

A Bi-objective Model to Quantify the Load-Latency Trade-off in LEO Constellation Routing

Alice De Guibert, Oana Hotescu, Emmanuel Lochin, Alain Haït
Fédération ENAC, ISAE-SUPAERO, ONERA, Université de Toulouse, France
{alice.de-guibert, oana.hotescu, alain.hait}@isae-supaero.fr, emmanuel.lochin@enac.fr

Abstract—Low Earth Orbit (LEO) satellite constellations enable global connectivity, but face unique routing challenges due to dynamic topologies and limited resources. Routing performance is mostly influenced by two conflicting metrics: the network load and the end-to-end latency. The former is closely tied to reliability and throughput, whereas the latter directly regards the user-perceived responsiveness. This paper introduces two bi-objective linear programming models for routing in LEO constellations, jointly minimizing congestion and path latency. While propagation delay is essentially fixed by links geometry, queuing delay constitutes the main variable contributor to latency. The novelty of this work lies in integrating this delay into a model through a piece-wise linear function of link load, capturing its impact on end-to-end performance. While this coupling implies that latency partially reflects load, minimizing latency alone does not ensure network-wide load balancing since shortest-path routing tends to concentrate traffic on a limited set of links; the proposed framework tackles this trade-off and enables a rigorous analysis of the conflict between load balancing and latency.

Index Terms—Optimization, LEO Satellites, Routing, Linear Programming, QoS

I. INTRODUCTION

Large-scale Low-Earth Orbit (LEO) satellite constellations, often referred to as mega-constellations, have emerged as a new paradigm for global broadband connectivity. Systems such as Starlink, Kuiper, or OneWeb deploy thousands of satellites orbiting at altitudes typically ranging from 340 to 1200km, organized in multiple orbital shells and planes. Unlike their geostationary predecessors, these constellations leverage proximity to the Earth surface to offer significantly lower propagation delays, on the order of a few milliseconds per hop, thereby positioning themselves as credible candidates for latency-sensitive applications. End-user traffic enters and exits the constellation through ground stations (i.e., gateways or Points of Presence) that establish radio links with the satellites passing overhead, inherently transient due to the high orbital velocity of LEO satellites. Thus, periodic handovers must be performed to transfer an active session from one satellite to the next as the constellation evolves, resulting in a dynamic network topology that continuously reconfigures on timescales of seconds to minutes.

Routing in such an environment raises challenges that go well beyond those encountered in terrestrial networks. The time-varying topology, the heterogeneity of link characteristics and the scale of the network (potentially tens of thousands of nodes) require routing strategies that are both computationally efficient and physically meaningful. Most existing approaches

optimize routes with respect to a single metric, predominantly latency, providing an incomplete view of the end-to-end performance: a minimum-latency path that crosses highly congested links may deliver poor goodput (i.e., application level throughput) leading to poor application performance. Conversely, a path that only maximizes available capacity may cause latencies incompatible with real-time service requirements.

This contradiction between latency and congestion appears at the transport layer. Modern congestion control algorithms, including widely deployed variants of TCP, determine their sending rate as a direct function of both the round-trip time (RTT) and the observed packet loss or queuing delay. For instance, in loss-based TCP variants, the steady state throughput is commonly approximated as inversely proportional to both the RTT and the square root of the loss rate. In rate based schemes such as BBR, the estimated bandwidth-delay product governs the sending rate, making it explicitly dependent on both the path latency and the available capacity. Consequently, a routing decision that minimizes latency at the expense of crossing a congested path may paradoxically reduce the achievable transport-layer throughput due to increased loss or queuing, while a decision that avoids congestion but inflates the RTT may equally degrade performance. Latency and capacity thus act as antagonistic metrics from a routing point of view: improving one often comes at the cost of the other. This motivates the need for a routing framework capable of navigating this trade-off, as the one presented in this paper, optimizing jointly latency and link utilization.

Our approach joins multiple objectives into a unique model to address the static routing problem. This enables the selection of paths that balance low latency with sufficient load-distributing. Furthermore, by incorporating a queuing delay penalization reflecting the current load on each link, we introduce a first step toward congestion impact awareness.

This paper is organized as follows. Section II presents the satellite system model and the proposed routing metrics. Section III reviews related work on routing in LEO constellations. Section IV details the routing optimization models. Section V presents simulation results illustrating the latency-capacity trade-off, and Section VI concludes the paper. A discussion on the limits of the models and on future work objectives is proposed in Section VII.

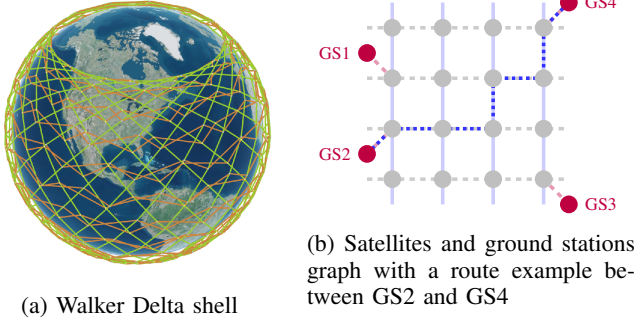


Fig. 1: Satellite shell of a LEO constellation

II. SYSTEM AND PROBLEM PRESENTATION

A. LEO constellation topology

Modern LEO satellite constellations are commonly deployed using Walker Delta configurations, typically organized into multiple shells, each defined by a common orbital altitude and inclination. Within a shell, satellites are distributed across several orbital planes, with a fixed number of satellites per plane moving in the same direction. Satellites are interconnected via Inter-Satellite Links (ISLs), enabling both intra-plane communications among adjacent satellites in the same orbit and inter-plane communications between satellites in neighboring orbital planes. Since all satellites within a shell orbit at the same angular velocity, the shell can be represented as a regular mesh-like topology with stable connectivity as shown in Fig. 1a. While the topology remains unchanged, the physical distances between satellites vary with latitude, decreasing near the poles as the orbital planes become closer.

The primary objective of LEO constellations is to provide connectivity to terrestrial users. This is achieved through Ground-to-Satellite Links (GSLs), which connect ground stations to visible satellites. Due to their low altitude, LEO satellites move rapidly relative to the Earth's surface, resulting in frequent changes in satellite visibility and the necessity of regular handovers between satellites. Therefore, while the ISL topology within a shell remains largely stable, the overall network topology is highly dynamic, with GSL connections lasting only few minutes depending on the orbital altitude.

Satellite constellations are commonly modeled as graphs representing a snapshot of the network at a given instant [1] as illustrated in Fig. 1b. In this representation, vertices correspond to satellites and ground nodes, while edges represent active ISLs and GSLs. As GSLs evolve over time, successive graph snapshots capture the dynamic nature of the overall network.

B. Routing in a LEO constellation

LEO satellite networks support heterogeneous services ranging from best-effort Internet traffic to delay-sensitive applications such as video conferencing, real-time streaming, and remote-control systems. These applications exhibit diverse Quality-of-Service (QoS) requirements in terms of latency, throughput, reliability, and packet loss tolerance. Routing

determines the path followed by packets from source to destination within the constellation and, as a result, directly affects these performance requirements.

End-to-end latency is one of the most critical metrics. In packet-switched networks, latency is generally composed of four components: (i) *Processing Delay* corresponding to the time required by a node (satellite or ground station) to handle the packet (*i.e.*, inspect headers and determine the outgoing interface); (ii) *Transmission Delay* representing the interval required to push the packet's bits onto the physical link, determined by the packet size and the link's bandwidth; (iii) *Queuing delay* constituting the time packets spend waiting in buffers before transmission; (iv) *Propagation Delay* being the time required for a signal to travel across the physical medium, it is strongly related to the distance and the speed of the link. The end-to-end latency of a route is obtained by summing these delays over all nodes and links along the selected path. Particularly in LEO constellations, propagation delays are largely determined by the physical distances between the objects (satellites or ground stations), whereas queuing delays depend on the current traffic load.

Another major indicator of network performance is link utilization. Since both ISLs and GSLs have finite transmission capacities, traffic demands exceeding the available capacity may generate congestion and increase queue occupancy. The resulting queuing delays can significantly impact end-to-end latency and may lead to packet losses due to buffer overflow.

Therefore, efficient routing strategies should not focus solely on minimizing path lengths but also account for load balancing in order to limit queuing delays and packet losses. In this paper, we consider the *static routing problem*, in which routing decisions are computed on a single graph snapshot of the constellation. Although real LEO networks evolve continuously over time, this snapshot-based model serves as the basis in understanding the behavior of the proposed optimization models. The dynamic aspects of the routing problem are left for future work; some directions are discussed in Section VII.

III. RELATED WORK

Routing in LEO satellite constellations raises new challenges that differ significantly from those of terrestrial networks (*e.g.*, handover with ground stations, limited onboard resources, ...). These specificities which emerged with the development of large satellite constellations [2], require substantial adaptations of existing routing approaches so that they can scale to networks of thousands of nodes while accounting for various, often conflicting, criteria [3]. Yet most existing approaches address only specific aspects of the problem and do not aim to jointly optimize all relevant performance metrics and resource utilization; moreover, traffic monitoring today is typically performed at the satellite scope rather than across the whole network. The goal of this work is to propose a centralized optimization approach for load-balancing across the satellite network.

In existing centralized solutions, the Shortest Path (SP) routing is widely used particularly to optimize the latency [4].

Some efficient algorithms exist such as Dijkstra's [5] or other Dijkstra's based implementations [6], optimizing one criterion only, generally latency.

Alternatively, the routing problem can be formulated as a Multi-Commodity Network Flow (MCNF) optimization problem where the primary objective is often to minimize the link utilization and balance network load. In this problem, several commodities must be routed from their respective source to destination nodes across a network, while satisfying demand requirements and network constraints such as arc capacity limitations. In the context of our work, the routing problem is modeled as an Unsplittable MCNF (UMCNF) [7], [8], in which each service corresponds to a commodity defined by a single source–destination pair, and each demand must be routed along a single and unsplittable path. In the telecommunication literature, this problem is also known as the Bandwidth Packing Problem (BPP) [9].

These optimization problems have different objective functions to model the throughput allocation optimization. As the goal is generally to avoid congestion, occurring when an arc exceeds its capacity, minimizing the overload is a natural choice. In [10], this objective function was proposed to develop an efficient heuristic, competitive with SP methods in terms of computation time and providing a good approximation of the optimum routing with respect to the load [11]. The authors also extended this model to solve the dynamic problem in [12].

Exploring these problems with the consideration of two opposed metrics is quite recent. For instance, [13] propose a MCNF model with two opposite objectives and multiple variables and constraints to model the transportation system. Other approaches consist in expressing a cost as a function of a load, such as [14], using a piece-wise linear cost function included in the UMCNF model in order to find the best trade-off between the QoS metrics. This cost is nevertheless a generic metric that does not adapt naturally to the context of the conflict between latency and capacity in LEO constellations.

In the domain of telecommunications, some papers have early started to integrate queuing delay expression and constraints in the models. This consideration has been seen in UMCNF [15] as well as in BPP models [16], [17], [18]. In these propositions, the queues on the links are considered as M/M/1 queues, then the models use the exponential formula to interpret the queuing delay.

The present paper aims to combine the two conflicting optimization criteria (latency and link load) within a static routing model by considering a penalization of the queuing delay based on the load. This combination leads to a novel model applied to the specificity of LEO constellations.

IV. ROUTING MODELS

In this section, two bi-objective Linear Programs (LP) for the static routing problem in LEO satellite constellations are presented. Both formulations are built upon the UMCNF model and aim to jointly optimize network load and end-to-end performance through different latency representations.

A. Definitions and notations

The LEO constellation network is represented as an oriented graph, $N = (V, A)$, where $V = V_S \cup V_G$ denotes the set of vertices, comprising both satellites (V_S) and ground stations (V_G). The set of arcs A represents the directed links where $u_a \in V$ is the origin and $v_a \in V$ the destination node of arc $a \in A$. Each arc a is associated with a positive propagation delay denoted d_a and a positive link capacity denoted c_a . For sake of simplification, we assume constant processing delay and fixed packet length resulting in constant transmission delay. Consequently, these delays are not considered in the proposed models. The set of commodities, K , defines all the services to be routed where, for a service $k \in K$, $S^k \in V_G$ is the starting ground station, $T^k \in V_G$ is the destination ground station and r^k is the required flow throughput for service k . The problem addressed here is to allocate the throughput r^k on a single route linking S^k and T^k for each service k .

B. Basic flow constraints

To find routes, the binary variables x_a^k indicate whether the path of service k passes through arc a . These binary variables ensure that a unique path is assigned to each service. The following basic flow constraints are defined to describe the routes from S^k to T^k . For each service k and node w :

$$\sum_{a \in A, v_a = w} x_a^k - \sum_{a \in A, u_a = w} x_a^k = \begin{cases} 1 & \text{if } w = T^k \\ -1 & \text{if } w = S^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

C. Load objective

With x variables, the load on the arc a , denoted l_a , is expressed as follows :

$$l_a := \sum_{k \in K} x_a^k r^k \quad (2)$$

In order to avoid congestion, a commonly adopted objective is the minimization of the Maximum Link Utilization (MLU). Routing solutions obtained by minimizing MLU with respect to flow constraints solely yield the theoretical minimum MLU value, which is an important indicator of the global network utilization. The MLU is expressed as:

$$MLU := \max_{a \in A} \frac{l_a}{c_a} \quad (3)$$

D. Latency objective

In absence of congestion, the end-to-end latency of a path corresponds to the sum of propagation delays along all arcs crossed. For the service k , the corresponding latency L^k is:

$$L^k := \sum_{a \in A} x_a^k d_a \quad (4)$$

Determining routes with minimized latencies is equivalent to solving the Shortest Path (SP) problem with propagation delay used as the link weight. However, if additional constraints or objectives are introduced, traditional SP algorithms become insufficient to obtain optimal solutions: linear optimization

models become necessary. To avoid potential disparities in the route lengths, the latency of each route is normalized by the corresponding shortest-path latency, L_{SP}^k . These normalized values are then aggregated to form a global latency score, L_{SCORE} . Minimizing this score aims to keep the latency of each route as close as possible to its shortest-path latency.

$$L_{SCORE} := \sum_{k \in K} \frac{L^k}{L_{SP}^k} \quad (5)$$

E. Routing in LEO constellation: bi-objective models

Considering latency and link utilization as separate objectives provides only a partial solution to the routing problem. Those two criteria are closely related, as high link utilization may result in congestion, which subsequently increases latency. Therefore, this section proposes two bi-objective routing models accounting for both aspects and identifying a suitable trade-off between efficient load distribution and low latency, one of which incorporating also a queuing delay penalization.

a) *Bi-objective Load - Latency with Propagation delay (BoLLP) model*: A first approach is to formulate a model that combines both metrics into a single objective function. A weight factor is added to equilibrate the contribution of each objective, as MLU is close or greater than 1 under congestion, whereas L_{SCORE} reaches $|K|$ at its optimum (shortest path for all services in K). The resulting objective is given by:

$$\min \quad |K|MLU + L_{SCORE} \quad (6)$$

b) *Bi-objective Load - Latency with Queuing delay (BoLLQ) model*: A refined modeling approach consists of directly incorporating the effect of congestion on latency. While most models of this type represent network links as M/M/1 queues and derive delays as a function of traffic load with the known exponential formula [17], we define a piecewise linear (PWL) cost to capture this effect linearly.

In our model, the PWL function q reprimands the latency penalty due to link overload as a monotonically increasing function of the load-to-capacity ratio ρ_a . When the link operates below capacity (ρ_a below 1), $q(\rho_a)$ remains minimal; in contrast, for higher ρ_a , $q(\rho_a)$ grows to impose severe penalties, ensuring that higher overloads incur disproportionately larger queuing delays and thus discouraging link saturation.

While q is not designed to capture exact queuing delays, it aims to provide a resilient congestion penalty. To this end, its definition is grounded in the *full queuing delay*, representing the waiting time when the buffer is fully occupied. This value is determined by the buffer size (*i.e.*, number of packets it can contain), packet length (*i.e.*, bytes per packet), and link capacity, which are fixed parameters in our model. Thus, the full queuing delay for an arc a is expressed by:

$$Full_qd_a := \frac{Packet_Length \times Buffer_Size}{c_a} \quad (7)$$

Now, the latency formulation is enhanced by incorporating the queuing delay cost q . Consequently, the latency of a path is computed as the sum of both propagation and queuing

delay costs along the traversed arcs. To maintain a linear formulation, two auxiliary variables are defined. The variable q_a corresponds to the queuing delay cost on arc a , determined by the q function based on the load ratio ρ_a , whereas q_a^k is the accounted cost for service k on arc a . q_a^k equals q_a if the path of service k contains arc a and 0 otherwise. These variables are introduced by the two following constraints, Max_q being the maximum value of q :

$$q_a^k \leq q_a \quad (8)$$

$$q_a^k \geq q_a - Max_q(1 - x_a^k) \quad (9)$$

The latency definitions integrating the queuing delay cost are given below:

$$L_q^k := \sum_{a \in A} x_a^k d_a + q_a^k \quad (10)$$

$$L_{SCORE_q} := \sum_{k \in K} \frac{L_q^k}{L_{SP}^k} \quad (11)$$

Finally, the *BoLLQ* model can be formulated as follows:

$$\begin{aligned} \min \quad & |K|MLU + L_{SCORE_q} \\ \text{s.t.} \quad & (1), (8), (9) \end{aligned}$$

It is important to note that the formulation remains bi-objective by maintaining the MLU component in addition to L_{SCORE_q} , which already accounts for network load through the queuing delay costs. Indeed, minimizing the latency of individual paths does not necessarily lead to balanced resource utilization across the network, as such an objective inherently favors shortest paths. Consequently, the congestion-related objective must be preserved to ensure global load balancing.

V. PERFORMANCE ANALYSIS

This section evaluates, in terms of load and latency, the proposed routing optimization models: (1) the *BoLLP* model, a bi-objective formulation where the latency calculation accounts solely for propagation delay, and (2) the *BoLLQ* model, which integrates queuing delay into the formulation. For comparison purpose, (3) the *ALL_SP* model, optimizing latency only by accounting all shortest paths and (4) the *MIN_MLU* model, which minimizes the MLU under flow constraints are also considered.

A. Evaluation setup

The evaluation is conducted on a satellite constellation comprising 13 orbital planes, each hosting 13 satellites, connected with the ground users via 100 ground stations located in major global cities, then non-uniformly distributed across the Earth's surface. We assume the link capacity is 1000 Mbps on all links. The satellite network is generated using *SatComTopology* simulator [19], a Python-based framework designed to generate realistic satellite communication topologies. The LP models are optimized thanks to Gurobi solver [20].

Service demands are randomly generated and uniformly distributed across the network. The required throughput r_k is



Fig. 2: MLU resulted for the four considered models for different number of services and throughput requirements

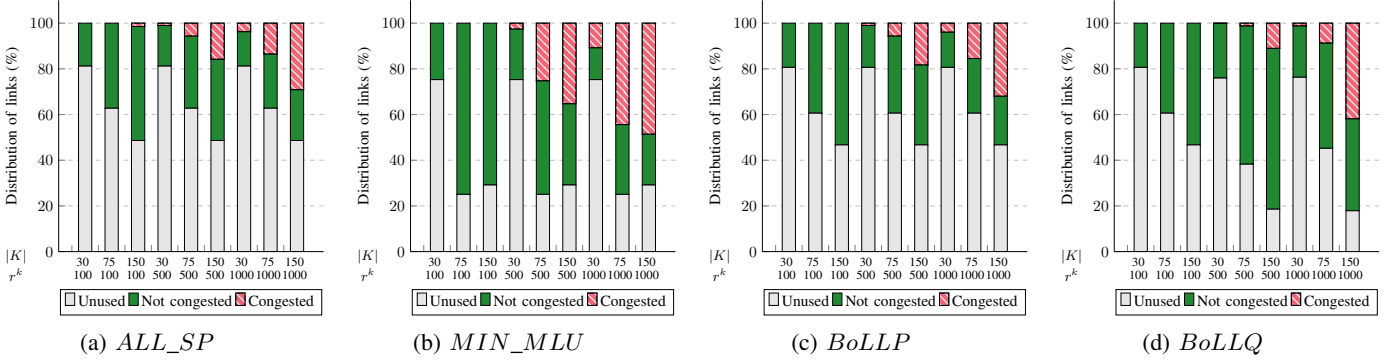


Fig. 3: Link utilization distributions for the four considered models

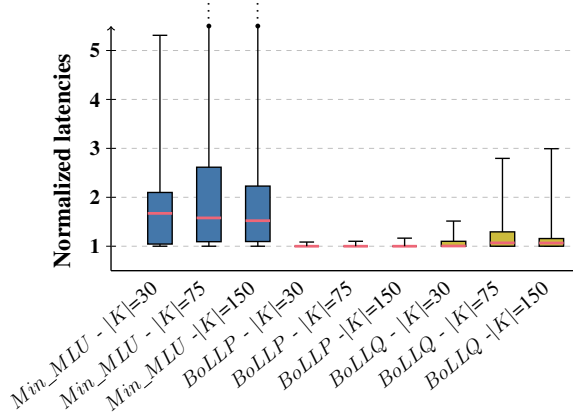


Fig. 4: Normalized latency distribution in solutions obtained on instances with service requirement $r_k = 500$ Mbps

constant across all services in an instance. The evaluation considers several instances with different network loads obtained by varying either the number of services or the throughput requirements of services sharing the same (S_k, T_k) pairs.

B. Results

First, it is essential to point out that, due to its computational complexity, the *BoLLQ* model could not produce results within a reasonable time frame for the most challenging instances, particularly those with high service requirements r_k . For these cases, a time limit was imposed in the solver, resulting in suboptimal solutions. Nevertheless, these solutions still

provide valuable insights into the model's efficiency, aligning with the paper's contribution which focuses on proposing innovative models rather than optimizing solving methods.

The evaluation analyzes load and latency metrics with a primary focus on the MLU, illustrated in Fig. 2 considering all instances and models. First, Fig. 2a and 2b reveal that *ALL_SP* exhibits higher MLUs (*i.e.*, higher congestion) compared to *MIN_MLU*. Indeed, *ALL_SP* tends to route multiple paths with close (S_k, T_k) pairs over the same links, whereas *MIN_MLU* distributes traffic to minimize link load, resulting in congestion only in high-traffic instances. In contrast, both *BoLLP* and *BoLLQ* (Fig. 2c, 2d) achieve optimal MLUs, demonstrating their effectiveness in load optimization.

The distribution of link utilization is further analyzed in Fig. 3, categorizing arcs into unused, not congested, and congested for the selected instances. *ALL_SP* uses fewer arcs than *MIN_MLU*, but with higher link utilization potentially leading to increased queuing delays. Conversely, *BoLLP* and *BoLLQ* almost match *ALL_SP* in arc usage while achieving lower congestion, as highlighted by their optimal MLU performance. Notably, *BoLLQ* (Fig. 3d) exhibits the fewest congested arcs in most cases, suggesting a better performance than *BoLLP* and promising results in load distribution.

Regarding end-to-end latency, considering propagation delay only, *MIN_MLU* produces routes significantly longer than the other models in all tested instances. This trend can be observed in Fig. 4, that presents normalized latency distribution for instances with throughput requirement $r_k = 500$ Mbps. In all instances, 25% of the routes generated by

MIN_MLU exceed twice the SP length, reaching routes 5 to 24 times longer than SP which results in delays that are impractical for operational use. In comparison, $BoLLP$ and $BoLLQ$ achieve competitive performance with normalized latencies close to 1 and below 2, indicating routes near-optimal to SPs. The $BoLLQ$ model exhibits slightly higher latency values due to its greater sensitivity to load. By prioritizing the impact of load on latency over absolute latency minimization, it tends to produce routes with reduced queuing delays, improving the end-to-end performance though.

VI. CONCLUSION

In this work, bi-objective optimization models for routing in LEO satellite constellations have been developed, with a focus on integrating queuing delay as a piecewise linear function of link load. Our analysis reveals a fundamental trade-off between latency and link utilization, highlighting the inherent conflict between minimizing end-to-end latencies and ensuring balanced network load.

The results demonstrate that bi-objective models are highly effective in reconciling these competing objectives. The resulting routes achieve latency close to those of shortest-path solutions while significantly reducing congestion compared to other methods. By explicitly accounting for queuing delays, our approach provides a more realistic representation of network behavior and enables routing decisions that better adapt to traffic conditions in LEO constellations. This work advances the theoretical understanding of routing in satellite networks and offers an insight into the integration of traffic optimization techniques within space-based communications systems.

VII. LIMITS AND FUTURE WORKS

The current problem formulation considers centralized routing under a known service scenario. As such, it does not account for several factors inherent to real-world LEO satellite networks, including potential disruptions in satellite constellations, traffic fluctuations and variations in link capacity.

Furthermore, while the bi-objective models tackle static routing, the operational problem is dynamic. A step-by-step optimization approach is adapted, but viable only if efficient solving methods are employed, as the heuristic proposed in [12].

Finally, exploring effective optimization methods, such as decompositions or heuristics, is also essential to evaluate the model on realistic instances and meet requirements of operational scenarios. Indeed, representative models necessitate large-scale networks and communication scenarios, introducing additional variables and constraints that may result in computationally intractable optimization models.

REFERENCES

- [1] M. Werner, "A dynamic routing concept for ATM-based satellite personal communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 8, pp. 1636–1648, 1997.
- [2] V. Gounder, R. Prakash, and H. Abu-Amara, "Routing in LEO-based satellite networks," in *1999 IEEE Emerging Technologies Symposium. Wireless Communications and Systems*, Apr. 1999, pp. 22.1–22.6. [Online]. Available: <https://ieeexplore.ieee.org/document/897340>
- [3] C. Li, Y. Zhang, Z. Cui, Y. Zhang, J. Liu, Z. Yu, and P. Zhang, "An Overview Of Low Earth Orbit Satellite Routing Algorithms," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, Jun. 2023, pp. 866–870. [Online]. Available: <https://ieeexplore.ieee.org/document/10182766/references>
- [4] Q. Chen, L. Yang, Y. Zhao, Y. Wang, H. Zhou, and X. Chen, "Shortest Path in LEO Satellite Constellation Networks: An Explicit Analytic Approach," *IEEE J.Sel. A. Commun.*, vol. 42, no. 5, pp. 1175–1187, May 2024. [Online]. Available: <https://doi.org/10.1109/JSAC.2024.3365873>
- [5] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959. [Online]. Available: <https://doi.org/10.1007/BF01386390>
- [6] W. Peng, X. Hu, F. Zhao, and J. Su, "A Fast Algorithm to Find All-Pairs Shortest Paths in Complex Networks," *Procedia Computer Science*, vol. 9, pp. 557–566, Dec. 2012.
- [7] C. Barnhart, C. A. Hane, and P. H. Vance, "Integer multicommodity flow problems," in *Integer Programming and Combinatorial Optimization*, W. H. Cunningham, S. T. McCormick, and M. Queyranne, Eds. Berlin, Heidelberg: Springer, 1996, pp. 58–71.
- [8] T. Takeuchi and N. Shinomiya, "An Approximate Solution Using Markov Approximation for a Load Balancing Problem in Communication Networks," in *2025 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)*. Seoul, Korea, Republic of: IEEE, Jul. 2025, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/11137741/>
- [9] R. B. C. Butkeraites, L. L. de Salles Neto, and M. Gendreau, "A sampling-based multi-objective iterative robust optimization method for Bandwidth Packing Problem," *Expert Systems with Applications*, vol. 203, p. 117337, Oct. 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417422006947>
- [10] F. Lamothe, E. Rachelson, A. Häit, C. Baudoin, and J.-B. Dupé, "Randomized rounding algorithms for large scale unsplittable flow problems," *Journal of Heuristics*, vol. 27, no. 6, pp. 1081–1110, Dec. 2021.
- [11] P. Grislain, A. Auddino, A. Barraqué, F. Lamothe, O. Hotescu, J. Lacan, J. Radzik, and E. Lochin, "Rethinking LEO constellations routing," *International Journal of Satellite Communications and Networking*, p. sat.1514, Apr. 2024. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/sat.1514>
- [12] F. Lamothe, E. Rachelson, A. Häit, C. Baudoin, and J.-B. Dupé, "Dynamic unsplittable flows with path-change penalties: New formulations and solution schemes for large instances," *Computers & Operations Research*, vol. 152, p. 106154, Apr. 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054823000187>
- [13] B. Bevrani, R. Burdett, A. Bhaskar, and P. K. D. V. Yarlagadda, "A multi-criteria multi-commodity flow model for analysing transportation networks," *Operations Research Perspectives*, vol. 7, p. 100159, Jan. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221471602030049X>
- [14] B. Fortz, L. Gouveia, and M. Joyce-Moniz, "Models for the piecewise linear unsplittable multicommodity flow problems," *European Journal of Operational Research*, vol. 261, no. 1, pp. 30–42, Aug. 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221717300863>
- [15] N. Huin, J. Leguay, and S. Martin, "Génération de colonnes pour le problème de routage à délai variable," in *ALGO 2020 – 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, Lyon, France, Sep. 2020. [Online]. Available: <https://hal.science/hal-02875446>
- [16] E. Rolland, A. Amiri, and R. Barkhi, "Queueing delay guarantees in bandwidth packing," *Computers & Operations Research*, vol. 26, no. 9, pp. 921–935, Aug. 1999. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054898000963>
- [17] N. Vidyarthi, S. Jayaswal, and V. B. Tirumala Chetty, "Bandwidth packing problem with queueing delays: modelling and exact solution approach," *Journal of Global Optimization*, vol. 65, no. 4, pp. 745–776, Aug. 2016. [Online]. Available: <https://doi.org/10.1007/s10898-015-0399-8>
- [18] S. Kim and C. Lee, "A branch and price approach for the robust bandwidth packing problem with queueing delays," *Annals of Operations Research*, vol. 307, no. 1, pp. 251–275, Dec. 2021. [Online]. Available: <https://doi.org/10.1007/s10479-021-04292-w>
- [19] "SatComTopology," <https://github.com/SatComProjects/satComTopology>.
- [20] "Gurobi," <https://www.gurobi.com/>.