Towards Efficient Traffic Engineering via Distributed Optimization in Large-Scale LEO Constellation

Linhui Wei[1], Tien-Thanh Le[2], Yusheng Ji[2], *Fellow, IEEE*, Mingqian Wang[1], Yu Liu[1], *Member, IEEE,* Yumei Wang[1], *Member, IEEE*, John C.S. Lui[3], *Fellow, IEEE*
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
[2]National Institute of Informatics, Tokyo, Japan
[3]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
Emails: {weilinhui, mingqianw, liuy, ymwang}@bupt.edu.cn, {lethanh, kei}@nii.ac.jp, cslui@cse.cuhk.edu.hk

2024 IEEE International Conference on Communications Workshops (ICC Workshops) | 979-8-3503-0405-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCWORKSHOPS59551.2024.10615550

***Abstract*—Space-air-ground integrated networks have the potential to offer extensive coverage beyond 5G and 6G services, as well as the cloud continuum. Managing network traffic in large-scale space-air-ground networks presents a significant challenge due to the dynamic nature of traffic demand and network topology. Additionally, the large-scale low earth orbit (LEO) constellation makes solving the traffic engineering optimization problem impractical. To tackle these challenges, we introduce the Cluster-based Elephant flow Splitting via Linear Programming (CESLP) algorithm, which can rapidly evaluate the maximum link utilization (MLU). CESLP classifies traffic flows into mice and elephant flows, and focuses on improving the splitting ratio for elephant flows. Using a divide and conquer strategy, CESLP divides the LEO constellation into multiple clusters, identifying the cluster with the most congested edge and minimizing its MLU by solving the Linear Programming (LP) sub-problems. Simulation is conducted on synthetic traffic matrices for Starlink constellation, and the results demonstrate the effectiveness of the CESLP algorithm in reducing MLU compared with other baseline routing algorithms.**

***Index Terms*—Large-scale LEO constellation, traffic engineering, elephant flow, minimize maximum link utilization**

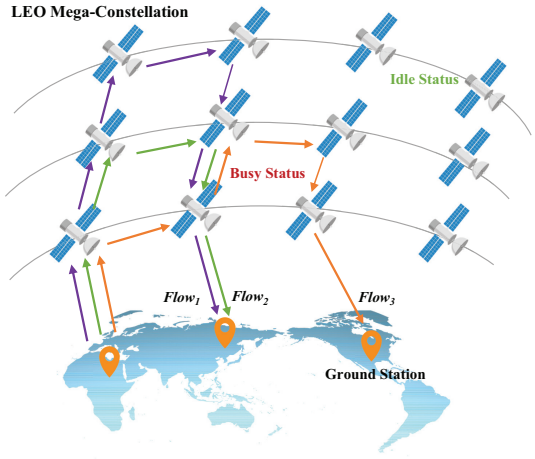

Fig. 1. A typical scenario of the traffic flows in the satellite networks.

I. Introduction

Space-air-ground integrated networks (SAGIN) offer the capability of global coverage and expand the reach of the cloud continuum and beyond 5G and 6G (B5G/6G) services [1]. The rapid increase of low earth orbit (LEO) satellites has revolutionized modern communication systems. The emergence of large-scale LEO constellations, such as SpaceX's Starlink and Amazon's Project Kuiper, provide high-speed data transmission and aim to deliver global broadband network services. By integrating large-scale LEO constellations, SAGIN enables low-latency and high-bandwidth connections necessary for the new emerging applications [2].

Traffic Engineering (TE) in satellite networks (SAGIN) is an intricate problem that is influenced by the diverse user requirements across the world. The global traffic pattern is significantly affected by the uneven distribution of population and disparities in economic development [3]. These factors directly impact the performance of satellite networks, giving rise to challenges such as unequal data traffic loads, heightened latency, and varying levels of congestion. As depicted in Fig. 1, a typical scenario of traffic flows in satellite networks illustrates these challenges. The disproportionate traffic distribution may lead to congestion on high-demand links, as well as underutilization in less active regions. Therefore, effective TE plays a critical role in optimizing performance in satellite networks.

Many existing works have proposed strategies for solving the TE problem in satellite networks. To provide satellite-based Internet service, Hu *et al.* [4] investigated the unicast and multicast TE for software-defined networking (SDN)-enabled LEO satellite networks to enhance the inter-satellite links (ISLs) performance. In the unicast scenario, they focused on minimizing the maximum link utilization (MLU) and formulated it as the k-segment routing problem in the grid topology. Zhang *et al.* [5] proposed to minimize the maximum satellite load instead of the MLU in each slot, and they put forward the delay-bounded traffic splitting method to limit the path length. When satellites move into polar regions, the connected links will be off, Chen *et al.* [6] presented the waypoint segment routing algorithm to avoid the broken links in the next snapshot and then provided stable packet forwarding methods. The position

of the ground gateway also impacts network congestion. Liu *et al.* [7] dynamically divided the light and heavy load zones according to the position between gateways and reverse slots. Light load zones and heavy load zones adopted different routing rules based on a load of outermost nodes in heavy load zones. However, these studies rarely address the TE challenges in large-scale LEO constellations, where the number of LEO satellites exceeds 1,000.

Therefore, the TE problem needs innovative approaches to effectively manage large-scale constellation scenarios. These include the deployment of advanced optimization algorithms and dynamic routing policies aimed at improving critical network performance metrics such as minimizing MLU and enhancing routing efficiency. Distributed TE has emerged as a promising approach for addressing the complexities of large-scale network management [8]. Dividing a large-scale network into multiple regions through distributed management significantly improves computation efficiency. This approach involves optimizing traffic both within and among these regions, where the influence of cross-traffic flows is a critical factor. Achieving local optimization within these regions is relatively straightforward. However, attaining a global optimum across the entire network presents a substantial challenge. Distributed optimization has never been exploited in previous work to achieve efficient TE in large-scale LEO constellations.

In this paper, we consider an efficient TE optimization method for large-scale LEO constellation. We target minimizing the MLU and formulate it as the linear programming (LP) problem. The traffic in the network is categorized into mice and elephant flows, and we focus on optimizing the splitting ratio for only elephant flows to reduce the complexity of the full-scale LP program. Moreover, we propose Cluster-based Elephant Flow Splitting via Linear Programming (CESLP) algorithm to achieve distributed TE optimization. Specifically, the large-scale LEO constellation is divided into multiple clusters. We iterate to find the cluster that contains the most congested edges and minimize the MLU within that cluster by solving the LP of the sub-problem within the cluster. This approach enables the LP solver to address the TE challenges in each cluster within a stringent time requirement.

The rest of this paper is organized as follows: Section II introduces the system model and the problem description. Section III introduces the proposed CESLP method. The performance evaluation and analysis are conducted in Section IV. The conclusion is drawn in Section V.

II. System Model and Problem Formulation

In this section, we give the LEO satellite network topology model, and then we formulate the TE problem as an LP problem to minimize the MLU.

A. LEO Satellite Network Topology

We consider a large-scale LEO constellation, wherein the satellites are interconnected using a +Grid configuration [9]. In the 2-D grid topology, each satellite is connected to its successor and predecessor within its orbit, as well as to the nearest satellite in its left and right orbits. It makes the topology remain unchanged as ISL pairs are kept constant. Despite the high mobility and continuous coverage across various service areas by LEO satellites, their roles within an orbit are interchangeable. Specifically, the successor satellite takes over the role of its predecessor satellite once it arrives at the predecessor's former location. The 2-D grid topology is defined as a graph $G = (V, E)$, where V represents the set of satellite nodes, and E represents the set of ISLs. The total number of satellites in the network is $|V| = N \times M$, where N denotes the number of orbits, and M represents the number of satellites per orbit.

B. Problem Formulation

In general, the main objective is to minimize congestion in the LEO network by splitting the traffic flow optimally among a set of shortest paths. Let $f_{i,j}$ denote the traffic demand from node $i \in V$ to node $j \in V$. The set of shortest paths by hop count from i to j is denoted as $\mathcal{P}_{i,j}$. The decision variable is $x^p_{i,j} \in [0, 1]$, which represents the fraction of traffic demand of flow $f_{i,j}$ that travels through the p^{th} path in the set $\mathcal{P}_{i,j}$.

The main goal is to minimize the MLU, denoted as θ, which can be formulated as follows:

Problem 1 (Optimal traffic splitting among shortest path):

$$\underset{x^p_{i,j}}{\text{minimize}} \quad \theta, \tag{1}$$

subject to

$$\sum_{i,j \in V} \sum_{p \in \mathcal{P}_{i,j}} g^p_{i,j}(e) x^p_{i,j} f_{i,j} \leq \theta c(e), \forall e \in E \tag{2}$$

$$\sum_{p \in \mathcal{P}_{i,j}} x^p_{i,j} = 1, \forall i, j \in V \tag{3}$$

$$x^p_{i,j} \geq 0, \forall i, j \in V, \forall p \in \mathcal{P}_{i,j} \tag{4}$$

Here, $g^p_{i,j}(e) \in \{0, 1\}$ represents whether the traffic on the p^{th} path of flow $f_{i,j}$ traverses link e. The first constraint in Eq. (2) imposes a limit on the load of every link $e \in E$, ensuring that it is smaller than the maximum link load $\theta c(e)$, where $c(e)$ is the capacity of link $e \in E$. Eq. (3) ensures that all traffic within each flow is routed through the network. Eq. (4) guarantees that the traffic travels on the shortest paths and traffic on each path is non-negative.

However, with the increase in possible paths between nodes, the number of possible shortest paths between distant nodes grows exponentially. In the following section, we present the proposed CESLP method for reducing the search space of the TE problem in a large-scale LEO constellation, thereby enabling feasible solutions to be found within a limited time.

III. Proposed CESLP Method

In this section, we discuss the proposed CESLP algorithm for addressing distributed TE optimizing problem in the large-scale LEO constellation. CESLP consists of three strategies: (i) quickly generating the set of shortest paths by utilizing

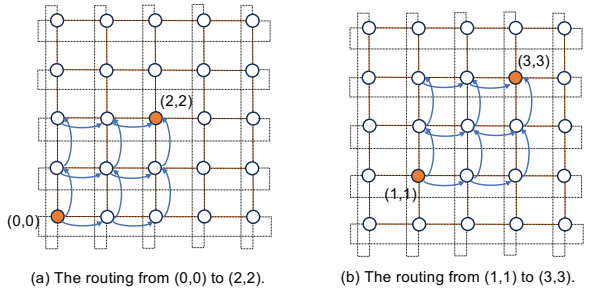

Fig. 2. The routing shifting initialization method in the 5×5 satellite network.

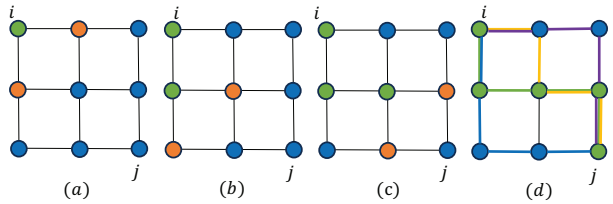

Fig. 3. The visualization of the modified randomized DFS algorithm in the 3×3 grid network.

the regular grid topology, (ii) refining traffic management through the mice and elephant flows, and (iii) dividing the whole problem into subproblems for cluster-based distributed optimization.

A. Fast Shortest Path Calculation

First, we initialize the optimization by calculating the set of shortest path $\mathcal{P}_{i,j}$. When the number of nodes in the network increases, the number of shortest paths among a distant pair of nodes (i, j) also grows exponentially. By leveraging the regularity of 2-D grid topology, a set of shortest path $\mathcal{P}_{i,j}$ can be computed rapidly. For example, as illustrated in Fig. 2, the set of shortest path of the flow from node $(0, 0)$ to node $(2, 2)$ has the same patterns as the flow from node $(1, 1)$ to node $(3, 3)$. For any node at (x_j, y_j) in the network, we can determine its relative position to node (x_i, y_i) as $(\Delta x, \Delta y)$, where $\Delta x = (x_j - x_i) \bmod N$ and $\Delta y = (y_j - y_i) \bmod M$. The path set $\mathcal{P}_{i,j}$ for any network node has the same patterns as the flow from node $(0, 0)$ to node Δ at the location $(\Delta x, \Delta y)$. $\mathcal{P}_{i,j}$ can be obtained by shifting all nodes in the path list $\mathcal{P}_{0,\Delta}$ an amount of x_i steps and y_i steps ahead in each dimension, respectively.

Furthermore, it is not necessary to identify all the shortest paths from i to j, as utilizing all paths generates overhead, thereby hindering the ability of the optimization solver to operate within a limited time. CESLP only optimize the traffic splitting ratios among the path in $\tilde{\mathcal{P}}_{i,j} \subseteq \mathcal{P}_{i,j}$. The requirement for the list of paths in $\tilde{\mathcal{P}}_{i,j}$ is that they can cover a diverse set of edges from i to j. It enables effective load balancing through optimized traffic splitting ratios. To generate $\tilde{\mathcal{P}}_{i,j}$, we need to traverse the sub-graph between i and j using a modified randomized DFS algorithm, which is given in Algorithm 1. The number of shortest paths in $\tilde{\mathcal{P}}_{i,j}$ is equal to the minimum hop counts between i and j, denoted as H. The algorithm begins by identifying the neighbors of i that are nearest to j, denoted as $N_j(i)$. Instead of visiting each neighbor in a fixed order, the next node to traverse to j is selected randomly from $N_j(i)$. The randomized path is added to the $\tilde{\mathcal{P}}_{i,j}$ when i equals j. The algorithm decrements H with each recursive call and terminates once H reaches zero or the number of paths in $\tilde{\mathcal{P}}_{i,j}$ equals the minimum hop counts between i and j.

The visualization of the modified randomized DFS algorithm is shown in Fig. 3. As depicted in Fig. 3(a) to Fig. 3(c), starting from node i, there are two nearest neighbors, marked as orange, to node j at each step. If a path has not been previously selected, it can be added to the set $\tilde{\mathcal{P}}_{i,j}$. Given that there are 4 minimum hop counts between nodes i and j, we define H as 4 accordingly. In the initialization phase, the set $\tilde{\mathcal{P}}_{i,j}$ is empty, with no elements in $path$. We then execute **Function** DFS$(i, j, path, 4, \emptyset)$ to acquire the set of shortest paths. The process is terminated when the accumulation of paths in $\tilde{\mathcal{P}}_{i,j}$ reaches 4, as illustrated in Fig. 3(d).

Algorithm 1 A modified randomized DFS algorithm

Function DFS$(i, j, path, H, \tilde{\mathcal{P}}_{i,j})$
if $H \leq 0$ **then**
  Return
else if $i = j$ **then**
  Append $path$ to $\tilde{\mathcal{P}}_{i,j}$
  $H = H - 1$
else
  Generate the set of neighbors $N_j(i)$ of i.
  Random permutation of $N_j(i)$ to obtain $\tilde{N}_j(i)$
  for h in $\tilde{N}_j(i)$ **do**
    DFS$(h, j, path + [h], H)$
  end for
end if
EndFunction

B. Mice and Elephant Flows Refinement

The traffic matrix exhibits inherent sparsity and low rank, which is not only true for LEO networks but also for core IP networking [10], [11]. This implies that the elements in the traffic matrix can be expressed as a linear combination of a limited set of underlying patterns or factors. Consequently, the traffic matrix can be accurately approximated or compressed using fewer dimensions [12]. Therefore, an alternative approach for simplifying the complexity of the problem involves identifying the elephant flows [13], characterized by the high-volume traffic. This approach allows TE to be performed only on these significant flows, simplifying the optimization process.

We employ the three-sigma rule to classify flow $f_{i,j}$ into elephant flows and mice flows. This means that a flow $f_{i,j}$ is classified as a mice flow if it satisfies the condition $f_{i,j} <$

$\mu(f_{i,j}) + 3\sigma(f_{i,j})$, where $\mu(f_{i,j})$ is the mean of $f_{i,j}$ and $\sigma(f_{i,j})$ is the standard deviation of $f_{i,j}$. For mice flows, all traffic is directed and routed through one randomized shortest path between the flow's source and destination. In contrast, elephant flows distribute their traffic across multiple paths in $\tilde{\mathcal{P}}_{i,j}$. This approach further reduces the search space by 99%, while adeptly addressing the most critical 1% of the problem.

C. Cluster-based Distributed Optimization

We divide the problem into sub-problems geometrically. The 2-D grid topology of the LEO constellation is split horizontally and vertically into K rectangles. Each rectangle is allocated with $\lceil \frac{N \times M}{K} \rceil$ satellites. Specifically, each orbit contributes half of its satellites $\lceil \frac{M}{2} \rceil$ to each rectangle, and there are $\lceil \frac{2N}{K} \rceil$ orbits in each rectangle. Moreover, to meet the effects of cross-region traffic, we built inter-connection regions that have the same number of edges in the rectangle. We regard each rectangle and the inter-connection regions as clusters $\mathcal{C}$. Each cluster $c \in \mathcal{C}$ is defined by the set of edges E_c, such that $\bigcap_{c \in C} E_c = \emptyset$, and $\bigcup_{c \in C} E_c = E$. The problem of distributing traffic across the set of shortest paths $\tilde{\mathcal{P}}_{i,j}$ for each cluster can be formulated as follows:

Problem 2 (Optimal cluster-based traffic splitting):

$$\underset{x^p_{i,j}}{\text{minimize}} \quad \theta_c, \tag{5}$$

subject to

$$\sum_{i,j \in V} \sum_{p \in \tilde{\mathcal{P}}_{i,j}} g^p_{i,j}(e) x^p_{i,j} l_{i,j} + l_{\text{mice}}(e) \leq \theta_c c(e), \forall e \in E_c \tag{6}$$

$$\sum_{p \in \tilde{\mathcal{P}}_{i,j}} x^p_{i,j} = 1, \forall i, j \in V, \forall p \in \tilde{\mathcal{P}}_{i,j} \tag{7}$$

$$x^p_{i,j} \geq 0, \forall i, j \in V, \forall p \in \tilde{\mathcal{P}}_{i,j} \tag{8}$$

Here, θ_c is the MLU in the cluster $c \in \mathcal{C}$. We optimize all flow $l_{i,j} \in \mathcal{L}_c$, in which $\mathcal{L}_c$ is the set of elephant flows, that pass through any link E_c of cluster $c \in \mathcal{C}$. The first constraint in Eq. (7) also takes into account mice flow $l_{\text{mice}}(e)$, which is the background traffic load that exerts on link $e \in E_c$. Similarly, Eq. (7) guarantees that all traffic for each flow is routed through the network, and Eq. (8) constrains that the traffic is routed through the shortest path and each path remains non-negative.

The procedure to obtain the results for *Problem 1* by solving *Problem 2* is as follows. We use the equal splitting ratio for $\tilde{\mathcal{P}}_{i,j}$ to initial the global link utilization. Identify the cluster E_c that contains the edge e_{max} with the highest θ_c. Then the traffic flow with the splitting ratio $x^p_{i,j}$ that passes through edges E_c is subtracted from the edges e_{max}. We use the LP solver to solve *Problem 2* to obtain the optimal $x^p_{i,j}$ in cluster E_c and add the traffic flow to corresponding edges. The procedure terminates when θ_c stabilizes or the limited running time is reached. Through iterative finding the cluster with the highest θ_c and solving *Problem 2*, we avoid constructing a complex LP problem and then tackle it from a global perspective. This approach leverages parallel computing to accelerate the optimization process while yielding near-optimal results of solving *Problem 1*.

TABLE I
SIMULATION PARAMETERS OF THE STARLINK CONSTELLATION

Parameter	Value
Height of LEO Satellites	550 km
Constellation Configuration	Walker
Total Number of LEO Satellites	1584
Number of Orbits (N)	72
Number of Satellites per Orbit (M)	22
Inclination of Orbits	55°
Phase Factor (F)	3

IV. SIMULATION RESULTS

In this section, we present the simulation setup and evaluate the performance of the proposed algorithm in comparison to other baseline approaches.

A. Simulation Setup

All experiments were conducted in Python 3.11 and the LP optimization is implemented on GPU to accelerate the LP solver's speed. Given that the snapshot is captured with a granularity of 5 minutes, the solver is allocated a maximum time of only 1 minute.

1) Satellite Constellation: We conduct a performance evaluation of the Starlink constellation (Phase-I, Shell-I) using a Python-based simulation [14]. The Starlink constellation consists of a total of 1,584 LEO satellites distributed across 72 orbits. Each orbit has 22 satellites. Other satellite parameters are listed in Table I. Given the dynamic topology of the LEO constellation, we capture snapshots of network topology every 5 minutes, starting from 12:00 PM GMT on January 1, 2024. There are 100 snapshots in total.

2) Traffic Matrix Generation: To effectively evaluate the TE problem in the satellite network, we generate traffic that resembles the real-world traffic distribution by reusing the implementation provided in [15]. To model and generate traffic patterns in satellite networks, we consider Gross Domestic Product (GDP) [16] and population [17] to model Internet traffic distribution. In this way, the performance of the TE algorithms in satellite networks can be evaluated under diverse traffic load scenarios. The steps for generating the traffic matrix are as follows:

- Generate city pairs around the world randomly based on the random Gravity model [15] (the probability of a city being selected is based on its GDP or population).
- Each city is connected to the nearest satellite.
- For each city pair, add 1 Mbps to the flow that connects the two endpoints.
- Aggregate the traffic between every pair of satellites to construct the final traffic matrix.

Regions with higher GDP or population contribute more to the overall traffic load. For each satellite snapshot, we generate one traffic matrix, resulting in a total of 100 traffic matrices

TABLE II
STATISTICAL COMPARISON OF THE GENERATED TRAFFIC MATRIX

Index	GDP-based	Population-based
Min	0.0	0.0
Max	1160	1800
Mean	11.59	159.46
Standard Deviation	29.25	242.34
Variance	855.31	58728.46
Percent of elephant flows	6.48%	2.59%

based on GDP and population, respectively. Table II shows the traffic statistic measured in Mbps. The GDP-based traffic matrices exhibit a higher ratio of standard deviation to mean compared to the population-based traffic matrices. It means that GDP-based traffic is more heavy-tailed than population-based traffic. Also, the GDP-based traffic matrices are more sparse than the population-based traffic matrices since there were 6.48% of flows in the GDP-based are classified as elephant flows, which is more than double the percentage of elephant flows in population-based traffic. We expect that the traffic is highly concentrated on a smaller number of elephant flows in a GDP-based traffic matrix.

3) Baseline and parameter setup: We compare the proposed CESLP algorithm with the other three algorithms, they are introduced as follows:

- SP (Shortest Path): The traffic in the network is routed through a randomly chosen shortest path.
- SE (Shortest path with Equal splitting ratio): Mice flows are not split and randomly routed on one of the shortest path $\mathcal{P}_{i,j}$, and the elephant flows can be split and routed with equal splitting ratios on the set of shortest paths $\tilde{\mathcal{P}}_{i,j}$.
- LS (Local Search): By iteratively selecting one of the flows that traverse the most congested edge in the network, and then randomly choosing another shortest path for that flow (in [18]). This method serves as a robust baseline for the TE problem in large-scale networks.

To implement the distribution optimization in the proposed CESLP method, the Starlink constellation is split into 12 rectangles, and each rectangle is allocated with 12×11 satellites. Besides, there are 2 inter-connection regions for the cross-region traffic.

B. Performance Analysis

Figure 4 shows the results on Starlink constellation with GDP-based traffic. The proposed CESLP algorithm with an average MLU of 0.579. In comparison, the SP, SE, and LS algorithms have average MLUs of 0.641, 0.653, and 0.651, respectively. Since the economic activities are highly concentrated on a small subset of cities, the approximated path set $\tilde{P}_{i,j}$ for elephant flows is frequently overlapped. Thus, flow splitting using the SE algorithm achieves the same performance as the SP algorithm. The CESLP algorithm improves about 10% performance compared with three baseline algorithms. Additionally, the CESLP algorithm demonstrates a slightly lower

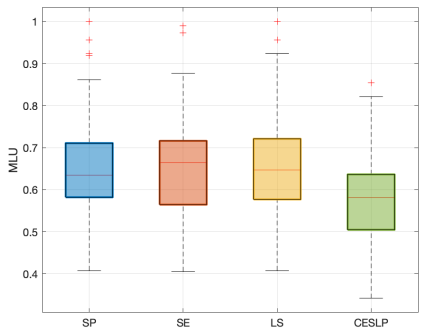

Fig. 4. The MLU results on Starlink constellation with GDP-based traffic matrix.

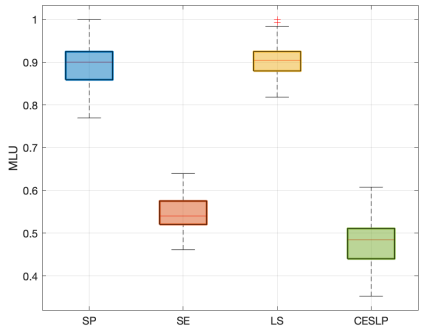

Fig. 5. The MLU results on Starlink constellation with population-based traffic matrix.

mean and minimum value than the LS algorithm, suggesting a more efficient traffic-handling capability. The LS algorithm needs a re-calculation of edge loads with each new traffic matrix, resulting in a computational cost exceeding 1 minute for initial calculations. This computational inefficiency may impact its practical deployment in scenarios where rapid traffic matrix updates are necessary. On the contrary, the CESLP algorithm leverages a set of precomputed paths for both mice and elephant flows, thereby simplifying the initial calculation process.

Figure 5 illustrates the results on Starlink constellation with population-based traffic. Since the population is more dispersed, the approximated path set $\tilde{P}_{i,j}$ for elephant flows is less overlapped. Thus, flow splitting using the SE and CESLP algorithms resulted in significant improvements compared to the SP and LS algorithms. The proposed CESLP algorithm with an average MLU of 0.479. The comparison algorithms, SP, SE, and LS algorithms, have average MLUs of 0.892, 0.547, and 0.903, respectively. The population-based traffic is less heavy-tailed and has fewer elephant flows than the GDP-based traffic. Therefore, the CESLP algorithm solves for the optimal solution within each cluster faster and provides significantly lower MLU than the SE algorithm. The LS algorithm shows the widest

range MLU suggesting significant variability. Conversely, the proposed CESLP algorithm not only exhibits the lowest median MLU but also the narrowest spread of results, showing a more stable and efficient management of network utilization.

V. Conclusion

Given the extensive number of satellites in large-scale LEO constellations, achieving efficient TE poses a significant challenge. In this paper, we studied the distributed TE optimization method in large-scale LEO constellations. We formulated the minimum MLU as an LP problem and tried to optimize the traffic in a distributed manner. Specifically, we divided the large-scale LEO constellation into multiple clusters and then solved the optimization problem by subproblems. We proposed CESLP algorithm to optimize the splitting ratios for elephant flows and minimize the MLU in the cluster that contains the most congested edge iteratively. The evaluation was conducted on Starlink constellation and the simulation results showed that the proposed CESLP algorithm was effective in minimizing the MLU.

Acknowledgement

This work was supported in part by JST ASPIRE under Grant No. JPMJAP2325, in part by National Key Research and Development Program of China under Grant No.2022YFB2902705, and in part by China Scholarship Council under Grant No.202206470050.

References

[1] R. Xing, X. Ma, A. Zhou, S. Dustdar, and S. Wang, "From earth to space: A first deployment of 5G core network on satellite," *China Communications*, vol. 20, no. 4, pp. 315–325, 2023.

[2] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[3] S. Karapantazis, E. Papapetrou, and F.-N. Pavlidou, "Multiservice on-demand routing in LEO satellite networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 107–112, 2009.

[4] M. Hu, M. Xiao, W. Xu, T. Deng, Y. Dong, and K. Peng, "Traffic engineering for software-defined LEO constellations," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5090–5103, 2022.

[5] S. Zhang, X. Li, and K. L. Yeung, "Segment routing for traffic engineering and effective recovery in low-earth orbit satellite constellations," *Digital Communications and Networks*, 2022.

[6] R. Chen, W.-N. Wang, X. Zhao, and G. Zhao, "Waypoint segment routing algorithm for LEO satellite network," *IET Communications*, vol. 16, no. 18, pp. 2133–2144, 2022.

[7] W. Liu, Y. Tao, and L. Liu, "Load-balancing routing algorithm based on segment routing for traffic return in LEO satellite networks," *IEEE Access*, vol. 7, pp. 112 044–112 053, 2019.

[8] F. Bannour, S. Souihi, and A. Mellouk, "Distributed sdn control: Survey, taxonomy, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 333–354, 2018.

[9] D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, ACM, 2019, pp. 341–354.

[10] V. A. Le, T. T. Le, P. Le Nguyen, H. T. T. Binh, R. Akerkar, Y. Ji, *et al.*, "GCRINT: network traffic imputation using graph convolutional recurrent neural network," in *ICC 2021-IEEE International Conference on Communications*, IEEE, 2021, pp. 1–6.

[11] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, and G. Zhang, "Accurate recovery of internet traffic data: A tensor completion approach," in *IEEE INFOCOM 2016-IEEE Conference on Computer Communications*, IEEE, 2016, pp. 1–9.

[12] V. A. Le, Y. Ji, H. H. Tran, P. L. Nguyen, and J. C. S. Lui, "Achieving multi-time-step segment routing via traffic prediction and compressive sensing techniques," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023.

[13] K. Xie, J. Tian, X. Wang, G. Xie, J. Wen, and D. Zhang, "Efficiently inferring top-k elephant flows based on discrete tensor completion," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, IEEE, 2019, pp. 2170–2178.

[14] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the "internet from space" with hypatia," in *Proceedings of the ACM Internet Measurement Conference*, ACM, 2020, pp. 214–229.

[15] G. Giuliari, T. Ciussani, A. Perrig, and A. Singla, "ICARUS: Attacking low earth orbit satellite networks," in *Proceedings of 2021 USENIX Annual Technical Conference (USENIX ATC 21)*, USENIX Association, 2021, pp. 317–331.

[16] W. D. Nordhaus and X. Chen., *Global gridded geographically based economic (g-econ) data set, version 4*, https://sedac.ciesin.columbia.edu/data/set/spatialecon-gecon-v4, 2016.

[17] *Center for international earth science information network-ciesin-columbia university. gridded population of the world, version 4 (gpwv4): Population count, revision 11.* https://sedac.ciesin.columbia.edu/data/set/gpwv4-population-count-rev11, 2018.

[18] S. Gay, R. Hartert, and S. Vissicchio, "Expect the unexpected: Sub-second optimization for segment routing," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, IEEE, 2017, pp. 1–9.