



**HAL**  
open science

# Monitoring Network Slices with a Genetic Algorithm Approach

Zahraa El Attar, Yassine Hadjadj-Aoul, Géraldine Texier

► **To cite this version:**

Zahraa El Attar, Yassine Hadjadj-Aoul, Géraldine Texier. Monitoring Network Slices with a Genetic Algorithm Approach. CCNC 2024 - IEEE Consumer Communications & Networking Conference, Jan 2024, Las Vegas, United States. pp.1-6. hal-04368540

**HAL Id: hal-04368540**

**<https://inria.hal.science/hal-04368540>**

Submitted on 1 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Monitoring Network Slices with a Genetic Algorithm Approach

Zahraa El Attar  
Univ Rennes, Inria, CNRS, IRISA  
Rennes, France  
zahraa.elattar@univ-rennes1.fr

Yassine Hadjadj-Aoul  
Univ Rennes, Inria, CNRS, IRISA  
Rennes, France  
yassine.hadjadj-aoul@irisa.fr

Géraldine Texier  
Univ Rennes, Inria, CNRS, IRISA  
IMT Atlantique, Rennes, France  
geraldine.texier@imt-atlantique.fr

**Abstract**—Network virtualization enables 5G slicing, a technique for sharing physical resources among isolated slices managed by different actors. However, monitoring slices’ performance is becoming challenging due to the significant network overhead associated with direct measurements. To overcome this problem, we propose using network tomography to estimate slices’ delays in the network. In particular, we investigate the problem of finding the minimal combination of end-to-end simple monitoring paths needed to minimize the estimation error of the slices’ delays in a network. We proposed a new genetic algorithm to identify the optimal monitoring paths required to achieve network tomography and minimize their number. To improve the search for the optimal solution, we investigated both a fixed mutation approach and our proposed adaptive mutation approach. Our evaluations show the effectiveness of both approaches; however, the adaptive mutation method outperforms the fixed method by exploring new solutions and avoiding local minima, leading to faster convergence and better results.

**Index Terms**—5G, Slice, Network monitoring, Tomography, Genetic algorithm.

## I. INTRODUCTION

In order to meet customers’ demands, networks are evolving and embracing virtualization, which enables the idea of network slicing introduced by 5G networks. Each slice is a logical network with distinct needs and requirements hosted on the same physical infrastructure. As a result, monitoring network slices is essential to assess performance, identify failures, and intervene when necessary [1], [2].

Monitoring 5G slices is complex due to the potential overload of direct measurement traffic and physical topology abstraction. In addition, the various stakeholders involved in different network slices may only have a partial view of the infrastructure [3]. To overcome these challenges, [4] proposes network tomography as a solution for network monitoring. In network tomography, we infer the internal state of a network from a few end-to-end measurements. The choice of measurement paths is critical in developing a network tomography solution, as these paths are used to estimate network metrics and evaluate performance.

Our contribution consists in proposing a new meta-heuristic that allows finding the optimal paths to be used by network tomography. Moreover, we suggested a new adaptive mutation strategy to improve the exploration of the space of solutions. By leveraging our approach, we were able to identify the op-

timal measurement paths and their numbers for more accurate network tomography.

The paper is structured as follows. Section II covers slices’ monitoring, including prior work and contributions. Section III presents the problem statement and the model developed to address such a challenge. Section IV outlines our proposed solution. In Section V, we evaluate the performance of our approach. Section VI concludes the paper.

## II. RELATED WORK

Slices’ monitoring is a problem that emerged with 5G and network slicing. Although the problem is recent, it may refer to an older issue, the monitoring of overlay networks [5].

Network tomography is probably one of the most promising solutions to this problem. Network tomography has been studied extensively from 1996 [6] to the present day [7], [8], with a review presented in [4] and [9].

A typical network tomography solution involves *nodes selection* for data collection, *measurement paths selection*, and *tomography* techniques to infer network metrics. Previous work has focused on one of the steps and considered the others as given. In our work, we concentrate on the second phase, which involves selecting measurement paths while choosing monitors and the tomography methodology as predetermined factors.

In the following, we summarize prior work to support our decisions in the first and the third steps of network tomography and review previous work on the paths’ selection problem.

Starting with the first step of devising a network tomography solution, which involves selecting nodes for data collection, several studies offer insights into optimal monitor selection. For instance, in [10], the authors propose an algorithm to determine the minimal number of needed monitors. In [11], the authors studied the problem of identifying additive link metrics from end-to-end measurements using only two monitors. Their findings suggest that only two monitors can effectively determine most links. Inspired by this outcome, we employed two nodes as our monitoring points.

In the third step of constructing a network tomography solution, current tomography techniques include algebraic, statistical, and heuristic approaches. Algebraic methods model the tomography problem as a system of linear equations and solve it

using linear algebra techniques to find an exact solution [12]–[14]. For instance, the Single Value Decomposition (SVD) method is presented as an algebraic approach in [12], which utilizes matrix decomposition to reduce the dimensionality of data and simplify the monitoring process. As for statistical approaches, network metrics are modeled as a random variable with a partially unknown probability distribution, and end-to-end measurements are used as input to do the estimation. Statistical methods, such as the Fourier transform, were also proposed to estimate the link metric’s distribution [15]. More recently, machine learning techniques have been introduced as a statistical solution for network tomography [16], [17], and [18]. In [16], the authors used a neural network to infer underlay resources from overlay network measurements, achieving accurate estimation. A neural network was also implemented in [17] to infer slices’ metrics. The authors show that their approach outperforms existing state-of-the-art methods in terms of accuracy and efficiency. In [18], the authors use Support Vector Regression and neural networks as solutions to estimate end-to-end path parameters, achieving significantly improved estimation accuracy compared to other approaches. The network tomography problem has also been modeled using heuristics. In [19], the paper proposes to use genetic algorithms as an optimization technique. The results show that genetic algorithms can effectively solve the network tomography problem, especially when dealing with large-scale networks or complex objective functions. As part of our monitoring task, we needed to select an appropriate tomography technique. We decided to use an algebraic method, specifically SVD, due to its powerful capabilities and ease of implementation in our monitoring case. By using SVD, we aimed to avoid further complexity in the monitoring process. It’s worth noting that all the previously cited studies treat the issue of measurement path selection—the problem we are focusing on—as a predetermined factor.

Now, regarding the problem of measurement paths’ selection, the authors in [5] treat the problem of finding the minimal subset of paths from which to infer loss rates and latencies for all other paths. To do so, they use end hosts as monitors and derive an algebraic method to solve the problem. This work has been improved in [14], where the authors work on the same general problem but manage to make the solution adaptive and scalable. Compared to our work, and inspired by [14], we are using only two nodes as monitors. Multicast and unicast probing methods were tested in [20], [21]. Using multicast and unicast probing dictates the paths we will use to monitor. The work presented in [20], [21] showed slow convergence times and was only tested on tree topologies. Compared to our work, our solution is particularly suitable for highly connected topologies.

To summarize, previous papers have assumed the paths are given, and to the best of our knowledge, no previous article explored using a genetic algorithm to find the combination of measurement paths.

### III. PROBLEM FORMULATION

As previously stated, our goal is to find the set of paths between two monitors that minimizes the estimation error for all slices’ metrics while minimizing the number of paths.

We assume that the topology of the physical network is known and modeled by an undirected graph  $G = (V, L)$ , where  $V$  and  $L$  represent, respectively, the sets of physical nodes and links.  $P$  denotes the set of all simple measurement paths in the network, and  $S$  is the set of slices we want to monitor and which are deployed on top of the physical network  $G$ .

We select the number of measurement nodes, set at two, along with their specific locations, and identify the slices that require monitoring. For the scope of this work, we focus exclusively on additive metrics. Under this assumption, we can express the metrics of slices  $Y_s$ , and paths  $Y_p$  can be expressed as linear combinations of the metrics of the links  $X$  as follows:

$$Y_s = A_s X \quad (1)$$

$$Y_p = A_p X \quad (2)$$

$A_s$  and  $A_p$  are boolean matrices.  $A_s(i, j) = 1$  if the link  $j \in L$  belongs to the slice  $i \in S$ .  $A_p(k, j) = 1$  if the link  $j \in L$  belongs to the path  $k \in P$  and 0 otherwise. This said, knowing links metrics leads us to slices’ metrics.

Given this formulation, it becomes clear that accurate knowledge of the link metrics  $X$  is crucial for determining the metrics of the slices  $Y_s$ . Therefore, our analysis will focus on estimating the metrics of all links in the network.

Let  $H_1, H_2, \dots, H_n$  be the subsets of paths to be selected from  $P$  to estimate all links’ metrics. The cardinality of each subset  $H_i$  is denoted as  $|H_i|$ , which represents the number of paths in  $H_i$ . We use the SVD method to solve Equation (2) and find  $X$ . The performance of each subset of paths is evaluated using the Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \left( \frac{100\%}{l} \right) \sum_{t=1}^l \left| \frac{X_{\text{real}} - X_{\text{estimated}}}{X_{\text{real}}} \right| \quad (3)$$

$X_{\text{real}}$  is the real value of links’ delays,  $X_{\text{estimated}}$  is their estimated value and  $l = |L|$  is the number of links in the physical network.

Our optimization problem can be formally stated as:

For  $H_i$  in  $\{H_1, H_2, \dots, H_n\}$ ,

$$\text{Minimize} \quad \left( \frac{100\%}{l} \right) \sum_{t=1}^l \left| \frac{X_{\text{real}} - X_{\text{estimated}}}{X_{\text{real}}} \right| \quad (4)$$

*Subject to* if two subsets  $H_i$  and  $H_j$  yield the same MAPE select the one with the smaller cardinality

Note that although we consider only the measurement of delays in what follows, this is generalizable to other additive or multiplicative metrics (such as loss) via the logarithms of the latter.

## IV. PROPOSED SOLUTION

To solve the problem introduced above, we propose a genetic algorithm combined with network tomography. The former generates path combinations, while the latter evaluates each combination. The different blocks needed in this work are defined above.

### A. Creation of the initial population

We build the initial population of individuals using a random combination of measurement paths. Each individual is represented as an integer list, where each integer is a monitoring path. The length of the list varies, resulting in different combinations of monitoring paths and the number of paths in each individual. The population is evaluated and evolved using a genetic algorithm.

### B. Evaluation of the population quality

The error of estimation of the links in the network is characterized by a fitness function that uses the Single Value Decomposition method (SVD) [12] and the Mean Absolute Percentage Error (3).

Having paths' delays and knowing the construction of paths in terms of links ( $A_p$ ). We can find the delays on all links in the network using Equation (5) to compute the pseudo-inverse matrix using the SVD method. Applying SVD to a matrix with an incomplete range results in an infinite number of solutions. We chose to find an average of 500 solutions. The following equation yields links' metrics:

$$X = A_p^{-1}Y_p \quad (5)$$

We calculate MAPE using (3). The real values of  $X$  were used to initially train our algorithm using SVD to determine the best paths. However, these values are not needed once the paths are determined. Additionally, other tomography techniques can be used, some of which do not require knowledge of  $X$  [16].

We calculate a fitness value for each individual in the population, we will use it to appraise the quality of individuals.

### C. Evolving the population

The initial population will undergo a selection process, a crossover process, and a mutation process. These operations will be repeated for a certain number of generations to eventually converge to the final global solution.

- **Selection:** To evolve the population, we implement tournament selection [22] as our selection method, where a small group of individuals (size of 3) is selected from the current population, and the fittest individual is chosen as the first parent. The fittest individual is the one that minimizes estimation error. If we have two or more individuals having the same fitness value, we select the one having the minimum number of paths. The process is repeated to find the second parent, and these two parents are crossed and mutated to generate a new child for the next generation.
- **Crossover:** The process of crossover involves combining two parents to generate a new offspring. We randomly

selected two crossover points and swapped the paths between the parents, resulting in two new offsprings.

- **Mutation:** Mutation, in our case, is the operation of replacing one monitoring path with another one. We implement two approaches for mutation: a classical fixed mutation approach and a proposed adaptive mutation approach.

### D. Fixed mutation rate

For fixed mutation, we fix a mutation probability and assign a probability to each element in the individual. If the probability of the individual is less than the mutation probability, we replace the path with another path, randomly chosen from the set of monitoring paths, and without repetition. However, the fixed mutation approach may not provide enough diversity to explore a wide range of solutions, leading the algorithm to get stuck in a local minimum. To address this issue, we suggest implementing an adaptive mutation rate.

### E. Adaptive mutation rate

The main idea behind the proposal of this approach is to ensure sufficient diversity in the population and, hence, a good exploration of the solution space.

To start the algorithm, we set an initial mutation rate and calculate the similarity index for each generation. The similarity index represents the similarity between individuals within the same population and is computed as follows: we create an  $N \times N$  matrix, where  $N$  is the size of the population, and identify the common links between individuals in the population. We then normalize the values in the matrix by dividing by the number of links in the graph and finding the average of the values below the diagonal of the matrix, which represents the similarity score.

Next, we set a fixed similarity threshold and compare it with the similarity index value. If the similarity index is less than the threshold, indicating a diverse set of solutions, we adapt the initial mutation rate by decreasing it one step at a time. On the other hand, if the similarity index is greater than the threshold, indicating similar solutions, we adapt the initial mutation rate by increasing it one step to make the solutions more diverse.

## V. PERFORMANCE EVALUATION

We test the performance of our algorithm on two topologies. The Topology A, taken from [17] and represented in Fig. 1, is formed of 9 nodes, 22 links, and a minimum node degree of 2. The second topology is the Gridnet network, taken from Zoo Topology <sup>1</sup> and represented in Fig. 2, is formed of 9 nodes, 20 links and has a minimum node degree of 4.

Two nodes are selected to exchange monitoring traffic; for topology A, nodes 1 and 4 are the monitoring nodes. The monitors for Gridnet are nodes 0 and 5. We identify all the direct paths between the monitoring nodes for each topology. A straight line with no loops that begins and ends at two

<sup>1</sup><http://www.topology-zoo.org/gallery.html>

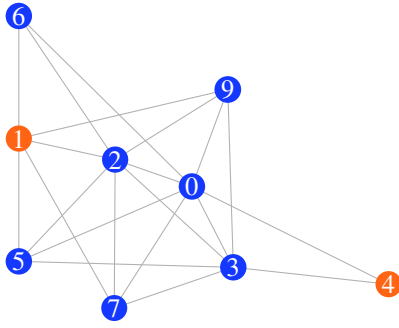


Fig. 1. Topology A

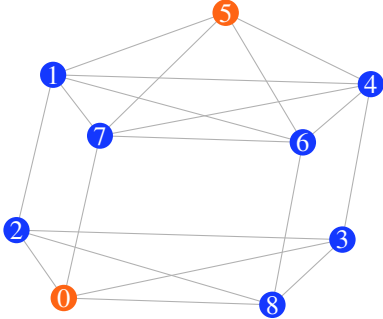


Fig. 2. Gridnet network topology

different nodes is referred to as a simple path. We assign a number to each path.

The population size of the topologies A and B are, respectively, 30 and 50, and we'll run the algorithms for 100 generations.

In the following, we assess how well our solution performs regarding different parameters.

#### A. Comparing fixed and adaptive mutation approach

First, we conduct a comparison between the approaches of fixed mutation and adaptive mutation to determine whether the strategy of adaptive mutation leads to an improvement in the solution. To do so, we start with one initial population and evolve it using the two strategies.

For topology A, in Fig. 3, the fitness value graph for the best individuals over 100 generations is shown. The blue graph represents the fixed mutation rate approach, while the green graph represents the adaptive mutation rate approach. The orange and magenta graphs represent the similarity score and associated mutation rate, respectively. Over the 100 generations, both the fixed and adaptive mutation strategies reduce the estimation error, with the adaptive mutation strategy converging to the solution faster. The orange and magenta graphs show that the adaptive mutation approach starts with a highly diverse population (similarity index  $< 0.5$ ) and adjusts the mutation rate with each generation based on the similarity score. When the similarity score increases, indicating less diversity, the mutation rate is increased to ensure more diversity, and when it decreases, indicating sufficient diversity, the mutation rate gradually decreases to a lower limit.

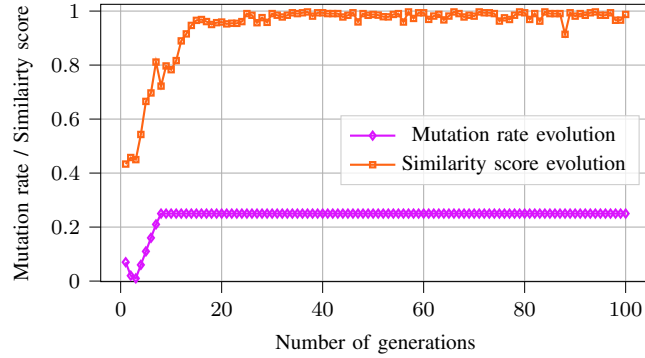
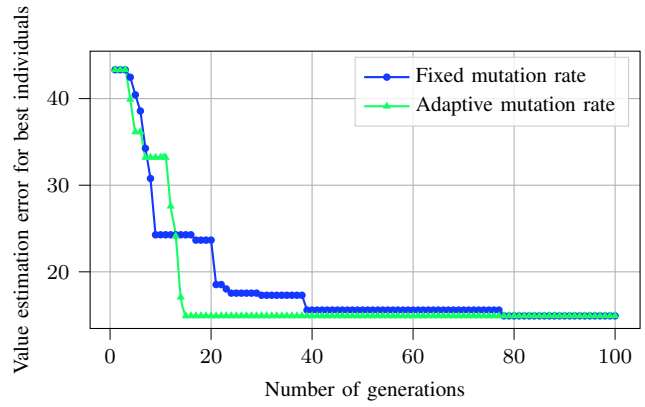


Fig. 3. **Topology A:** Evolution of best individuals

We run the same tests on the Gridnet topology. Fig. 4 shows the fitness value of the best individuals throughout 100 generations. The estimation error in both approaches is decreasing. In adaptive mutation approaches, we reached a smaller error faster than in the fixed mutation approach.

Estimation errors for topology A and Gridnet were compared, with topology A having a higher error of 14% and Gridnet having a lower error of 8%. The difference in error can be attributed to the variation in connectivity between the topologies. In topologies with a minimum degree of three, monitoring paths share more links, which increases correlation and reduces estimation error. However, in topologies with numerous nodes of degrees 1 and 2, correlation is less present, making us more vulnerable to inaccuracies. Even with network tomography, we are unable to precisely estimate link delays in such topologies.

It should be noted that an important step we did in this work to put the solution into practice is fine-tuning by experiment the mutation rate, lower and upper bounds of the adaptive mutation, size of the population, number of generations, and The number of simple paths between monitors.

#### B. Individuals length

The length of the best individuals in each strategy and each topology is also tested and visualized, and it is contrasted with the length of the individuals of the first generation. In Fig. 5 and Fig. 6, we plotted the evolution of the best individuals' lengths of topology A and B, respectively.

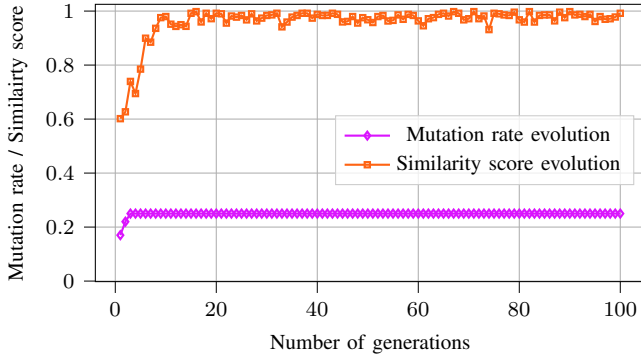
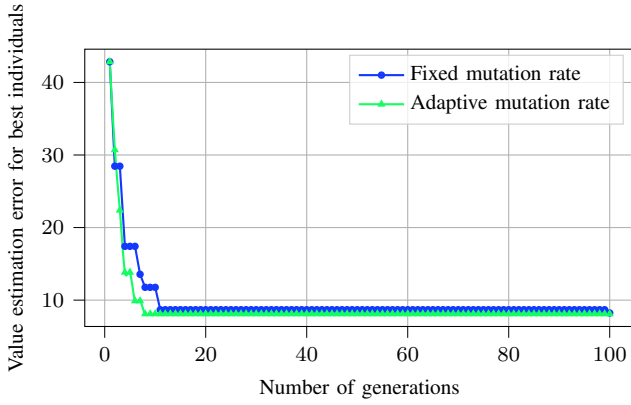


Fig. 4. **Gridnet network**: Evolution of best individuals

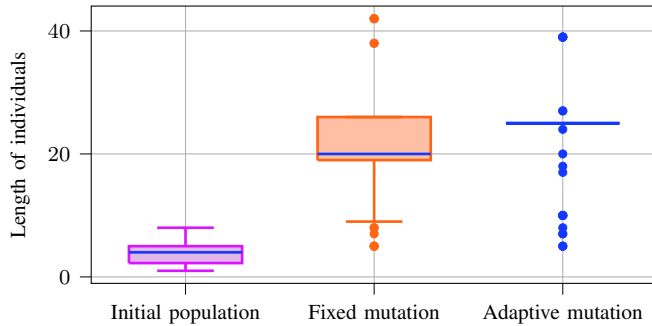


Fig. 5. **Topology A**: Evolution of best individuals' length

In topology A, the population initially has a mean individual length of 4, and both fixed and adaptive mutation methods are used to evolve it. As generations progress, the individual length increases, and the resulting population has a mean individual length of 20 and 25 for fixed and adaptive mutation approaches, respectively. In Gridnet, the initial population has a mean individual length of 5, and the individual length expands to a mean of 17 paths and 19 paths with the fixed and adaptive mutation approaches, respectively.

The average individual length is expected to be higher in adaptive mutation strategies than in fixed mutation strategies because adaptive mutation explores a wider solution space, resulting in better outcomes and more measurement paths. However, the number of paths in the combination of measure-

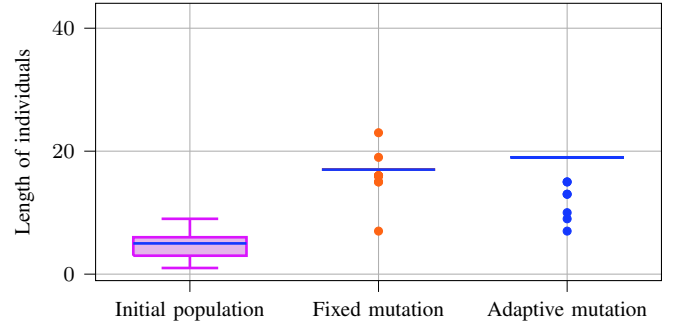


Fig. 6. **Gridnet network**: Evolution of best individuals' length

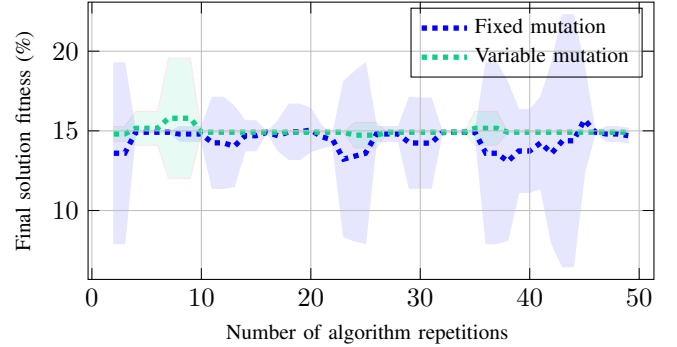


Fig. 7. **Topology A**: Algorithm performance with 95% CI

ment paths must be balanced against reducing the estimation error. It should be noted that in both topologies, only 3% of the paths between the monitors were used to create individuals.

### C. Confidence interval

A confidence interval is an interval within which we are reasonably confident to discover the true value of the variable we are measuring. We usually search for the 95% confidence interval. To calculate the confidence interval, we used the following formula:

$$\left[ m - t \frac{s}{\sqrt{N}}; m + t \frac{s}{\sqrt{N}} \right] \quad (6)$$

where  $m$  is the mean value,  $s$  is the sample standard deviation,  $N$  is the sample size, and  $t$  is a parameter that is related to the level of confidence we want.

The genetic algorithm was repeated 50 times to study the variability of the solutions generated. The final solution from each iteration was saved, and the mean and standard deviation were calculated. The confidence interval was then calculated using these values (equation (6)). The moving average of the 50 best solutions was plotted for topology A and Gridnet, and a confidence interval was drawn around it in Fig. 7 and Fig. 8, respectively.

As shown in Fig. 7, in blue, we have the fixed mutation approach, and in green, we have the adaptive mutation one. The blue line shows more variability compared to the green line, which produced consistent values throughout the executions. Both approaches have a narrow confidence interval.

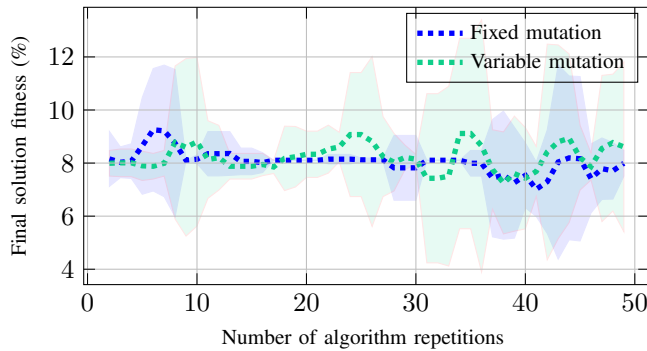


Fig. 8. **Gridnet network**: Algorithm performance with 95% CI

For Gridnet, Fig. 8, both lines start with high fitness values, indicating that the initial population is not very fit. However, as the algorithm repetitions increase, both lines decrease, indicating that both approaches are becoming more effective in minimizing the fitness value.

The fixed mutation approach is initially slightly better than the adaptive mutation approach, as the blue line decreases faster than the green line. However, as the number of repetitions increases, the blue line levels off while the green line continues to decrease. This suggests that the adaptive mutation approach is better at exploring new solutions and avoiding local minima than the fixed mutation approach.

Around repetition 30, the green line crosses the blue line, indicating that adaptive mutation is more effective in minimizing the fitness value. After that point, the green line continues to decrease faster than the blue line.

## VI. CONCLUSION AND FUTURE WORK

In this work, we aimed to minimize the estimation error and the number of paths used to estimate slices' delays. We modeled slices as a linear combination of links in the network and proposed a genetic algorithm solution combined with network tomography. We compared two approaches, fixed mutation rate and adaptive mutation rate, and evaluated the algorithm's convergence, length of individuals in the population, and confidence interval. Our results suggest that when the optimal solution is difficult to find and requires extensive exploration, a fixed mutation rate may not be sufficient. In contrast, the adaptive mutation rate is more effective in exploring the search space by increasing the mutation rate when the population stagnates and decreasing it when the population is diverse.

Future work will include testing additional network tomography methods, such as machine learning techniques. Additionally, an objective function will be added to the optimization problem.

## REFERENCES

- [1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.
- [2] N. Al-Falahy and O. Y. Alani, "Technologies for 5g networks: Challenges and opportunities," *It Professional*, vol. 19, no. 1, pp. 12–20, 2017.
- [3] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, and C.-S. Yang, "Network monitoring in software-defined networking: A review," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3958–3969, 2018.
- [4] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi, "Network tomography: A review and recent developments," *Frontiers in statistics*, pp. 345–366, 2006.
- [5] Y. Chen, D. Bindel, and R. H. Katz, "Tomography-based overlay network monitoring," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 2003, pp. 216–231.
- [6] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American statistical association*, vol. 91, no. 433, pp. 365–377, 1996.
- [7] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Inferring link metrics from end-to-end path measurements: Identifiability and monitor placement," *IEEE/ACM transactions on networking*, vol. 22, no. 4, pp. 1351–1368, 2014.
- [8] N. G. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurements," in *IWDC*, vol. 1. Springer, 2001, pp. 576–595.
- [9] G. Kakkavas, D. Gkatzoura, V. Karyotis, and S. Papavassiliou, "A review of advanced algebraic approaches enabling network tomography for future network infrastructures," *Future Internet*, vol. 12, no. 2, p. 20, 2020.
- [10] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 906–916, 2011.
- [11] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Link identifiability in communication networks with two monitors," in *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2013, pp. 1513–1518.
- [12] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Linear algebra*, vol. 2, pp. 134–151, 1971.
- [13] Y. Chen, D. Bindel, H. H. Song, and R. H. Katz, "Algebra-based scalable overlay network monitoring: algorithms, evaluation, and applications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1084–1097, 2007.
- [14] Y. Chen, D. Bindel, H. Song, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004, pp. 55–66.
- [15] A. Chen, J. Cao, and T. Bu, "Network tomography: Identifiability and fourier domain estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6029–6039, 2010.
- [16] M. Rahali, J.-M. Sanner, and G. Rubino, "Tom: a self-trained tomography solution for overlay networks monitoring," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–6.
- [17] A. Rkhami, Y. Hadjadj-Aoul, G. Rubino, and A. Outtagarts, "On the use of machine learning and network tomography for network slices monitoring," in *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2021, pp. 1–7.
- [18] I. Sartzetakis and E. Varvarigos, "Machine learning network tomography with partial topology knowledge and dynamic routing," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4922–4927.
- [19] S. Qazi, R. A. Memon, and A. A. Farooqui, "Evaluation of performance of genetic algorithms for network tomography," *Journal of Internet Technology*, vol. 16, no. 1, pp. 75–85, 2015.
- [20] C. Liu, T. He, A. Swami, D. Towsley, T. Salonidis, A. I. Bejan, and P. Yu, "Multicast vs. unicast for loss tomography on tree topologies," in *MILCOM 2015-2015 IEEE Military Communications Conference*. IEEE, 2015, pp. 312–317.
- [21] E. Lawrence, G. Michailidis, and V. N. Nair, "Network delay tomography using flexicast experiments," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 5, pp. 785–813, 2006.
- [22] T. Blickle, "Tournament selection," *Evolutionary computation*, vol. 1, pp. 181–186, 2000.