IEEE Open Journal of the Communications Society

Received 13 February, 2024; revised 18 March, 2024; accepted 27 March, 2024; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/OJCOMS.2024.xxxxxx

Enhancing satellite Non-Terrestrial Networks through advanced constellation management: Optimizing in-orbit resources for NB-IoT

ARNAU SINGLA[1,2], ANNA CALVERAS[2], FRANCESC BETORZ[1] AND JOAN A. RUIZ-DE-AZUA[1]

[1]Space Communications Research Group, i2CAT Foundation, Barcelona, Spain
[2]Department of Network Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain

CORRESPONDING AUTHOR: Arnau Singla (e-mail: arnau.singla@i2cat.net).

This work was supported by the Generalitat de Catalunya; the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union – NextGeneration EU; the Spanish MCIN/AEI/ 10.13039/501100011033 through project PID2019-106808RA-I00; and by Departament d'Innovació, Universitats i Empresa, Generalitat de Catalunya 2021 SGR 00330.

ABSTRACT This paper presents a novel autonomous management system for satellite-based Non-Terrestrial Networks (NTN). It addresses the challenges of managing the operations of Low Earth Orbit (LEO) constellations for the Internet of Things (IoT), emphasizing intermittent connectivity and resource constraints. This work contributes with the description of the scheduler module alongside a customized formulation of a novel multi-objective optimization problem and its associated mathematical framework. The resultant optimization algorithm is elaborated. This Constellation Management System (CMS) is validated through a real-world scenario, revealing substantial improvements of up to 40% in task assignment and throughput rates when compared to a best-effort baseline. This study underscores the CMS's potential to enhance IoT communication with LEO constellations through resource-efficient and business-aware satellite operations scheduling.

INDEX TERMS Autonomous management, IoT, Non-Terrestrial Networks, Operations scheduling, Resource optimization, Satellite communications.

I. Introduction

Integrating satellite systems with terrestrial telecom infrastructures, a goal pursued through initiatives such as the Third Generation Partnership Project (3GPP), has gained prominence. This is evident in endeavors like the development of Non-Terrestrial Networks (NTN) and the expansion of radio protocols like New Radio (NR) and Narrow-Band Internet of Things (NB-IoT) for satellite connectivity [1] [2]. This trend is also reflected in the research community, where concerted efforts are underway to establish a seamless integration between satellite and terrestrial networks, commonly referred to as the Satellite-Terrestrial Integrated Network (STIN) [3]. A significant focus lies on augmenting the growing Internet of Things (IoT) ecosystem with satellite connectivity facilitated by NTN. In pursuit of this integration, both Low Earth Orbit (LEO) and Geosynchronous Equatorial Orbit (GEO) solutions are being explored [4]. This paper focuses on LEO constellations as NTN. Often deployed to enable global coverage, these constellations may use Store and Forward (S&F) mechanisms for data delivery in sparse constellations [5]. However, managing the operations of these LEO constellations, particularly for telecom purposes, presents significant challenges. The complexity arises from integrating diverse elements of a heterogeneous environment, the resource-constrained nature of New Space satellite platforms, and the need to integrate mobile network business criteria into satellite operations.

Satellite mission planning has historically been the purview of governmental space agencies, leveraging sophisticated tools such as the Advanced Planning and Scheduling Initiative (APSI) developed by the European Space Agency (ESA) [6], and the Automated Planning/Scheduling

Environment (ASPEN) and the Extensible Universal Remote Operations Planning Architecture (EUROPA) by the National Aeronautics and Space Administration (NASA) [7] [8]. A pivotal shift occurred at the dawn of the 20th century, marked by the proliferation of commercial satellite constellations. With this surge, private companies emerged as key players, developing their own mission planning tools or adapting existing ones from governmental agencies [9]. In contemporary times, notable commercial constellations, like those from Planet, have garnered significant attention. Despite their advancements, challenges persist in satellite mission planning [10]. Therefore, current research in the field aims to streamline and automate mission planning processes like the Hands-Off Operations Platform (HOOP) tool [11].

Existing planning tools from space agencies suffer from a lack of adaptability, often remaining static after the conclusion of specific missions. Conversely, commercial solutions pose limitations due to their closed and proprietary nature, hindering their utility for research and innovation. Moreover, current commercial constellations rely on in-house management tools, often incorporating semi-manual procedures, which can introduce inefficiencies and complexities into operations. Furthermore, while research in satellite mission planning is ongoing, a notable gap exists in the coverage of telecommunication constellation use cases compared to Earth Observation (EO) missions. Addressing these challenges is imperative for advancing the state of the art in satellite mission planning and realizing the full potential of commercial satellite constellations across various domains.

The need for efficient and autonomous management systems for the New Space telecom LEO constellations becomes clear in this context. Though available, traditional satellite-by-satellite operations and commercial solutions often fall short of meeting the intricate requirements of this evolving landscape. As the focus of current operations management research tends to gravitate toward EO missions, tailored solutions are needed to effectively address the distinct challenges of the telecom use case within the New Space paradigm [12]. Therefore, the authors proposed an autonomous and reactive constellation management system, integrated with the 6th Generation (6G) Core Network (CN) to enhance the NTN architecture by extending IoT roaming services with a LEO satellite constellation using S&F. This innovative system and its architecture referred to as the Constellation Management System (CMS), is described in [13].

The CMS aims to bridge the gap between satellite and network operations, particularly within a satellite-based NTN. Given that satellite operations directly influence network service quality within such environments, it is noteworthy that, to the authors' knowledge, no existing management framework for satellite operators adequately addresses this impact on service provision. The novelty of the CMS is that it addresses this gap by introducing a novel approach that not only considers the ramifications of satellite operations on service delivery but also provides a mechanism for integrating all requisite agents to automate NTN operations. Additionally, the CMS facilitates seamless integration of terrestrial networks with satellite-based NTN, offering the capability to generate 3GPP-compliant operation plans.

This paper contributes to thoroughly describe the scheduler module, a pivotal component within the CMS architecture. Furthermore, the paper delves into formulating a novel Multi-Objective Optimization (MOO) problem, specifically tailored for the satellite and Ground Station (GS) task scheduling problem in the NTN use case. A detailed mathematical framework is presented to address this optimization challenge. This work also presents the optimization algorithm derived from the model and mathematical formulation, showcasing its relevance and applicability within the CMS framework. To substantiate the proposed solutions, the paper offers a validation phase utilizing a real-world scenario by a satellite operator named Sateliot [14]. The performance of the CMS is assessed through a comparative analysis, pitting its capabilities against those of a manual best-effort operator, thus providing insights into its efficacy and potential advantages.

II. Challenges

In resource-constrained constellations, the intricate relationship between satellite operations and service provision becomes particularly pronounced. The challenge lies in maintaining a delicate balance, where the limitations of available resources must be navigated to ensure the seamless delivery of intended services. A central predicament in this landscape revolves around the formulation of operations plans that align with critical business metrics. Effectively integrating operational intricacies with overarching business objectives is imperative to optimize satellite performance and meet the desired service goals. The development of a comprehensive task schedule emerges as a crucial facet in addressing these challenges. This schedule not only dictates the sequence of satellite operation modes but also serves as the foundational contact plan for the satellites' routing and forwarding protocols, intricately connecting the operational and communicative aspects of satellite constellations. These challenges encompass various aspects such as architecture design, scenario modeling, algorithmic implementation, and space testing. This paper primarily deals with the challenges regarding scenario modeling, including (1) metrics and constraints for IoT scenarios, (2) managing link discontinuities, (3) incorporating business criteria and policies, and (4) modeling protocol procedures, particularly those specified by 3GPP in the context of NB-IoT.

Figure 1 shows the different procedural steps for user authentication and registration to a 3GPP network with a discontinuous link [5]. These steps are later used in the model as tasks. The figure shows that the authentication steps in a discontinuous NTN are separated at four different points in time, according to satellite contacts with the ground nodes. At t_1, during a contact between the satellite and

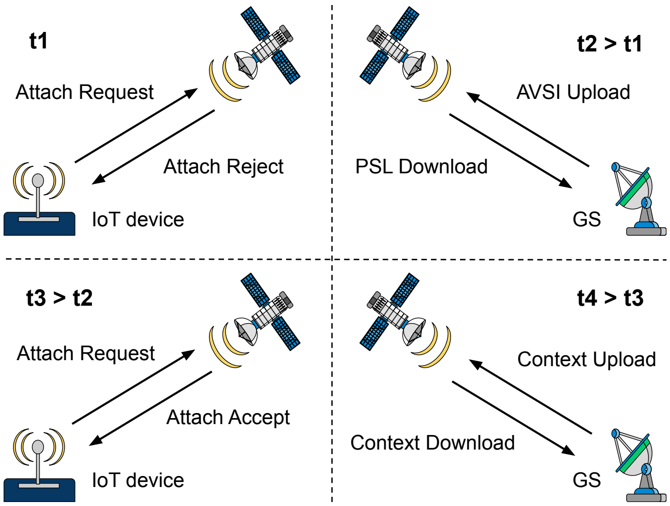

FIGURE 1. 3GPP authentication procedures with discontinuous link.

the IoT device or User Equipment (UE), the device sends an attach request to the satellite. Since the satellite does not have connectivity with a GS connected to the CN at that moment, the satellite rejects the request while storing the device for pending authentication. At t_2, upon contact between the satellite and a GS with connection to the CN, the satellite requests authentication to the CN on the ground of the pending devices (Pending Subscriber List (PSL)). The CN provides the satellite with the Authentication Vectors and Subscriber Information (AVSI) of the subscribed devices. At t_3, the satellite has another contact with the previous device from t_1. If the device was subscribed and the satellite is carrying its AVSI, the attach request will result in the satellite accepting the request and registering the device. Finally, at t_4 the satellite has a contact with a GS. The satellite downloads the generated contexts from newly registered devices to the CN on the ground. During these contacts, the CN may exchange context updates with the satellite from registered devices.

III. Scheduler model

Within the CMS, a modular architecture comprising various modules is employed, with particular emphasis in this study placed on the scheduler module. This pivotal module generates and optimizes task plans disseminated to different system agents, including GS, satellites, and CN. As previously discussed, the CMS is a conduit between the satellite and network operations. However, satellite constellations are ultimately managed by satellite operators. Therefore, the selected approach leverages historical methodologies used by satellite operators, adapting them to encompass network operations requirements. Traditionally, satellite scheduling has been addressed through task scheduling methodologies, typically called the satellite task scheduling problem [15] [16]. Numerous techniques exist for addressing this challenge, with Constraint Satisfaction Programming (CSP) emerging as the chosen method within the CMS framework. It's imperative to note that the primary focus of this study is not to ascertain the optimal solution for the satellite scheduling problem per se, but rather to tailor it to accommodate network metrics and constraints, thereby enhancing the efficiency of NTN operations.

The scheduler is implemented using an open-source scheduling framework named Optaplanner [17], which is built upon a constraint satisfaction engine. The model has three different hierarchical planning categories: (1) the planning solution, (2) the planning entity, and (3) the planning variable. The basic structure is that the planning solution object contains one or more planning entities, which in turn contain one or several planning variables. The planning solution object is both the problem and the solution of the optimizer. A planning solution with unassigned planning entities is called the scheduling problem, and a planning solution with planning variables assigned to their planning entities is a potential solution. This means that the input of the scheduler is a planning solution object with unassigned or partially unassigned planning entities, and the scheduler's output is a planning solution with its planning entities assigned. The planning variable object is the one that the solver will change during the optimization process. The solver changes the planning entity object by assigning different planning variables to it. Therefore, the planning variable object is the range of all the possible values the planning entity might assign.

The data structure of the implemented model is presented in Figure 2. The planning solution object, named *TaskAssignment*, has the following attributes.

- A list of task objects, which are the main planning entities of the model (TaskList). Tasks are assigned by setting a satellite, target, and start time.
- A list of satellite objects, which are planning variables (SatelliteList). These objects contain the visibility windows list obtained from the orbital propagation simulator and relevant information and resources.
- A list of target objects, which are planning variables (TargetList).
- The score, which holds the computed quality evaluated from the constraints of the current solution object, will determine the best solution (Score).

Note that the Task object has as attributes the three planning variables of the model: the assigned satellite, start time, and target. Additionally, each object has a series of attributes that do not change during solving time and are fixed for each object. These kinds of attributes are called problem variables. Examples of the proposed model are the task resource requirements (e.g., battery and memory), task type, and solving information like the heuristic difficulty of task assignment. There are 10 different task types currently modeled: Attach Request (AR), PSL download, Authentication Vectors (AV) upload, UE registration, context download, context upload, Mobile Originated Download (MODL), Mobile Originated

Upload (MOUL), Mobile Terminated Download (MTDL), and Mobile Terminated Upload (MTUL) data upload. These task types are defined according to 3GPP procedures and the S&F scenario (see Figure 1).

The satellite planning variable's attributes are the satellite's initial available resources, like memory and battery levels, overall satellite information like ID, and a list of windows with the coming contacts from that satellite with all the system's targets. These windows are defined with the corresponding target, start, and end times. The contact windows come from a simulator software for orbital propagation [18]. The start time is a pure integer object with no attributes.

It is also important to remark that from the three planning variable objects of the model, the satellite and target ones are unique. Note in figure 2 that one of their attributes is a list of the tasks they have been assigned to by the scheduler, named *TaskSet*. This attribute is key to evaluating model constraints for individual satellites and targets. This list of tasks must be automatically updated every time the solver changes the planning entity. As such, these two planning variables also change during solving time. This type of attribute (the *TaskSet*) is known as a shadow variable, and the planning variables are called a shadow planning entity because they now work as a planning entity, which changes during solving time, but only to update the *TaskSet* field.

A practical example is presented here to illustrate this data model. The algorithm within the scheduler changes the planning variables of the planning entities with each optimization step. The object containing all the planning variables, planning entities, and other necessary scenario data is the planning solution. For this specific model, the scheduler changes the satellite, target, and start time of one or more tasks in each step. This means that the satellite, target, and start time are the planning variables of each task, which are the planning entities of the planning solution.

A. Score and Constraints

The score attribute of the planning solution is computed by evaluating the defined system constraints with the current satellite, target, and start time assignation to the different tasks of the solution object. The solution score is one of the main parameters used by the optimization algorithm to generate new moves and take steps towards a higher score solution. Due to the nature of the different constraints of the considered system, the score is modeled as a hierarchical three-level structure. This means the highest score is the one with the higher first score level. In the case of a tie in the first score level, the higher score is the one with the higher second level score, and the same in the case of a tie in the second score level. The three score levels are called hard, medium, and soft scores:

- *Hard score*: The hard score is the aggregated score resulting from evaluating the feasibility constraints of the problem and the first score level. These constraints dictate whether or not a given solution is feasible. These include both temporal and physical constraints of the domain. All feasibility constraints are modeled to reduce the score if broken. Therefore, a feasible solution will have a hard score of zero, and an unfeasible solution will have a negative hard score.
- *Medium score*: The medium score is the aggregated score resulting from the evaluation of the business constraints of the problem. It is also the second priority level. These constraints represent the highest level of optimization criteria, and they are tied to the business goals of the constellation and service operator. The value of the business constraints rewards is always positive. Thus, the higher the value, the better the solution is.
- *Soft score*: The soft score is the aggregated score resulting from the evaluation of the operational constraints of the problem and the last priority level. These constraints are usually tied to improving the overall operation of the constellation and its survivability.

1) Feasibility constraints

This model implements the following feasibility constraints, which impact the hard score level:

- Temporal bounds: It bounds the start and end time of the tasks within a contact window from the simulation timespan.
- Task uniqueness: It ensures that the task is assigned a defined number of times. For example, a UE registration shall only be conducted by a single satellite once, and then the generated context shall be uploaded just once to each other satellite.
- Correct contact: It ensures that the assigned task is within a contact window between the assigned satellite and the correct task target.
- Task precedence: It ensures that the correct task types precede specific tasks or meet the correct sequence related to 5th Generation (5G) procedures and protocols. There are three implemented precedence constraints for the modeled scenario of S&F. The first one assures that a satellite tasked with downloading the PSL of a UE has been assigned earlier with an attachment request from that same UE. The second one assures that a satellite tasked with authenticating a UE upon an attachment request is carrying the AV for that UE, either because an earlier task has uploaded them or the satellite had them on board from the start of the simulation. The third one assures that every satellite with a data transfer task to/from a UE has this UE context registered on the local onboard database. This is modeled for the S&F scenario [5].
- Overlapping tasks: It ensures that no time-overlapping tasks are assigned to the same satellite or GS.

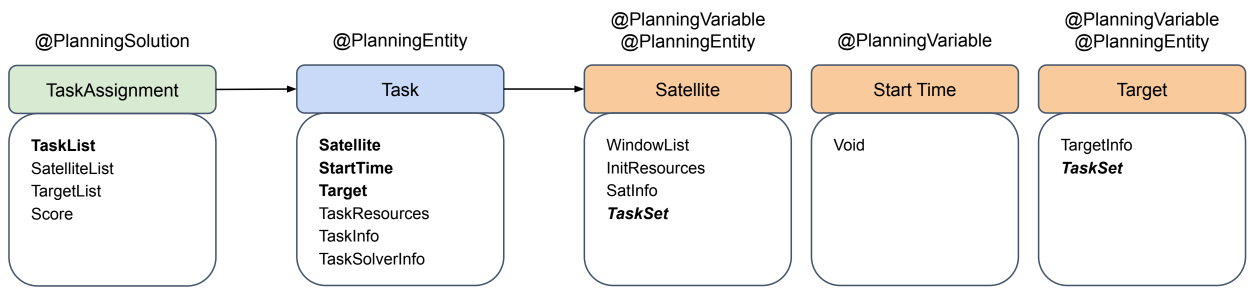

FIGURE 2. **Satellite scheduling problem data model**

- Resource bounds: It ensures that each satellite's specific energy and memory resource bounds are not broken with any assigned task.
- Target-specific memory: This constraint is similar to the memory resource bounds constraint but for specific UE data on-board the satellite. This assures, for example, that a satellite tasked with an Mobile Terminated (MT) download for a specific UE has some data previously stored targeting that same UE.

2) Business constraints

The model implements the following business constraints, which impact the medium score.

- Task assignation: It is necessary to solve the overconstrained nature of the problem. To give input to the scheduler a higher number of tasks that can be assigned, the scheduler has to be able to avoid assigning tasks. As such, the solver will naturally not assign any task to avoid constraint conflicts. To solve this, a reward has to be given in the second score level per task assigned. This way, the best score will have the most assigned tasks while not breaking any feasibility constraints.
- Task prioritization: The operator can prioritize specific tasks over others by tuning the reward scored when assigning those tasks.

3) Operational constraints

The model implements the following operational constraints, which impact the soft score.

- Early tasks: Rewards solutions with early start times assigned to their tasks. This way, a schedule will have all its tasks as early as possible.
- Final resource state: Rewards solutions with task sequences that leave the constellation satellites with better memory and battery states.
- Mean latency: Rewards solutions with lower overall mean latency in the assigned data transmission tasks.
- Synchronization time: Rewards solutions with the lower time between the start of the scheduling and the start time of the last of the pending synchronization tasks. A synchronization task is a task to update the local databases of the satellites.

IV. Mathematical formulation

This section mathematically defines the constraints and optimization criteria of the model.

Theorem IV.1 (MOO problem)**.** The MOO problem is expressed as follows [19]:

$$\begin{aligned} \text{Minimize / Maximize} \quad & F(x) = [F_1(x), F_2(x), ..., F_C(x)] \\ \text{subject to} \quad & g_a(x) \leq 0, \quad \forall a = 1, 2, ..., m \\ \text{and} \quad & h_b(x) = 0, \quad \forall b = 1, 2, ..., e, \end{aligned} \tag{1}$$

where $g_a(x)$ are inequality constraints, $h_b(x)$ are equality constraints and $F(x)$ is a vector of objective functions. Indices C, m, and e are the number of objective functions, inequality constraints, and equality constraints respectively.

To relate our model with this mathematical formulation of the MOO problem, the inequality and equality constraints are the hard constraints, and the objective function is composed of the medium and soft constraints.

First, the mathematical definitions required for the mathematical expression of the constraints and optimization criteria are presented. The model constraints are developed, and finally, the optimization criteria are developed.

A. Definitions

1) Indexes

To iterate through the main problem entities and variables, the following indexes are defined:

- $\forall i \in I \subset \mathbb{N}$ to represent each task within I maximum tasks.

- $\forall j \in J \subset \mathbb{N}$ to represent each contact time window within J maximum windows.
- $\forall k \in K \subset \mathbb{N}$ to represent each satellite within K maximum satellites.
- $\forall l \in L \subset \mathbb{N}$ to represent each node within L maximum nodes.
- $\forall y \in Y \subset \mathbb{N}$ to represent each user node within Y maximum user nodes.
- $\forall r \in R \subset \mathbb{N}$ to represent each task type within R maximum types.

2) Spaces

With these indexes, the following spaces are defined:

Definition A.1 (Task types)**.** This space is composed of the different task types of the model. The current task types of the model are defined here as AR, PSL download, AVSI upload, attach, Context Download (CDL), Context Upload (CUL), MOUL, MODL, MTUL, and MTDL.

$$\forall r \in R \quad Q = \{q_1, q_2, ..., q_{|R|}\} = \\ = \{AR, PSL, AVSI, Attach, CDL, CUL, \\ MOUL, MODL, MTUL, MTDL\} \quad (2)$$

Definition A.2 (Nodes)**.** This space is composed of the different nodes of a specific scenario. Each node is defined by the node type ρ, which in this model can either be a UE or a GS.

$$\forall l \in L \quad \Psi = \{\psi_1, \psi_2, ..., \psi_{|L|}\} \\ \psi_l = \{\rho_l\} \quad \rho_l = \begin{cases} 1 & \text{if } \psi_l \text{ is UE} \\ 0 & \text{if } \psi_l \text{ is GS} \end{cases} \quad (3)$$

Definition A.3 (User nodes)**.** This space is composed of the different user nodes of a specific scenario. Each user node is defined by the node type ρ. Since all nodes from this space are user nodes, ρ is always a UE.

$$\forall l \in L, \, \forall y \in Y \quad \Psi^{UE} = \{\psi_1^{UE}, \psi_2^{UE}, \psi_{|Y|}^{UE}\} \\ \psi_y^{UE} = \psi_l \mid \rho_l = 1 \quad (4)$$

Definition A.4 (Satellites)**.** This space is composed of the different satellites of a specific scenario. Each satellite is defined by the initial memory and energy state (m^{init} and e^{init}), the threshold memory and energy levels (m^{max}, e^{max}, and e^{min}), and a list of the onboard AVSI and contexts at the start of the simulation (A_k, and C_k).

$$\forall k \in K, \, A_k, C_k \subset Y \quad S = \{s_1, s_2, ..., s_{|K|}\} \\ s_k = \{m_k^{init}, e_k^{init}, m_k^{max}, e_k^{max}, e_k^{min}, A_k, C_k\} \quad (5)$$

Definition A.5 (Contact windows)**.** This space is composed of the different contact windows of a specific scenario. Each contact window is defined by the start time of the window τ^*, the end time of the window ϵ, the satellite s, and node ψ, which have potential contact opportunity during that time window.

$$\forall j \in J \quad \exists! k \in K, \, \exists! l \in L \quad W = \{w_1, w_2, ..., w_{|J|}\} \\ w_j = \{\tau_j^*, \epsilon_j, s_k, \psi_l\} \quad (6)$$

Definition A.6 (Tasks)**.** This space is composed of the different tasks of a specific scenario. Each task is defined by its start time τ, its memory and energy requirements (m and e), its duration δ, its type q, its assigned window w, its destination node ψ^{UE}, and its weight ϖ.

$$\forall i \in I \quad \exists! j \in J \quad \exists! y \in Y \quad T = \{t_1, t_2, ..., t_{|I|}\} \\ t_i = \{\tau_i, m_i, e_i, \delta_i, q_i, w_j, \psi_y^{UE}, \varpi_i\} \quad (7)$$

Definition A.7 (Specific type tasks)**.** This space comprises the all tasks of a specific type.

$$\forall i \in I, \, \forall r \in R \quad T^r \subset T \mid \varphi_i^r = 1 \quad (8)$$

Definition A.8 (Satellite chronologically-assigned tasks)**.** This space comprises the different tasks ordered chronologically and assigned to a specific satellite.

$$\forall i \in I, \, \forall k \in K \quad T^k \subset T \mid \nu_i^k = 1 \wedge \tau_i < \tau_{i+1} \quad (9)$$

Definition A.9 (Satellite, destination, and type chronologically-assigned tasks)**.** This space is composed of the different tasks of a specific type, ordered chronologically and assigned to a specific satellite and destination.

$$\forall i \in I, \, \forall k \in K, \, \forall y \in Y \, \forall r \in R \\ T^{k,y,r} \subset T^k \mid \zeta_i^y \cdot \varphi_i^r = 1 \quad (10)$$

3) Support parameters

The following support definitions are required for the spaces, constraints, and optimization criteria. For better understanding, all support parameters are summarized in Table 1.

Definition A.10. A parameter to indicate whether a contact window is assigned to a task is defined as follows:

$$\forall i \in I, \, \forall j \in J \quad \lambda_{i,j} = \begin{cases} 1 & \text{if } \exists j \mid w_j \subset t_i \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Definition A.11. A parameter to indicate whether a satellite is involved in a contact window is defined as follows:

$$\forall k \in K, \, \forall j \in J \quad \chi_{k,j} = \begin{cases} 1 & \text{if } \exists j \mid s_k \subset w_j \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Definition A.12. A parameter to indicate whether a node is involved in a contact window is defined as follows:

$$\forall l \in L, \, \forall j \in J \quad \beta_{l,j} = \begin{cases} 1 & \text{if } \exists j \mid \psi_l \subset w_j \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Definition A.13. A parameter to indicate whether a certain task is assigned to a certain satellite is defined using λ and

χ as follows:

$$\forall i \in I,\ \forall j \in J\ \forall k \in K$$

$$\nu_i^k = \sum_{j=1}^{J} \lambda_{i,j} \cdot \chi_{k,j} = \begin{cases} 1 & \text{if } t_i \text{ is assigned to } s_k \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Parameter ν is certain to be either 0 or 1 due to constraint 54.

Definition A.14. A parameter to indicate whether a certain task is assigned to a certain node is defined using variables λ and β as follows:

$$\forall i \in I,\ \forall j \in J\ \forall l \in L$$

$$\gamma_i^l = \sum_{j=1}^{J} \lambda_{i,j} \cdot \beta_{l,j} = \begin{cases} 1 & \text{if } t_i \text{ is assigned to } \psi_l \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Parameter γ is certain to be either 0 or 1 due to constraint 54.

Definition A.15. A parameter to express whether a certain task type is assigned to a certain task is defined as follows:

$$\forall i \in I\ \forall r \in R \quad \varphi_i^r = \begin{cases} 1 & \text{if} \quad q_r \subset t_i \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Definition A.16. A parameter to express whether at least one task of a certain type is assigned to a satellite is defined using ν and φ as follows:

$$\forall i \in I\ \forall k \in K\ \forall r \in R \quad \xi_k^r = \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{I} \nu_i^k \cdot \varphi_i^r \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Definition A.17. A parameter to express whether two tasks are assigned to the same satellite is defined as follows:

$$\forall i_1, i_2 \in I \mid i_1 \neq i_2,\ \forall k \in K$$

$$\phi_{i_1,i_2}^k = \begin{cases} 1 & \text{if} \quad \nu_{i_1}^k \cdot \nu_{i_2}^k = 1 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Definition A.18. A parameter to express whether two tasks are assigned to the same node is defined as follows:

$$\forall i_1, i_2 \in I \mid i_1 \neq i_2,\ \forall l \in L$$

$$\varrho_{i_1,i_2}^l = \begin{cases} 1 & \text{if} \quad \gamma_{i_1}^l \cdot \gamma_{i_2}^l = 1 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Definition A.19. A parameter to express whether the start time of a task comes earlier than the start time of another task is defined as follows:

$$\forall i_1, i_2 \in I \mid i_1 \neq i_2 \quad \alpha_{i_1,i_2} = \begin{cases} 1 & \text{if} \quad \tau_{i_1} < \tau_{i_2} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Definition A.20. A parameter to express whether the end time of a task comes earlier than the start time of another task is defined as follows:

$$\forall i_1, i_2 \in I \mid i_1 \neq i_2 \quad \vartheta_{i_1,i_2} = \begin{cases} 1 & \text{if} \quad \tau_{i_1} + \delta_{i_1} < \tau_{i_2} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Definition A.21. Two parameters to express the resource state of a satellite after n tasks are defined as follows:

$$\forall i, n \in I,\ \forall k \in K,\ m_k^{init} \in \mathbb{R}^+,\ m_i \in \mathbb{R}$$

$$m_n^k = m_k^{init} + \sum_{i=i}^{n} m_i \cdot \nu_i^k \tag{22}$$

$$\forall i, n \in I,\ \forall k \in K,\ e_k^{init} \in \mathbb{R}^+,\ e_i \in \mathbb{R}^-,\ E \in \mathbb{R}^+$$

$$e_n^k = e_k^{init} + \sum_{i=1}^{n} (e_i \cdot \nu_i^k + E(\tau_i - \tau_{i-1})) \tag{23}$$

Variable E represents the mean solar panels charge over time. Variables m_i and e_i represent the specific task memory and energy resource consumption respectively.

Definition A.22. Four parameters to check if resources are kept within the boundaries are defined as follows:

$$\forall n \in I,\ \forall k \in K \quad \eta_n^k = \begin{cases} 1 & \text{if} \quad m_n^k \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

$$\forall n \in I,\ \forall k \in K \quad \kappa_n^k = \begin{cases} 1 & \text{if} \quad m_n^k \leq m_k^{max} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

$$\forall n \in I,\ \forall k \in K \quad \mu_n^k = \begin{cases} 1 & \text{if} \quad e_n^k \geq e_k^{min} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

$$\forall n \in I,\ \forall k \in K \quad \sigma_n^k = \begin{cases} 1 & \text{if} \quad e_n^k \leq e_k^{max} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Definition A.23. A parameter to express whether a task i has user node y as a destination is defined as follows:

$$\zeta_i^y = \begin{cases} 1 & \text{if} \quad \psi_y^{UE} \subset t_i \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

Definition A.24. Two parameters to express whether a user node is within the node collection of A_k and C_k of a satellite are defined as follows:

$$B_y^k = \begin{cases} 1 & \text{if} \quad \psi_y^{UE} \subset A_k \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

$$G_y^k = \begin{cases} 1 & \text{if} \quad \psi_y^{UE} \subset C_k \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Definition A.25. Two parameters to express whether a destination node of a task is within the node collections of a satellite are defined as follows:

$$\Lambda_i^k = \begin{cases} 1 & \text{if} \quad \sum_{y=1}^{Y} \zeta_i^y \cdot B_y^k \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

$$\Gamma_i^k = \begin{cases} 1 & \text{if} \quad \sum_{y=1}^{Y} \zeta_i^y \cdot G_y^k \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

Definition A.26. A parameter to express whether at least one task of type r is assigned to satellite k and with destination node y is defined as follows:

$$\forall i \in I,\ \forall k \in K,\ y \in Y,\ r \in R$$
$$\iota_y^{k,r} = \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{I} \nu_i^k \cdot \zeta_i^y \cdot \varphi_i^r \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

Definition A.27. A parameter to express the memory state of a satellite for a given user node, and after the completion of n tasks is defined as follows:

$$\forall i, n \in I,\ \forall k \in K, \forall y \in Y,\ m_k^{init} \in \mathbb{R}^+,\ m_i \in \mathbb{R}$$
$$m_n^{k,y} = m_k^{init} + \sum_{i=i}^{n} m_i \cdot \nu_i^k \cdot \zeta_i^y \tag{34}$$

Definition A.28. A parameter to express whether all CUL tasks (q_6) have been assigned is defined as follows:

$$\forall i \in I,\ \forall l \in L \quad A = \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{I} \varsigma_i \cdot \varphi_i^{q_6} = \sum_{i=1}^{I} \varphi_i^{q_6} \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

The parameter ς is defined in constraint 54. It represents whether a task has been assigned.

Definition A.29. A parameter to express the latest start time of the assigned synchronization tasks (q_6) is defined as follows:

$$\tau_{sync} = \begin{cases} max(\tau_i) \mid \tau_i \subset t_i \subset T^{q_6} & \text{if} \quad A = 1 \\ \Delta_{sim} & \text{otherwise} \end{cases} \tag{36}$$

For the mean latency computation, a series of variables are defined. The latency is divided into Mobile Originated (MO) and MT. For the MO latency, the MOUL (q_7) and MODL (q_8) task types are used. Using the $T^{k,y,r}$ task subset for q_7 and q_8 is not enough since there must be at least an upload and download task for the same destination node to compute the latency. Furthermore, some tasks will not be paired and thus need to be purged from the subset. This is because the satellite can hold pending data in its memory.

Definition A.30. The unpaired task subset of MODL to purge the MODL task set from non-paired download tasks within the satellite is defined as the subset $T^{k,y,r}$ of tasks of type q_8, whose start time is earlier than the earliest start time of the q_7 task subset for the same user node and satellite. Let $\tau_i(T^{k,y,r})$ be the start time of an element of a $T^{k,y,r}$ subset, and $\tau_1(T^{k,y,r})$ be the start time of the first element of a $T^{k,y,r}$ subset. Then:

$$\forall i \in I,\ \forall k \in K\ \forall y \in Y$$
$$T_{unpaired}^{k,y,q_8} \subset T^{k,y,q_8} \mid \tau_i(T^{k,y,q_8}) < \tau_1(T^{k,y,q_7}) \tag{37}$$

Definition A.31. The purged task subset of MODL is defined with the unpaired subset as follows:

$$T_{purged}^{k,y,q_8} = (T^{k,y,q_8} \cap T_{unpaired}^{k,y,q_8})^c \tag{38}$$

Definition A.32. The unpaired task subset to purge the MOUL task set from non-paired download tasks within the satellite is defined as the subset of tasks of type q_7, whose index are out of bounds of the T_{purged}^{k,y,q_8} task subset.

$$\forall i \in I,\ \forall k \in K\ \forall y \in Y$$
$$T_{unpaired}^{k,y,q_7} \subset T^{k,y,q_7} \mid i > \left\| T_{purged}^{k,y,q_8} \right\| \tag{39}$$

Definition A.33. The purged task subset of MOUL is defined with the unpaired subset as follows:

$$T_{purged}^{k,y,q_7} = (T^{k,y,q_7} \cap T_{unpaired}^{k,y,q_7})^c, \tag{40}$$

Definition A.34. The unpaired task subset of MTDL to purge the MTDL task set from non-paired download tasks within the satellite is defined as the subset of tasks of type q_{10}, whose start time is earlier than the earliest start time of the q_9 task subset for the same node and satellite:

$$\forall i \in I,\ \forall k \in K\ \forall y \in Y$$
$$T_{unpaired}^{k,y,q_{10}} \subset T^{k,y,q_{10}} \mid \tau_i(T^{k,y,q_{10}}) < \tau_1(T^{k,y,q_9}) \tag{41}$$

Definition A.35. The purged task subset of MTDL is defined with the unpaired subset as follows:

$$T_{purged}^{k,y,q_{10}} = (T^{k,y,q_{10}} \cap T_{unpaired}^{k,y,q_{10}})^c \tag{42}$$

Definition A.36. The unpaired task subset to purge the MTUL task set from non-paired download tasks within the satellite is defined as the subset of tasks of type q_9, whose index are out of bounds of the $T_{purged}^{k,y,q_{10}}$ task subset:

$$\forall i \in I,\ \forall k \in K\ \forall y \in Y$$
$$T_{unpaired}^{k,y,q_9} \subset T^{k,y,q_9} \mid i > \left\| T_{purged}^{k,y,q_{10}} \right\| \tag{43}$$

Definition A.37. The purged task subset of MTUL is defined with the unpaired subset as follows:

$$T_{purged}^{k,y,q_9} = (T^{k,y,q_9} \cap T_{unpaired}^{k,y,q_9})^c, \tag{44}$$

Definition A.38. A set of $\Delta_\tau^{k,y}$ is defined if there exists a MO or MT pair for a given satellite and node (UE):

$$\begin{aligned} \Delta_{\tau,MO}^{k,y} = \{d_1, d_2, ..., d_{I_{MO}}\} \mid I_{MO} = \left\| T_{purged}^{k,y,q_8} \right\| \iff \\ \iff T_{purged}^{k,y,q_7}, T_{purged}^{k,y,q_8} \neq \varnothing \\ d_i = (\tau_i^{q_8} - \tau_i^{q_7}) \mid \tau_i^{q_8} \subset t_i \subset T_{purged}^{k,y,q_8} \wedge \\ \wedge\ \tau_i^{q_7} \subset t_i \subset T_{purged}^{k,y,q_7} \end{aligned} \tag{45}$$

$$\begin{aligned} \Delta_{\tau,MT}^{k,y} = \{d_1, d_2, ..., d_{I_{MT}}\} \mid I_{MT} = \left\| T_{purged}^{k,y,q_10} \right\| \iff \\ \iff T_{purged}^{k,y,q_9}, T_{purged}^{k,y,q_{10}} \neq \varnothing \\ d_i = (\tau_i^{q_{10}} - \tau_i^{q_9}) \mid \tau_i^{q_{10}} \subset t_i \subset T_{purged}^{k,y,q_{10}} \wedge \\ \wedge\ \tau_i^{q_9} \subset t_i \subset T_{purged}^{k,y,q_9} \end{aligned} \tag{46}$$

Definition A.39. The mean latency for a given UE and satellite is defined as follows:

$$\overline{lat}_{MO}^{k,y} = \frac{1}{I_{MO}} \sum_{i=1}^{I_{MO}} d_i \mid d_i \in \Delta_{\tau,MO}^{k,y} \tag{47}$$

$$\overline{lat}_{MT}^{k,y} = \frac{1}{I_{MT}} \sum_{i=1}^{I_{MT}} d_i \mid d_i \in \Delta_{\tau,MT}^{k,y} \quad (48)$$

Definition A.40. The mean latency for a given satellite is defined as follows:

$$\overline{lat}_{MO}^{k} = \frac{\sum_{y=1}^{Y} \overline{lat}_{MO}^{k,y}}{\sum_{y=1}^{Y} \iota_y^{k,q_7} \cdot \iota_y^{k,q_8}} \quad (49)$$

$$\overline{lat}_{MT}^{k} = \frac{\sum_{y=1}^{Y} \overline{lat}_{MT}^{k,y}}{\sum_{y=1}^{Y} \iota_y^{k,q_9} \cdot \iota_y^{k,q_{10}}} \quad (50)$$

Definition A.41. The overall mean latency is defined as follows:

$$\overline{lat}_{MO} = \frac{1}{K} \sum_{k=1}^{K} \overline{lat}_{MO}^{k} \quad (51)$$

$$\overline{lat}_{MT} = \frac{1}{K} \sum_{k=1}^{K} \overline{lat}_{MT}^{k} \quad (52)$$

Note that this latency computation assumes that all satellites in the scenario have been assigned with at least one MO and one MT data transfer task pair for at least one destination UE since it takes into account all satellites K to average the latency.

B. Constraints

Constraints in the mathematical formulation correspond to the hard constraints of the model. The medium and soft constraints correspond to the optimization criteria and are explained in the next section. Constraints are divided into general and 3GPP-specific constraints.

1) General

Definition B.1. The temporal bounds of the task start and end time are defined as follows:

$$\forall i \in I \quad \exists! j \in J \quad | \quad \tau_j^* \leq \tau_i < \tau_i + \delta_i \leq \epsilon_j \quad (53)$$

Definition B.2. The unique assignation of a task to a contact window is defined using the λ parameter. This constraint also defines the support parameter ς, which indicates whether a task is assigned:

$$\forall i \in I \quad \sum_{j=1}^{J} \lambda_{i,j} \leq 1 \Rightarrow \varsigma_i = \begin{cases} 1 & \text{if} \quad \sum_{j=1}^{J} \lambda_{i,j} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (54)$$

Definition B.3. The overlapping tasks constraint in satellites is defined as follows:

$$\forall k \in K,\ \forall i_1, i_2 \in I \mid i_1 \neq i_2 \quad \alpha_{i_1,i_2} \cdot \phi_{i_1,i_2}^{k} \cdot \vartheta_{i_1,i_2} = 1 \quad (55)$$

Definition B.4. The overlapping tasks constraint in terrestrial nodes is defined as follows:

$$\forall l \in L,\ \forall i_1, i_2 \in I \mid i_1 \neq i_2 \quad \alpha_{i_1,i_2} \cdot \varrho_{i_1,i_2}^{l} \cdot \vartheta_{i_1,i_2} = 1 \quad (56)$$

Definition B.5. The memory limits constraint of a satellite is defined as follows:

$$\forall k \in K,\ \forall n \leq \left\|T^k\right\| \quad \eta_n^k \cdot \kappa_n^k = 1 \quad (57)$$

Definition B.6. The energy limits constraint of a satellite is defined as follows:

$$\forall k \in K,\ \forall n \leq \left\|T^k\right\| \quad \mu_n^k \cdot \sigma_n^k = 1 \quad (58)$$

Definition B.7. The memory limits constraint of a satellite and for a specific node is described as follows:

$$\forall k \in K,\ \forall n \leq \left\|T^k\right\|,\ \forall y \in Y \quad m_n^{k,y} \geq 0 \quad (59)$$

2) 3GPP-specific

Definition B.8. This constraint binds each task type to a specific node type (i.e., UE or GS):

$$\forall i \in I,\ \forall l \in L,\ \forall q \in \{q_1, q_4, q_7, q_{10}\} \quad \varphi_i^q \cdot \gamma_i^l \cdot \rho_l = 1 \quad (60)$$

$$\forall i \in I,\ \forall l \in L,\ \forall q \in \{q_2, q_3, q_5, q_6, q_8, q_9\} \quad \varphi_i^q \cdot \gamma_i^l \cdot (1 - \rho_l) = 1 \quad (61)$$

Definition B.9. The precedence constraint between a PSL download task (q_2) and an AR to the same satellite (q_1) is defined as follows:

$$\forall k \in K,\ \forall i_1, i_2 \in I \mid i_1 \neq i_2,\ q_1, q_2 \in Q \quad \varphi_{i_1}^{q_1} \cdot \varphi_{i_2}^{q_2} \cdot \alpha_{i_1,i_2} \cdot \phi_{i_1,i_2}^{k} = 1 \iff \xi_k^{q_2} = 1 \quad (62)$$

Definition B.10. The precedence constraint between a context download (q_5) and an attach (q_4) is defined as follows:

$$\forall k \in K,\ \forall i_1, i_2 \in I \mid i_1 \neq i_2,\ q_4, q_5 \in Q \quad \varphi_{i_1}^{q_4} \cdot \varphi_{i_2}^{q_5} \cdot \alpha_{i_1,i_2} \cdot \phi_{i_1,i_2}^{k} = 1 \iff \xi_k^{q_5} = 1 \quad (63)$$

Definition B.11. The precedence constraint between an attach (q_4) and an AVSI upload task (q_3) to the same satellite uses the support parameter Λ to check if the authentication vectors are already onboard the satellite:

$$\forall k \in K,\ \forall i_1, i_2 \in I \mid i_1 \neq i_2,\ q_3, q_4 \in Q \quad \varphi_i^{q_4} \cdot \nu_i^k \cdot \Lambda_i^k + \varphi_{i_1}^{q_3} \cdot \varphi_{i_2}^{q_4} \cdot \alpha_{i_1,i_2} \cdot \phi_{i_1,i_2}^{k} \geq 1 \iff \xi_k^{q_4} = 1 \quad (64)$$

Definition B.12. The precedence constraint between a data transfer to and from a UE (q_7 and q_{10}) and the context upload task (q_6) uses the support variable Γ to check if the context is already onboard the satellite:

$$\forall k \in K,\ \forall i_1, i_2 \in I \mid i_1 \neq i_2,\ q_6, q_r \in Q,\ q_r = \{q_7, q_{10}\} \quad \varphi_i^{q_r} \cdot \nu_i^k \cdot \Gamma_i^k + \varphi_{i_1}^{q_6} \cdot \varphi_{i_2}^{q_r} \cdot \alpha_{i_1,i_2} \cdot \phi_{i_1,i_2}^{k} \geq 1 \iff \xi_k^{q_r} = 1 \quad (65)$$

C. Optimization criteria

The optimization criteria is the approach followed to minimize or maximize the objective function vector of the problem ($F(x)$). In our case, the objective function is computed using the space of all tasks (T). There are three different

TABLE 1. Support parameters.

Parameter	Description	Parameter	Description
$\lambda_{i,j}$	Boolean indicating if contact window j is assigned to task i	σ_n^k	Boolean indicating if the energy resource of satellite k is below its maximum value at n tasks
$\chi_{k,j}$	Boolean indicating if satellite k is involved in contact window j	ζ_i^y	Boolean indicating if task i has user node y as destination
$\beta_{l,j}$	Boolean indicating if node l is involved in contact window j	B_y^k	Boolean indicating if user node y is within the node collection A_k of satellite k
ν_i^k	Boolean indicating if task i is assigned to satellite k	G_y^k	Boolean indicating if user node y is within the node collection C_k of satellite k
γ_i^l	Boolean indicating if task i is assigned to node l	Λ_i^k	Boolean indicating if the destination node of task i is within the node collection A_k of satellite k
φ_i^r	Boolean indicating if task type r is assigned to task i	Γ_i^k	Boolean indicating if the destination node of task i is within the node collection C_k of satellite k
ξ_k^r	Boolean indicating if at least one task of type r is assigned to satellite k	$\iota_y^{k,r}$	Boolean indicating if at least one task of type r is assigned to satellite k and with destination node y
ϕ_{i_1,i_2}^k	Boolean indicating if two tasks i_1 and i_2 are assigned to satellite k	$m_n^{k,y}$	Memory state of satellite k for user node y, and after the completion of n tasks
ϱ_{i_1,i_2}^l	Boolean indicating if two tasks i_1 and i_2 are assigned to node l	A	Boolean indicating if all CUL tasks (q_6) have been assigned
α_{i_1,i_2}	Boolean indicating if the start time of task i_1 comes earlier than the start time of task i_2	τ_{sync}	Latest start time of the assigned synchronization tasks (q_6)
ϑ_{i_1,i_2}	Boolean indicating if the end time of task i_1 comes earlier than the start time of task i_2	$\overline{lat}_{MO}^{k,y}$	Mean MO latency for satellite k and destination y
m_n^k	Memory resource state of satellite k after n tasks	$\overline{lat}_{MT}^{k,y}$	Mean MT latency for satellite k and destination y
e_n^k	Energy resource state of satellite k after n tasks	$\overline{lat}_{MO}^{k}$	Mean MO latency for satellite k
η_n^k	Boolean indicating if the memory resource of satellite k is above zero at n tasks	$\overline{lat}_{MT}^{k}$	Mean MT latency for satellite k
κ_n^k	Boolean indicating if the memory resource of satellite k is below its maximum value at n tasks	$\overline{lat}_{MO}$	Mean overall MO latency
μ_n^k	Boolean indicating if the energy resource of satellite k is above its minimum value at n tasks	$\overline{lat}_{MT}$	Mean overall MT latency

solving approaches: (1) a priori articulation of preferences, (2) a posteriori articulation of preferences, and (3) no articulation of preferences. The proposed optimization criteria for this work is a priori articulation of preferences. Several global optimization methods exist for the a priori articulation of preferences approach [20]. The proposed method for this optimization criteria is a combination of the lexicographic and weighted sum methods.

1) Lexicographic method

This method involves ordering the objective functions based on their importance and solving a sequence of optimization problems one at a time, with each problem optimizing one objective while respecting constraints introduced by previous solutions. This means that the search space for each subsequent objective function is constrained by the solutions obtained by minimizing the previous objective functions in the order of importance. In our model, we have two different score levels for optimization constraints: medium and soft. As such, we define two objective functions $F_{medium}(T)$ and $F_{soft}(T)$ and proceed with an iterative optimization. In the first iteration, the optimization problem maximizes $F_{medium}(T)$ subject to the problem equality and inequality constraints:

$$\begin{aligned} &\text{Step 1: Maximize } F_{medium}(T) \\ &\qquad \text{subject to} \quad g_a(T) \leq 0, \quad \forall a = 1,2,...,m, \\ &\qquad\qquad \text{and} \quad h_b(T) = 0, \quad \forall b = 1,2,...,e \end{aligned} \tag{66}$$

The solution of this optimization is T^*_{medium}. For the second iteration, the optimization problem is to maximize $F_{soft}(T)$ subject to the same problem constraints and additionally, subject to a new constraint to not worsen the value of the first optimization step:

$$\begin{aligned} &\text{Step 2: Maximize } F_{soft}(T) \\ &\qquad \text{subject to} \quad g_a(T) \leq 0, \quad \forall a = 1,2,...,m, \\ &\qquad\qquad \text{and} \quad h_b(T) = 0, \quad \forall b = 1,2,...,e, \\ &\qquad\qquad \text{and} \quad F_{medium}(T) \geq F_{medium}(T^*_{medium}) \end{aligned} \tag{67}$$

2) Weighted sum method

The weighted sum method is a particular case of the weighted global criterion method with $p = 1$. This method combines all the objective functions into a global function using the weight vector (ω) and parameter (p). The weights are unique for each objective function, and they indicate each objective's relative importance or priority according to the decision-maker's preferences. The parameter p is common to all objective functions and governs the overall trade-offs among them in the scalarization process. Higher p values yield a focused global function on heavily weighted objective functions, and low p values produce a more balanced global function.

$$U = \sum_{c=1}^{C} \omega_c [F_c(T)]^p \xRightarrow{p=1} U = \sum_{c=1}^{C} \omega_c F_c(T) \tag{68}$$

This approach allows the operator to select the appropriate weights according to their business criteria. For that, the objective functions have to be comparable, and have to be transformed into normalized non-dimensional functions. For this formulation, the proposed transformation method is as follows:

$$F^{norm}(T) = \frac{F(T)}{|F^{max}(T)|} \tag{69}$$

Following the weighted sum method, the functions that aggregate all objective functions (F_{medium} and F_{soft}) can be expressed as:

$$F_{medium}(T) = \omega_{assign} F_{assign}(T) \tag{70}$$

$$\begin{aligned} F_{soft}(T) = \omega_{early} F_{early}(T) + \omega_{state} F_{state}(T) + \\ + \omega_{lat} F_{lat}(T) + \omega_{sync} F_{sync}(T) \end{aligned} \tag{71}$$

These functions correspond to the score levels and their respective constraints explained in section III.

Definition C.1. The F_{assign} contribution to the score is defined using the task reward (ϖ) as follows:

$$F_{assign}(T) = \frac{\sum_{i=1}^{I} \varpi_i \cdot \varsigma_i}{\sum_{i=1}^{I} \varpi_i} \tag{72}$$

Definition C.2. The F_{early} contribution to the score is defined using the simulation duration (Δ_{sim}) as follows:

$$F_{early}(T) = \frac{1}{\Delta_{sim}} \sum_{i=1}^{I} (\Delta_{sim} - \tau_i) \tag{73}$$

Definition C.3. The F_{state} contribution to the score is defined using the final states of the satellite (m_k^{final} and e_k^{final}) as follows:

$$F_{state}(T) = F_{memory} + F_{energy} \tag{74}$$

$$F_{memory}(T) = \sum_{k=1}^{K} \frac{m_k^{max} - m_k^{final}}{m_k^{max}} \quad \text{where}$$

$$m_k^{final} = m_k^{init} + \sum_{i=i}^{I} m_i \cdot \nu_i^k \tag{75}$$

$$F_{energy}(T) = \sum_{k=1}^{K} \frac{e_k^{final}}{e_k^{max}} \quad \text{where} \quad e_k^{final} = e_k^{init} + \sum_{i=i}^{I} e_i \cdot \nu_i^k \tag{76}$$

Definition C.4. The F_{lat} contribution to the score is defined using mean MO and mean MT latency ($\overline{lat}_{MO}$ and $\overline{lat}_{MT}$) as follows:

$$F_{lat}(T) = F_{MO}(T) + F_{MT}(T) \tag{77}$$

$$F_{MO}(T) = \frac{\Delta_{sim} - \overline{lat}_{MO}}{\Delta_{sim}} \tag{78}$$

$$F_{MT}(T) = \frac{\Delta_{sim} - \overline{lat}_{MT}}{\Delta_{sim}} \tag{79}$$

Definition C.5. The F_{sync} contribution to the score is defined using the maximum start time (τ_{sync}) as follows:

$$F_{sync}(T) = \frac{\Delta_{sim} - \tau_{sync}}{\Delta_{sim}} \tag{80}$$

V. Optimization algorithm

The proposed scheduler uses a combination of two different algorithms: a Construction Heuristics (CH) algorithm and a Local Search (LS) algorithm. CH, like greedy search, are good algorithms for finding a quick, feasible solution using a presumably known underlying rule of the system. In this way, the solution is not just feasible but also at least coherent following this underlying rule. LS algorithms, on the other hand, shine in finding better solutions by avoiding getting stuck in local optima and constantly searching the optimization space. This takes longer but delivers better solutions than the CH algorithms. Since the results of a LS algorithm depends on the initial solution or initial guess, the best approach is to initialize the solution with a CH algorithm and then refine it using a LS algorithm [21].

The algorithmic architecture is presented in Figure 3. The CH algorithm takes an uninitialized solution object and iterates over all its planning entities (i.e., tasks), assigning them a satellite, target, and start time. Then, it produces an initialized feasible solution to serve as the initial guess for the LS algorithm. Note that the CH algorithms do not require any termination criteria since the process ends when all planning entities are initialized. The scheduler is modeled for an over-constrained problem, meaning it might have more planning entities than those that can be assigned without breaking any feasibility constraint. This means, for example, that the initial task number introduced to the scheduler as input is larger than the maximum amount of tasks that the scheduler can assign with the given constraints. The CH algorithm can assign null objects to the planning entities to solve this. Then, an additional constraint is required in the second score level, a business constraint to force the maximum amount of assigned tasks.

The LS algorithm takes the initialized (yet unoptimized) solution produced by the CH algorithm as an initial guess for the LS optimization. During this optimization process, the algorithm changes the planning variables assigned to the planning entities and computes the score of the new solution. This is called an optimization move. One of the generated moves is selected as the following optimization step, from which the next set of possible moves will be generated. The generated moves and the acceptance criteria based on the calculated score to take the next optimization step depend on the LS algorithm used and its parameters. The LS phase requires a termination criterion to end the optimization process. This termination criterion can be a fixed duration of the LS phase, a fixed number of optimization steps, or setting a threshold when a specific score value is reached. A more advanced termination criterion may be to end the LS phase as soon as the score has not improved over a fixed amount of time or optimization steps. The latter is used in combination of a global time for the simulations of this paper. The global time is set to five minutes and the unimproved score time termination criteria is set to one minute.

A. Construction heuristics

The proposed CH algorithm used for the simulations of this study is the first fit decreasing [22]. First-fit algorithms do not require any strength comparison function like the weakest and strongest fit since they assign planning entities with the original given order. A strength comparison function evaluates the planning value strength of planning entities. In our case, this depends on the business criteria of the constellation and service operator and is left out of the construction heuristics phase because it is evaluated by the business constraints' score later on. The planning entities are then only ordered by allocation difficulty, which is an invariable notion of the scenario, regardless of the changing business criteria that the operator might have. The first-fit decreasing algorithm orders the planning entity list, namely the unassigned *TaskList*, according to the allocation difficulty assigned to each task. Decreasing means that the most difficult entities to allocate are ordered first. The algorithm is presented in Algorithm 1.

Algorithm 1 First fit decreasing algorithm.

```
Input: Uninitialized TaskAssignment
Output: Initialized TaskAssignment
Sort taskList according to task.difficulty
LOOP Process
for task in taskList do
  if (task.satellite == null) then
    for satellite in satelliteList do
      for window in windowList do
        Select a target and start time combination from
        the window
        if (feasible combination) then
          Assign satellite, target, and start time to task
          Break window and satellite loops
        end if
      end for
    end for
  end if
end for
return Assigned taskList
```

The difficulty assigned to each task depends entirely and solely on the task type and objective target. This means that a task targeting a specific UE has a higher allocation difficulty assigned than a task targeting any system GS, for example. There are four different difficulty levels, number one being the most complex entities to allocate. The last difficulty level is assigned to those entities or tasks targeting GS or general target areas like countries. This is because the constellation satellites will have more potential contact windows with those targets than with a specific UE, for example. Table 2 presents the proposed difficulty assignation to the different task types of the model.

TABLE 2. Entity allocation difficulty

Type	Target	Difficulty
MT download	UE	1
MO upload	UE	2
Registration	UE	3
MT upload	GS	4
MO download	GS	4
Attach request	TA	4
PSL download	GS	4
AVSI upload	GS	4
Context download	GS	4
Context upload	GS	4

B. Local search

Same as with the CH algorithms, the implemented scheduling framework allows the usage of different LS algorithms. The proposed LS algorithm used for the simulations of this study is tabu search [23] [24]. The difference between other algorithms is how the neighbor solutions and the next optimization step are chosen. The tabu search algorithm approach uses a tabu list of planning entities or variables that cannot be changed as possible moves, to avoid coming back to visited solutions and encourage exploration. This is a useful approach to avoid getting stuck in a local optimum. The implemented tabu list size is 5% of the system planning entities, which means that five percent of last step planning entities are included in the tabu list for the following steps and that the acceptor function will cull any move generated by the move selector function involving these planning entities. The forager function is set to evaluate all the accepted moves and select the higher-scoring solution as the next optimization step. This parameter and others can be tweaked using a benchmark to increase the algorithm performance for this specific scenario, but this is out of the scope of the current study and is left to future research. Algorithm 2 describes in pseudocode the selected LS algorithm.

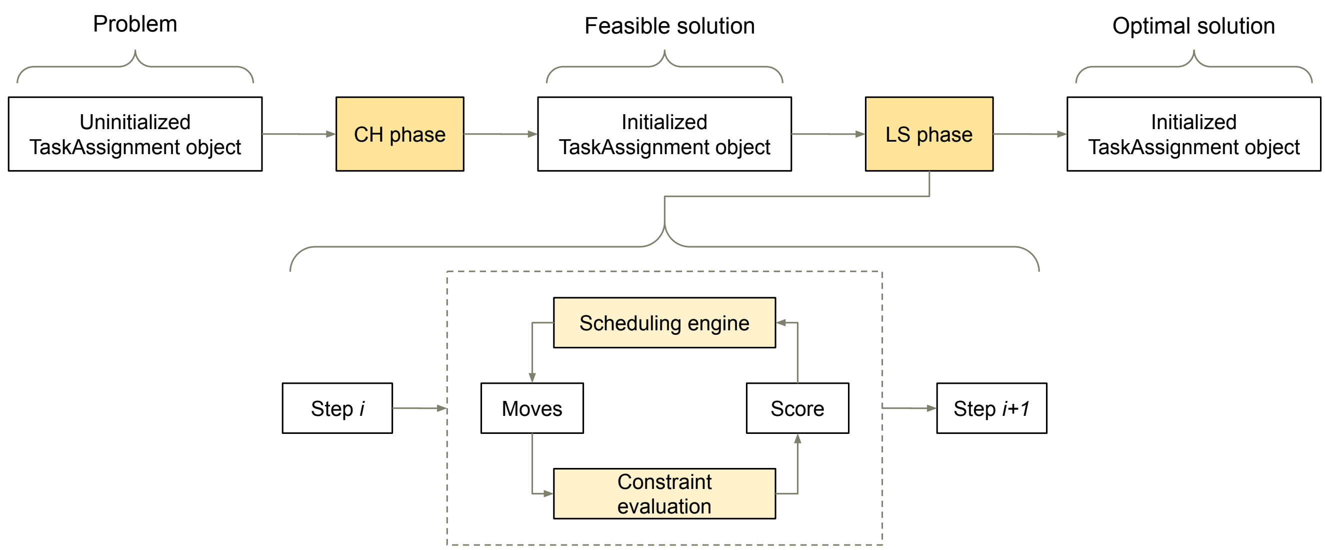

FIGURE 3. Scheduler algorithm

Algorithm 2 Tabu search algorithm.

Input: Initialized TaskAssignement
Output: Optimized TaskAssignment
Initialisation : Initialize tabu list
1: Evaluate initial solution
LOOP Process
2: **for** $i = 0$ to $maxSteps$ **do**
3: Generate neighborhood
4: Select winning move
5: Update tabu list with that move
6: Update current solution with selected move
7: Check termination criteria
8: **if** (criteria met) **then**
9: Break optimization loop
10: **end if**
11: **end for**
12: **return** Final solution

VI. Scenario

The selected scenario to validate and test the presented CMS is a 16-satellite constellation of Sateliot. The constellation comprises four polar planes separated by 45º, each filled with four equally separated satellites (90º phase difference). The orbital altitude of all the LEO satellites is 600 km, and the satellites' true anomalies are offset 5º. The keplerian orbital elements of the described orbits can be seen in Table 3. The five ground stations are KSAT Inuvik, KSAT Tromso, AWS Punta Arenas, AWS CapeTown, and AWS Sydney. The thirteen UE are located around the world in the following countries: China, EEUU, Brazil, Canada, Spain, India, Japan, Germany, France, UK, Norway, Australia, and United Arab Emirates (UAE) (see Table 4). The minimum elevation angles considered for such UE are 50º, 30º for all GS, and 60º Field of View (FoV) for all satellites. Last but not least, this scenario assumes that all users are authenticated and their contexts are already onboard all constellation satellites. Therefore, the scenario features only data transfer task types, namely MOUL, MODL, MTUL, and MTDL.

TABLE 3. Satellite Keplerian orbital elements

ID	a (km)	e	i (º)	Ω (º)	ω (º)	θ (º)
1	6978	0	90	0	0	0
2	6978	0	90	0	90	0
3	6978	0	90	0	180	0
4	6978	0	90	0	270	0
5	6978	0	90	45	0	5
6	6978	0	90	45	90	5
7	6978	0	90	45	180	5
8	6978	0	90	45	270	5
9	6978	0	90	90	0	10
10	6978	0	90	90	90	10
11	6978	0	90	90	180	10
12	6978	0	90	90	270	10
13	6978	0	90	135	0	15
14	6978	0	90	135	90	15
15	6978	0	90	135	180	15
16	6978	0	90	135	270	15

[1] The Keplerian elements are semimajor axis (a), eccentricity (e), inclination (i), the longitude of the ascending node (Ω), the argument of periapsis (ω), and the true anomaly (θ)

These are the fixed elements of the scenario and are mainly used to compute the contact opportunities between the constellation satellites and the rest of the elements, namely the Sun, UE, and GS. However, a series of scenario elements

TABLE 4. UE coordinates

ID	Country	Latitude	Longitude
1	China	29.56	106.35
2	EEUU	29.19	-81.05
3	Brasil	-8.70	-64.42
4	Canada	49.66	-123.51
5	Spain	41.39	2.11
6	India	32.88	75.39
7	Japan	42.89	142.63
8	Germany	47.83	12.04
9	France	43.77	7.43
10	UK	57.14	-6.10
11	Norway	62.52	7.07
12	Australia	-31.95	116.13
13	UAE	23.50	54.78

will vary depending on the constraint being validated or the metric being evaluated.

- Initial task number: This is the number of tasks in the input list of the scheduler. This number, combined with the contact opportunities, dictates whether the scenario is saturated. A non-saturated scenario is one where all the tasks can be assigned to the satellites. On the other hand, a saturated scenario is one where there are more tasks to be assigned than the ones possible with the available contact windows.
- Satellite resources: Instead of modifying the task resource consumption, satellite resources are modified to test resource saturation scenarios. The satellite has two main resources: energy and memory capacity. Each satellite starts with a defined initial battery, memory level, and maximum capacity value.
- Priorities: Each task object has an attribute defined as the task weight. This is the value that is going to be used to compute some of the business constraints' scores. Depending on the weight assigned to each task, different types of task type and task target prioritization are possible. For this study, tasks targeting priority UE are assigned a weight value of 20, while tasks targeting non-priority UE are assigned a weight value of 10.

The core focus of this paper is on the saturation of the contact time and onboard reserved storage capacity resources. As such, two scenarios are simulated: (1) a contact time-constrained scenario and (2) a memory-constrained scenario. Both scenarios use the aforementioned fixed elements and a 24-hour temporal horizon. The assumption for these scenarios is that the maximum capacity of the contact link is used with the maximum number of served users. To compute the maximum number of possible users, the channel's maximum capacity is defined according to hardware tests. The capacity of the feeder link (channel for satellite-GS communications) is 1.5 Mbps. The capacity of the service link (channel for satellite-UE communications) is 253.6 Kbps. Another aspect to be modeled is the memory required per UE. This depends on the type of task being conducted with the UE. For instance, the required data from a context download task will differ from the required data from a user data transfer task. Table 5 presents the memory consumption per UE of every modeled task. For this study, it is assumed that the average packet size is 33 Bytes and that both MO and MT tasks consume the same amount of memory resources per message and UE. The rest of the values from the table are taken from the testbed results presented in [5].

TABLE 5. Task memory consumption per UE [5]

Task type	Memory consumption per UE (bytes)
MOUL	33
MODL	-33
MTUL	33
MTDL	-33
AR	9
PSL download	-9
AVSI upload	12.5
Registration	6600
Context download	-6600
Context upload	6600

The baseline model is designed to emulate a semi-manual traditional operator using a best-effort approach without considering business criteria for the offered service. This model is taken as the reference for this comparison. As such, the main characteristics of this model are as follows:

- Only feasible constraints are considered. This means that the model's business and operational constraints are not considered for the computation of the score.
- No local search refinement phase. The solver has a unique CH phase.
- The CH algorithm used is simple first fit. This means that the initial task list is not arranged following any criteria (it is random).

Since the simple first fit algorithm has a certain degree of randomness in the initial task list order, several simulations shall be conducted for statistical performance.

The main objective of the validation is not to assess the difference in performance between algorithms but rather to evaluate the difference in performance when using a classical satellite operations model (the one used by the baseline) versus when using the presented novel model for NTN. The comparison between the obtained schedules is based on business criteria metrics like the throughput, latency, and task prioritization of certain UE.

A. Contact time constrained scenario

For this scenario, the reserved onboard capacity of the satellites is set at a constant value and high enough not to saturate the satellite's memory. For the given scenario parameters, the reserved capacity per satellite is set to 20 MB[1]. The parameter that changes is the amount of served UE. Since the assumption is that the maximum capacity of the channel is being used, incrementing the amount of users results in incrementing the required task time to serve the selected number of users. Table 6 presents the maximum number of served UE for a one-minute duration task.

The most restricting tasks are the context uploads and downloads. However, it is unreasonable to expect all UE to generate all contexts at the time since this is only necessary during registration procedures. Furthermore, for the comparison simulations, only user data transmissions tasks are simulated (i.e., MOUL, MODL, MTUL, and MTDL), and the same amount of MO and MT tasks are used as initial input. Therefore, the most restrictive tasks are considered to be the user data transmission tasks using the service link (MOUL and MTDL). Table 7 shows three different task parameters depending on the maximum UE served due to the maximum capacity of the service link for user packets. These scenarios use the same contact information for the serving nodes as the one presented in this section (see Table 4). Instead of a single UE, these coordinates now represent a high-density area to saturate the scenario.

B. Memory constrained scenario

The domain of IoT applications is characterized by relatively modest data traffic, typically obviating concerns regarding onboard memory saturation within the system. Nevertheless, constraints on the satellite's onboard memory may arise from some considerations. Primary among these is the need to control Quality of Service (QoS) and adapt to different subscription models, requiring careful memory allocation strategies. Additionally, potential deviations from nominal functionality within the satellite constellation and ground segment introduce further complexity. Operational disruptions may render certain ground stations temporarily inaccessible or compromise satellite memory integrity. In the scope of this research, we propose that memory limitations primarily emanate from the intricacies of QoS management and the diverse profiles of subscribers.

For this scenario, the number of served UE is set as a constant value. The service link capacity gives the chosen value with a one-minute task duration (see Table 7). The parameter that changes is the reserved onboard capacity to serve these UE. Remember that this is just a portion of the satellite's memory capacity. The chosen values to test the memory saturation are 20 MB, 15 MB, 10 MB, and 5 MB. This is the reserved capacity to serve these thirteen specific areas, hence the low capacity values.

[1]In this paper, the prefixes used for memory measures are decimal, instead of the binary prefixes defined by the International Electrotechnical Commission (IEC).

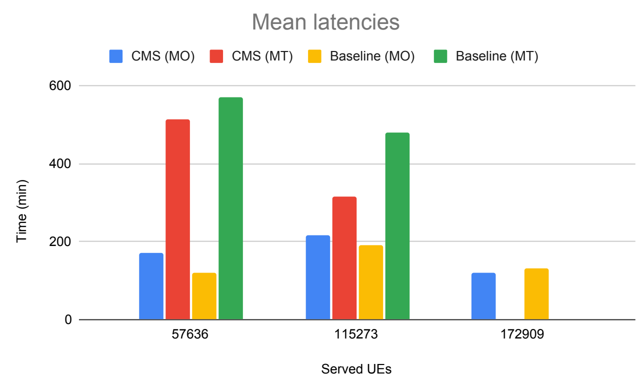

FIGURE 4. Mean latency results over served UE from the contact saturation simulations.

VII. Results

This section compares and analyzes the results obtained using the CMS and a baseline model for the scheduling of the resource-constrained scenarios explained in section VI.

A. Contact time constrained scenario

The results obtained for the different maximum serviced UE are presented in Table 8. As can be seen, the evaluated metrics are latency, throughput, and task completion percentage. The mean latency is computed by averaging the elapsed time between a data upload task and a data download task for each UE in each constellation satellite. The throughput metric is computed by adding all the completed transmission data (data upload and data download tasks assigned) and dividing it by the scheduler duration. The schedule is considered finished at the end of the last assigned task. Finally, the task completion percentage is computed by comparing the assigned tasks with the initial input number of those task types.

Figure 4 compares the mean latency obtained for MO and MT traffic using the manual algorithm and the CMS. Remember that the latency constraint contributes to the soft score of the scheduler when using the CMS. This means that the solver will always prioritize increasing the number of tasks assigned and the system throughput over minimizing the latency. Therefore, the latency metric heavily depends on the number of assigned tasks on the schedule. This is why there seems to be no clear trend between using the CMS with the latency constraint and using the manual solver without the constraint. However, it is worth noting that the latency results obtained using the manual solver are more unbalanced between MO and MT traffic than the ones obtained using the CMS. Another observable result is that the scenario to serve 172909 UE has no MT mean latency because the scenario is heavily saturated in contact time and the solvers (both CMS and manual) did not assign any MTDL tasks.

Figure 5 presents the throughput obtained by task type, using the CMS and the manual solver. Note that the throughput for MTDL tasks is zero for both the CMS and the

TABLE 6. Number of maximum served UE per task

Task type	Memory per UE (KB)	Link capacity (KBps)	Number of UE
MOUL	0.033	31.7	57636
MODL	-0.033	187.5	340909
MTUL	0.033	187.5	340909
MTDL	-0.033	31.7	57636
AR	0.009	31.7	211333
PSL download	-0.009	187.5	1250000
AVSI upload	0.0125	187.5	900000
Context download	-6.6	187.5	1705
Context upload	6.6	187.5	1705

TABLE 7. Task parameters depending on served UE for the considered link capacities

Served UE	Memory consumption (KB)	Task duration (s)
57636	1902	60
115273	3804	120
172909	5706	180

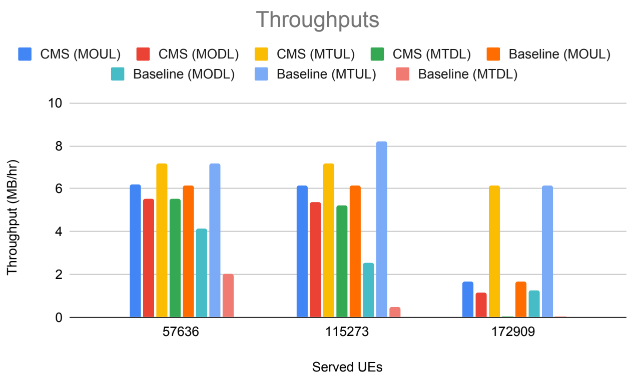

FIGURE 5. Data throughput over served UE from the contact saturation simulations.

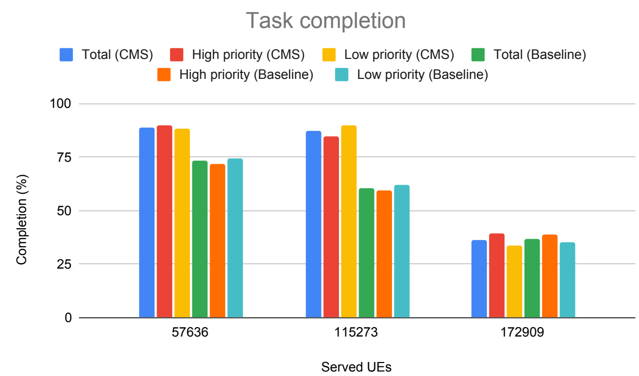

FIGURE 6. Task completion percentage over served UE from the contact saturation simulations.

manual solver for the scenario to serve 172909 UE. The first observable result is that the overall throughput obtained by the CMS is always equal to or greater than the one obtained with the manual solver. The general trend is that the throughput is reduced when increasing the number of served UE due to the saturation in contact time. Even though the throughput of MTUL computed with the manual scheduler is higher when serving 115273 UE than compared to the throughput obtained when serving 57636 UE, the overall MT throughput is reduced considering the decrease in MODL and MTDL throughput.

Figure 6 presents the task completion percentage, with different served UE numbers obtained by the CMS and the manual algorithm. The observable trend is that task completion is reduced when increasing the number of served UE, both for the CMS and the manual solver. This is because the task requires more time to serve that amount of UE. Another relevant result is that the difference in task completion obtained with the two algorithms is reduced when saturating the scenario with contact opportunities. This is because tasks can only be assigned to a few contact opportunities, and the CMS scheduler has no room for improvement by using business and operational constraints. Even in the constrained scenario when serving 172909 UE, the CMS is able to prioritize the task assignment of high priority targets and achieve a higher percentage than that of the baseline. Moreover, it is observed that for non-saturated scenarios, the CMS can increase the percentage of total assigned tasks. For the case of 57636 served UE, the total task completion is increased by 17.87%; for the case of 115273 served UE, the total task completion is increased by 44.02%. Furthermore, the results obtained with the manual scheduler cannot increase the percentage of assigned priority tasks over non-priority tasks for non-saturated scenarios, like the CMS does for the scenarios when serving 57636 and 115273 UE.

B. Memory constrained scenario

Table 9 presents the results obtained for the different reserved capacity simulations. As in the contact time saturation case, the results of the memory-constrained scenario can be grouped into mean latency, throughput, and task assignment.

TABLE 8. Contact saturation results

Served UEs: 57636			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	171	121.4	-29.01
MT mean latency (min)	514	570.8	11.05
MOUL throughput (MB/hr)	6.18	6.17	0.16
MODL throughput (MB/hr)	5.54	4.15	25.09
MTUL throughput (MB/hr)	7.2	7.2	0
MTDL throughput (MB/hr)	5.51	2.042	62.94
Total task completion (%)	89.05	73.136	17.87
High priority task completion (%)	89.74	71.664	20.14
Low priority task completion (%)	88.86	74.394	15.90
Served UEs: 115273			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	216	190.2	-11.94
MT mean latency (min)	316	478.6	51.46
MOUL throughput (MB/hr)	6.17	6.17	0
MODL throughput (MB/hr)	5.38	2.564	109.83
MTUL throughput (MB/hr)	7.2	8.23	-12.52
MTDL throughput (MB/hr)	5.22	0.506	931.62
Total task completion (%)	87.36	60.66	44.02
High priority task completion (%)	84.52	59.536	41.96
Low priority task completion (%)	89.80	61.83	45.24
Served UEs: 172909			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	121	123.2	1.82
MT mean latency (min)	-	-	-
MOUL throughput (MB/hr)	1.66	1.66	0
MODL throughput (MB/hr)	1.18	1.234	-4.58
MTUL throughput (MB/hr)	6.175	6.17	0.08
MTDL throughput (MB/hr)	0	0	0
Total task completion (%)	36.54	36.732	-0.53
High priority task completion (%)	39.58	38.748	2.10
Low priority task completion (%)	33.93	34.998	-3.15

TABLE 9. Memory constrained scenario results

Reserved memory of 20 MB			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	171	121.4	-29.01
MT mean latency (min)	514	570.8	11.05
MOUL throughput (MB/hr)	6.18	6.17	0.16
MODL throughput (MB/hr)	5.38	4.31	19.89
MTUL throughput (MB/hr)	7.2	7.2	0
MTDL throughput (MB/hr)	5.51	3,04	44.83
Total task completion (%)	89.05	76.92	13.62
High priority task completion (%)	89.74	75	16.43
Low priority task completion (%)	88.86	78.57	11.18
Reserved memory of 15 MB			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	180	113.2	-37.11
MT mean latency (min)	457	535.6	17.20
MOUL throughput (MB/hr)	6,1	6,17	-1.15
MODL throughput (MB/hr)	5.23	4.15	20.65
MTUL throughput (MB/hr)	7.2	7.2	0
MTDL throughput (MB/hr)	5.07	2.53	50.06
Total task completion (%)	86.09	75.62	12.16
High priority task completion (%)	87.18	73.2	16.03
Low priority task completion (%)	85.16	77.69	8.77
Reserved memory of 10 MB			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	125	92.8	-25.76
MT mean latency (min)	460	537.8	16.91
MOUL throughput (MB/hr)	6.1	5.6	8.3
MODL throughput (MB/hr)	4.51	3.94	12.59
MTUL throughput (MB/hr)	7.13	6.87	3.59
MTDL throughput (MB/hr)	4.12	2.39	41.99
Total task completion (%)	81.66	70.35	13.85
High priority task completion (%)	85.9	67.69	21.20
Low priority task completion (%)	78.02	72.64	6.90
Reserved memory of 5 MB			
Metric	*CMS*	*Manual*	*Diff. (%)*
MO mean latency (min)	109	97	-11.01
MT mean latency (min)	479	454.6	-5.09
MOUL throughput (MB/hr)	2.85	2.29	19.51
MODL throughput (MB/hr)	2.3	1.85	19.48
MTUL throughput (MB/hr)	4.12	3.51	14.71
MTDL throughput (MB/hr)	2.22	1.47	33.69
Total task completion (%)	42.9	34.14	20.41
High priority task completion (%)	53.21	35.77	32.78
Low priority task completion (%)	34.07	32.75	3.88

Figure 7 presents the mean latency obtained for MT and MO data transfers for the memory-constrained simulations. There is no general observable trend regarding the latency and the memory limits. It is worth mentioning that for the manual simulations, the obtained latency is slightly more unbalanced between MO and MT traffic than the ones obtained with the CMS.

Figure 8 presents the throughput of different task types obtained from the memory-constrained simulations by both the manual and CMS solvers. The most remarkable result is that the obtained throughput with the CMS scheduler is always equal to or higher than the ones obtained with the manual scheduler. The CMS can allocate tasks more efficiently. Another observable trend is that the task throughput is reduced when the reserved memory for the served UE is increasingly constrained. Again, this is because fewer data transmission tasks can be assigned due to the satellite's lack of assigned memory.

Figure 9 presents the resulting completion percentage from the memory-constrained simulation by the manual and CMS scheduler. The task completion percentage is tightly related to the throughput obtained and discussed in the previous

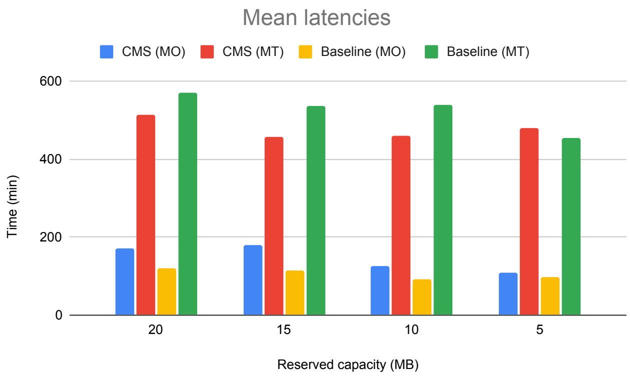

FIGURE 7. Mean latency results over served UE from the memory constrained simulations.

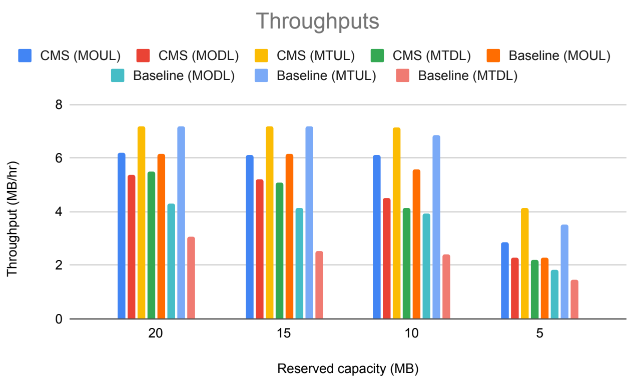

FIGURE 8. Data throughput over served UE from the memory constrained simulations.

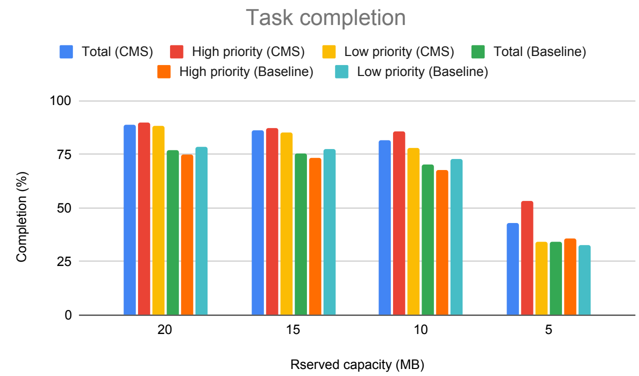

FIGURE 9. Task completion percentage over served UE from the memory saturation simulations.

figure. The general trend is the same as the throughput: it is reduced with increasing memory limitation. The most remarkable result is that the task completion percentage obtained using the CMS scheduler is always greater than that obtained using the manual scheduler. Another remarkable result is that the CMS scheduler considers business criteria and prioritization mechanisms. This is observed by the high-priority task completion percentage always being higher than the low-priority task completion in the CMS results. This is not the case for the manual results. Moreover, it can be seen that this effect becomes more relevant the more limited the scenario. In the 20 MB scenario, the high-priority completion percentage is just one percent higher than the low-priority completion percentage. However, in the 5 MB scenario, this gap is increased to 36%.

VIII. Conclusion

This paper has addressed some of the challenges associated with leveraging S&F LEO constellations for IoT use cases. The multifaceted nature of these challenges, encompassing the discontinuity of communication links, constraints in system resources, and the imperative to align satellite operations with mobile network business criteria, underscores the complexities inherent in achieving an efficient IoT communication through a LEO constellation. We proposed a novel architecture, the CMS, to tackle this scenario. This architecture offers a comprehensive framework that integrates various components to ensure optimized and business-aware communication within the context of NTN IoT.

This paper has contributed to the detailed model of the scheduler module within the CMS. The mathematical formulation developed in this work for solving the MOO problem within the CMS framework is a crucial step towards its practical implementation. Tailored to the specific needs of the S&F IoT use case, the proposed formulation provides a foundation for orchestrating communication tasks, optimizing resources, and adhering to constraints inherent in the satellite communication environment. The optimization algorithm derived from the proposed model and mathematical formulation provides a computational means to implement and evaluate the CMS architecture.

A real-world scenario presented by Sateliot serves as a concrete backdrop against which to evaluate the performance of the CMS. We compared the results of the CMS with those obtained from an alternative algorithm simulating a best-effort scheduling strategy from a satellite operator. The outcomes of these comparative experiments demonstrate a substantial enhancement, with the CMS showcasing an increase of up to 40% in task assignment and throughput rates. The demonstrated performance improvements highlight the CMS as a pivotal enabler for efficient and effective IoT communication in the realm of satellite-based NTN.

Acknowledgment

This work has been funded by the Government of Catalonia in the scope of the NewSpace Strategy for Catalonia. Additionally, this work was funded by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union – NextGeneration EU, in the framework of the Recovery Plan, Transformation and Resilience (PRTR) (Call UNICO I+D 5G 2021, ref. number TSI-063000-2021-1-6GSatNet-GS). This research was funded in part by the Spanish MCIN/AEI/ 10.13039/501100011033 through project PID2019-106808RA-I00, and by Secretaria d'Universitats i Recerca del departament d'Empresa i Coneixement de la Generalitat de Catalunya with the grant number 2021 SGR 00330. Assistance from Franco Criscola and David Canales in the Space Trajectories and Applications Research group at Embry-Riddle Aeronautical University is acknowledged, as is the support from the Aerospace Engineering department. Assistance in the review process from Juan A. Fraire in INRIA research institute is acknowledged.

REFERENCES

[1] 3rd Generation Partnership Project. RadioAccess Network; Study on Narrow-Band Internet of Things (NB-IoT)/ Enhanced Machine Type Communication (eMTC) Support for NonTerrestrial Networks (NTN) (Release 17). Technical report, Technical Specification Group, 2022.

[2] 3rd Generation Partnership Project. Solutions for NR to support non-terrestrial networks (NTN) (TR 23.737 Release 17). Technical report, Technical Specification Group, 2023.

[3] Zhi Lin, Min Lin, Benoit Champagne, Wei-Ping Zhu, and Naofal Al-Dhahir. Secrecy-Energy Efficient Hybrid Beamforming for Satellite-Terrestrial Integrated Networks. *IEEE Transactions on Communications*, 69(9):6345–6360, September 2021.

[4] Juan A. Fraire, Oana Iova, and Fabrice Valois. Space-Terrestrial Integrated Internet of Things: Challenges and Opportunities. *IEEE Communications Magazine*, 60(12):64–70, dec 2022.

[5] Timo Kellermann, Roger Pueyo Centelles, Daniel Camps-Mur, Ramon Ferrus, Marco Guadalupi, and Anna Calveras Auge. Novel Architecture for Cellular IoT in Future Non-Terrestrial Networks: Store and Forward Adaptations for Enabling Discontinuous Feeder Link Operation. *IEEE Access*, 10:68922–68936, 2022.

[6] R Steel, M Niézette, A Cesta, S Fratini, A Oddi, G Cortellessa, R Rasconi, G Verfaillie, C Pralet, M Lavagna, et al. ADVANCED PLANNING AND SCHEDULING INITIATIVE MRSPOCK AIMS FOR XMAS. *Language*, 1050:3, 2009.

[7] Alex Fukunaga, Gregg Rabideau, Steve Chien, and David Yan. Aspen: A framework for automated planning and scheduling of spacecraft control and operations. In *Proc. International Symposium on AI, Robotics and Automation in Space*, pages 181–187, 1997.

[8] Javier Barreiro, Matthew Boyce, Minh Do, Jeremy Frank, Michael Iatauro, Tatiana Kichkaylo, Paul Morris, James Ong, Emilio Remolina, Tristan Smith, et al. EUROPA: A platform for AI planning, scheduling, constraint programming, and optimization. *4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*, 2012.

[9] J.-P. Chamoun, S. Risner, T. Beech, and G. Garcia. Bridging ESA and NASA Worlds: Lessons Learned from the Integration of hifly®/SCOS-2000 in NASA's GMSEC. In *2006 IEEE Aerospace Conference*. IEEE, 2006.

[10] Mohamed Khalil Ben-Larbi, Kattia Flores Pozo, Tom Haylok, Mirue Choi, Benjamin Grzesik, Andreas Haas, Dominik Krupke, Harald Konstanski, Volker Schaus, Sándor P. Fekete, Christian Schurig, and Enrico Stoll. Towards the automated operations of large distributed satellite systems. part 1-2: Review and paradigm shifts and classifications and tools. *Advances in Space Research*, 67(11):3598–3637, jun 2021.

[11] Gregory Stock, Juan A. Fraire, Holger Hermanns, Eduardo Cruz, Alastair Isaacs, and Zhana Imbrosh. On the automation, optimization, and in-orbit validation of intelligent satellite constellation operations, 2022.

[12] Cezar Antônio Rigo, Laio Oriel Seman, Eduardo Camponogara, Edemar Morsch Filho, and Eduardo Augusto Bezerra. Task scheduling for optimal power management and quality-of-service assurance in CubeSats. *Acta Astronautica*, 179:550–560, feb 2021.

[13] Arnau Singla, Franco Criscola, Elena Ponce, David Canales, Anna Calveras, and Joan A. Ruiz-De-Azua. Towards 6G Non-Terrestrial Networks - An Autonomous Constellation Management Engine. 33rd AAS/AIAA conference. Pending proceedings. URL: https://www.researchgate.net/publication/369541726_Towards_6G_Non-Terrestrial_Networks_-An_Autonomous_Constellation_Management_Engine, 2022.

[14] Sateliot. https://sateliot.space/en/, 2023. Accessed: 2023-06-12.

[15] Jian Wu, Jiawei Zhang, Jinghui Yang, and Lining Xing. Research on Task Priority Model and Algorithm for Satellite Scheduling Problem. *IEEE Access*, 7:103031–103046, 2019.

[16] Huilong Fan, Zhan Yang, Xi Zhang, Shimin Wu, Jun Long, and Limin Liu. A novel multi-satellite and multi-task scheduling method based on task network graph aggregation. *Expert Systems with Applications*, 205:117565, November 2022.

[17] RedHat. OptaPlanner. https://www.optaplanner.org/, 2023 (accessed june 2023).

[18] Franco Criscola, Arnau Singla, Elena Ponce, Anna Calveras, Joan A. Ruiz-De-Azua, and David Canales. Development of a simulator for coverage planning of a 6G/IoT constellation. 33rd AAS/AIAA conference. Pending proceedings., 2022.

[19] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, apr 2004.

[20] J. S. Arora, O. A. Elwakeil, A. I. Chahande, and C. C. Hsieh. Global optimization methods for engineering applications: A review. *Structural Optimization*, 9(3-4):137–159, jul 1995.

[21] Geoffrey De Smet and open source contributors. *OptaPlanner User Guide*. Red Hat, Inc. or third-party contributors, 2006. OptaPlanner is an open source constraint solver in Java.

[22] Wei Mike Li, Victoria Mitchell, Barrie R Nault, and Denise Brind. Operating room scheduling and adaptive control using a priority first fit decreasing heuristic. *Engineering Management Research*, 4(2):46–53, 2015.

[23] Fred Glover. Tabu Search—Part I. *ORSA Journal on Computing*, 1(3):190–206, aug 1989.

[24] Fred Glover. Tabu Search—Part II. *ORSA Journal on Computing*, 2(1):4–32, feb 1990.