



Evolutionary Approaches to Fog Node Placement in LV Distribution Networks

Gilbert M. Gilbert[†], Naiman Shililiandumi^{**}, Honest Kimaro^{***},

^{*}Department of Information Systems and Technology, The University of Dodoma, P.O.Box 490, Dodoma, Tanzania

^{**}Department of Electronics and Telecommunication Engineering, University of Dar es Salaam, P.O.Box 33335, Dar es Salaam, Tanzania

^{***}Department of Computer Science and Engineering, University of Dar es Salaam, P.O.Box 33335, Dar es Salaam, Tanzania

(gilly.map@gmail.com, smakere@gmail.com, honestck@gmail.com)

[†]Corresponding Author; Gilbert M. Gilbert, Tel: +255716728542, gilly.map@gmail.com

Received: 03.12.2020 Accepted: 11.01.2021

Abstract- Smart grids merge intelligent computing technologies and electrical grid networks for better monitoring, control and management of electrical energy and facilities. The maturity of cloud computing has been the major driving factor for its adoption in smart grid deployments. Despite the elasticity of cloud resources, centrality and long distances to remote data centers cause high latency, high bandwidth consumptions and unstable connectivity, which are undesirable for IoT-based smart grid applications. Fog computing as an extension of cloud computing services close to the sources of data overcomes these challenges and perfectly suits the distributed nature of the low voltage (LV) electrical distribution networks as part of a smart grid. The pressing issue with the uptake of fog computing for smart grid services is finding the best placement plan for fog node locations in LV distribution networks to enhance monitoring and control. The main goal of this work is to present a mathematical model to address the aforementioned issues focusing on minimizing deployment cost and network delay. In addressing this multi-objective problem, a new algorithm, namely Future Search Particle Swarm Non-dominated Sorting Genetic Algorithm (FPNSGA), is proposed and it combines the best features of the evolutionary algorithms NSGA-II, SMPSO, and Future Search. The effectiveness of the algorithm is evaluated based on the benchmarking technique (Weighted Sum approach), the convergence and diversification of the solutions using HyperVolume indicators and CPU time. The results from simulations show that the proposed mechanism is very competitive and outperforms other fog planning network schemes by more than 8% in terms of HV indicators.

Keywords Smart Grid, Multi-objective, LV distribution, Optimization, Fog Computing, Network Planning.

Table 1 : A table of Nomenclature

Symbol	Meaning
I	Potential sensor/edge cluster locations set
m	Number of sensor cluster locations
C	Edge Node/sensor cluster
c_j	j^{th} sensor cluster $j \in [1, m]$
$P(c)$	Required processing power
$M(c)$	Required Memory
$K(c)$	Sensor cluster kind
$P_c^{K(c)}$	The total processing power required is by edge node devices of the kind $K(c)$
$M_c^{K(c)}$	The sum of the memory required by the edge node cluster of kind $K(c)$,

T_c	Packets needed by a sensor cluster of kind $K(c)$ from the fog devices or data center
F	A set of fog devices
f_i	i^{th} fog node (device) $i \in [1, k]$
$M_F^{K(f)}$	The memory capacity for fog device of the kind $K(f)$
$P_F^{K(f)}$	The number of CPUs available
$N_F^{K(f)}$	The network interface capacity
$\Phi_F^{K(f)}$	The number of fog devices of the kind $K(f)$
$\phi_F^{K(f)}$	The cost of a fog device of the kind $K(f)$
l	A set of possible physical links installed
ℓ_i	i^{th} physical link $i \in [1, k]$

B^ℓ	Bandwidth (in Gbps/Mbps) of a link
σ^ℓ	The price of a link of a type (in dollars per meter)
P	Set of possible geographical position for deployment of fog nodes
ρ_i	i^{th} fog node location $i \in [1, k]$
δ	Average packet size coming from the edge devices
Γ	The maximum budget allocated for the installation of fog devices
h	The average number of hops
d	Average processing delay per hop
q	A quota of traffic directed to the data center from the fog devices
d^{ab}	the distance between points a and b (using x, y coordinates)
$x^{\rho k} \in [0, 1]$	$x^{\rho k} = 1$ when a fog node $f_i \in F$ is deployed at the position $\rho_i \in P$, 0 otherwise
$y^{c\rho} \in [0, 1]$	$y^{c\rho} = 1$ when a sensor node cluster $c_i \in C$ is linked with fog device at the position $\rho_i \in P$.
$v^{cD} \in [0, 1]$	$v^{cD} = 1$ when a sensor node cluster $c_i \in C$ is connected to data center D .
$b^{\rho l} \in [0, 1]$	$b^{\rho l} = 1$ when a fog device $f_i \in F$ deployed at position $\rho_i \in P$ is interconnected to data center using link $\ell_i \in l$
$c_i^{\text{Site}}, c_k^{\text{Fog}}$	site rent cost, fog node cost,
$c_i^{\text{Link}}, d_{\rho, \text{cloud}}$	link cost per meter and distance from fog node to the cloud.
$D(d^{\rho c})$	network function that represents end-to-end delay

1. Introduction

Low voltage (LV) distribution networks have been operating in “Connect and Forget” approach and is still predominant in most developing countries [1]. Despite improvements such as that of digitization of consumer energy meters, these networks are still largely neglected and isolated in terms of monitoring and control, as there are no sensing devices and two-way communication systems for exchange of data with distribution substations. Further, owing to their complexity and geographical coverage, these networks have poor visibility and thus are prone to interruptions caused by faults of various nature. The poor visibility in developing countries, particularly, Tanzania is a major cause of unreliable power supply and loss of income.

There are efforts to modernize legacy and newly electric grids by making use of latest sensing and computing advancements to improve visibility and hence provide intelligent monitoring and control [2, 3]. From the literature, different aspects of methodologies, technologies, and techniques have been and are being proposed by industries and academia to and have proved to work effectively in transmission and medium voltage (MV) networks [4, 5]. These smart-grid-based solutions consume and generate a huge amount of data from sensors, substations, smart meters, SCADA, customers’ devices, and historical data [6] which

would require sound computing infrastructure to process them.

Cloud computing architectures have been widely adopted for the deployment in smart grids [7, 8] as they deliver dynamically scalable computing services on demand and in a subscription manner with convenience. Usually data collected from the power grid systems are sent to the centralized cloud data centers for processing and archival. However, as cloud centers are remotely placed and serve a lot of devices, it leads to poor quality of service as a result of undesirable network bandwidth and latency [9]. As new IoT-based smart grid applications are sensitive to latency, geo-distributed, security-aware and in need of near-real-time processing, there is a need of introducing a new platform that can respond to these requirements which can not be guaranteed by cloud computing.

The proposed new platform has been named as Fog computing [9], and its theoretical foundations promote very well the concept of distributed processing in which services and applications can be locally managed and globally controlled. An architecture based on fog computing consists of distributed computing units called fog nodes in which fog-based services are deployed. Typically, these fog nodes are positioned in proximity to data sources or users of fog services and act as a bridge or near-end proxies between front-end IoT devices (and sensors) and the backend cloud services. They don’t replace but extend cloud systems to the edges of the network.

LV distribution networks can leverage fog-based computing architecture to enhance visibility by placing fog nodes to identified customer areas, and process data received from IoT sensor devices installed along the distribution system for immediate feedback. Practically, deploying a lot of fog nodes at a particular geographical area is the same as committing more resources to the edges of the network enabling processing of latency-sensitive applications while heavy and latency-tolerant tasks are forwarded to the cloud.

Taking into consideration the distributed nature of the LV networks, the deployment of fog nodes needs to be carefully planned as their utilization need to be maximized, installation costs to be minimized and being able to serve a large number of assets in LV networks while minimizing service latency. There is a need of comprehensive planning tools to design optimal fog-based computing architecture to serve the LV distribution networks.

Optimizing the fog network planning problem by considering service delay and deployment cost, leads to a multi-objective optimization problem (MOOP) which may constitutes several non-dominated solutions. A solution (proposed network plan) that can minimize the overall delay but corresponds to the high deployment cost of fog node devices is not optimal.

This research paper is inspired by the potential deployment of fog nodes in LV distribution networks and extends authors’ previous work in [1]. The major objective is to propose fog network plan that achieves a good tradeoff between the overall service delay and deployment cost. It extends the planning process by also considering the effects of smart-grid-based application resource requirements. This work makes use of three multi-objective nature-inspired

algorithms that are combined to produce a set of non-dominated solutions in a single run. The technique is adopted because it searches for the optimal solution set that takes into account all the objectives of the multi-objective optimization problem and also converges well toward Pareto-front. The proposed model addresses the issues of the planning by considering the amount and capacity of fog nodes, and cloud interconnection.

It is also important to point out that latest surveys on the fog computing research have noted very few works about fog network design and planning [10] in relation to LV network distribution control and monitoring. Specifically, the contributions of this paper can be summarized as:

- A mathematical model formulation based on multi-objective optimization to address the issues of the planning and design of fog facility placement.
- To demonstrate the application of the weighted sum method and three evolutionary algorithms, Non-dominated Sorting Genetic Algorithm (NSGA-II), Speed- Constrained Multi-Objective Particle Swarm Optimization (SMPSO), and Future Search algorithm (FSA) to solve the MOOP.
- To propose the new algorithm, namely, Future Search Particle Swarm Non-dominated Sorting Genetic Algorithm (FPNSGA) that is constituted by combining (jointing) the three metaheuristic algorithms and show that it can greatly improve the results of the MOOP efficiently.

To evaluate the proposed solutions, the model considers two sets of services in terms of computing resource demands. The first set involves services that are only able to be processed in the fog nodes, and the other set includes services that can be hosted in either fog nodes or cloud servers. The number of fog nodes and demands were varied based on the proposed budget (cost) and delay. Solutions represented in the Pareto front were selected to capture the application requirements.

Real data representing possible user and fog node locations in a geographical area within a metropolitan city were employed as input to the model. Solutions that were obtained after the evaluation of the model were compared using various metrics and benchmarks (exact models). In the end, it can be shown that the proposed model, through extensive simulations, achieves better performance and is more cost-effective.

This paper is organized as follows. Section 2 gives an overview of the background and related work on fog network planning. Section 3 provides the mathematical model formulation of the proposed fog node placement. The following section, Section 4, elaborates further on the optimization procedure which entails algorithm design and the proposed approach. Results are described in Section 5, which also includes evaluations and comparisons. Finally, Section 6, presents the conclusion and future work.

2. Background and Related Work

As pointed out in the work by [11], LV distribution networks provide the interface between the consumers and service providers and they constitute a large part of the power system. The author proposes the methods that can be used to improve the smartness of the distribution network.

Similarly, research works by [12, 13] introduce a smart-meter based method to monitor and located fault in LV network. A work by [14] also presents an LV monitoring system based on enterprise system.

In the literature, a lot of research work has been done on exploring the concept of fog computing from different perspectives and themes. To regulate and agree on common research grounds, a body, namely OpenFog Consortium has been established to influence the development of standards across the industry and academia [15].

The aspect of fog network planning has also been approached in different angles in the literature but with similar related objectives at the end which mostly are minimizing delay, cost, bandwidth and overall energy consumption [16–22]. Some scholars have addressed the problem of fog network planning from the purely communication point of view by considering the network topologies and protocols to find minimize the number of cluster head nodes [16, 23–25]. Most of these research works were evaluated based on performance metrics functional metrics and non-functional metrics listed in [26].

Placement of data centers for cloud computing is also comprehensively researched, although big companies like Google, Amazon, etc. treat their procedures as top secret. For the few available public papers, the authors in [27] compose the problem of data center placement as a linear programming model aiming at minimizing total deployment cost. They considered the maximum number of servers to be deployed and user per server ratio as inputs. Similarly, the other authors [28], proposed their fog network design based on the deployment cost, power awareness, optical link degradation factors and resource of fog devices to establish a placement baseline.

Extensive research work about facility location problem (FLP) have been done by [29, 30] in which, generally the main optimization parameters are demands and distance, although for capacitated FLPs limited facility capacities can be considered as problem constraints.

Most of the visited literature are under the assumption that there are already well-planned underlying fog networks to build their proposed solutions on top of them. Very few of these works of literature considered the issues of design and planning of fog network for field deployment, let alone the consequential issues [17]. Other researchers [18, 19] have addressed issues similar to this work, however, their proposed solutions needed further improvements with regard to solution quality and time complexity. This has been approached by proposed a new hybrid algorithm that combines three metaheuristic algorithms.

Since the fog network planning problem is considered as an NP-Hard, it is challenging to find a single solution that could properly define the characteristics of the required design plan. Depending on the formulation of the problem, such as MILP, exact approaches can be used to address the problem. However, as scalability requirements and other network parameters are taken into consideration, the problem complexity increases exponentially. For this case, it is wise to use heuristics (or metaheuristic) methods to solve the problem. Evolutionary algorithms as optimization methods have been used predominantly by authors such as [31] and [32] to minimize the number of gateways in wireless

networks. The same methods could be adapted to plan for fog networks with some little modifications.

Closely related research studies with this work include that of [18] and [19] in which they introduced the problem of fog network design by proposing mathematical models for realizing planning parameters. The authors in [18] proposed a model that could determine the optimal location, the capacity and the number of fog node(s) as well as the interconnection between the installed fog nodes and the cloud. They used the weighted sum, the hierarchical and the trade-off methods to optimize the solutions. Though their proposed approach provides the best fog network baseline, but were time insensitive and limited to small scale problems due to complexity. The authors in [19], extended the work in [18] by proposing the inclusion of a combination of two evolutionary algorithms SMPSO and NSGAI1 to improve computation time. However, the proposed hybridized algorithm leaves some solutions, particularly those that are close to Y-margin, as the problem grows in size.

In this work, the investigation is made on the fog nodes placement problem to minimize cost as well as overall network delay while meeting the application and user requirements in LV distribution networks.

3. Fog Node Placement Problem

In this work, it is envisioned that the decision over where to place data received from IoT devices depends on the cloud and fog node's location and their capacities. It is therefore important to have a mechanism to perform fog nodes planning over the given area. To properly plan for the fog nodes the following factors have to be taken into account: number of edge device clusters, number of hops, network bandwidths, storage, processing speeds, and coordinates. Characteristics such as distance between edge devices, fog nodes and cloud data centers could be derived and in that case, delay costs could also be established. In this work edge devices are also referred to as IoT devices, sensor or edge nodes and are organized as clusters.

During the planning process, it is assumed that the site is a greenfield; positions of sensor node clusters and fog nodes are expressed in terms of (x, y) coordinates; bandwidth availability and cost of network links between the cloud and fog nodes are known; fog node computing resources are known; the cloud is remotely located, and serves the sensor node clusters could not connect to the fog nodes; and a portion of fog node traffic is sent to the cloud.

3.1 Fog Nodes Planning Model

The main aim of the fog nodes planning is to determine the optimal locations (placement) of the fog node devices to minimize the total deployment cost and the delay experienced by the edge node devices and applications. The cost of deploying fog devices is related to device purchasing, site installation and maintenance costs and computation demands.

In the process of modelling it is assumed that applications for monitoring LV networks will have components running in the fog nodes and cloud.

The data from sensor clusters may be processed through the fog nodes and the cloud depending on the data type and sensitivity.

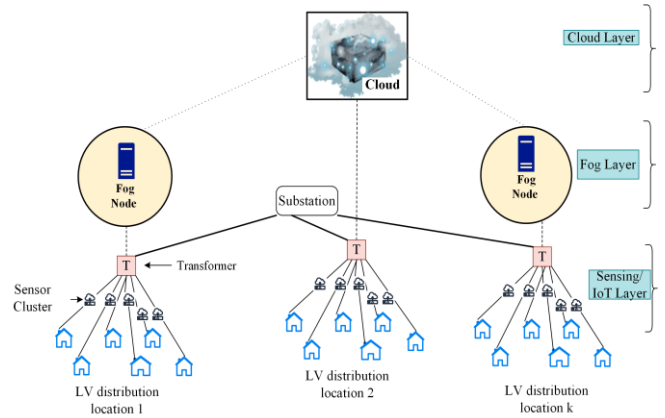


Fig. 1. Cloud-Fog Nodes Planning Notations for Modelling

Fig. 1 represents the proposed architecture of the fog-based computing system. As it can be seen, the figure illustrates the 3-layered architecture where edge devices, fog nodes and cloud are linked with each other using wired or wireless communication link. Data received from sensors is processed by fog nodes being forwarded to the cloud. The fog nodes are installed on distribution transformers [33]. An important condition that is to be noted throughout the planning process is that, once a fog node is deployed in a given area it should be able to connect to the data center.

The main objective of the fog network planning is to minimize the overall cost of deployment of fog devices in Equation (1) and the total delay in Equation (2). Based on variables described in Table 1, the total delay is obtained by summing transmission delay, propagation delay, and processing delay. The total cost is made of cost due to fog devices, network links and site cost. Therefore,

Minimize cost:

$$\left[\sum_{\rho=1}^m \sum_{k \in F} x^{\rho k} c_i^{Site} + \sum_{\rho=1}^m \sum_{k \in F} x^{\rho k} c_k^{Fog} + \sum_{\rho=1}^m \sum_{l \in l} b^{\rho l} c_l^{Link} d_{\rho, cloud} \right] \quad (1)$$

Minimize Delay:

$$\left[\sum_{\rho=1}^m \sum_{c \in C} D(d^{\rho c}) y^{\rho c} \right] \quad (2)$$

As can be seen, the problem is multi-objective optimization, and NP-hard as improvement of one objective leads to worsening of the other.

The objective functions are subject to the constraints represented by Equations (3) to (13). Equation (3) ensures that a user or application is served by only one fog node or cloud. Equation (4) enforces that there should not be more than one fog device deployed at the given area. Practically, there is a possibility of deploying several fog device at each location under consideration with different hardware or software configurations. Without loss of generality and to reduce complexity, only one fog device is deployed in the

$$\sum_{\rho=1}^m y^{\rho c} = 1 \quad (\forall c \in C) \quad (3)$$

$$\sum_{k=K}^m x^{\rho k} \leq 1 \quad (\forall \rho \in I) \quad (4)$$

$$\sum_{\ell \in I} \ell^{\rho \ell} \leq 1 \quad (\forall \rho \in I) \quad (5)$$

$$y^{\rho c} - \sum_{k \in K} x^{\rho k} \leq 0 \quad (\forall \rho \in I, \forall c \in C) \quad (6)$$

$$\sum_{\ell \in I} \ell^{\rho \ell} \leq \sum_{k \in K} x^{\rho k} \quad (\forall \rho \in I) \quad (7)$$

$$\sum_{c \in C} y^{\rho c} P_c^{K(c)} \leq \sum_{k \in K} x^{\rho k} P_F^{K(f)} \quad (\forall \rho \in I) \quad (8)$$

$$\sum_{c \in C} y^{\rho c} M_c^{K(c)} \leq \sum_{k \in K} x^{\rho k} M_F^{K(f)} \quad (\forall \rho \in I) \quad (9)$$

$$\sum_{c \in C} y^{\rho c} T_c \leq \sum_{\ell \in I} b^{\rho \ell} B^{\ell} \quad (\forall \rho \in I) \quad (10)$$

$$x^{\rho c} \in \{0,1\} \quad (\forall \rho \in I, \forall c \in C) \quad (11)$$

$$y^{\rho k} \in \{0,1\} \quad (\forall \rho \in I, \forall k \in K) \quad (12)$$

$$z^{ik} \in \{0,1\} \quad (\forall \rho \in I, \forall k \in K) \quad (13)$$

selected area. In case the left side of the equation evaluates to zero, it signifies that the area has not been considered for selection, and therefore there will be no installation of fog device.

Equation (5) exemplifies that not more than one link type can be used at selected area. Equation (6) epitomizes that only users or (applications) can use the fog node that has been opened (deployed). Equation (7) enforces that any fog device that is deployed at a given area ρ_i is interlinked with the data center. Equations (8) and (9) represent the conditions imposed on the availability of CPU and memory at the fog layer. These equations will put a check on computing resources demanded by applications or users are limited to the capacities of fog nodes. Equation (10) ensures that the total bandwidth from a fog device at a given site to the data center is within the limits of the link bandwidth. Equation (11), Equations (12) and (13) impose the decision variables as binary.

4. Methods for Optimizing Fog Network Planning

In addressing the multi-objective problem, various techniques may be employed. The most popular methods that are commonly used include Pareto and Scalarization, in which for the former, the aim is to find a set solutions that are non-dominated and update it by various iterations of the algorithm. For the case of scalarization, the idea is to merge the multi-objective into a single objective by assigning weights to the objectives using methods such as equal weights, rank order centroid weights and rank-sum weights [34].

4.1 Exact Method

4.1.1 Weighted Sum Optimization

This optimization method casts the multi-objective functions as a single-objective mathematical optimization

model by assigning weight values between 0 and 1 to the objectives as shown in the equations (14) to (16).

$$\min \max \left\{ \sum_{i=1}^M w_i f^i(x) : x \in X \right\} \quad (14)$$

$$0 \leq w_i \leq 1 \quad (15)$$

$$\sum_{i=1}^M w_i = 1 \quad (16)$$

where w_i is the weight coefficient that is applied to each objective for computing its relative importance. The single-objective function can now be written as:

$$\min \left\{ \chi_1 Cost_{total}^{norm} + \chi_2 Delay_{total}^{norm} \right\} \quad (17)$$

where χ_1 and χ_2 are weighted coefficients and, $Cost_{total}^{norm}$ and $Delay_{total}^{norm}$ are normalized objective functions. The normalized functions have been obtained through Equations (18) and (19).

$$Cost_{total}^{norm} = \frac{Cost_{total}^{max} - Cost_{total}}{Cost_{total}^{max} - Cost_{total}^{min}} \quad (18)$$

$$Delay_{total}^{norm} = \frac{Delay_{total}^{max} - Delay_{total}}{Delay_{total}^{max} - Delay_{total}^{min}} \quad (19)$$

The weight values represented by the coefficients gauge the preference of each objective value. Assignment of proper weight values to the objective functions is not trivial regardless of the one's expertise.

For solving the fog network planning problem using this approach, firstly inputs are generated, by initializing the weight values of χ_1 and χ_2 by 1 and 0 respectively. This is because cost and delay have different units and scales, and therefore, to evaluate them in the comparable range, normalization is needed to avoid one variable (cost) outweigh the other (delay). Then the problem is solved by obtaining the optimal solutions to form the Pareto optimal set in step of 0.1 for each phase until stopping criteria is met.

4.2 Approximate Methods

For higher dimension problems, exact methods take a longer time to compute the optimal solutions, which is infeasible in production settings. Besides, in the case of concave Pareto fronts, exact methods may tend to select only extremal solutions, where solutions are optimal in one of the objectives [35]. Most of the approximate methods that are widely adopted for addressing multi-objective optimization problems are evolutionary algorithms (EA), which nature-inspired and use natural evolution to steer a population towards optimal or near-optimal solutions. With evolutionary algorithms, multiple Pareto optimal solutions can be found within a single simulation. Three EAs have been adopted in this work (Future Search, NSGA-II and SMPSO) as they exhibit different patterns about coverage and convergence. The behaviors of these algorithms are then used to propose a method that combines these three as a single hybrid algorithm, namely Future Search Particle Swarm Non-dominated Sorting Genetic Algorithm (FPNSGA).

4.2.1 Future Search Algorithm

Future Search Algorithm (FSA) is the newest optimization algorithm proposed by Elsis [36], in which, it works by mimicking how people adapt to challenges as they look the best life. Anyone looking for improved life will look to follow the steps made by the successful ones. Features that make this new algorithm distinguishable are: it avoids tuned parameters, it has low computation complexity, quick convergence and high local optimal avoidance. Results from the benchmark functions and comparisons made with other well-known metaheuristic algorithms proved the superiority of the algorithm [36][37].

$$S(i,:) = LB + (UB - LB) \times rand(1, D) \quad (20)$$

$$S(i,:)_L = [LS(i,:) - S(i,:)] \times rand \quad (21)$$

$$S(i,:)_G = [GS - S(i,:)] \times rand \quad (22)$$

$$S(i,:) = S(i,:) + S(i,:)_L + S(i,:)_G \quad (23)$$

$$S(i,:) = GS + [GS - LS(i,:)] \times rand \quad (24)$$

The operating principle of FSA is to utilize both local and global search mechanisms for updating the randomized initial population to obtain the optimal solutions based on simple mathematical formulations. The steps in using the FSA for solving the fog network planning problem are illustrated in Fig. 2. The algorithm starts by initializing the solution based on Equation (20), in which S is a solution, i is the pointer to the current solution, LB and UB are lower and upper bounds of the optimized parameters respectively, D is the dimension of the problem and $rand$ is the function that generates uniformly distributed pseudo-random numbers.

Then each solution is treated as a local best solution (LS), and the best overall is recorded as the global best solution (GS). The search in a country or state depends on LS which supports the exploitation characteristic of the algorithm, and therefore the local convergences are determined based on Equation (21). Similarly, the search in the countries of the world depends on the GS , which characterizes the exploration. Further, in the next step, the algorithm updates LS and GS , which are used along with the newfound solutions in the behavior of the algorithm, and it is established in Equation (22) as global convergences. Hence, the individual solution is then determined by Equation (23) using the current iteration to update the random initial in Equation (20) as shown in Equation (24). In each iteration, the algorithm checks the status of LS and GS after updating initial random and updates them if they are better than ones in the main loop of the algorithm.

It should be emphasized here that, the original FSA algorithm was benchmarked based on a single-objective optimization problem. To adapt it for the multi-objective optimization problem, the straightforward approach would have been to convert the problem into a single-objective optimization problem and then use the weighted sum approach. However, as discussed in Section 4.1.1, this approach has some drawbacks, therefore, the authors of this

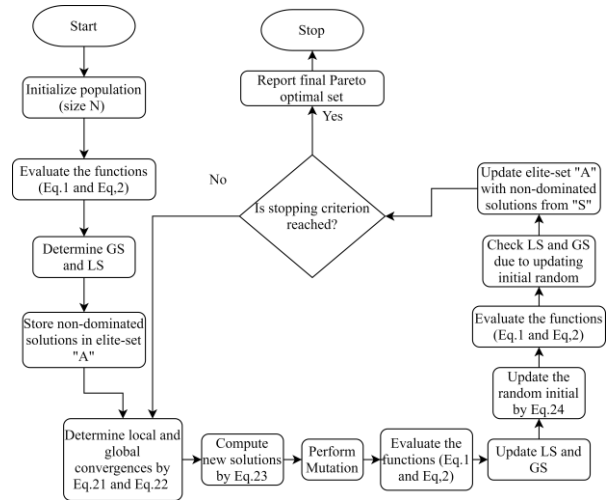


Fig. 2. Flow chart for FSA for fog network planning

manuscript introduced the concepts of capturing the non-dominated solutions and mutation for maintaining diversity to fit into a multi-objective context.

To the best of the authors' knowledge, this is the first time that this algorithm is adapted for a multi-objective optimization problem in this manner, as now it is easier to compare with other similar evolutionary algorithms on the same grounds.

From the preliminary results, FSA performed better when it comes to speed and convergence to the Pareto front. Besides, it covers the points close to the Y-axis which are missed by NSGA-II and SMPSO as problem size increases. However, the algorithm suffers from poor coverage of optimal solutions as one moves away along the X-axis.

4.2.2 NSGA-II

NSGA-II is among the most effective EAs for addressing multi-objective optimization problems [38]. It uses the ranking procedure which is useful in preserving the best non-dominated solutions in the latest solutions.

The steps to solve the fog network planning problem using NSGA-II are summarized in Fig. 3. Firstly, a population is given initial values and then evaluation is performed using the objective functions specified in Equation (1) and Equation (2). The obtained solutions are ranked in the next step, and then undergo the processes of selection, crossover and mutation to generate offspring population. In the next step, parents and generated children are combined, evaluated and ranked again. The steps are repeated until a stopping condition is met, in which a final Pareto front is obtained.

From the preliminary results it could be observed that NSGA-II performed better with regard to convergence to the Pareto Front, but in the case of coverage, it unevenly distributed the solution points. This behavior was also echoed by authors in [39], who proposed elitism strategy to improve the situation, but their proposed solution comes with high time-complexity which is undesirable for large-scale problems.

4.2.3 SMPSO

SMPSO, proposed by Nebro et al. [40] is one of the improved versions of the Particle Swarm Optimization (PSO) algorithm adapted for multi-objective problems with salient features for convergence speed towards Pareto front and competitive solution quality.

SMPSO uses a randomly-generated set of N populations, in which the positions of populations are iteratively updated in the search space towards local and ideal global best position by adjusting the velocity of the population members.

The steps for using SMPSO in solving the fog network planning problem are summarized in Fig. 4. It starts by randomly generating the solution particles in accordance with the problem dimensions, and assigns to each particle its initial velocity and position. The particles are then evaluated based on the objective functions and the resulting non-dominated solution particles stored in the elite set.

The next stage, the algorithm finds new speed of the particles and the corresponding new positions as they move through the search space. In each iteration, the elite-particle-set is updated with the newfound non-dominated solutions, if any, until a stopping criterion is met, in which the final set is obtained. The experience from the results of the preliminary experiments suggests that SMPSO is very effective in exploring the solution search space and preserving the diversity of the Pareto front. Just like NSGA-II, SMPSO was found to struggle to converge to the true optimal front as the problem size and number of iterations increase.

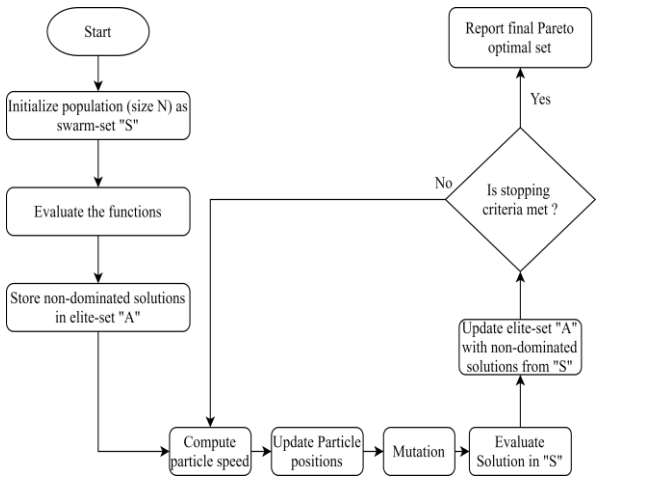


Fig. 3: Flowchart for the NSGA-II algorithm

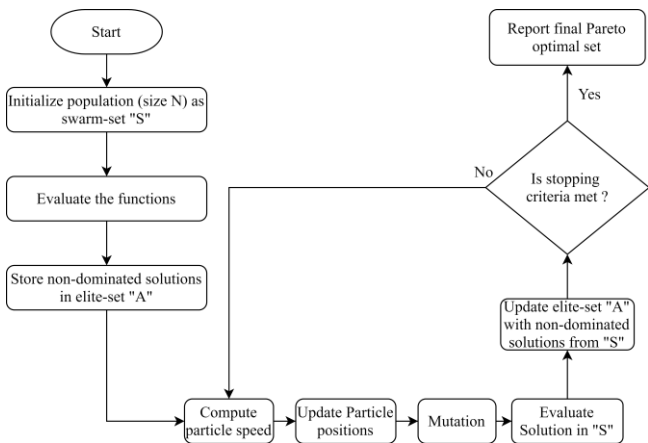


Fig. 4: Flowchart for SMPSO algorithm

4.3. Proposed Evolutionary Algorithm: (FPNSGA)

This algorithm is proposed following the drawbacks observed in FSA, NSGA-II and SMPSO in solving the optimization the fog network planning as a multi-objective problem. The new proposed evolutionary algorithm proposed is named Future Search Particle and Swarm Optimized Non-dominated Sorting Genetic algorithm. The proposed FPNSGA brings together strong features of the three evolutionary algorithms (FSA, NSGA-II and SMPSO) to increase the search efficiency. Specifically, as shown in Algorithm 1, it consists of three phases, in which, in the first phase FSA is introduced to perform initial processing of the Pareto set. In the next phase, SMPSO is used to extend in terms of coverage the obtained new Pareto set, the third phase, the other algorithm, NSGA-II reinforces the convergence of the Pareto set.

With the FSA stage, the exploration of solution space is done in order to avoid stacking early in local optima as shown in rows 6 to 8. The computation of the new solutions is equivalent to updating the individual solutions toward the best solution which improves the speed at which the solutions converge. The introduction of mutation in line 10, helps to preserve the diversity of the solutions, which is useful in the next phase.

At the second stage, the SMPSO algorithm is applied to further explore the solution space by extending the coverage of the solution points that were either missed or avoided in the first phase. SMPSO mechanism is useful in ensuring that the searching process converges fast and preserving diversity through the use of polynomial mutation as a turbulence factor (line 27).

In this third stage, which involves the use of NSGA-II, the aim is to distribute evenly the obtained solution set, and then move them closer to the true Pareto front (line 33-46). The ranking technique used in NSGA-II for non-dominated solutions facilitates the coverage of the solution space. This phase also makes use of the aggressive selection process to impose the fitness pressure on the solutions [19].

Algorithm 1: FPNSGA algorithm

- 1 Set N population as S
- 2 Evaluate the objective functions
- 3 Determine solutions that are the non-dominated (ND) in S
- 4 Place obtained ND solutions into archive E
- 5 Loop1 (FSA stage):
- 6 Determine global solution (GS) and local solution (LS)
- 7 while iteration < first phase max iterations do
- 8 Determine local and global convergences
- 9 Compute new solutions
- 10 Use polynomial mutation (PM) to 6% of the population
- 11 Determine the objective functions
- 12 Update GS and LS
- 13 Update the random initial
- 14 Evaluate the objective functions
- 15 Test the GS and LS of the updating initial
- 16 Update the archive E with ND solutions
- 17 End while
- 18 Return An archive E of non-dominated solutions
- 19 Put obtained ND solutions in U from FSA's archive E
- 20 V ← U (Use U as initial archive for leaders SMPSO)
- 21 Loop2 (SMPSO stage):
- 22 while iterations ≤ second stage iteration do
- 23 Apply constrained binary tournament to choose a solution in V
- 24 Make this solution as the global best particle
- 25 Calculate the new speed of particle s.
- 26 Calculate the new the position of particle s
- 27 Perform Mutation

```

28 Perform evaluation
29 Update leaders archive B:
30 Update particles memory
31 End while
Return Leaders archive B
32 Assign to K as the initial ranked individual from SMPSO's
leaders archive B
33 P ← K (Set K as initial population for NSGA-II)
34 Loop3(NSGA-II stage)
35 while iterations ≤ third stage iteration do
36     D ← Initial child population as empty
37     Use constrained binary tournament to select parents in P.
38     while not enough individuals in P do
39         Reproduce a population of offspring
40         Use Binary tournament selection
41         Apply crossover and mutation
42         Calculate the objective values of the offspring
43         Use Aggressive Selection Process
44         Merge the offspring into D
45         Determine the non-dominated-sorting over population P
and offsprings population D.
46         Select new parents based on ranks and crowding distance
47 return Pareto Optimal solutions in population P
48 End while

```

NSGA-II algorithm involves the following main operations that constitutes different time complexities: Non-dominated sorting impose a time complexity of $O(M(2N)^2)$, Crowding distance calculation introduced a time complexity of $O(M(2N)\log(2N))$ and Sorting and selection contributed a time complexity of $O(2N)\log(2N)$. M represents the number of objective functions, N stands for population size and K signifies the total number of iterations. Therefore, the entire time complexity is (KMN^2) .

SMPSO algorithm includes three parts that affect the time complexity: Particle speed computation $O(N)$, Application of polynomial mutation $O(\tau\%N)$ and Elite set polling $O(MN^2)$, whereby $\tau\%$ represents the proportional of solution mutation. Therefore, in K iterations, the time complexity of SMPSO is $O(KMN^2)$.

FSA also consists of the following parts that contribute to the overall time complexity of the algorithm: Computation of local and global convergences $O(N)$, Application of polynomial mutation $O(\tau\%N)$, Polling of Non-dominated solution set $O(MN^2)$. For the case of the FPNSGA algorithm, if there is a maximum of K iterations is specified and since it consists of three optimization phases (FSA, SMPSO and NSGA-II), then the overall complexity of the FPNSGA algorithm is $O(KMN^2)$.

5. Numerical Results and Analysis

5.1 Experiment Setup and Input:

To demonstrate the performance of the proposed method, it is used for fog network planning. It is assumed that a multitude of voltage and current sensors and actuators (circuit breakers/closers) are deployed along transformers and distribution lines leading to customer premises in LV distribution networks in a selected geographical area. Data received from the sensors is used to monitor electrical faults

related to current and voltages. In the cloud-based setup, the sensors send data directly to the utility control data center for analysis. Similarly, actuators can receive commands from the data center. At the data center, the received data might go through several layers of processing such as pre-processing, sizing, extraction, etc.

However, owing to the centralized nature and the distance between the data center and sensors, there is a greater chance that some applications and data may suffer from high latencies and communication bandwidth. Therefore it is best to leverage the fog computing infrastructure so that some of the data processing tasks can take place closer to the source or edges of the communication networks. Further, it is reasonable not to have all tasks be processed in a single computing device due to the resource constraints, and therefore a distributed deployment is needed where multiple fog computing devices share the load of tasks processing.

From the utility's best practices (in Tanzania), customers get connections through nearby service lines that are carried through utility poles originating from the distribution transformers. The locations of the utility poles supplying power to the customers correspond to customer premises. Therefore, it is wise to assume that each customer premise location may also act as a possible sensor cluster or actuator location. Based on the data collected from the site and supplied by the local utility company, it is assumed that an edge cluster should consist of a group of collocated customers with a shared unique IP address.

The potential locations for placing fog nodes correspond to the locations of transformers in the selected area. The assumption is that practically all the service lines which carry sensor/edge clusters for measuring electrical parameters are originating around the transformers to the customers. In this work, it is required to pick the locations for placing fog computing devices out of listed distribution transformer locations.

Each sensor cluster has computing resource demands in terms of the number of CPUs, amount of memory and network bandwidths as shown in Table 2. Fog node capacities are represented in Table 3.

In this work, it is assumed that these resource requirements follow uniform distribution and these should be fulfilled by fog nodes. Each sensor cluster has a geographical coordinate expressed in Cartesian coordinate (x, y) which corresponds to the customer locations. The distance from the sensor cluster to the fog device is used to determine the propagation delay.

5.2 Experiment Environment

For the exact method which used the weighted sum approach, CPLEX 12.10.0 solver was used to optimize the problem. All parameters of the solver were left in their default values except for a time limit which was set to be 1.5 hours. This is to ensure that, if a solution is not found within that limit, then CPLEX would return any solution obtained at that point in time.

All the experiments were conducted on a HP computer with the following specifications: Intel Quadcore i7 Processor at 1.8 GHz, 16 GB of Physical Memory. The simulations were developed through Java programming

language using Java 14 Platform, Eclipse 2020-06 and jMetal Framework [41].

24 problems were defined as shown in Table 5 ranging from small size problem (problem 1 to 8), medium-size problem (problem 9 to 18) and big size problem (problem 19 to 24). The first column represents the problem ID, the second column represents the number of sensor clusters and the last column stands for the number of potential locations for deploying fog nodes.

To evaluate the algorithms, problems with the similar parameters were used for each. Selected multiple instances of the problems were run several times and average values were taken for comparisons.

For a fair comparison between algorithms, the following set of parameter settings were chosen as shown in Table 3. The number of iterations was set to 20,000 for each algorithm.

Table 2: Sensor Cluster Resource Demands

Size of the area	10 x 10 km sq.
Coordinates of the sensor cluster	(x, y)
Number of sensor cluster	(10-200)
Number of CPUs	(1-4)
Amount of Memory	(1-40) GB
Number of Packets sent per second	(1-64)
Network access bandwidth	(20 – 70) Mbps

Table 3 : Fog Node Resource Availability

Fog Type	Number of CPUs	Memory (GB)	NIC Capacity (Mbps)	Cost
1	90	480	360	67200
2	180	800	1024	120000
3	360	1600	1024	170000
4	720	3200	10240	250000

Table 4: Parameterization (D is individual length)

Parameterization used in NSGA-II	
Population Size	100 individuals
Selection of Parents	binary tournament
Crossover	Simulated binary, p=0.9
Mutation	Polynomial, p=1.0/D
Parameterization used in SMPSO	
Swarm Size	100 particles
Mutation	Polynomial, p=1.0/D
Archive Size	100 individuals
Parameterization used in Future Search	
Population Size	100 individuals
Mutation	Polynomial, p=1.0/D
Archive Size	100 individuals

Table 5 : Problem Definition

Problem ID	Number of sensor cluster	Numer of possible locations
1	10	5
2	15	5
3	20	5
4	25	5
5	30	5
6	35	5
7	40	5

8	45	5
9	50	5
10	55	5
11	60	5
12	65	5
13	70	10
14	75	10
15	80	10
16	85	10
17	90	10
18	95	10
19	100	10
20	105	10
21	120	10
22	135	10
23	150	10
24	200	10

5.3 Results Analysis

This section presents the results related to the evaluation of the proposed algorithms.

5.3.1 Weighted Sum Approach

With the Weighted Sum approach, the idea is to combine the objective functions into a single objective function using weight coefficients. The results shown here are based on coefficient values of 0.2 and 0.8 for cost and delay functions respectively. FSA was able to post the lowest values of the objective function for the small size problem (15 cluster sensors and 5 fog nodes) for simulation numbers 9, 10, 20 and 25. However, the FSA algorithm could not find feasible solutions in simulation numbers 6 and 19. NSGA-II and SMPSO were much more robust and consistent throughout simulation runs. Fig. 5 depicts the comparison of the SMPSO, NSGA-II and FSA in terms of objective function values through simulations. Although FSA can find the lowest values, it can also record the highest values of objectives as is the case in simulations 3 and 19. NSGA-II constitutes the lowest average values of the objective functions followed by SMPSO, and then FSA.

Further, Fig. 6 presents the result of the comparison between FSA, SMPSO and NSGA-II in solving medium size problem (90 Sensor clusters on 10 fog devices), in which it can be noticed that FSA is superior to other algorithms in finding the smallest value of the objective function. Besides, as observed in solving of small size problem, out of 30 simulations that were ran for each algorithms, NSGA-II could not find the feasible values most of the times, with a probability of 0.433 compared with FSA (0.233) and SMPSO (0.0). This justifies the inclusion of FSA in the proposed algorithm as a result of its quick convergence.

5.3.2 Diversification and Convergence Comparisons

The performance of the proposed algorithm is compared with another algorithm proposed by [19], namely PSONSGA (Particle Swarm Optimized Non-dominated Sorting Genetic Algorithm). The comparisons in terms of convergence and divergence are shown in Fig. 7 and Fig. 9 to Fig. 12. Based on simulations, as the problem space and dimension grow bigger PSONSGA struggles to find low value solutions, at the left-hand top corner, attested by Fig. 10.

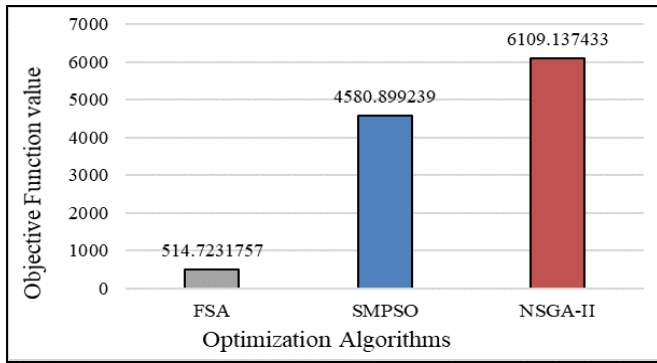


Fig. 5: Weighted Sum approach for problem (90 sensor clusters on 10 fog devices)

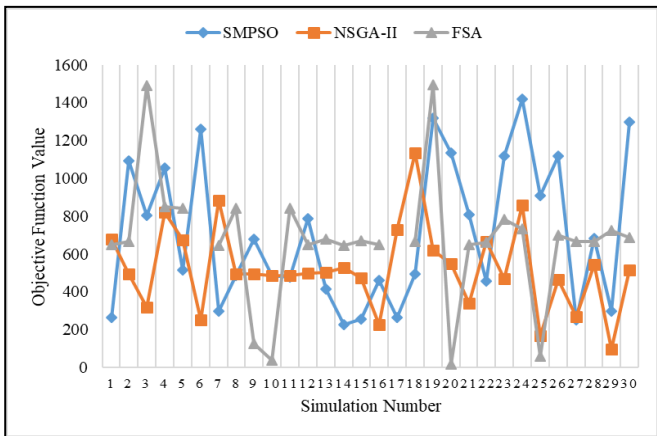


Fig. 6: Weighted Sum Approach for small size problem (15 sensor nodes on 5 fog devices)

The proposed method clearly offer the better convergence compared to PSONSGA with fewer iterations. The introduction of the FSA in the proposed FPNSGA, have added the exploration capability significantly. For convergence, the Future Search algorithm can guide the solution points toward the Pareto Front as it is generally designed to avoid local optima.

5.3.3 Test for HyperVolume indicator

HyperVolume (HV) indicator [42] is one of the metrics that is used to measure the searching performance of multi-objective algorithms. For minimization problems, a HyperVolume that returns a higher value provides the indication that the solutions have spread better and well converged [43].

HV results of the 24 problems for SMPSO, FSA, PSONSGA and FPNSGA algorithms are shown in Fig. 13. During the evaluation of algorithms through HV indicator, FPNSGA algorithm offered better performance in terms of finding quality solutions as it can be revealed by the 29% that returned best results from the problems compared to SMPSO (21%), Future Search (12.5%), PSONSGA (12.5%) and NSGA-II(4%).

It could be noted that in most problems NSGA-II performed the worst, which is attributed to the fact that it concentrated more on non-dominated sorting for convergence.

5.3.4 Comparisons based on the number of fog nodes

Figure Fig. 9 illustrates the comparisons of FPNSGA and PSONSGA based on the number of fog nodes on 100 users. The performances of the algorithms are slightly comparable for the case of 1 and 5 fog nodes. FPNSGA is far better in solution diversification as the number of fog nodes increases. This is attributed to the strong features of FSA.

5.3.5 CPU Time Comparisons

Comparisons in terms of CPU time usage is given in Fig. 10 in which the overall average time an algorithm spends solving a problem is presented. As it can be seen, for problems 1 to 12, algorithms spend a fairly equal amount of CPU time in its category, however, NSGA-II, SMPSO, PSONSGA impose a slight jump in the use of CPU time from problems 13 to 24. Overall, FPNSGA constitutes the largest share of CPU time consumption, followed by Future Search algorithms. For instance, in solving problem number 20, FPNSGA uses 116473ms which is 8 times more than that of SMPSO, and 2.7 times longer than PSONSGA. Overall, there is a linear increase in solution time as problem size increases.

6. Conclusion

In this work, the concepts and processes of fog network planning are discussed with regard to deployment for smart grid initiatives. The design of the fog network is considered with goals to minimize overall deployment costs and delays in which a mathematical model is presented.

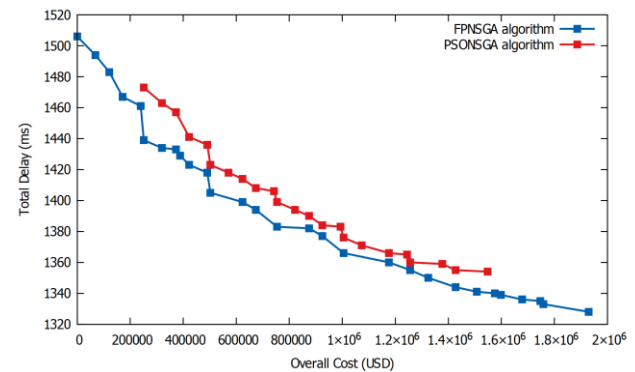


Fig. 7: FPNSGA and PSONSGA based on the medium sized problem (95 sensor clusters on 10 fog nodes)

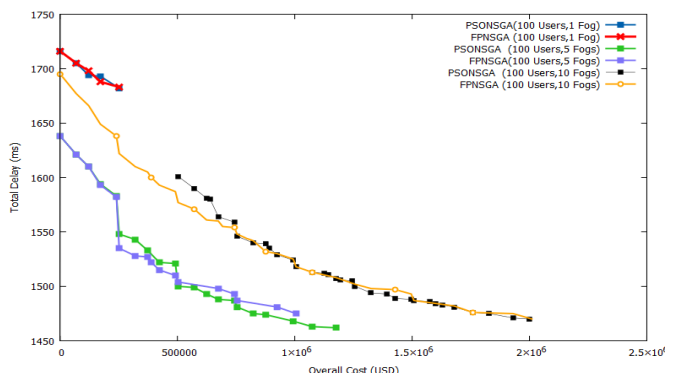


Fig. 8: Comparisons based on the impact of number of fog nodes on the delay and cost

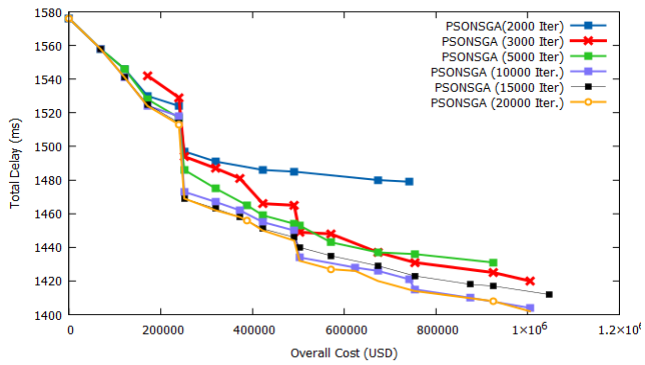


Fig. 9 : Convergence curve for PSONSGA (100 sensor clusters, 5 Fog nodes)

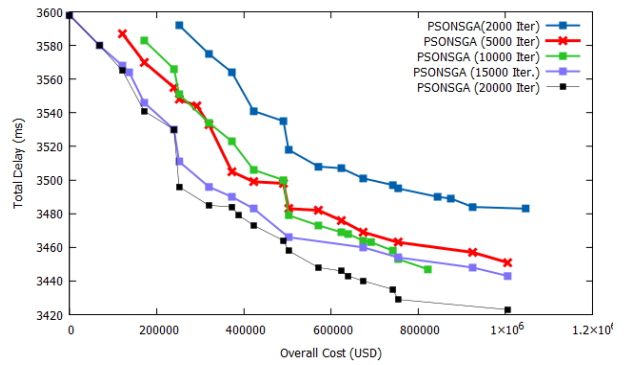


Fig. 10 : Convergence curve for PSONSGA (200 sensor clusters, 5 Fog nodes)

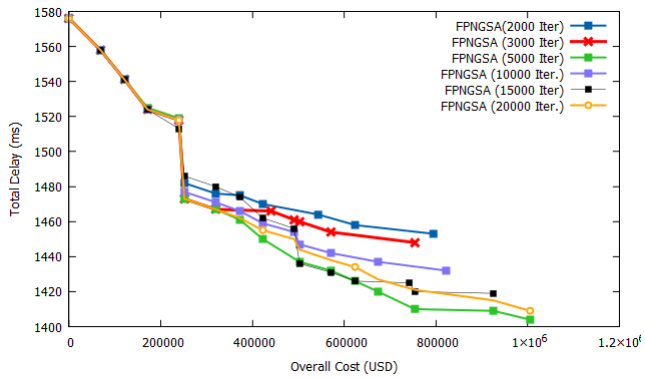


Fig. 11 : Convergence curve for FPNSGA (100 sensor clusters, 5 Fog nodes)

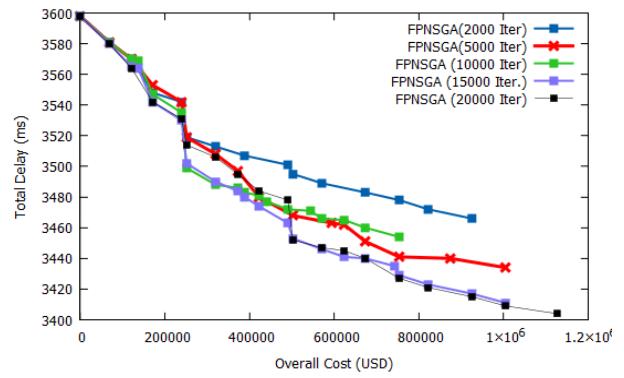


Fig. 12 : Convergence curve for FPNSGA (200 sensor clusters, 5 Fog nodes)

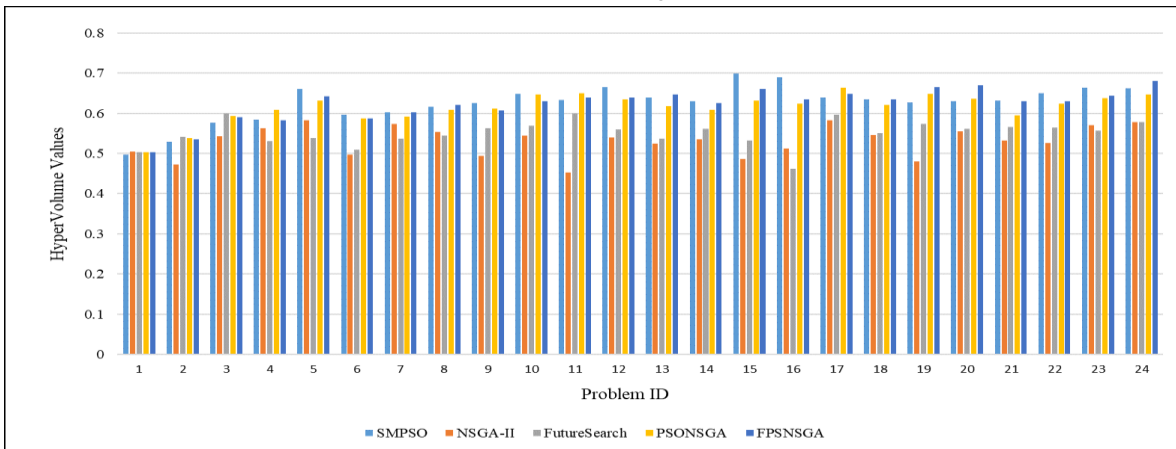


Fig. 13: HyperVolume Comparisons

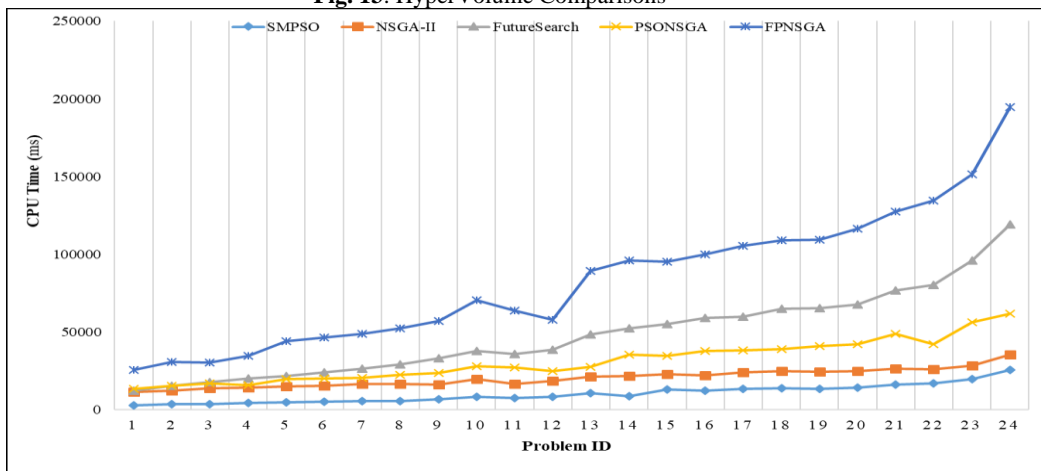


Fig. 14 : CPU Time Comparisons

The model is evaluated based on three evolutionary algorithms NSGA-II, SMPSO and FSA, and a proposed method (FPSNSGA) that combines all of them. The planning problem, which is NP-hard, required to find optimal locations for installing fog nodes in LV distribution networks for addressing the computing needs.

The evolutionary algorithms that were tested returned mixed performances with each performing better in some aspects and weaker in other cases as evidenced by HV indicators, CPU time and diversification and convergence comparisons. From experiments it could be revealed that, the multi-objective-based fog network planning problem can be addressed thoroughly by combining the best features of SMPSO, NSGA-II and Future Search algorithms into a single algorithm. The new formed algorithm has shown to outperform other methods by more than 8% in terms convergence and proved to be useful in practical cases where solutions diversity and convergence are of the highest priority with little sacrifice in computation time.

Since fog network planning is an NP-hard problem in which the problem complexity increases as problem size gets bigger, still viable multi-objective algorithms can improve it further close to true Pareto solutions. In that regard, more factors such as mobility and application workloads can be added to the problem and then with the introduction of parallel features, the processing time for the solution can be improved and reflect reality.

Acknowledgments

Authors acknowledge SIDA (Swedish International Development Agency) through the University of Dar es Salaam iGRID project for supporting this work.

References

- [1] G.M. Gilbert, S. Naiman, H. Kimaro, N. Mvungi, "A Cloud-Fog Based System Architecture for Enhancing Fault Detection in Electrical Secondary Distribution Network," In: Pandian, A.P., Palanisamy, R., and Ntalianis, K. (eds.) Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCB - 2019), pp. 845–855, Springer International Publishing, Cham (2020), https://doi.org/10.1007/978-3-030-43192-1_92.
- [2] Y. Andegelile, N. Mvungi, M. Kissaka, "SDN Based High Availability Communication Network Architecture for Secondary Distribution Electric Power Grid," In: 7th International Conference on Smart Grid, icSmartGrid 2019, pp. 52–57, IEEE (2019), <https://doi.org/10.1109/icSmartGrid48354.2019.8990828>.
- [3] F. Ayadi, I. Colak, R. Bayindir: "Interoperability in Smart Grid," In: 7th International Conference on Smart Grid, icSmartGrid 2019, pp. 165–169, IEEE (2019), <https://doi.org/10.1109/icSmartGrid48354.2019.8990680>.
- [4] D. Mnyanghwal, S. Kawambwa, R. Mwifunyi, G.M. Gilbert, D. Makota, N. Mvungi, "Fault Detection and Monitoring in Secondary Electric Distribution Network Based on Distributed Processing," In: 2018 20th International Middle East Power Systems Conference, MEPCON 2018 - Proceedings, pp. 84–89 (2019), <https://doi.org/10.1109/MEPCON.2018.8635141>.
- [5] I. Colak, R. Bayindir, S. Sagioglu, "The Effects of the Smart Grid System on the National Grids," In: 8th International Conference on Smart Grid, icSmartGrid 2020, pp. 122–126, IEEE (2020), <https://doi.org/10.1109/icSmartGrid49881.2020.9144891>.
- [6] H. Daki, A. El Hannani, A. Aqqal, A. Haidine, A. Dahbi, "Big Data management in smart grid: concepts, requirements and implementation," J. Big Data. 4, 1–19 (2017), <https://doi.org/10.1186/s40537-017-0070-y>.
- [7] S. Bera, S. Misra, J.J.P.C. Rodrigues, "Cloud Computing Applications for Smart Grid: A Survey," IEEE Trans. Parallel Distrib. Syst, 26, 1477–1494 (2015), <https://doi.org/10.1109/TPDS.2014.2321378>.
- [8] Y. Allahvirdizadeh, M.P. Moghaddam, H. Shayanfar, "A survey on cloud computing in energy management of the smart grids," Int. Trans. Electr. Energy Syst, 29, (2019), <https://doi.org/10.1002/2050-7038.12094>.
- [9] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog computing and its role in the internet of things," In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12, p. 13, ACM, NY, USA (2012), <https://doi.org/10.1145/2342509.2342513>.
- [10] A. Yousefpour ... J.P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," J. Syst. Archit, 98, 289–330 (2019), <https://doi.org/10.1016/j.sysarc.2019.02.009>.
- [11] S.N. Saxena, "Smart Distribution Grid – and How to Reach the Goal," Int. J. Smart Grid - ijSmartGrid, 3, 188–200 (2019).
- [12] A. Estebasari, E. Pons, E. Bompard, A. Bahmanyar, S. Jamali, "An improved fault location method for distribution networks exploiting emerging LV smart meters," In: EESMS 2016 - 2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems, Proceedings. pp. 1–6, IEEE (2016), <https://doi.org/10.1109/EESMS.2016.7504815>.
- [13] S. Jamali, A. Bahmanyar, E. Bompard, "Fault location method for distribution networks using smart meters," Meas. J. Int. Meas. Confed, 102, 150–157 (2017), <https://doi.org/10.1016/j.measurement.2017.02.008>.
- [14] A. Stanimirović, M. Bogdanović, M. Frtunić, L. Stoimenov, "Low-voltage electricity network monitoring system: Design and production experience.," Int. J. Distrib. Sens. Networks, 16, 155014772090362 (2020), <https://doi.org/10.1177/1550147720903629>.
- [15] OpenFog Consortium Architecture Working Group, "OpenFog Architecture Overview," Open Fog Consortium, 1–35 (2016).
- [16] E. Balevi, R.D. Gitlin, "Optimizing the Number of Fog

- Nodes for Cloud-Fog-Thing Networks,” *IEEE Access*, 6, 11173–11183 (2018), <https://doi.org/10.1109/ACCESS.2018.2808598>.
- [17] Z. He, Y. Zhang, B. Tak, L. Peng, “Green Fog Planning for Optimal Internet-of-Thing Task Scheduling,” *IEEE Access*, 8, 1224–1234 (2020). <https://doi.org/10.1109/ACCESS.2019.2961952>.
- [18] F. Haider, D. Zhang, M. St-Hilaire, C. Makaya, “On the Planning and Design Problem of Fog Computing Networks” (2018).
- [19] D. Zhang, F. Haider, M. St-Hilaire, C. Makaya, “Model and Algorithms for the Planning of Fog Computing Networks,” *IEEE Internet Things J*, 6, 3873–3884 (2019). <https://doi.org/10.1109/JIOT.2019.2892940>.
- [20] A. Yousefpour ... J.P. Jue, “FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework,” *IEEE Internet Things J*, 6, 5080–5096 (2019), <https://doi.org/10.1109/JIOT.2019.2896311>.
- [21] G. Premsankar, B. Ghaddar, M. Di Francesco, R. Verago, “Efficient Placement of Edge Computing Devices for Vehicular Applications in Smart Cities,” *NOMS 2018 - 2018 IEEE/IFIP Netw. Oper. Manag. Symp*, 1–9.
- [22] A. Santoyo-González, C. Cervelló-Pastor, “Latency-aware cost optimization of the service infrastructure placement in 5G networks,” *J. Netw. Comput. Appl.* 114, 29–37 (2018), <https://doi.org/10.1016/j.jnca.2018.04.007>.
- [23] I. Gravalos, P. Makris, K. Christodoulopoulos, E.A. Varvarigos, “Efficient Network Planning for Internet of Things with QoS Constraints,” *IEEE Internet Things J*, 5, 3823–3836 (2018), <https://doi.org/10.1109/JIOT.2018.2849327>.
- [24] A. Santoyo-Gonzalez, C. Cervello-Pastor, “Edge Nodes Infrastructure Placement Parameters for 5G Networks,” *2018 IEEE Conf. Stand. Commun. Networking, CSCN 2018*, 1–6 (2018), <https://doi.org/10.1109/CSCN.2018.8581749>.
- [25] S. Mondal, G. Das, E. Wong, “An Analytical Cost-Optimal Cloudlet Placement Framework over Fiber-Wireless Networks with Quasi-Convex Latency Constraint,” *Electronics*, 8, 404 (2019), <https://doi.org/10.3390/electronics8040404>.
- [26] ETSI, “ETSI GS MEC-IEG 006; Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines,” *Etsi*, 1, 1–29 (2017).
- [27] I. Goiri, K. Le, J. Guitart, J. Torres, R. Bianchini, “Intelligent Placement of Datacenters for Internet Services,” In: *2011 31st International Conference on Distributed Computing Systems*, pp. 131–142, *IEEE* (2011), <https://doi.org/10.1109/ICDCS.2011.19>.
- [28] W. Zhang, B. Lin, Q. Yin, T. Zhao, “Infrastructure deployment and optimization of fog network based on MicroDC and LRPON integration,” *Peer-to-Peer Netw. Appl.* 10, 579–591 (2017), <https://doi.org/10.1007/s12083-016-0476-x>.
- [29] D. Celik Turkoglu, M. Erol Genevois, “A comparative survey of service facility location problems,” *Ann. Oper. Res.* 292, 399–468 (2020), <https://doi.org/10.1007/s10479-019-03385-x>.
- [30] M. Saleh, Y. Esa, N. Onuorah, A.A. Mohamed, “Optimal microgrids placement in electric distribution systems using complex network framework,” In: *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 1036–1040, *IEEE* (2017), <https://doi.org/10.1109/icrera.2017.8191215>.
- [31] T. Vanhatupa, M. Hannikainen, T.D. Hamalainen, “Genetic Algorithm to Optimize Node Placement and Configuration for WLAN Planning,” In: *2007 4th International Symposium on Wireless Communication Systems*, pp. 612–616, *IEEE* (2007), <https://doi.org/10.1109/ISWCS.2007.4392413>.
- [32] S.C. Ho, “An iterated tabu search heuristic for the Single Source Capacitated Facility Location Problem,” *Appl. Soft Comput.* 27, 169–178 (2015), <https://doi.org/10.1016/j.asoc.2014.11.004>.
- [33] G. Chugulu, F. Simba, S. Lujara, “Proposed Practical Communication Architecture for Automatic Fault Detection and Clearance in Secondary Distribution Power Network,” *Int. J. Smart Grid - ijSmartGrid*, 4, (2020).
- [34] N. Gunantara, “A review of multi-objective optimization: Methods and its applications,” *Cogent Eng.* 5, 1–16 (2018), <https://doi.org/10.1080/23311916.2018.1502242>.
- [35] M.T.M. Emmerich, A.H. Deutz, “A tutorial on multiobjective optimization: fundamentals and evolutionary methods,” *Nat. Comput.* 17, 585–609 (2018), <https://doi.org/10.1007/s11047-018-9685-y>.
- [36] M. Elsis, “Future search algorithm for optimization,” *Evol. Intell.* 12, 21–31 (2019), <https://doi.org/10.1007/s12065-018-0172-2>.
- [37] M. Elsis, M. Soliman, “Optimal design of robust resilient automatic voltage regulators,” *ISA Trans.* (2020), <https://doi.org/10.1016/j.isatra.2020.09.003>.
- [38] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.* 6, 182–197 (2002), <https://doi.org/10.1109/4235.996017>.
- [39] J. OuYang, F. Yang, S.W. Yang, Z.P. Nie, “The improved NSGA-II approach,” *J. Electromagn. Waves Appl.* 22, 163–172 (2008), <https://doi.org/10.1163/156939308784160703>.
- [40] A.J. Nebro, J.J. Durillo, G. Nieto, C.A.C. Coello, F. Luna, E. Alba, “SMPSO: A new pso-based metaheuristic for multi-objective optimization,” In: *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009 - Proceedings*, pp. 66–73, *IEEE* (2009), <https://doi.org/10.1109/MCDM.2009.4938830>.
- [41] J.J. Durillo, A.J. Nebro, “JMetal: A Java framework for

- multi-objective optimization,” *Adv. Eng. Softw.* 42, 760–771 (2011), <https://doi.org/10.1016/j.advengsoft.2011.05.014>.
- [42] E. Zitzler, L. Thiele, “Multiobjective optimization using evolutionary algorithms - A comparative case study,” In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 292–301 (1998), <https://doi.org/10.1007/bfb0056872>.
- [43] N. Padhye, K. Deb, “Multi-objective Optimisation and Multi-criteria Decision Making for FDM Using Evolutionary Approaches,” In: *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, pp. 219–247, Springer London, London (2011), https://doi.org/10.1007/978-0-85729-652-8_7.