. If a particle is dominated by other particles, *Dominated* flag will be true, otherwise it remains false. In the MOPSO algorithm, the objective space is tabulated, so that the two properties, *Grid Index* and *Grid Sub Index*, are used to locate each particle. If a single number is in the table houses, it is shown with a *Grid Index* and if two numbers are in table houses, the row and column number, *Grid Sub Index* is used. For example, in a two-dimensional objective space , GridIndex can be reached through the number of rows and columns (GridSubIndex) with the $(m(i-1)+j)$ mapping equation in which, m is the number of rows, i and j are counters of rows and columns.

4.3. Parameter Tuning

The efficiency and computational convergence speed of metaheuristic methods are highly dependent on the accurate tuning of the components and parameters within these method (Tatsis & Parsopoulos, 2019). However, it has been shown that there are no set of generic parameters that can be utilized for any of metaheuristic methods (Talbi, 2009).

The majority of methods for tuning of the parameters within metaheuristic algorithms can be classified into the *online* and *offline* methods (Fallahi et al., 2014). In the online

approach the parameters are dynamically or adaptively updated during the execution of the metaheuristic (Akbaripour & Masehian, 2013). This method is efficient when the number of tuning parameters are very limited. In the offline parameter tuning, for each parameter a set of values are indicated from the literature or empirical experiments. Correspondingly, the metaheuristic is executed with the fixed-values for the parameters and the outcomes are compared to find the best set of parameters. In most of the offline methods, the metaheuristic designer tunes each parameter individually, also called sequential optimization strategy. Correspondingly, the potential interactions between the parameters are overlooked and finding the optimal or near optimal settings cannot be reached.

Design of Experiment (DOE) is one of the well studied statistical approaches to evaluate the interactions between parameters (Ozkan et al., 2019). Once the number of tuning parameters and their associated value levels becomes large, DOE methods are no longer able to provide best tuned parameters' value in timely manner. As a result, in this study, Taguchi method, also called orthogonal array (Roy, 2001), an alternative to full-experimental design in DOE, are utilized for parameter tuning in offline mode. Taguchi design requires much fewer number of experiments compared to DOE, and find best solutions without loss of generality.

4.3.1. Taguchi Design for Parameter Tuning

For each of the implemented metaheuristic algorithm, several parameters are required to be fully calibrated prior to final execution. For this purpose, three initial values for each unknown parameters are extracted from the literature.Using Minitab software, the Taguchi method is utilized to build an associated design, corresponding to number of parameters that need to simultaneously tuned and number of levels for each parameter. Utilizing set of proposed parameters levels in each design, results of metaheuristic methods, are calculated and then combined using Relative Percent Deviation(RPD), as a measure of precision. As a result, best set of parameters are derived and utilized for solving our problem. More specifically, for each NSGA-II and NSGA-III, a $L9(3^4)$ Taguchi design is used, correspond to four parameters each with three levels. In these algorithms both genetic-related parameters and termination condition parameters are investigated. The four parameters that are tuned are *probability of mutation* (P_m), *probability of crossover* (P_c), *maximum number of iterations* (M_{it}), and *population size* (n_P). Table 2 shows the initial and the finalized value for calibrated parameters in NSGA-II and NSGA-III.

Regarding other parameters of NSGA-III, for a three-objectives problem, developers of MOEA/D suggest the number of divisions ($\mathrm{n_D}$) over each objective to be $\mathrm{n_D} = 18$ (Bhesdadiya et al., 2016), and as a result, the number of reference points (H) can be calculated using Eq. (2) with $H = 190$.

For MOPSO metaheuristic algorithm, 7 parameters are calibrated, Similar to NSGA parameter tuning, three levels of value for each parameter has been obtained from the literature. As a result, a $L27(3^7)$ Taguchi design is utilized to tune the parameters. The parameters of interest are: (1) size of the particle population (n_P); (2) maximum number of search iterations (M_{it}) before algorithm termination; (3) number of repository (n_r); (4) mutation rate for each particle (μ_{rate}); (5) inertia weight (w); (6) local learning coefficient (c_l), and (7) global learning coefficients (c_g) for updating velocity of the particles.

Table 2: Parameter tuning for NSGA-II, NSGA-III, and MOPSO. For each parameter, three levels are considered and the interaction among the parameters are evaluated using Taguchi method. The final parameter values are indicated in bold

Parameters	NSGA-II			NSGA-III			MOPSO		
	low level	base level	high level	low level	base level	high level	low level	base level	High level
population size (n_P)	150	**250**	350	150	**200**	250	150	200	**250**
Maximum iterations (M_{it})	**200**	500	1000	200	**500**	1000	200	500	**1000**
Probability of crossover (P_c)	**0.7**	0.8	0.9	0.7	**0.8**	0.9	–	–	–
Probability of mutation (P_m)	0.1	0.2	**0.3**	0.1	**0.2**	0.3	–	–	–
Number of repository (n_r)	–	–	–	–	–	–	100	**150**	200
Mutation rate (μ_{rate})	–	–	–	–	–	–	0.2	0.3	**0.5**
Local learning coefficient (c_l)	–	–	–	–	–	–	0.5	1.5	**2**
Global learning coefficient (c_g)	–	–	–	–	–	–	**0.5**	1.5	2
Inertia weight (w)	–	–	–	–	–	–	0.4	0.7	**0.9**

Note that, only the initial value of w is calibrated while it will be decreased at the end of each iteration by:

$$w = w \times w_{damp}, \tag{3}$$

where w_{damp} is an inertia weight damping rate, (e.g., $w_{damp} = 0.99$).

4.4. Best Compromise Solution

The main feature of Pareto optimal solutions is that these solutions are incapable of improving one of the objectives without sacrificing other objectives. However, in practice, we are interested to select one solution from Pareto optimal set of solutions, also called coordinated solution. Finding coordinated solution becomes harder when the objective functions are measured by different unit scales. (Niimura & Nakashima, 2003). In order to develop a solution method that presents a good compromise between the objectives, we apply a fuzzy mechanism. As shown in the following, a simple linear membership function is considered for each of the objective functions:

$$\mu_i(F_i) = \begin{cases} 0 & F_i \leq F_i^{min} \\ \frac{F_i^{max} - F_i}{F_i^{max} - F_i^{min}} & F_i^{min} < F_i < F_i^{max} \\ 1 & F_i \geq F_i^{max} \end{cases} \tag{4}$$

F_i^{min} and F_i^{max} indicate the least and most satisfactory solutions for the i^{th} objective function among Pareto front, respectively. The range for the membership function μ is between 0 and 1, where $\mu = 0$ denotes the incompatibility of the solution and $\mu = 1$ donates full compatibility. The reliability and flow maximization objective functions are equivalently analysed as minimizing the negative value for reliability and flow. As a result, all objectives correspond to the smaller, the better by the degrees measured in Eq. (4). The normalized membership function μ_k for each Pareto member k is computed

based on:

$$\mu_k = \frac{\sum_{i=1}^{\mathrm{n_O}} \mu_i^k}{\sum_{k=1}^{M} \sum_{i=1}^{\mathrm{n_O}} \mu_i^k}, \tag{5}$$

where M is the number of Pareto members and $\mathrm{n_O}$ is the number of objective functions. The function μ_k can be realized as the membership function for non-dominated solutions in a fuzzy set. In that case, the best compromise solution is the solution with the maximum membership, following the fuzzy decision process provided by Zimmermann (2011).

$$\text{Maximize} \min_{\mathrm{i=1,\ldots,k}} \{\omega_\mathrm{i} \mu_\mathrm{i}(\mathrm{f_i(x)})\} \tag{6}$$

subject to same constraints as of the original problem, where k denotes the number of objective functions and ω_i is the weight factor ($0 \leq \omega_i \leq 1$) corresponding to each objective function.

4.5. Comparison Criteria for Pareto Optimal Solutions

Among several commonly used performance metrics for multi-objective problems (Tavana et al., 2016), five highly-reported metrics are used to assess the convergence rate and the diversity of Pareto optimal solutions for each of the algorithms: (1) average number of Pareto members (NOM), (2) Diversity (Δ) (Zitzler, 1999), (3) mean ideal distance (MID) (Karimi et al., 2010), (4) hyper-volume indicator ($\mathrm{HV_I}$) and (5) CPU time for achieving the final solutions.

Average number of Pareto members (NOM) shows capability of an algorithm to provide diverse compromised solutions. Note that, the number of Pareto members cannot exceed the population size (n_p) of an algorithm.

The Diversity (Δ) metric is formulated as follows:

$$\Delta = \sqrt{\sum_{j=1}^{n_O} (\max_n f_n^j - \min_n f_n^j)^2} \tag{7}$$

where in Eq. (7), n_O represents the number of objective functions and n represents the number of Pareto solutions. Δ demonstrate the difference between the best and the worst solution for each objective function among all Pareto optimal solutions.

Another metric to compare the diversity of solution is called MID and is formulated as follows:

$$MID = \sum_{i=1}^{n} \frac{l_i}{n}, \tag{8}$$

Similarly, in Eq. (8), n is representing the number of Pareto solutions and l_i forms the Euclidean distance of i^{th} Pareto solution from the ideal point for each objective function. The ideal point for maximizing reliability, the first objective as shown in eq. 1a is $\{1\}$, describing a fully-reliable network. The ideal point for minimizing cost, as shown in eq.

1c, can be considered $\{0\}$, albeit unrealistic. However, for maximizing the flow, eq. 1b, an ideal solution is not intuitively available and $(\max(z_3))$ is used for the calculations.

Hypervolume (HV), also known as S metric, is a measure to quantify the amount of objective-space which is being dominated by the obtained Pareto front. HV has been a favored performance metric in multi-objective literature since it captures both the relative distance of the solutions to the optimal set and the spread of the solutions across objective space using a single scalar (While et al., 2006).

Considering a reference point (A) as the anti-ideal or the worst solution within the space of the objective function, the volume that is delimited by anti-ideal point and an Pareto optimal points forms the hypervolume. One limitation of HV metric is objective function values must be normalized prior to determining the hyper-volume (Ciro et al., 2016).

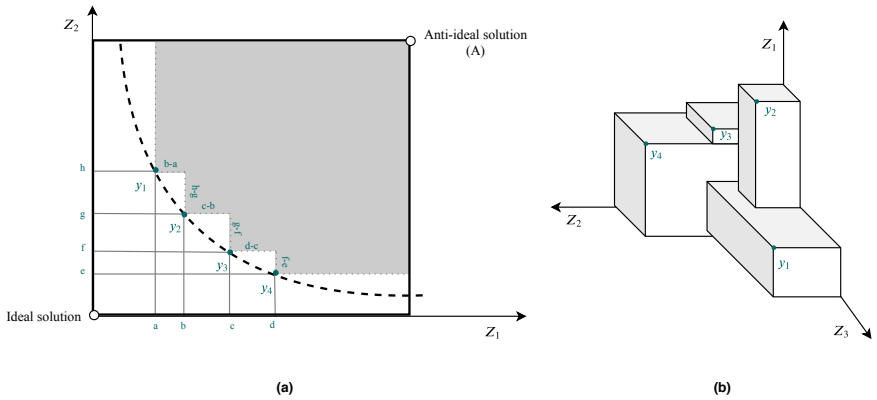


Figure 3: a) HV surface for a problem with two objective functions, b) HV volume for a problem with three-objective functions and multiple Pareto front solutions.

In Fig. 3.a, the calculation for hyper-volume of a problem with two minimization objective functions is a surface provided by aggregating the rectangles created by the reference point (A) and the Pareto front solution (in this example, y_1, y_2, y_3, and y_4). In a three-objective problem, the hypervolume realization create a volume as shown in Fig. 3.b

In order to calculate hypervolume indicator, denoted as $\mathrm{HV_I}$, for a Pareto optimal solution set X, bounded with a reference point $A = (a_1, \ldots, a_k) \in \mathbb{R}^k$, the following formulation can be utilized:

$$\mathrm{HV_I(X)} = \lambda\left(\bigcup_{\mathrm{a \in A}} [\mathrm{f_1(x), a_1}] \times [\mathrm{f_2(x), a_2}] \times \cdots \times [\mathrm{f_k(x), a_k}] \right), \tag{9}$$

where, $\lambda(S)$ is the Lebesgue measure of a set S and $[f_1(x), a_1] \times [f_2(x), a_2] \times \cdots \times [f_k(x), a_k]$ is the k-dimensional hypercuboid that consists all non-dominated points compared to the reference point. In recent years, both exact and approximate methods are used to calculate $\mathrm{HV_I}$(Zitzler et al., 2007; Jaszkiewicz, 2018; Oliveira et al., 2019). In this research, the hypervolume is approximated using Monte-Carlo (MC) method, as proposed by Kruisselbrink (2015).

5. Results and Performance Comparison of Solution Methods

As previously elaborated with details, the multi-objective optimization problem (introduced in eq. 1) has been solved by three metaheuristic algorithms, namely NSGA-II, NSGA-III, and MOPSO and the results are compared. All algorithms are implemented in MATLAB R2010a and solved on a Core 2 Duo/2.66 GHz processor and 2GB of RAM memory. Twelve numerical instances for the small-scale and twelve numerical instances for the large-scale are randomly generated to compare the performance of these algorithms. The scale of the instances is defined as a function of decision variables and the constraints. With an increase in the size of the network, number of nodes (n), and number of associated arcs $\left(2 \times \binom{n}{2}\right)$ also increase. Previous studies suggest that a problem can be referred to large-scale, once the size of the problem exceeds 1000 (Benson et al., 2003). Hence, the increment in number of decision variables and constraints render the problem intractable and provide good candidates to compare the solution methods for more realistic instances.

Table 3: Comparison between results of different metaheuristic algorithms in small-scale and large-scale problems (P = Problem, N = Nodes, R = Reliability, u_{ij} = Upperbound for flows, NOM = Number of Pareto members, MID = Mean Ideal Distance, Δ = Diversity, HV_I =Hyper-Volume index)

Scenario properties				**MOPSO with 1000 iterations and $n_P = 250$**					**NSGA-II with 200 iterations and $n_P = 250$**					**NSGA-III with 500 iterations and $n_P = 200$**				
P	N	R	u_{ij}	NOM (avg)	MID	Δ	HV_I	Time (min)	NOM (avg)	MID	Δ	HV_I	Time (min)	NOM (avg)	MID	Δ	HV_I	Time (min)
S1	**6**	**0.75**	**[50,600]**	203	1.26E+06	1.36E+06	1.13E+08	31.6	249	1.38E+06	2.88E+06	1.89E+08	11.1	200	7.33E+05	3.64E+07	2.81E+08	10.9
S2	**6**	**0.75**	**[2k,6k]**	195	3.65E+07	5.62E+07	3.79E+08	31.9	217	5.98E+07	7.94E+06	4.23E+08	12.3	197	1.38E+07	9.06E+06	6.76E+08	9.8
S3	**6**	**0.85**	**[50,600]**	165	1.34E+06	3.11E+06	9.16E+07	31.1	250	1.43E+06	3.00E+06	8.73E+07	11.8	200	9.59E+05	4.88E+06	9.84E+07	9.7
S4	**6**	**0.85**	**[2k,6k]**	172	3.21E+07	6.53E+07	9.88E+07	32.0	198	5.25E+07	7.94E+06	1.85E+08	11.9	196	1.85E+07	9.48E+06	2.53E+08	11.0
S5	**6**	**0.95**	**[50,600]**	177	1.95E+06	2.44E+06	9.60E+06	31.4	246	2.89E+06	5.02E+06	1.10E+07	11.0	198	1.06E+06	2.85E+07	9.89E+06	10.8
S6	**6**	**0.95**	**[2k,6k]**	185	3.68E+07	5.04E+06	2.07E+07	32.5	198	7.23E+07	7.94E+06	1.37E+07	11.8	197	2.25E+07	8.90E+06	4.51E+07	11.2
S7	**10**	**0.75**	**[50,600]**	176	4.18E+06	7.29E+06	7.55E+08	39.2	246	3.93E+06	8.47E+06	1.27E+09	12.9	176	1.44E+06	7.56E+06	8.84E+08	16.9
S8	**10**	**0.75**	**[2k,6k]**	210	4.50E+07	5.08E+07	2.44E+09	39.9	192	5.48E+07	7.22E+07	5.85E+09	14.1	196	2.87E+07	9.12E+07	7.30E+09	17.6
S9	**10**	**0.85**	**[50,600]**	209	3.71E+06	5.90E+06	2.23E+08	39.0	138	4.63E+06	7.33E+06	3.14E+08	12.2	190	1.97E+06	8.71E+06	5.24E+08	16.8
S10	**10**	**0.85**	**[2k,6k]**	235	5.93E+07	3.08E+07	8.10E+08	30.1	236	7.44E+07	3.96E+07	9.16E+08	13.7	197	3.78E+07	6.77E+07	1.06E+09	17.7
S11	**10**	**0.95**	**[50,600]**	241	5.01E+06	5.34E+06	5.19E+07	38.9	210	6.30E+06	4.94E+06	3.54E+07	12.6	194	2.37E+06	6.88E+06	6.68E+07	17.0
S12	**10**	**0.95**	**[2k,6k]**	226	6.23E+07	7.75E+07	4.86E+08	39.8	203	8.19E+07	8.25E+07	3.00E+08	13.9	199	4.13E+07	1.15E+08	7.11E+08	17.1
Average				**199.5**	**2.41E+07**	**2.59E+07**	**4.57E+07**	**34.78**	**195.58**	**3.47E+07**	**2.08E+07**	**8.00E+08**	**12.44**	**195**	**1.43E+07**	**3.29E+07**	**1.5E+09**	**13.83**
L1	**30**	**0.75**	**[100,800]**	223	6.75E+06	8.91E+06	2.61E+07	120.8	234	3.73E+06	5.86E+07	6.93E+07	32.4	196	1.08E+06	4.73E+07	3.31E+08	55.3
L2	**30**	**0.75**	**[3k,7k]**	217	7.67E+07	3.24E+06	8.90E+07	122.2	223	6.66E+07	3.13E+07	3.21E+08	34.1	193	2.16E+07	5.18E+06	4.15E+08	57.2
L3	**30**	**0.85**	**[100,800]**	197	7.08E+06	1.02E+06	1.76E+07	120.5	217	4.47E+06	6.44E+05	2.77E+07	31.9	197	1.14E+06	8.31E+05	3.60E+07	56.1
L4	**30**	**0.85**	**[3k,7k]**	242	7.83E+07	2.35E+07	2.25E+08	122.0	250	6.93E+07	7.51E+06	8.58E+07	34.3	200	2.23E+07	9.09E+06	5.35E+08	58.4
L5	**30**	**0.95**	**[100,800]**	203	6.54E+06	6.59E+07	6.64E+08	119.9	222	4.32E+06	8.43E+08	9.72E+08	30.8	200	1.18E+06	7.34E+08	2.19E+09	55.8
L6	**30**	**0.95**	**[3k,7k]**	227	6.63E+07	4.70E+06	1.11E+08	121.8	236	5.88E+07	7.29E+06	6.34E+07	34.5	182	2.72E+07	6.93E+06	5.08E+08	57.9
L7	**40**	**0.75**	**[100,800]**	187	7.68E+06	9.87E+05	8.87E+07	133.7	196	8.40E+06	8.04E+06	2.06E+08	49.0	200	2.22E+06	1.65E+07	4.91E+08	68.3
L8	**40**	**0.75**	**[3k,7k]**	175	7.89E+07	2.46E+07	3.22E+08	135.4	188	5.12E+07	2.76E+07	5.85E+08	50.6	194	4.48E+07	4.80E+07	3.62E+09	72.3
L9	**40**	**0.85**	**[100,800]**	218	7.94E+06	2.89E+06	3.74E+07	134.3	223	6.05E+06	8.68E+05	4.12E+07	47.8	200	2.27E+06	9.77E+05	7.24E+07	69.7
L10	**40**	**0.85**	**[3k,7k]**	176	8.27E+07	6.77E+06	5.71E+08	137.7	197	8.14E+07	7.66E+06	8.39E+08	50.3	189	5.33E+07	1.18E+07	3.41E+09	72.0
L11	**40**	**0.95**	**[100,800]**	215	7.71E+06	8.69E+07	9.79E+08	133.4	217	4.79E+06	5.37E+08	2.76E+09	49.2	190	2.51E+06	2.08E+08	4.68E+09	71.3
L12	**40**	**0.95**	**[3k,7k]**	199	10.9E+07	9.57E+08	6.19E+08	135.0	195	8.5E+07	8.40E+07	7.95E+08	49.9	191	6.09E+07	3.21E+09	1.88E+09	71.7
Average				**206.5**	**4.46E+07**	**9.89E+07**	**3.12E+08**	**128.05**	**236.32**	**3.7E+07**	**1.34E+08**	**5.64E+08**	**41.23**	**194.2**	**2.0E+07**	**3.58E+08**	**1.51E+09**	**63.83**
overall Average				**203**	**3.4E+07**	**6.2E+07**	**3.8E+08**	**81.4**	**215.8**	**3.6E+07**	**7.8E+07**	**6.8E+08**	**26.84**	**194.6**	**1.7E+07**	**2.0E+08**	**1.3E+09**	**38.85**

The numerical instances are introduced in Table 3, where the number of nodes (N), the arc reliability (R), and upper-bound of flow (u_{ij}) vary among instances. Number of nodes for small-scale and large-scale problems are defined as (6 and 10) and (30 and 40), respectively. Arc reliability ($R \in \{0.75, 0.85, 0.95\}$) is defined as an inverse of probability of arcs failure ($1-q$). Correspondingly transmission factor (a_{ij}) for each arc is randomly

generated in form of $N \times N$ matrices in an acceptable interval. For all $\binom{n}{2}$ arcs of a complete graph, the upper-bound flow of arc ij, (u_{ij}), are randomly generated from a Uniform distribution. In small-scale problems, the random variable is chosen such that $u_{ij} \sim U(50, 600)$ and $u_{ij} \sim U(2000, 6000)$. Similarly, the upper bound for large-scale problems are randomly generated by the uniform distributions with the range such that $u_{ij} \sim U(100, 800)$ and $u_{ij} \sim U(3000, 7000)$. Since cost of flow ($\mathrm{CF}_{ij}$) depends directly on the amount of flow (f_{ij}), the per-unit cost of flow through the arcs and cost of building arcs (CB_{ij}) are defined as a random fraction of u_{ij} in a predefined interval.

Each instance is solved by all metaheuristic methods with the calibrated parameters values as previously shown in Section 4.3.1. Correspondingly, all performance metrics, reported in Section 4.5, are calculated. As shown in Table 3, the performance metrics provide an accurate platform to compare the efficiency and diversity of candidate solutions provided by the solution algorithms.

As shown in Table 3, the scales of instances have significant impact on the CPU time and Hypervolume index ($\mathrm{HV_I}$). For example, the required time to run the algorithm, or the CPU time, vary from 12.44 to 34.78 minutes for small-scale instances, while for large-scale instances the range is escalated to be between 41.23 and 128.05 minutes. MID performance metric has shown to be more robust in terms of the scale of the instances where the value has not significantly changed once the size of the problems increased. the values for average NOM, diversity(Δ), $\mathrm{HV_I}$, and CPU time for NSGA-II, NSGA-III and MOPSO vary based on the size and property of instances. Comparing the overall performance of the three metaheuristic, the percentage of NOM, or ratio of number of Pareto members compared to population size (n_p) is 82.6%, 86.6%, and 97.3% for MOPSO, NSGA-II, and NSGA-III respectively. The trend is consistent for both small large-scale problems.

Table 4: Statistical Comparison of performance metrics using Kruskal-Wallis test with ($\alpha = 0.05$)

	Kruskal-Wallis Test			
	Small-scale instances		**large-scale instances**	
Performance Metrics	**P-Value**	**Results**	**P-Value**	**Results**
NOM ratio	0.004	NSGA-III is preferred	0.000	NSGA-III is preferred
Diversity (Δ) ratio	0.073	There is no significant difference	0.054	There is no significant difference
MID ratio	0.049	NSGA-III & MOPSO are preferred	0.040	NSGA-III & MOPSO are preferred
$\mathrm{HV_I}$ **ratio**	0.061	There is no significant difference	0.003	NSGA-II & NSGA-III are preferred
CPU time ratio	0.000	NSGA-III & MOPSO are preferred	0.000	NSGA-III & MOPSO are preferred

In order to compare the performance of three methods statistically, Kruskal-Wallis test which is a non-parametric test to determine whether the median of two or more groups of independent variables are statistically significantly different is utilized. Kruskal-Wallis test can efficiently rank the sample data and show whether samples are originated from the same distribution.

Since the value of some of the performance metrics are dependent to the properties of metaheuristic method that are individually calibrated (e.g. number metaheuristic iterations (M_{it}) and population size (n_p)), the results are normalized prior to performing Kruskal-Wallis test. The normalized values of performance metrics result in a fair

and unbiased comparison among the three metaheuristic algorithm. The results of the Kruskal-Wallis test for each performance metric is presented in Table 4.

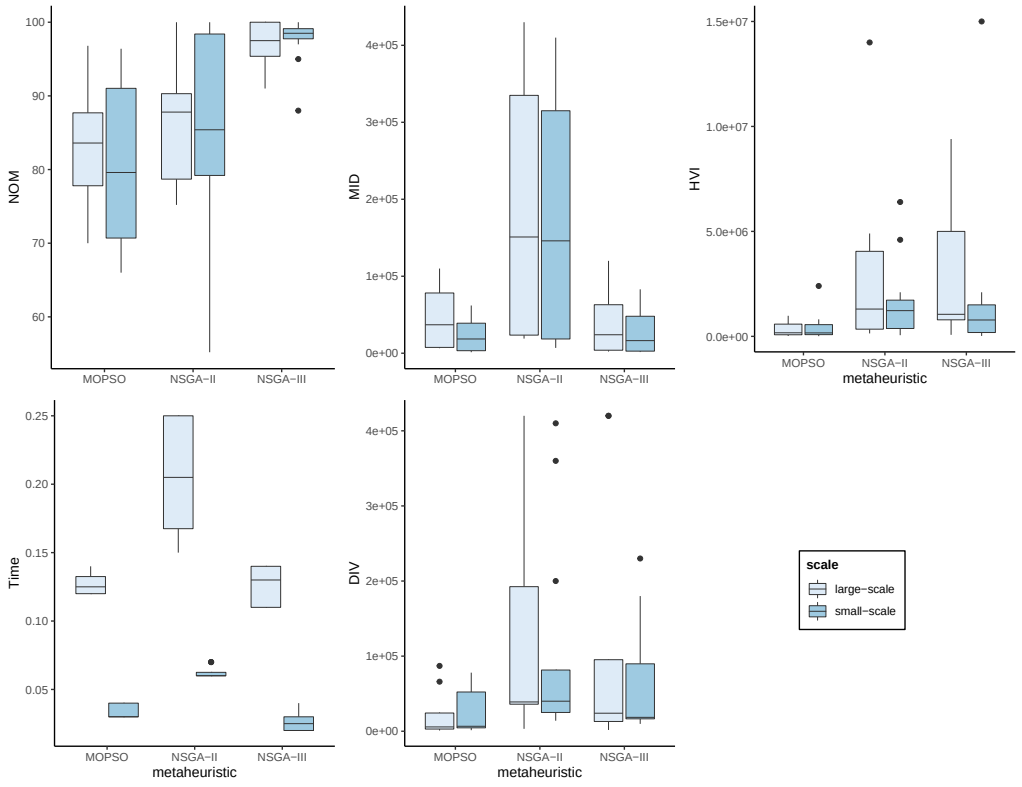
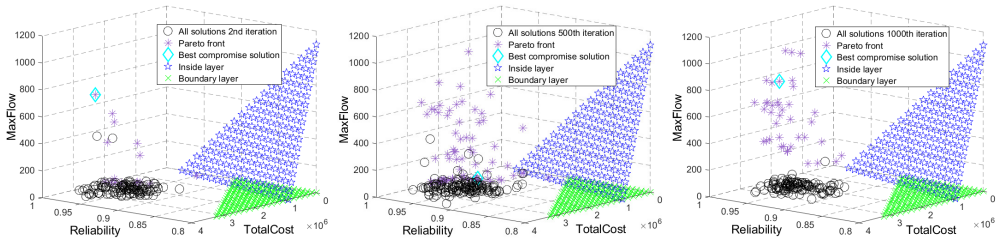


Figure 4: Statistical analysis of performance comparison metrics using Box Plot.

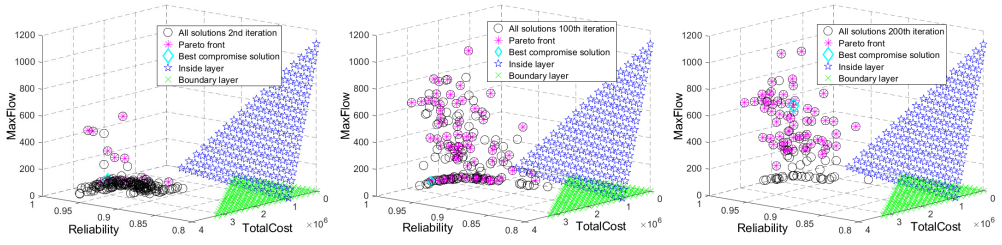
The test findings indicate that NSGA-III is majorly in favor compared to both NSGA-II and MOPSO in the viewpoint of providing higher Pareto members(NOM). NSGA-III and MOPSO share the best performance regarding CPU time ratio and MID ratio analysis. However, NSGA-II and NSGA-III outperforms MOPSO in terms of achieving higher hyper-volume for large-scale problems.There is no significant difference in the distribution of solutions for the Diversity (Δ) performance metric and HV_I in small-scale instances.

Finally, performance metrics properties are compared via box-plots to show the deviation from 1^{st} quartile, median, and 3^{rd} quartile of the results. Fig. 4 represents the box-plots for both small scale and large scale instances, while for DIV, some of the outliers are truncated for better scaling purposes. As expected, once the size of the problem increase, the CPU time increase as well. Also, there is a direct relation between the size of the problem and the hypervolume that NSGA approaches can achieve. By looking into the number of Pareto optimal members (NOM), NSGA-III outperforms both NSGA-II and MOPSO, not only by median value, but also by lower deviation among all instances. The outcomes for MID is different where both NSGA-III and MOPSO provide

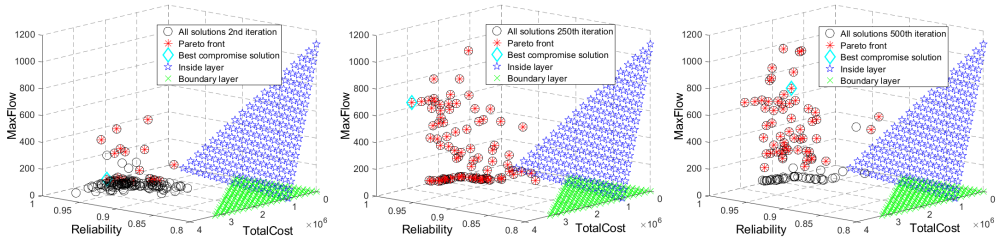
better solutions. overall NSGA-III is the preferred solution approach for both small and large-scale problems.



(a) MOPSO results for 2^{nd}, 500^{th} and 1000^{th} iterations.



(b) NSGA-II results for 2^{nd}, 100^{th} and 200^{th} iterations.



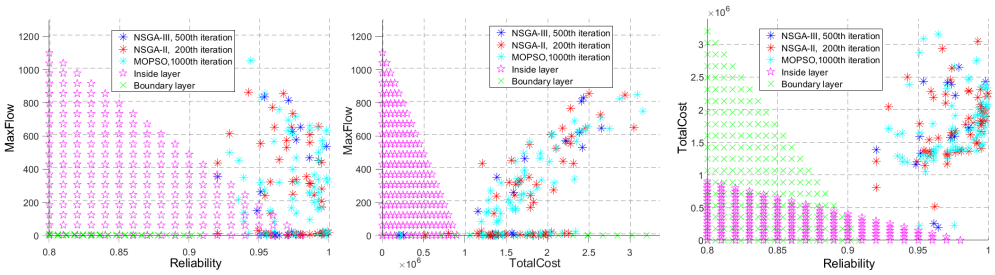
(c) NSGA-III results for 2^{nd}, 250^{th} and 500^{th} iterations.

Figure 5: All solutions, Pareto front members and best compromise solution at first, midst, and final iterations of problem instance **S3** by a) MOPSO, b) NSGA-II, and c) NSGA-III.

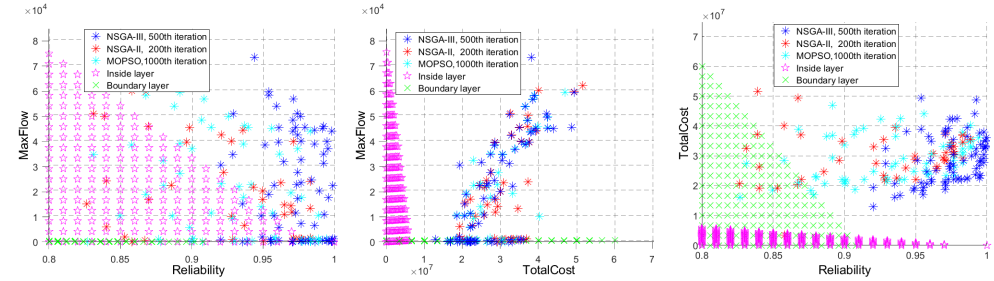
In order to visualize the progression of algorithms and their strength in finding high-quality Pareto-front members, the solutions of first, midst, and final iteration of each metaheuristic method for problem instance **S3** is illustrated in Fig. 5. In these figures, in addition to Pareto front members and corresponding best compromised solution, *Inside layer* and *Boundary layer* are demonstrated. *Inside layer* is a surface composed of reference points that intersects with the best values of the objective function, taken from the ideal point, (Z_i^{min}) on the 3D coordinate axes, and *boundary layer* is a surface composed of reference points that intersects with the worst values of the objective function (Z_i^{max}) (Deb & Jain, 2014). The figures demonstrate that at the very first iterations, the number and diversity of Pareto optimal solutions are limited. As number of iterations increase, more Pareto optimal solutions are reached, covering larger solution space. Note that

although MOPSO iteration was set to 1000, the number of Pareto optimal solutions are lower compared to NSGA-II and NSGA-III.

Another implication of Fig. 5 is the quality of solutions provided by NSGA-II, NSGA-III, and MOPSO in different iterations. As the iterations advance, the distance between the solutions and the *boundary layer* increase. Concurrently, the Pareto front solutions become closer to *inside layer*, as an indication of higher quality results.



(a) Pair-wise comparisons of objective functions for problem **S3**.



(b) Pair-wise comparisons of objective functions for problem **L4**.

Figure 6: Pair-wise comparison of Pareto front members at the final iteration of different algorithms for a) small-scale instance **S3**, and b) large-scale instance **L4**.

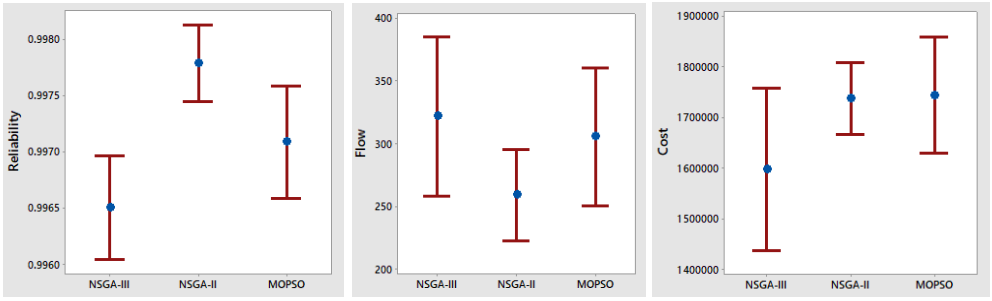
For the purpose of comparing the quality of Pareto members computed by the three metaheuristics in one platform, Fig. 6 illustrates a pair-wise comparison of objective functions for one small-scale (**S3**) and one large-scale (**L4**) problems. In general, NSGA-III Pareto members are closer to the intersection of ideal points or *inside layer*. For example, right subfigure of Fig. 6.b depicts the comparison between reliability and cost where NSGA-III provides higher reliability solutions with lower cost, compared to NSGA-II and MOPSO. Similarly, when reliability is compared against, as shown in left subfigure of Fig. 6.b, NSGA-III outperforms NSGA-II and MOPSO. Note that, a consistent trend in the results have been observed for both small and large-scale problems.

In addition to comparing performance evaluation among metaheuristic methods, the best compromise solution of each sample instance, correspond to each objective function, are shown in Table 5. In this table, bold numbers represent the best compromised solution for each objective function among the three metaheuristic methods. For a more robust comparison analysis, the percentage difference for each objective function from the best objective function is calculated and shown. While the final compromised solution of MOPSO, NSGA-II and NSGA-III differs in several instances, there are some

cases that the solutions of one metaheuristic algorithm are completely dominated by another algorithm. More specifically, for several large-scale instances, MOPSO is unable to provide solutions that are not completely dominated by NSGA-II and NSGA-III.

Table 5: Best compromise solution of sample instances from each metaheuristic algorithms. (%) indicates the difference ratio between the provided solution and the best solution among different methods (*). (P = Problem, N = Nodes, R = Reliability)

Scenario properties				MOPSO			NSGA-II			NSGA-III		
P	N	R	u_{ij}	Reliability	Cost	Flow	Reliability	Cost	Flow	Reliability	Cost	Flow
S1	**6**	**0.75**	**[50,600]**	0.88(-4%)	**3.20E+06(*)**	**9.80E+02(*)**	**0.92(*)**	2.20E+06(-31%)	6.30E+02(-36%)	0.9(-2%)	2.40E+06(-25%)	8.10E+02(-17%)
S2	**6**	**0.75**	**[2k,6k]**	0.9(-1%)	**8.80E+06(*)**	**1.40E+04(*)**	**0.91(*)**	8.30E+06(-6%)	1.30E+04(-7%)	0.9(-1%)	**8.80E+06(*)**	**1.40E+04(*)**
S3	**6**	**0.85**	**[50,600]**	0.93(-2%)	**3.00E+06(*)**	**9.40E+02(*)**	**0.95(*)**	2.30E+06(-23%)	6.50E+02(-31%)	0.93(-2%)	**3.00E+06(*)**	**9.40E+02(*)**
S4	**6**	**0.85**	**[2k,6k]**	0.91(-5%)	8.90E+06(-4%)	1.40E+04(-7%)	**0.96(*)**	9.00E+06(-3%)	1.40E+04(-7%)	0.94(-2%)	**9.30E+06(*)**	**1.50E+04(*)**
S5	**6**	**0.95**	**[50,600]**	0.95(-3%)	2.40E+06(-17%)	7.80E+02(-12%)	**0.98(*)**	2.80E+06(-3%)	8.50E+02(-4%)	0.97(-1%)	**2.90E+06(*)**	**8.90E+02(*)**
S6	**6**	**0.95**	**[2k,6k]**	0.95(-2%)	9.30E+06(-2%)	**1.50E+04(*)**	0.96(-1%)	8.70E+06(-8%)	1.20E+04(-20%)	**0.97(*)**	**9.50E+06(*)**	**1.50E+04(*)**
S7	**10**	**0.75**	**[50,600]**	0.87(-3%)	5.20E+06(-2%)	2.30E+03(-4%)	0.89(-1%)	5.20E+06(-2%)	2.30E+03(-4%)	**0.9(*)**	**5.30E+06(*)**	**2.40E+03(*)**
S8	**10**	**0.75**	**[2k,6k]**	**0.89(*)**	1.40E+07(-7%)	5.20E+04(-17%)	0.86(-3%)	1.20E+07(-20%)	4.90E+04(-22%)	0.86(-3%)	**1.50E+07(*)**	**6.30E+04(*)**
S9	**10**	**0.85**	**[50,600]**	0.93(-3%)	6.70E+06(-1%)	2.10E+03(-5%)	0.92(-4%)	**6.80E+06(*)**	**2.20E+03(*)**	**0.96(*)**	6.70E+06(-1%)	2.10E+03(-5%)
S10	**10**	**0.85**	**[2k,6k]**	**0.91(*)**	1.50E+07(-25%)	6.10E+04(-27%)	0.9(-1%)	**2.00E+07(*)**	7.30E+04(-13%)	**0.91(*)**	1.90E+07(-5%)	**8.40E+04(*)**
S11	**10**	**0.95**	**[50,600]**	**0.98(*)**	8.10E+06(-1%)	2.50E+03(-4%)	0.97(-1%)	4.70E+06(-43%)	1.70E+03(-35%)	**0.98(*)**	**8.20E+06(*)**	**2.60E+03(*)**
S12	**10**	**0.95**	**[2k,6k]**	0.97(-1%)	1.60E+07(-27%)	5.90E+04(-23%)	**0.98(*)**	1.80E+07(-18%)	6.50E+04(-16%)	**0.98(*)**	**2.20E+07(*)**	**7.70E+04(*)**
L1	**30**	**0.75**	**[100,800]**	0.84(-7%)	5.70E+07(-7%)	9.30E+04(-7%)	0.88(-2%)	5.60E+07(-8%)	9.10E+04(-9%)	**0.9(*)**	**6.10E+07(*)**	**1.00E+05(*)**
L2	**30**	**0.75**	**[3k,7k]**	0.84(-1%)	9.30E+07(-1%)	2.70E+06(-7%)	**0.85(*)**	9.30E+07(-1%)	2.70E+06(-7%)	**0.85(*)**	**9.40E+07(*)**	**2.90E+06(*)**
L3	**30**	**0.85**	**[100,800]**	0.92(-3%)	5.10E+07(-18%)	8.80E+04(-20%)	0.94(-1%)	5.20E+07(-16%)	8.90E+04(-19%)	**0.95(*)**	**6.20E+07(*)**	**1.10E+05(*)**
L4	**30**	**0.85**	**[3k,7k]**	0.88(-3%)	**9.40E+07(*)**	**2.80E+06(*)**	**0.91(*)**	8.90E+07(-5%)	2.60E+06(-7%)	0.89(-2%)	9.30E+07(-1%)	**2.80E+06(*)**
L5	**30**	**0.95**	**[100,800]**	0.97(-1%)	**6.30E+07(*)**	9.70E+04(-12%)	**0.98(*)**	5.70E+07(-10%)	9.20E+04(-16%)	0.96(-2%)	6.20E+07(-2%)	**1.10E+05(*)**
L6	**30**	**0.95**	**[3k,7k]**	0.93(-1%)	9.50E+07(-1%)	2.90E+06(-6%)	**0.94(*)**	9.40E+07(-2%)	2.80E+06(-10%)	0.92(-2%)	**9.60E+07(*)**	**3.10E+06(*)**
L7	**40**	**0.75**	**[100,800]**	0.89(-1%)	**7.60E+07(*)**	1.70E+05(-11%)	**0.9(*)**	7.40E+07(-3%)	**1.90E+05(*)**	**0.9(*)**	7.50E+07(-1%)	**1.90E+05(*)**
L8	**40**	**0.75**	**[3k,7k]**	0.87(-3%)	**3.40E+08(*)**	4.00E+06(-9%)	**0.9(*)**	3.30E+08(-3%)	4.10E+06(-7%)	0.86(-4%)	**3.40E+08(*)**	**4.40E+06(*)**
L9	**40**	**0.85**	**[100,800]**	**0.92(*)**	**7.80E+07(*)**	**2.20E+05(*)**	0.91(-1%)	7.70E+07(-1%)	1.70E+05(-23%)	**0.92(*)**	**7.80E+07(*)**	**2.20E+05(*)**
L10	**40**	**0.85**	**[3k,7k]**	**0.93(*)**	3.10E+08(-3%)	**4.20E+06(*)**	**0.93(*)**	**3.20E+08(*)**	3.90E+06(-7%)	**0.93(*)**	3.10E+08(-3%)	4.10E+06(-2%)
L11	**40**	**0.95**	**[100,800]**	0.96(-2%)	7.50E+07(-4%)	1.80E+05(-10%)	**0.98(*)**	**7.80E+07(*)**	1.90E+05(-5%)	**0.98(*)**	7.60E+07(-3%)	**2.00E+05(*)**
L12	**40**	**0.95**	**[3k,7k]**	0.96(-1%)	3.00E+08(-9%)	4.20E+06(-5%)	**0.97(*)**	3.10E+08(-6%)	4.00E+06(-9%)	0.96(-1%)	**3.30E+08(*)**	**4.40E+06(*)**



(a) Maximizing reliability (b) Maximizing flow (c) Minimizing Cost

Figure 7: Comparison between 95% confidence interval for mean of each objective function.

NSGA-III has proven to be robustly better or equally good, in approximately 34% of cases, compared to both MOPSO and NSGA-II in small and large-scale instances. Majority of NSGA-III superiority is as a result of providing higher flow for the network, in over 87% of the cases, compared to the other algorithms. The number of cases that

NSGA-III is outperforming NSGA-II and/or MOPSO in terms of minimizing cost and maximizing reliability objective functions are 66% and 54%, respectively.

Finally, the 95% confidence intervals for the mean value of each objective function is shown in Figure 7. This analysis provides the opportunity to compare the confidence interval of Pareto front solutions provided by each metaheuristic methods. The Pareto front solutions of NSGA-II provide higher reliability, compared to MOPSO and NSGA-III. However, in terms of the cost and flow, NSGA-III Pareto front solutions provide more robust solutions. Note that individual standard deviations are used to calculate the intervals.

6. Discussion

In this Section, we first provide a brief, yet holistic view of the specific gaps covered by this research and the areas which still need to be investigated. Then, we outline the challenges associated with the solution methods and elaborate on their computational complexity.

6.1. Strengths and Limitations of the Proposed Model

Enhancing the reliability of networks is one of the most fundamental elements of many physical and virtual networks such as wired, wireless, supply chain, and social networks. However, the network reliability comes with a price. This research examined two additional objectives of minimizing the total cost and maximizing the flow, while designing reliable networks. Our proposed mixed integer linear model, seeks to find the best topology and design of a network such that the three conflicting objectives are addressed. The practicality of the model relies in taking important features of the network design into account such as both cost of building new arcs and the cost of flow, and transmission rates on the edges rather than nodes.

Our research, similar to other research studies, encounters several limitations that could be further explored by the interested readers. First, assumptions in the proposed mathematical model can be revised based on the characteristics of future problems. For instance, reliability of all arcs are assumed to be similar. However, this assumption limits the use of the model for complex networks. Also, in some networks, arc failure increases as adjacent arcs are not functional. The contingency among arcs' failure are mostly seen in disruptive extreme events, such as earthquakes and hurricanes. Taking uncertainty into account, the model can further expanded with stochastic probability of arc failure and the cost of the repair. Additional exploration can consist of adding multi-period and multi-commodity flows representing more advanced networks.

6.2. Strengths and Limitations of the Proposed Methods

The proposed mathematical model is first solved with CPLEX for a very small instance and the optimal solution is verified to ensure the accuracy of the model. However, for large-scale problems, CPLEX is either unable to solve the problem or attain an acceptable optimality gap in a reasonable time-frame. Hence, three metahueristic methods (i.e. MOPSO, NSGA-II, and NSGA-III) are employed and compared to solve the proposed problem.

Several advantages and disadvantages are associated with each of these methods. MOPSO's greatest advantage, compared to NSGA-II and NSGA-III is the simplicity of its concept, ease of implementation, and good convergence rate (Coello et al., 2007). However, MOPSO is unable to control diversity, compared to the other two metaheuristic algorithms. NSGA-II is mainly criticized for exploratory capabilities and search biases when the number of objective functions increases. In the meantime, the algorithm strength is to spread properly when a certain non-dominated region is found, providing diverse solutions (Coello & Pulido, 2001). Overall, NSGA-III has shown major superiority, compared to MOPSO and NSGA-II, in providing higher number of Pareto front and non-dominated solutions with reasonable diversity and convergence.

While metaheuristic methods are highly utilized to solve large-scale problems, one of their limitations is the associated computational complexity. In general, complexity of multi-objective evolutionary algorithms is a function of number of objective functions and the third power of population size but can vary based on the sorting algorithm and domination rules. Specifically, computational complexity of MOPSO and NSGA-II in each iteration is $\mathcal{O}(n_o n_p^2)$ where n_o is the number of objective functions and n_p is the population size (Tripathi et al., 2007). For NSGA-III, the overall worst-case complexity is $\mathcal{O}(n_p^2 \log^{n_o-2} n_p)$ or $\mathcal{O}(n_o n_p^2)$ whichever is larger (Deb & Jain, 2013). While the computational complexity of the metaheuristic methods are in favor of lower n_p, the solution method may not find near-optimal solutions once the size of population is not sufficient. Therefore, it is important to find the best population size to balance the computational complexity and the quality of solutions. This in turn introduces another limitation of the proposed solution which is computationally expensive parameter tuning for these methods. Despite the fact that there are several guidelines on the best range of parameters to be used by NSGA-II, NSGA-III, and MOPSO, these parameters need to be calibrated for each specific problem. The tuning procedure requires additional effort to investigate which are sometimes neglected in other research studies. As a result of the calibration process, one realization is the capability of NSGA-III to provide higher quality solution with lower computational complexity. In this research, the population level for NSGA-III is 200, compared to 250 for MOPSO and NSGA-II.

7. Conclusion

Flow networks play a vital role in many industries. Moreover, the security and reliability of the flow networks, especially infrastructure networks, has become the major concern for both industry managers and government policy makers. Howbeit, the networks' reliability is achieved with a higher cost, and a design of the model that can compromise among several conflicting objectives of a flow network is required.

In this paper, a three-objective mixed-integer linear mathematical model is developed that simultaneously optimizes three goals of minimizing total cost, maximizing flow, and maximizing reliability. The minimization of cost comprises the cost of construction of network arcs and the cost of flow. To maximize the flow of the network, a generalized flow network problem with positive multiplier for each arc is considered to account for the flow loss. The network reliability is estimated by the Jans upper bound, used in large-scale networks where calculating the network reliability with exact methods are not feasible. Since computational complexity of the proposed model is NP-hard, three multi-objective metaheuristic methods are used; Multi-objective Particle Swarm Optimization

(MOPSO), and two non-dominated sorting genetic algorithms (i.e. NSGA-II and NSGA-III). Each of the methods has shown capabilities and strengths on providing near optimal solutions for large-scale multi-objective problems.

To decide on selecting best compromise solution from the Pareto front members, a fuzzy-based mechanism is applied. Therefore, the proposed model leads to a final reliable generalized flow network design which would represent the desired compromise among the different objectives from the decision-makers perspective. To ensure the results' quality of the metaheuristic methods, the efficiency of MOPSO, NSGA-II and NSGA-III are compared using several performance metrics. Our results indicate that NSGA-III outperforms MOPSO and NSGA-II in terms of convergence and run-time for both small-scale and large-scale instances.

While the solutions of NSGA-III is promising for different instances, our analysis indicates that: (1) the selection of the method to solve such large-scale complex problems is important and (2) hybrid methods that are carefully calibrated for the specific objectives can be beneficial. For instance, MOPSO and NSGA-III have specific strengths and a hybrid model that can utilize strengths of both these methods and overcome the potential shortcomings of each individual method can be a direction for future research.

The proposed mathematical model can be extended in several ways to be applicable for infrastructure networks with different properties. One recommendation is the applications of these models in the design of wireless computer networks and electric power networks. These networks may have stochastic demand that need to be fulfilled within each time slot. In these cases, the model can be expanded to stochastic, multi-time period, and multi-commodity network flow problems.

Appendix A.

The pseudocode for calculating the penalty function for the reliability objective function considering fully-connected and two-connectivity conditions is shown in Algorithm 2.

Algorithm 2 Consideration of Fully-Connected and Two-Connectivity Conditions using Penalty Function

Compute V_{AC} ▷ All-Connectivity Violation
Compute graph degree sequence (d)
$V_{2C} \leftarrow \text{mean}(d, 2)$ ▷ 2-Connectivity Violation
$H \leftarrow 1$ ▷ H is the Jan's Upper Bound
for $i \leftarrow 1$ to V **do**
    $m_i \leftarrow \min(d_i, i-1)$ ▷ m_i: min edges connected to i
    $dH \leftarrow q^{d(i)}$ ▷ q: probability of failure
    if $m_i \geq 1$ **then**
        $dH \leftarrow dH * (1 - q^{d_i - 1})^{m_i}$
    end if
    if $i - 1 \geq m_i + 1$ **then**
        $dH \leftarrow dH * (1 - q^{d_i - 1})^{i-1-m_i}$
    end if
    $H \leftarrow H - dH$
end for
Reliability $\leftarrow$ H
Violation $\leftarrow V_{AC} + V_{2C}$
$\beta \leftarrow 5$
$z \leftarrow R - \beta *$ Violation

References

Akbaripour, H., & Masehian, E. (2013). Efficient and robust parameter tuning for heuristic algorithms. *International Journal of Industrial Engineering and Production Research*, .

Azghandi, R., Griffin, J., & Jalali, M. S. (2018). Minimization of drug shortages in pharmaceutical supply chains: A simulation-based analysis of drug recall patterns and inventory policies. *Complexity*, *2018*.

Bagherinejad, J., & Dehghani, M. (2016). Optimising the bi-objective multidimensional integer knapsack problem using non-dominated sorting particle swarm optimisation. *International Journal of Industrial and Systems Engineering*, *23*, 263–289.

Beltrán, H. F., & Skorin-Kapov, D. (1994). On minimum cost isolated failure immune networks. *Telecommunication Systems*, *3*, 183–200.

Benson, H. Y., Shanno, D. F., & Vanderbei, R. J. (2003). A comparative study of large-scale nonlinear optimization algorithms. In *High performance algorithms and software for nonlinear optimization* (pp. 95–127). Springer.

Bhesdadiya, R. H., Trivedi, I. N., Jangir, P., Jangir, N., & Kumar, A. (2016). An NSGA-III algorithm for solving multi-objective economic/environmental dispatch problem. *Cogent Engineering*, *3*, 1269383.

Cancela, H., Murray, L., & Rubino, G. (2019). Efficient estimation of stochastic flow network reliability. *IEEE Transactions on Reliability*, .

Chen, X. G. (2017). A novel reliability estimation method of complex network based on monte carlo. *Cluster Computing*, *20*, 1063–1073.

Ciro, G. C., Dugardin, F., Yalaoui, F., & Kelly, R. (2016). A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine*, *49*, 1272–1277.

Coello, C. A., & Pulido, G. T. (2001). Multiobjective optimization using a micro-genetic algorithm. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (pp. 274–282). Morgan Kaufmann Publishers Inc.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A. et al. (2007). *Evolutionary algorithms for solving multi-objective problems* volume 5. Springer.

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, *8*, 256–279.

Cohen, E., & Megiddo, N. (1994). New algorithms for generalized network flows. *Mathematical Programming*, *64*, 325–336.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature* (pp. 849–858). Springer.

Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, *18*, 577–601.

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, *18*, 577–601.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*, 182–197.

Deeter, D. L., & Smith, A. E. (1997). Heuristic optimization of network design considering all-terminal reliability. In *Reliability and Maintainability Symposium. 1997 Proceedings, Annual* (pp. 194–199). IEEE.

Dengiz, B., & Smith, A. E. (2000). Evolutionary methods for design of reliable networks. *Telecommunications Optimization: Heuristics and Adaptive Methods*, .

Dubourg, V., Sudret, B., & Deheeger, F. (2013). Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics*, *33*, 47–57.

El Khadiri, M., & Yeh, W.-C. (2016). An efficient alternative to the exact evaluation of the quickest path flow network reliability problem. *Computers & Operations Research*, *76*, 22–32.

Elshqeirat, B., Soh, S., Rai, S., & Lazarescu, M. (2015). Topology design with minimal cost subject to network reliability constraint. *IEEE Transactions on Reliability*, *64*, 118–131.

Esmailpour, A., Salehi, S., & Safavi, N. (2013). Quality of service differentiation measurements in 4g networks. In *2013 Wireless Telecommunications Symposium (WTS)* (pp. 1–5). IEEE.

Fallahi, M., Amiri, S., & Yaghini, M. (2014). A parameter tuning methodology for metaheuristics based on design of experiments. *International Journal of Engineering and Technology Sciences*, *2*, 497–521.

Garey, M. R. (1979). A Guide to the Theory of NP-Completeness. *Computers and intractability*, .

Gertsbakh, I. B., & Shpungin, Y. (2016). *Models of network reliability: analysis, combinatorics, and Monte Carlo*. CRC press.

Guo, J., Li, Z. S., & Jin, J. J. (2018). System reliability assessment with multilevel information using the Bayesian melding method. *Reliability Engineering & System Safety*, *170*, 146–158.

Hochbaum, D. S. (2016). Min cost flow on unit capacity networks and convex cost K-flow are as easy as the assignment problem with All-Min-Cuts algorithm. *arXiv preprint arXiv:1610.04012*, .

Jan, R.-H., Hwang, F.-J., & Chen, S.-T. (1993). Topological optimization of a communication network subject to a reliability constraint. *IEEE Transactions on Reliability*, *42*, 63–70.

Jane, C.-C., & Laih, Y.-W. (2017). Distribution and reliability evaluation of max-flow in dynamic multi-state flow networks. *European Journal of Operational Research*, *259*, 1045–1053.

Jaszkiewicz, A. (2018). Improved quick hypervolume algorithm. *Computers & Operations Research*, *90*, 72–83.

Jian, G., Zhaojun, L., & Keyser, T. (2018). A Bayesian approach for integrating multilevel priors and data for aerospace system reliability assessment. *Chinese Journal of Aeronautics*, *31*, 41–53.

Karimi, N., Zandieh, M., & Karamooz, H. (2010). Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. *Expert Systems with Applications*, *37*, 4024–4032.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on* (pp. 1942–1948). IEEE volume 4.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, *91*, 992–1007.

Konak, A., & Smith, A. E. (2006). Network reliability optimization. In *Handbook of optimization in telecommunications* (pp. 735–760). Springer.

Konak, A., & Smith, A. E. (2011). Efficient optimization of reliable two-node connected networks: a biobjective approach. *INFORMS Journal on Computing*, *23*, 430–445.

Kruisselbrink, J. (2015). Hypervolume Computation. `http://www.sourcecodeonline.com/details/hypervolume_computation.html`. [Online; accessed 19-July-2018].

Kuipers, F. A., Yang, S., Trajanovski, S., & Orda, A. (2014). Constrained maximum flow in stochastic networks. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on* (pp. 397–408). IEEE.

Larsson, C. (2014). *Design of Modern Communication Networks: Methods and Applications*. Academic

Press.

Li, Z., Mobin, M., & Keyser, T. (2016). Multi-objective and multi-stage reliability growth planning in early product-development stage. *IEEE Transactions on Reliability*, *65*, 769–781.

Lin, Y.-K., & Chang, P.-C. (2013). A novel reliability evaluation technique for stochastic-flow manufacturing networks with multiple production lines. *IEEE Transactions on Reliability*, *62*, 92–104.

Lin, Y.-K., & Yeh, C.-T. (2015). System reliability maximization for a computer network by finding the optimal two-class allocation subject to budget. *Applied Soft Computing*, *36*, 578–588.

Malinowski, J. (2016). Reliability analysis of a flow network with a series-parallel-reducible structure. *IEEE Transactions on Reliability*, *65*, 851–859.

Mobin, M., Li, Z., & Komaki, G. (2017). A multiobjective approach for multistage reliability growth planning by considering the timing of new technologies introduction. *IEEE Transactions on Reliability*, *66*, 97–110.

Mousavi, S., Taghiabadi, M. M. R., & Ayanzadeh, R. (2019). A survey on compressive sensing: Classical results and recent advancements. *arXiv preprint arXiv:1908.01014*, .

Namin, A. T., Kamarthi, S., Eckelman, M. J., & Isaacs, J. A. (2019). Comparison of us manufacturing locations for solar pvs. *Procedia CIRP*, *80*, 434–439.

Nguyen, D., Vo, B., & Vu, D.-L. (2016). A parallel strategy for the logical-probabilistic calculus-based method to calculate two-terminal reliability. *Quality and Reliability Engineering International*, *32*, 2313–2327.

Niimura, T., & Nakashima, T. (2003). Multiobjective tradeoff analysis of deregulated electricity transactions. *International Journal of Electrical Power & Energy Systems*, *25*, 179–185.

Oliveira, S., Wanner, E. F., de Souza, S. R., Bezerra, L. C., & Stützle, T. (2019). The hypervolume indicator as a performance measure in dynamic optimization. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 319–331). Springer.

Ozkan, O., Ermis, M., & Bekmezci, I. (2019). Reliable wireless multimedia sensor network design: comparison of hybrid metaheuristics and a matheuristic. *Computational and Applied Mathematics*, *38*, 106.

Pant, S., Kumar, A., Kishor, A., Anand, D., & Singh, S. B. (2015). Application of a multi-objective particle article swarm optimization technique to solve reliability optimization problem. In *Next Generation Computing Technologies (NGCT), 2015 1st International Conference on* (pp. 1004–1007). IEEE.

Pasandideh, S. H. R., Niaki, S. T. A., & Sharafzadeh, S. (2013). Optimizing a bi-objective multi-product epq model with defective items, rework and limited orders: Nsga-ii and mopso algorithms. *Journal of Manufacturing Systems*, *32*, 764–770.

Praks, P., Kopustinskas, V., & Masera, M. (2017). Monte-carlo-based reliability and vulnerability assessment of a natural gas transmission system due to random network component failures. *Sustainable and Resilient Infrastructure*, *2*, 97–107.

Provan, J. S., & Ball, M. O. (1983). The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, *12*, 777–788.

Rong-Hong, J. (1993). Design of reliable networks. *Computers & Operations Research*, *20*, 25–34.

Roy, R. K. (2001). *Design of experiments using the Taguchi approach: 16 steps to product and process improvement*. John Wiley & Sons.

Ruiwen, H., Jianhua, D., & Lai, L. L. (2017). Reliability evaluation of communication-constrained protection systems using stochastic-flow network models. *IEEE Transactions on Smart Grid*, *9*, 2371–2381.

Salehi, S., Li, L., Shen, C.-C., Cimini, L., & Graybeal, J. (2018). Traffic differentiation in dense wlans with csma/eca-dr mac protocol. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)* (pp. 1–5). IEEE.

Sherman, J. (2017). Generalized preconditioning and undirected minimum-cost flow. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 772–780). SIAM.

Srivaree-ratana, C., Konak, A., & Smith, A. E. (2002). Estimation of all-terminal network reliability using an artificial neural network. *Computers & Operations Research*, *29*, 849–868.

Suman, B. (2003). Simulated annealing-based multiobjective algorithms and their application for system reliability. *Engineering Optimization*, *35*, 391–416.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* volume 74. John Wiley & Sons.

Tatsis, V. A., & Parsopoulos, K. E. (2019). Dynamic parameter adaptation in metaheuristics using gradient approximation and line search. *Applied Soft Computing*, *74*, 368–384.

Tavana, M., Li, Z., Mobin, M., Komaki, M., & Teymourian, E. (2016). Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS. *Expert Systems*

with Applications, *50*, 17–39.

Tripathi, P. K., Bandyopadhyay, S., & Pal, S. K. (2007). Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. *Information sciences*, *177*, 5033–5049.

Watcharasitthiwat, K., & Wardkein, P. (2009). Reliability optimization of topology communication network design using an improved ant colony optimization. *Computers & Electrical Engineering*, *35*, 730–747.

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE transactions on evolutionary computation*, *10*, 29–38.

Xu, J., Tu, Y., & Zeng, Z. (2012). A nonlinear multiobjective bilevel model for minimum cost network flow problem in a large-scale construction project. *Mathematical Problems in Engineering*, *2012*.

Yeh, W.-C. (2013). Evaluating the reliability of a novel deterioration-effect multi-state flow network. *Information Sciences*, *243*, 75–85.

Yeh, W.-C. (2015). A fast algorithm for quickest path reliability evaluations in multi-state flow networks. *IEEE Transactions on Reliability*, *64*, 1175–1184.

Yeh, W.-C., & Chu, T.-C. (2018). A novel multi-distribution multi-state flow network and its reliability optimization problem. *Reliability Engineering & System Safety*, *176*, 209–217.

Zimmermann, H.-J. (2011). *Fuzzy set theoryand its applications*. Springer Science & Business Media.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications* volume 63. Citeseer.

Zitzler, E., Brockhoff, D., & Thiele, L. (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 862–876). Springer.

Zuev, K. M., Wu, S., & Beck, J. L. (2015). General network reliability problem and its efficient solution by subset simulation. *Probabilistic Engineering Mechanics*, *40*, 25–35.