Received 28 August 2023; revised 10 October 2023; accepted 6 November 2023. Date of publication 9 November 2023; date of current version 30 November 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3331622

Fairness-Aware Inter-Slice Scheduler for IoT Services Over Satellite

ILORA MAITY (Member, IEEE), HOUCINE CHOUGRANI (Member, IEEE), AND SYMEON CHATZINOTAS (Fellow, IEEE)

Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, 1855 Esch-sur-Alzette, Luxembourg

CORRESPONDING AUTHOR: I. MAITY (e-mail: ilora.maity@uni.lu)

This work was supported by the European Space Agency (ESA)—ANCORSAT Project under Contract ESA 4000133827/21/NL/MM (https://connectivity.esa.int/projects/ancorsat).

ABSTRACT This paper addresses the problem of scheduling the uplink bandwidth of Low Earth Orbit (LEO) satellites among multiple Internet of Things (IoT) slices with diverse Quality of Service (QoS) requirements. The scheduling process involves twofold decisions — the amount of bandwidth allocated and the allocation duration. Resource scheduling for satellite IoT services is challenging because of limited bandwidth availability during a satellite pass, especially for LEO satellites. Another challenge is to compute a fair allocation schedule for IoT services with different latency demands, packet transmission frequency, required data volume, and the number of IoT devices. To address these challenges, we propose a fairness-aware inter-slice scheduler for satellite IoT services in this work. The proposed scheduler computes service priority based on respective traffic demands. We propose two algorithms for the scheduler based on weighted greedy and Simulated Annealing (SA), respectively. The weighted greedy algorithm schedules the services greedily based on the priority order. The SA algorithm enhances the greedy solution by ensuring the allocated bandwidth is proportional to the respective priority values. The simulation results show that the proposed SA algorithm achieves up to 21.11% more proportional fairness than the Simulated Annealing and Monte Carlo (SA-MC) benchmark scheme.

INDEX TERMS Resource allocation, Internet of Things (IoT), proportional fairness, scheduling, simulated annealing (SA).

I. INTRODUCTION

THE ADVANCEMENT of Internet of Things (IoT) technology enables smart devices to interconnect and access different services [1]. Some examples of IoT services are smart home IoT, smart city IoT, agriculture IoT, vehicular IoT for tracking, and vehicular IoT for traffic control [2]. However, these IoT services are divergent regarding latency requirements, periodicity or frequency of packet transmission, data size, and the number of IoT devices requesting the service. Moreover, the number of IoT devices is proliferating [3], and the data requested by each device is also increasing [4]. Therefore, the available bandwidth must be scheduled efficiently among network slices comprising the IoT services. This issue intensifies in the satellite IoT scenario [5], where the bandwidth for processing the IoT slices is fixed during a satellite pass. Hence, it is essential to distribute the available bandwidth rapidly and fairly among the slices.

A. MOTIVATION

Present communication networks are expanding from terrestrial broadband networks to integrated satellite-terrestrial networks to address diverse Quality of Service (QoS) demands of IoT services [6]. In an integrated satellite-terrestrial network, a satellite collects IoT service requests from the IoT devices and forwards them to a ground station or another satellite. However, the limited uplink bandwidth and the increasing volume of IoT traffic with diverse QoS demands during a satellite pass necessitates an optimal scheduling scheme to distribute the bandwidth among heterogeneous IoT services. Scheduling the available bandwidth between heterogeneous slices is challenging because we need to decide the allocation time in parallel to allocating the required bandwidth to the slices. Each slice refers to the service requests generated by the IoT devices for a particular IoT service [7]. Therefore, each slice has different bandwidth and latency demand based on the characteristics of the respective service [8]. For example, a latency-sensitive slice should be scheduled before another latency-tolerant slice. However, the prioritization of IoT services is a complex decision due to the presence of other relevant parameters related to IoT services. A fair scheduling strategy should consider latency and all the other relevant parameters in combination. Additionally, the scheduler should consider the trade-off between the scheduling algorithm's convergence time and the generated schedule's fairness for a satellite IoT scenario.

B. CONTRIBUTION

In this work, we propose an Inter-Slice Scheduler (ISS) to schedule the uplink bandwidth during a satellite pass between IoT devices requesting different IoT services. In the context of this work, slicing refers to a network management strategy that enables the efficient allocation of the bandwidth by logically dividing it into distinct segments or "slices." Each slice corresponds to a specific service, allowing for tailored resource allocation. Moreover, the ISS performs scheduling between the slices to determine the time period of allocating the bandwidth slice to each service. The proposed ISS considers service-specific parameters and generates a scheduling strategy that is proportionally fair [9], [10]. The specific contributions of the work are outlined as follows:

- We formulate the inter-slice scheduling problem as a $0-1$ mixed integer programming problem (MIP) that maximizes proportional fairness. Subsequently, we relax the MIP and formulate a penalized optimization problem to achieve the solution of the relaxed problem closer to the optimal solution of the MIP.
- We propose a distributed approach for scheduling slices to reduce the scheduling time in proportion to the latency demands of the slices.
- We propose a weighted greedy heuristic ISS with a low convergence time.
- A Simulated Annealing (SA)-based ISS algorithm is presented to improve the proportional fairness of the greedy solution.
- We performed extensive simulations to evaluate the effectiveness of the proposed ISS. Simulation results depict that the proposed ISS achieves higher proportional fairness with low convergence time compared to the benchmark solutions.

C. PAPER ORGANIZATION

The remainder of the paper is organized as follows. First, we discuss the state-of-the-art in Section II. Then, Section III presents the system model. Section IV discusses the formulation of the optimization problem. Section V presents the proposed heuristic solution for the ISS. Section VI evaluates the performance of the proposed solution. Finally, Section VII concludes the document.

II. RELATED WORK

This section discusses the related works on resource scheduling in terrestrial and non-terrestrial networks, fair resource scheduling, and SA-based resource scheduling.

A. RESOURCE SCHEDULING IN TERRESTRIAL AND NON-TERRESTRIAL NETWORKS

Recent works investigate different resource scheduling approaches in terrestrial and non-terrestrial networks. Cao et al. [11] proposed an algorithm that identifies the type of network slice and allocates resources based on the slice type in 6G networks. This work considers multiple resources, including CPU, storage, and bandwidth. Kodheli et al. [12] proposed an uplink resource scheduling strategy for narrowband IoT via Low Earth orbit (LEO) satellites. The authors formulate a 0-1 2D knapsack problem to assign users physical resource blocks (PRBs). A PRB comprises subcarriers in the frequency domain and slots in the time domain. The users are prioritized based on coverage time, channel, and buffer conditions. Based on the problem formulation, an exact solution, a greedy heuristic solution, and an approximate solution were proposed. Wang et al. [6] addressed the problem of scheduling the uplink channels of LEO satellites for IoT traffic. The authors proposed a joint SA and Monte Carlo algorithm to solve the problem.

B. FAIR RESOURCE SCHEDULING

Some of the existing works for resource scheduling focus on fairness. Fossati et al. [9] proposed a framework for the fair allocation of multiple resources among network slices. This work considers dependency between resources and congestion for the unavailability of resources. The proposed framework selects appropriate resource allocation based on the fairness goal. He et al. [13] proposed a scheduling scheme to allocate subcarriers among users requiring radio resources for 5G downlink transmission. The proposed approach is based on adaptive short-term traffic prediction and provides fair treatment to each user. Zhang et al. [14] formulate a joint user clustering, scheduling, and power allocation problem to maximize the proportional fairness of the users for downlink millimeter wave multi-user multiple input and single output non-orthogonal multiple access systems. For fair user scheduling, this work aims to maximize the long-term average throughput of the users.

C. SA-BASED RESOURCE SCHEDULING

SA is a meta-heuristic algorithm that finds a better neighborhood solution after evaluating candidate solutions based on acceptance criteria [15]. SA is a widely applied technique for scheduling problems [15], [16], [17]. For scheduling problems, the primary steps of the SA algorithm are the

TABLE 1. Differences between the proposed ISS and related works.

Work	*Non-Terrestrial Network*	*Fair Resource Allocation*	*Heterogeneous Traffic*
Cao *et al.* [11]	×	×	✓
Kodheli *et al.* [12]	✓	×	×
Wang *et al.* [6]	✓	✓	×
Fossati *et al.* [9]	×	✓	✓
He *et al.* [13], Zhang *et al.* [14]	×	✓	×
The Proposed ISS	✓	✓	✓

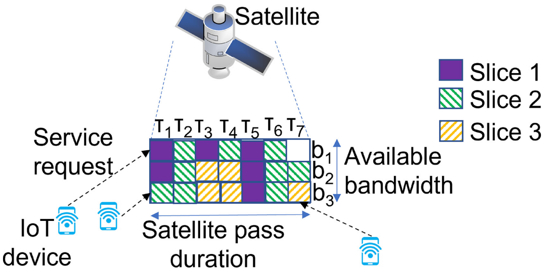

FIGURE 1. Satellite uplink bandwidth allocation during a satellite pass.

TABLE 2. Table of symbols.

Symbol	*Definition*
h	Altitude of a LEO satellite
E	Radius of earth
θ_{min}	Minimum elevation angle at an IoT device
γ	IoT device density
c	Packet size
S	Set of services
l_i	Latency-sensitivity of $s_i \in S$
r_i	Packet transmission frequency per hour for $s_i \in S$
b_i	Requested packet count per day per IoT device for $s_i \in S$
σ	Scheduling duration
$U_i[t]$	Set of IoT devices corresponding to s_i during the t^{th} satellite pass
p_i	Priority of s_i
B	Total available uplink bandwidth (Bps) of the satellite
y_k	k^{th} bandwidth block
τ_l	l^{th} time block
b_{block}	Size of a bandwidth block
t_{block}	Duration of a time block
g	Size of a bandwidth-time block
ξ	Spectral efficiency
$\alpha_{k,l}$	2D block referring to k^{th} bandwidth block and l^{th} time block

representation of a schedule as a state, identification of an initial schedule, selection of a neighbor solution represented by the next state, and the acceptance criteria defined by cost function and acceptance probability. In addition, SA escapes local optima by allowing an inferior move so that a suitable solution can be found in a later iteration. Initial temperature, cooling rate, and the length of Markov are the metrics used for defining the stopping criteria for SA in terms of the number of iterations. Khurshid et al. [15] use SA for the flow shop scheduling problem to schedule jobs to machines so that the makespan is minimized. Zhou et al. [16] proposed an SA-based solution for the colored traveling salesman problem, a particular class of scheduling problems. Liu et al. [17] compute a task offloading schedule based on SA for multiaccess edge computing in IoT systems. Table 1 shows the difference between the proposed ISS and related works.

As depicted in Table 1, the existing literature does not address fair resource allocation in non-terrestrial networks in the presence of heterogeneous traffic. Therefore, this paper proposes an efficient scheme to schedule the uplink bandwidth fairly among multiple IoT devices requesting satellite IoT services. The proposed solution includes a weighted-greedy solution with low convergence time and an SA-based solution proportionally fair for heterogeneous IoT slices. Also, different from related works, in this work, we consider iterative resource allocation for multiple passes of the satellite with leftover user demands carried forward to the next iteration or pass.

III. SYSTEM MODEL

For the system model, we consider a Low Earth Orbit (LEO) satellite that covers multiple on-ground IoT devices during its pass. The IoT devices request different services from different slices based on the QoS demands. However, the uplink bandwidth of the satellite is limited and available only for the satellite pass duration. Therefore, the LEO satellite allocates subchannels to the IoT devices. A subchannel refers to a portion of the uplink bandwidth for an interval within the satellite pass duration. Figure 1 shows the scenario for allocating satellite uplink bandwidth to multiple slices during a satellite pass. Table 2 shows the symbols used in this paper.

A. IOT DEVICE MODEL

The LEO satellite accepts service requests from IoT devices within its coverage area during a pass. The area covered by the satellite during the t^{th} pass is given by:

$$A[t] = 2\pi E^2(1 - \cos\psi[t]), \tag{1}$$

where E is earth's radius in kilometers and $\psi[t]$ is the angular radius of the coverage circle during the t^{th} pass which is estimated as:

$$\psi[t] = \arccos\left(\frac{E}{E + h[t]}\cos\theta_{min}\right) - \theta_{min}, \tag{2}$$

where $h[t]$ is the altitude of the satellite in kilometers during the t^{th} pass and θ_{min} is the minimum elevation angle at an IoT device [18].

Let γ be the IoT device density per square kilometer area. Therefore, the total number of IoT devices covered by the satellite during the t^{th} pass is:

$$U[t] = A[t]\gamma \tag{3}$$

B. SERVICE REQUEST MODEL

Let S denote the set of services where each service corresponds to a network slice. The i^{th} service $s_i \in S$ is represented by a tuple $<l_i, r_i, b_i>$, where l_i denotes the latency-sensitivity of the service with higher value signifying highly latency-sensitive service, r_i is the frequency of packet transmission per hour, and b_i denotes the requested packet count per day per IoT device for the service request. We assume that an IoT device requests a single type of service. Let $U_i[t] \in U[t]$ be the set of IoT devices requesting the i^{th} service during the t^{th} satellite pass. In case of limited resource availability, e.g., end of the satellite pass, some IoT device demands can be addressed in the subsequent passes of the same satellite or another satellite covering the same area. Therefore, the set $U_i[t]$ also includes the IoT devices with residual demands from the previous satellite passes.

We assign priority values to each service for fair allocation of the available resources. Service priority is proportional to the packet transmission frequency, the requested number of packets per day per IoT device for the service, and the number of IoT devices requesting the service. Additionally, the priority of a service is high if it is latency-sensitive.

Definition 1 (Service Priority): The priority of s_i at the t^{th} satellite pass is defined as:

$$p_i[t] = \mu\left(\frac{l_i}{\sum_{s_k \in S} l_k} + \frac{r_i}{\sum_{s_k \in S} r_k} + \frac{b_i}{\sum_{s_k \in S} b_k} + \frac{|U_i[t]|}{U[t]}\right), \tag{4}$$

where $\mu > 1$ is the multiplication factor to increase the priority of leftover demand for processing in the subsequent pass of the satellite. Further, we consider $\mu = 1$ for the traffic that has required bandwidth fully allocated during the satellite pass when the service is requested.

The definition of service priority based on factors such as packet transmission frequency, requested packets per day per IoT device, and the number of IoT devices requesting the service is justified as it accounts for the diverse needs of IoT applications. These factors reflect the varying data requirements and resource demands across different services and devices, ensuring fair allocation of resources and efficient network utilization. Additionally, considering the latency-sensitivity of services for determining the service priority aligns with the critical nature of real-time data delivery in IoT applications, further optimizing the quality of service for time-sensitive tasks.

C. BANDWIDTH-TIME GRID

Let B denote the total uplink bandwidth available during a pass. We divide B into m bandwidth blocks such as $y_1, y_2, \dots, y_m$ each with size $b_{block} = \frac{B}{m}$ units. Similarly, we divide the satellite pass duration in minutes σ into n time blocks such as $\tau_1, \tau_2, \dots, \tau_n$ each with a duration of $t_{block} = \frac{\sigma}{n}$ minutes. The size of one bandwidth-time block is $g = b_{block}t_{block}$. The uplink data demand of the i^{th} service during σ is $D_i = \frac{b_i|U_i[t]|c}{24*60}\sigma$, where c denotes the size of a packet. Therefore, the number of blocks required by a service s_i is $\lceil\frac{D_i}{g\xi}\rceil$, where ξ is the spectral efficiency of the adopted communication protocol. For simplicity, we assume that the signal-to-noise ratio (SNR) does not vary significantly during the satellite pass, and hence the spectral efficiency is assumed fixed.

We define the two-dimensional (2D) bandwidth-time block to be α and the symbol $\alpha_{k,l}$ refers to the 2D block representing the bandwidth block y_k and time block τ_l. Figure 1 shows the bandwidth-time grid for $m = 3$ and $n = 7$.

ISS allocates the 2D blocks to the IoT devices. The following binary variable defines the allocation of a 2D block $\alpha_{k,l}$ to a user $u_{i,j}$ during the t^{th} satellite pass:

$$\beta_{i,j}^{k,l}[t] = \begin{cases} 1 & \text{if block } \alpha_{k,l} \text{ is allocated to } u_{i,j} \text{ in the } t^{th} \text{ pass,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Therefore, $\beta_{i,j}^{k,l}[t] = 1$ signifies that the bandwidth block y_k is allocated to $u_{i,j}$ during the time block τ_l.

IV. PROBLEM FORMULATION

The work aims to allocate the 2D blocks among the IoT devices requesting different services and maximize the proportional fairness for all services. The proportional fairness signifies weighted block allocation according to respective service priority [9]. For the t^{th} satellite pass, the problem is formulated as:

$$\textbf{OP1:} \ \underset{\beta}{\text{Maximize}} \sum_{s_i \in S} p_i[t]R^i[t] \tag{6}$$

subjectto

$$\sum_{s_i \in S} \sum_{u_{i,j} \in U_i} \beta_{i,j}^{k,l}[t] \le 1, \forall k \le m, \forall l \le n, \tag{7}$$

$$\sum_{s_i \in S} R^i[t] \le mn, \tag{8}$$

$$\left\lceil \frac{D_i}{g\xi} \right\rceil \le R^i[t], \forall s_i \in S, \tag{9}$$

$$\beta_{i,j}^{k,l}[t] \in \{0, 1\}, \forall u_{i,j} \in U_i[t], \forall s_i \in S, \ k \le m, l \le n, \tag{10}$$

where $R^i[t] = \sum_{u_{i,j} \in U_i[t]} \sum_{k=1}^{m} \sum_{l=1}^{n} \beta_{i,j}^{k,l}[t]$ gives the number of blocks allocated to s_i during the t^{th} satellite pass. Equation (6) provides the expression for maximizing the sum of weighted block allocation for all services where the weights refer to the service priorities. Equation (7) states that only one IoT device can be allocated to each block. Equation (8) ensures that the allocated number of blocks does not exceed the available mn, which signifies the available uplink bandwidth. Equation (9) ensures that each service's allocated number of blocks satisfies the service requirement. Equation (10) expresses the binary decision variable. The formulated optimization problem (OP1) is a $0-1$ MIP (or combinatorial) problem due to the binary nature of the decision variable β, which is intractable to solve in polynomial time.

A. RELAXED FORMULATION

In this subsection, we present the relaxed formulation for the MIP problem OP1 stated in Equation (10) to circumvent the combinatorial nature of OP1. For this purpose, the decision variable $\beta[t]$ has to be relaxed between 0 and 1. The relaxation makes the problem convex. Nevertheless, the relaxed problem does not guarantee binary value for $\beta[t]$. However, the binary value is essential in bandwidth-time block allocation. Therefore, to acquire binary values for $\beta[t]$, the relaxed problem is further penalized as follows:

$$\textbf{OP2:}\ \underset{\beta}{\text{Maximize}} \quad \sum_{s_i\in S} p_i[t]R^i[t] + \rho \sum_{s_i\in S}\sum_{u_{i,j}\in U_i[t]} P\Big(\beta^{k,l}_{i,j}[t]\Big) \tag{11}$$

$$\text{subjectto} \quad (7),\ (8),\ (9)$$
$$0 \le \beta^{k,l}_{i,j}[t] \le 1, \forall u_{i,j} \in U_i[t], \forall s_i \in S,$$
$$k \le m, l \le n, \tag{12}$$

where ρ is the penalty parameter controlling the binary nature of $\beta[t]$. The penalty function $P(\beta^{k,l}_{i,j}[t])$ is considered as $(\beta^{k,l}_{i,j}[t])^2 - \beta^{k,l}_{i,j}[t]$, which is a convex function in the region of $[0, 1]$ that produces no penalty at either 0 or 1 and increases the penalty as $\beta[t]$ moves away from 0 or 1 with the highest penalty at $\beta[t] = 0.5$. The formulation of the penalty function ensures that the relaxed optimization problem produces the output favorable to 0 or 1.

Since OP2 is a subtraction of two convex functions with linear constraints, the problem OP2 is a class of difference of convex (DC) programming. Therefore, we utilize the iterative convex-concave procedure (CCP) technique [19] for solving OP2. In CCP, the following two steps are executed successively till its convergence: (1) Taylor series approximation around the second convex function in the objective, and (2) the next update $(\beta^{k,l}_{i,j})^{f+1}[t]$ is obtained by solving the following:

$$\textbf{OP3:}\ \underset{\beta}{\text{Maximize}} \quad \sum_{s_i\in S} p_i[t]R^i[t]$$
$$+\rho \sum_{s_i\in S}\sum_{u_{i,j}\in U_i[t]} \left(\beta^{k,l}_{i,j}[t] - \left(\beta^{k,l}_{i,j}\right)^{f-1}[t]\right)$$
$$\nabla P\Big(\beta^{k,l}_{i,j}[t]\Big)$$
$$\text{subject to} \quad (7),\ (8),\ (9),\ (12) \tag{13}$$

The high convergence time of OP3 motivates us to investigate heuristic solutions for the ISS, which we discuss in the next section. Further, Section VI shows the analysis of the convergence time of OP3.

V. THE PROPOSED SCHEME

In this section, we discuss the heuristic solution proposed for ISS. For efficient scheduling, the ISS considers N groups of service requests in the order of latency demands and allocates them to a bandwidth-time block with total bandwidth B and duration of $\sigma' < \sigma$, where $N\sigma' = \sigma$. Let m and n be the numbers of bandwidth and time blocks, respectively, for each group. The schedules for all groups are generated in parallel at the beginning of a satellite pass. However, a generated schedule for the q^{th} group can be assigned to respective IoT devices after a duration of $(q-1)\sigma'$ minutes. Therefore, the IoT devices are scheduled according to respective latency sensitivity. For each group, we propose a weighted greedy algorithm for fast scheduling the slices based on the respective priority. However, the greedy algorithm is not optimal for maximizing proportional fairness in adverse conditions, such as when the available resource is insufficient to address the aggregated demand. Therefore, based on the initial solution generated by the greedy algorithm, we propose an SA-based algorithm to generate a proportionally fair solution when the available resource is insufficient compared to the demand. For both algorithms, we define a state as $\Omega = \{\beta^{k,l}_{i,j}[t] | u_{i,j} \in U'_i[t], s_i \in S, k \le m, l \le n\}$, where $U'_i[t] \subset U_i[t]$ is the IoT device count in a group.

Figure 2 shows the workflow of the proposed scheme for a case where the total requests are grouped into 3 groups. It is worth mentioning that in the context of our work, a "group" is different from a "slice". As mentioned, a "slice" maps to a specific service. Whereas a "group" can contain requests for different services. The bandwidth-time grids for each group are computed in parallel. The weighted greedy ISS generates the initial solution for each group, upgraded by the SA ISS. The final bandwidth-time grid is obtained by sequentially combining the bandwidth-time grids for all groups. In the example shown in Figure 2, the generated schedule for group 1 can be applied as soon as the algorithms for weighted greedy ISS and SA ISS converge, which depends on the IoT device count and grid size for the group. However, the generated schedules for groups 2 and 3 can be applied only after σ' and $2\sigma'$ duration, respectively.

A. WEIGHTED GREEDY ISS SOLUTION FOR A GROUP

The weighted greedy ISS mainly focuses on service priority to allocate available blocks and schedule the services. Algorithm 1 shows the steps of the proposed weighted greedy heuristic algorithm for generating the initial solution for the satellite's t^{th} pass. Initially, the IoT devices are sorted in descending order of the priorities defined in Equation (4). For each IoT device, if the allocated numbers of bandwidth-time blocks multiplied by block size are less than the required

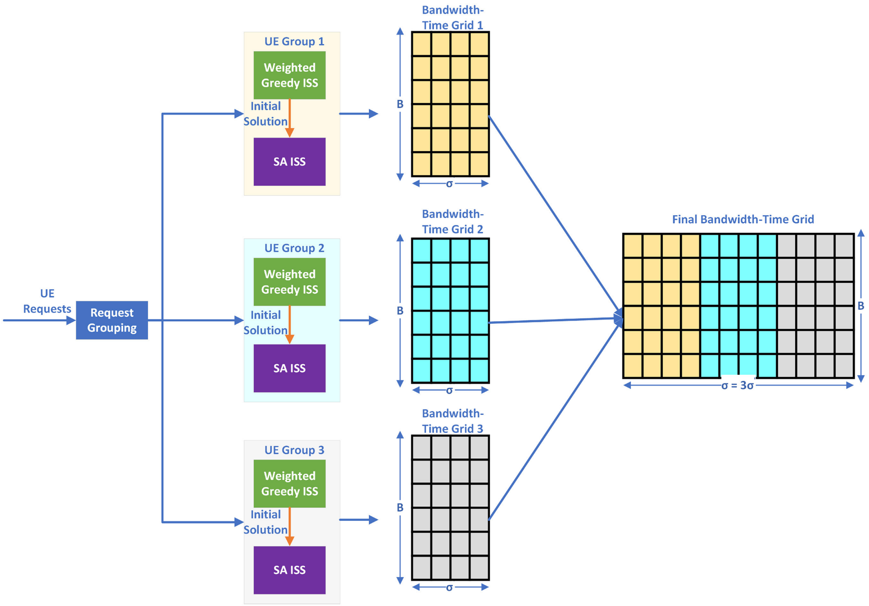

FIGURE 2. Example showing the workflow of the proposed scheme. The total requests are grouped into 3 groups. The bandwidth-time grids for each group are computed in parallel. The final bandwidth-time grid is obtained by sequentially combining the bandwidth-time grids for all groups.

Algorithm 1 Weighted Greedy ISS Algorithm

```
INPUTS: $S$, $U'[t]$, $B$, $x$, $m$, $n$
OUTPUT: $\Omega^0$                                  ▷ Initial state
PROCEDURE:
  $SortedList \leftarrow$ Sort IoT devices in descending order of priorities
  for $u_{i,j} \in SortedList$ do
      if $x_{i,j}[t] = 1$ then
          $s^{alloc} \leftarrow 0$      ▷ Allocated number of bandwidth-time blocks
          for each $k \leq m$ do      ▷ Visit bandwidth blocks in ascending order
              for each $l \leq n$ do      ▷ Visit time blocks in ascending order
                  if $(s^{alloc}g < b_i$ and $\sum_{s_i \in S} \sum_{u_{i,j} \in U_i[t]} \beta^{k,l}_{i,j}[t] = 0$ then
                      $\beta^{k,l}_{i,j}[t] \leftarrow 1$, $s^{alloc} \leftarrow s^{alloc} + 1$
                  end if
              end for
          end for
      end if
  end for
  return $\Omega^0 \leftarrow \{\beta^{k,l}_{i,j}[t] | u_{i,j} \in U'_i[t], s_i \in S, k \leq m, l \leq n\}$
```

Algorithm 2 SA ISS Algorithm

```
INPUTS: $\Omega^0$, $T_0$, $z$, $q$, $S$, $U'[t]$, $B$, $x$, $m$, $n$
OUTPUT: $\Omega$
PROCEDURE:
  Set $T \leftarrow T_1$, $\Omega \leftarrow \Omega^0$
  while $T > 1$ do
      $\Omega^{next} \leftarrow GenerateNextState(\Omega)$
      if $exp\left(\frac{Cost(\Omega) - Cost(\Omega^{next})}{T}\right) > q$ then
          Set $\Omega \leftarrow \Omega^{next}$
      end if
      Set $T \leftarrow z \times T$
  end while
  return $\Omega \leftarrow \{\beta^{k,l}_{i,j}[t] | u_{i,j} \in U'_i[t], s_i \in S, k \leq m, l \leq n\}$
```

data size, blocks without any assigned IoT device are allocated to the IoT device, and $\beta[t]$ is updated accordingly. Finally, the output of the algorithm is the initial state $\Omega^0 = \{\beta^{k,l}_{i,j}[t] | u_{i,j} \in U'_i[t], s_i \in S, k \leq m, l \leq n\}$.

B. SA ISS SOLUTION FOR A GROUP

The schedule generated by the greedy ISS does not consider the fair allocation of bandwidth demand among heterogeneous slices during high traffic volume. Therefore, for scheduling the bandwidth-time blocks to the IoT devices fairly, an SA-based meta-heuristic algorithm is designed. For each pass of the satellite, the SA algorithm starts with an initial solution determined by the weighted greedy heuristic algorithm.

Algorithm 2 depicts the steps of the SA-based optimal inter-slice scheduling algorithm. The algorithm's inputs include the initial state generated by Algorithm 1 and parameters required for SA. The parameters needed for SA are initial temperature T_0, the cooling rate z, and acceptance probability q. The initial temperature determines the convergence time of the SA algorithm, where a high value of T_0 signifies that the time to reach the global optimal solution is high, and a low T_0 may direct the algorithm to a locally optimal solution. The cooling rate determines the decrease in the temperature, and the algorithm terminates when the temperature reaches 0. The acceptance probability

Algorithm 3 GenerateNextState

```
INPUT: $\Omega$, $S$, $U'[t]$, $B$, $x$, $m$, $n$
OUTPUT: $\Omega^{next}$                                    ▷ Next state
PROCEDURE:
$\Omega^{next} \leftarrow \Omega$
Randomly select a block $\alpha_{k,l}$
$U''[t] \leftarrow$ IoT devices which have allocated blocks less than required blocks
if $U''[t] \neq \phi$ then                                  ▷ Non-empty set
    Select the IoT device $u_{i,j} \in U''[t]$ that has the minimum value for the expression $(\frac{\text{allocated blocks}}{\text{required blocks}})(\frac{1}{\text{service priority}})$
else
    Select the IoT device $u_{i,j} \in U'[t]$ that has the minimum value for the expression $(\frac{\text{allocated blocks}}{\text{required blocks}})(\frac{1}{\text{service priority}})$
end if
Set $\beta_{i,j}^{k,l} \leftarrow 1$
return $\Omega^{next} \leftarrow \{\beta_{i,j}^{k,l}[t] | u_{i,j} \in U_i'[t], s_i \in S, k \leq m, l \leq n\}$
```

q decides whether a solution is acceptable or not. We define the value of q based on the simulation outcome.

The SA algorithm starts with the initial state as the current state. The algorithm executes until the value of the temperature T reaches terminating threshold 1 as mentioned in the loop in Line 2 of Algorithm 2. The value of T decreases at the rate of z at each step. In each loop step, a new state or schedule is generated using Algorithm 3. Subsequently, the cost, which refers to the fairness of resource allocation, is evaluated both for the current and the new states. If the value of the expression $exp(\frac{Cost(\Omega)-Cost(\Omega^{next})}{T})$ is higher than the acceptance probability q, then the new state is acknowledged as the current state. The algorithm continues until the termination condition is reached. This signifies that the new state or block allocation is significantly better than the previous allocation regarding proportional fairness.

Algorithm 3 demonstrates the steps of generating the next state given a current state. We randomly select a block $\alpha_{k,l}$ and alter its allocation status. The set $U'[t]$ lists the set of IoT devices that do not have the required number of blocks allocated. The selected block is allocated to the IoT device $u_{i,j} \in U''[t]$ that has the minimum value for the expression $(\frac{\text{allocated blocks}}{\text{required blocks}})(\frac{1}{\text{service priority}})$. Let all IoT devices have the required number of blocks allocated. In that case, The selected block is allocated to the IoT device $u_{i,j} \in U'[t]$ that has the minimum value for the expression $(\frac{\text{allocated blocks}}{\text{required blocks}})(\frac{1}{\text{service priority}})$. The cost of a state is the negative of proportional fairness. For a state Ω, the amount of blocks allocated to a service s_i during the t^{th} pass of the satellite is given by:

$$R_{\Omega}^{i}[t] = \sum_{u_{i,j} \in U_i[t]} \sum_{k=1}^{m} \sum_{l=1}^{n} \beta_{i,j}^{k,l}[t]. \tag{14}$$

Therefore, the cost of a state Ω is given by:

$$Cost(\Omega) = -\sum_{s_i \in S} p_i[t] R_{\Omega}^{i}[t] \tag{15}$$

If the number of blocks allocated is less than the required number, the leftover block demand is forwarded over the satellite's next pass. We assume the allowable delay within which the leftover traffic should be processed exceeds the interval between consecutive satellite passes. Additionally, as stated in Equation (4), we increase the priority of the leftover demand by multiplication factor μ so that it is processed as early as possible. However, it is unusual for services with high priority to have leftover demand because the ISS prefers these services regarding allocating the required bandwidth-time blocks.

C. TIME COMPLEXITY ANALYSIS

In this subsection, we analyze the time complexity of the proposed ISS solution.

The time complexity of Algorithm 1 involves two main steps – (1) sorting the IoT devices in descending order of priorities which take $O(|U'[t]|log(|U'[t]|))$ time and (2) allocating bandwidth-time blocks to each IoT device which takes $O(|U'[t]|mn)$ time. Therefore, the time complexity of Algorithm 1 is $O(|U'[t]|log(|U'[t]|) + |U'[t]|mn) \approx O(|U'[t]|mn)$ as the number of bandwidth-time blocks is usually higher than the value $log(|U'[t]|$.

For Algorithm 2, the parameters T_0 and L influence the time complexity. The two loops in Line 2 and Line 3 iterate together for $L \log_{\frac{1}{z}} T_0$ times approximately [21]. Line 4 takes $O(|U'[t]|)$ time to select the IoT device for generating the next state as mentioned in Algorithm 3. Line 5 takes $O(|S||U'[t]|mn)$ time to compute the cost of a state as mentioned in Equations (14)–(15). Therefore, the time complexity for Lines $4-7$ in Algorithm 2 is $O(|U'[t]|| + |S||U'[t]|mn) \approx O(|S||U'[t]|mn)$. Accordingly, the time complexity of Algorithm 2 is $O(|S||U'[t]|mnL \log_{\frac{1}{z}} T_0)$. As Algorithm 2 starts with the initial state generated by the greedy ISS, the time complexity of the SA ISS solution includes the time required by Algorithm 1.

Therefore, the computational complexity of the proposed ISS solution depends on multiple factors, such as the number of services, the number of IoT devices, the size of the bandwidth-time grid, and the SA parameters. These factors can be defined carefully based on the available processing capacity of the servers.

VI. PERFORMANCE EVALUATION

A. SIMULATION SETTINGS

We evaluate the performance of ISS by performing simulations in the MATLAB platform. For the simulation, we consider 5 type of services such as smart home IoT, smart city IoT, agriculture IoT, vehicular IoT for tracking, and vehicular IoT for traffic control. Each IoT device requests one service only. Therefore, the number of IoT devices is equivalent to the traffic demand or the number of service requests. The requirements of these IoT services are described in Table 3. Table 4 depicts other parameters considered for the simulation. The ISS solution can be

TABLE 3. Requirements of IoT services.

Service	Latency-sensitivity	r_i (per hour) [20]	b_i (packets per day per device) [20]	Percentage of devices [20]
Smart home IoT	Medium	0.5	12	37
Smart city IoT	Medium	0.04	1	12
Agriculture IoT	High	1	24	24
Vehicular IoT for tracking	Low	6	144	9
Vehicular IoT for traffic control	High	10	240	18

TABLE 4. Simulation parameters.

Parameter	Value
Number of satellite passes	$1-3$
Altitude of LEO satellite (h)	900 km [18]
Number of services	5
IoT device density per square kilometer (γ)	$[0.1-25]\times10^{-4}$
Uplink bandwidth (B)	200 MHz
Spectral efficiency (ξ)	1 bps/Hz
Minimum elevation angle at the IoT device	15 degrees [18]
Satellite pass duration (σ)	15 minutes
Number of bandwidth blocks (m)	200
Number of time blocks for a group (n)	60
Initial temperature (T_0)	100
Cooling rate (z)	0.95
Acceptance probability (q)	0.85
Multiplication factor for leftover demand (μ)	2

implemented on the ground station where the schedule for different groups is determined in parallel, as discussed in Section V.

B. BENCHMARK SCHEME

We consider the scheme SA-MC proposed in [6] for performance evaluation of the greedy ISS and SA ISS. The greedy ISS only refers to the solution with the weighted greedy algorithm (Algorithm 1). In contrast, the SA ISS refers to the SA-based solution described in Algorithm 2 where the initial state considered is the solution of Algorithm 1. We select the greedy ISS as a benchmark to emphasize the need for the proposed SA-based solution. We prefer SA-MC as a benchmark because it considers a satellite IoT scenario and aims to schedule uplink transmission based on an SA-based algorithm. SA-MC seeks to minimize the difference between demanded resources and allocated resources. SA-MC starts from a randomly generated initial schedule based on which an SA-based algorithm iterates. The next state for the SA-MC algorithm is generated simply by perturbing the current state schedule. The authors propose SA-MC as a single-shot algorithm without considering residual demands. Therefore, for comparison with our proposed weighted greedy and SA-based algorithms, we consider that for SA-MC, the residual demand is carried forward to the next satellite pass.

C. PERFORMANCE METRICS

We consider the following performance metrics to evaluate the performance of the proposed ISS solutions:

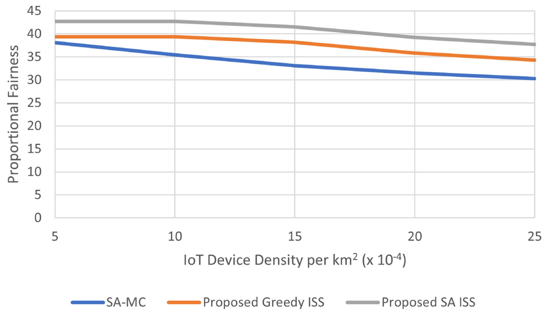

FIGURE 3. Effect of IoT device density on proportional fairness.

- *Proportional Fairness:* This metric ensures fair resource allocation among services with diverse requirements.
- *Service Allocation Ratio:* For each service, we compute the Service Allocation Ratio $= \frac{\text{Blocks allocated to a service}}{\text{Blocks required by a service}}$. The allocation ratio is an important metric for a scenario when the available resource is insufficient for the aggregated resource requirement of service. In this case, efficient resource scheduling ensures that the allocation ratio of each service is proportional to the respective service priority.
- *Residual Demand:* This metric refers to the aggregated demand for unaddressed services. For service s_i, the residual demand is expressed as $(b_i|U_i[t]|c\sigma - R^i[t]g)$.
- *Convergence Time:* This metric quantifies the time a scheduling algorithm requires to converge on a solution. Low convergence time is essential for satellite IoT scenarios due to time-varying topology.

D. RESULT AND DISCUSSION

1) PROPORTIONAL FAIRNESS

We investigate weighted proportional fairness as a performance metric to ensure that all slices share the available resource fairly. For the simulation results shown in Figure 3, we consider 3 satellite passes. Figure 3 shows that SA ISS is fairer than the weighted greedy ISS and SA-MC. The performance gap between the greedy ISS and the SA ISS increases as the IoT device count rises. For 5×10^{-4} IoT devices per square kilometer, the SA solution is 8.46% and 12.14% fairer than the greedy ISS and SA-MC, respectively. This is because the greedy ISS allocates the IoT devices available bandwidth-time blocks in priority order. However, as the IoT device count increases, the available resource becomes insufficient to address the service demands. In this case, the greedy ISS prefers high-priority IoT devices,

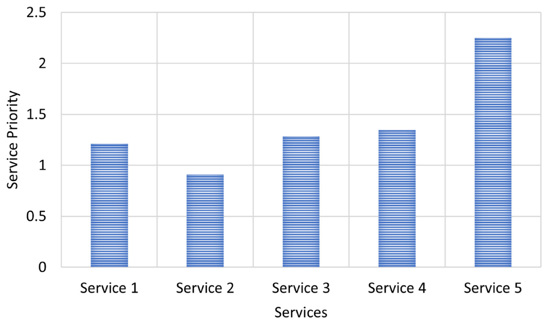

FIGURE 4. Priority of services.

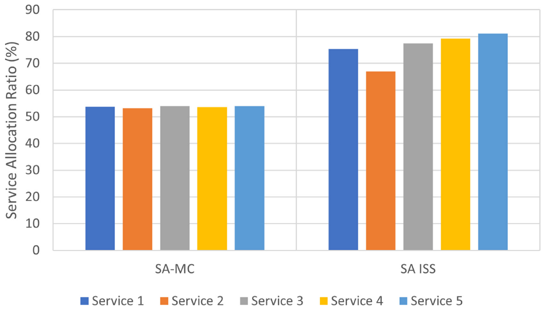

FIGURE 5. Service allocation ratio.

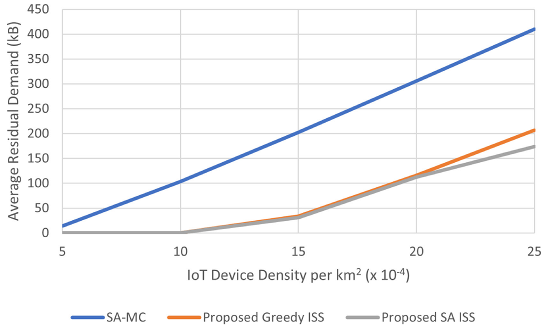

FIGURE 6. Residual demand for 100 IoT devices.

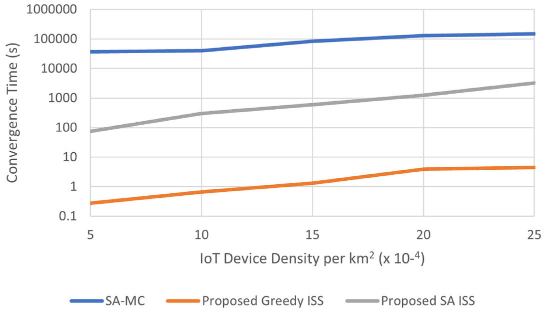

FIGURE 7. Effect of IoT device density on convergence time.

and the low-priority IoT devices suffer starvation. On the other hand, the SA ISS considers the negative of weighted proportional fairness as cost and aims to reduce the cost, eventually resulting in a fair allocation of limited resources among the services. In particular, for 25×10^{-4} IoT devices per square kilometer, the SA ISS solution is 9.92% fairer than the Greedy ISS solution. However, the initial state or schedule for SA-MC is selected randomly, and the cost function considers only the remaining demand based on the allocation. Therefore, the SA-MC solution cannot guarantee high fairness for high traffic demand or IoT device count.

2) ALLOCATION RATIO

Figure 4 depicts the priorities of the 5 IoT services we consider for performance evaluation. Figure 5 depicts the allocation ratio for the 5 IoT services for SA-MC and SA ISS solutions, respectively. The simulation results show that the SA ISS solution is more accurate than the benchmark in allocating the resources to all services proportionally to their respective priority. This is because the SA ISS solution distributes the available bandwidth in the order of service priority. This is done by considering the service priority and allocated resources as parameters for generating the next state as described in Algorithm 3. In contrast, SA-MC does not consider service priority, which results in a similar allocation ratio for all services. Also, SA-MC shows a lower allocation ratio than SA ISS because SA-MC starts with a random allocation for which a sufficiently high number of SA iterations are required to reach an optimal state. However, SA ISS begins with the solution provided by the Greedy ISS and achieves an optimal solution sooner than SA-MC, given the same parameter for SA.

3) RESIDUAL DEMAND

As shown in Figure 6, the residual demand for SA-MC is higher than greedy ISS and SA ISS. This is because SA-MC randomly selects and inverts the initial state to generate the next state. Due to this, some of the resources remain unallocated. On the other hand, the average residual demand for SA ISS is less because the SA ISS algorithm prioritizes the IoT devices with residual demands in each pass to generate the next state, as stated in Algorithm 3. Moreover, the average residual demand shows a non-linear trend for high IoT device density. As IoT device density grows, the strain on the available resources grows, and the number of unaddressed services increases with the corresponding residual demand.

4) CONVERGENCE TIME

Figure 7 shows the convergence time for varying IoT device numbers. The proposed weighted greedy ISS and SA ISS converge significantly faster than the SA-MC. This is because the weighted greedy ISS and the SA ISS generate the parallel schedule for multiple groups, as stated in Section V. The SA algorithm takes more time to converge than the weighted greedy algorithm because of the next state generation and cost calculation sub-processes. The SA-MC takes the highest time to converge because SA-MC considers all IoT devices together, and the dimension of a state is high. Therefore, SA-MC takes more time to generate the next state and calculate the cost of a state.

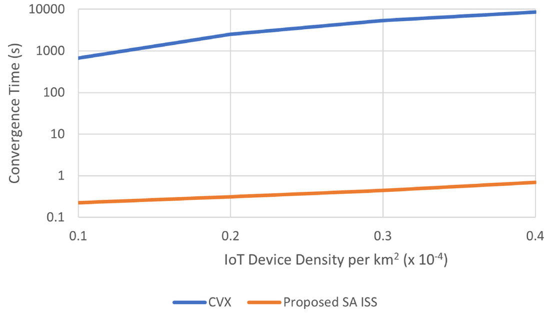

FIGURE 8. Comparison with relaxed solution: convergence time.

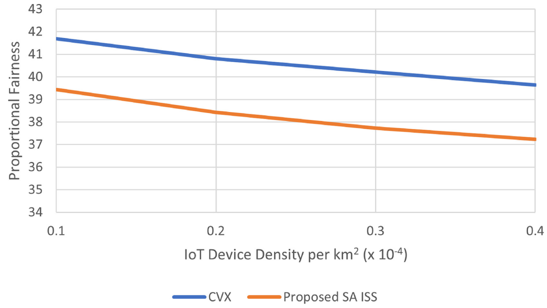

FIGURE 9. Comparison with relaxed solution: proportional fairness.

5) COMPARISON WITH THE RELAXED SOLUTION

We implemented the relaxed optimization problem OP3 stated in Equation (13) using CVX [22]. Figure 8 and Figure 9 compare the CVX solution with SA ISS regarding convergence time and proportional fairness, respectively. As shown in Figure 8, the convergence time for the CVX solution increases exponentially with the increasing number of IoT devices. Combining the simulation result shown in Figure 9, we yield that the CVX solution achieves higher proportional fairness than SA ISS at the expense of convergence time. In particular, the CVX solution is 6.24% fairer than SA ISS.

Based on the extensive simulation, we deduce that compared to the benchmarks, the SA ISS is fairer for high demand and low resource availability. Moreover, the proposed SA ISS addresses QoS demands of latency-sensitive requests by scheduling them earlier than other requests. In addition, for low demand, the proposed weighted greedy ISS solution can be used to save computation power.

6) PRACTICAL IMPLICATIONS, LIMITATIONS, AND APPLICATIONS OF THE RESULTS

The practical implication of the proposed scheme lies in its ability to ensure fair and efficient allocation of uplink bandwidth for IoT slices with diverse latency demands. However, in this work, we assumed that an IoT device requests a single type of service. This is a limitation of the proposed scheme because, in a real scenario, each device can request multiple services. Therefore, this simplification can affect the applicability of the proposed scheme in a highly dynamic satellite IoT environment. The applications of this research extend to various domains where satellite-based IoT services are employed. These applications include environmental monitoring, precision agriculture, disaster management, and remote sensing. In these contexts, the proposed inter-slice scheduling scheme can enhance the reliability and efficiency of data transmission, leading to more effective decision-making processes.

VII. CONCLUSION

In this paper, we proposed an approach for resource scheduling among heterogeneous satellite IoT services. The proposed scheme maximizes proportional fairness for resource allocation. We proposed a weighted greedy solution, which is a faster and lighter solution with low accuracy in terms of fairness concerning service demands. In addition, we proposed an SA-based solution with higher accuracy. Simulation results show that the SA ISS takes significantly less time to schedule highly latency-sensitive flows and achieves higher proportional fairness for high resource demand with insufficient availability. Therefore, the proposed greedy ISS solution is appropriate for a low IoT device density as it consumes less computation power. In contrast, the proposed SA ISS is suitable for a high IoT device density because of higher fairness in scheduling. The simulation results show that the SA ISS solution achieves 9.01% and 21.11% more proportional fairness than the weighted greedy ISS and the existing SA-MC scheme, respectively.

In the future, we aim to enhance the proposed inter-slice scheduling scheme by incorporating and managing multiple resources with complex inter-dependencies. This extension will enable us to address more intricate real-world scenarios and optimize resource allocation in broader network environments. Furthermore, we intend to investigate the application of inter-slice scheduling techniques to satellite constellations, where unique challenges related to orbital dynamics, handovers using inter-satellite links, and latency-sensitive communications require specialized scheduling solutions.

ACKNOWLEDGMENT

As part of this consortium, headed by OQ Technology Sarl, SnT and Leafspace Srl are partners and Subcontractors of this consortium. This work acknowledges the consortium partners in evolution of this activity. The views of the authors of this paper do not necessarily reflect the views of ESA or the other members of the consortium.

REFERENCES

[1] X. He, H. Lu, M. Du, Y. Mao, and K. Wang, "QoE-based task offloading with deep reinforcement learning in edge-enabled Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2252–2261, Apr. 2021.

[2] G. Ruggeri, M. Amadeo, C. Campolo, A. Molinaro, and A. Iera, "Caching popular transient IoT contents in an SDN-based edge infrastructure," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3432–3447, Sep. 2021.

[3] J. A. Ansere, G. Han, L. Liu, Y. Peng, and M. Kamal, "Optimal resource allocation in energy-efficient Internet-of-Things networks with imperfect CSI," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5401–5411, Jun. 2020.

[4] J. Huang, M. Wang, Y. Wu, Y. Chen, and X. Shen, "Distributed offloading in overlapping areas of mobile edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13837–13847, Aug. 2022.

[5] L. Yan, X. Ding, and G. Zhang, "Dynamic channel allocation aided random access for SDN-enabled LEO satellite IoT," *J. Commun. Inf. Netw.*, vol. 6, no. 2, pp. 134–141, Jun. 2021.

[6] L. Wang, S. Liu, W. Wang, and Z. Fan, "Dynamic uplink transmission scheduling for satellite Internet of Things applications," *China Commun.*, vol. 17, no. 10, pp. 241–248, Oct. 2020.

[7] D. Marabissi and R. Fantacci, "Highly flexible RAN slicing approach to manage isolation, priority, efficiency," *IEEE Access*, vol. 7, pp. 97130–97142, 2019.

[8] Y. Li, Y. Wang, Y. Jin, X. Cheng, L. Xu, and G. Liu, "Research on wireless resource management and scheduling for 5G network slice," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2021, pp. 508–513.

[9] F. Fossati, S. Moretti, P. Perny, and S. Secci, "Multi-resource allocation for network slicing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1311–1324, Jun. 2020.

[10] S. Wu, Y. Wei, S. Zhang, and W. Meng, "Proportional-fair resource allocation for user-centric networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1549–1561, Feb. 2022.

[11] H. Cao et al., "Toward tailored resource allocation of slices in 6G networks with softwarization and virtualization," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6623–6637, May 2022.

[12] O. Kodheli, N. Maturo, S. Chatzinotas, S. Andrenacci, and F. Zimmer, "NB-IoT via LEO satellites: An efficient resource allocation strategy for uplink data transmission," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5094–5107, Apr. 2022.

[13] Q. He, G. Dán, and G. P. Koudouridis, "Semi-persistent scheduling for 5G downlink based on short-term traffic prediction," in *Proc. GLOBECOM*, 2020, pp. 1–6.

[14] M. Zhang, Y. Guo, L. Salaün, C. W. Sung, and C. S. Chen, "Proportional fair scheduling for downlink mmWave multi-user MISO-NOMA systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6308–6321, Jun. 2022.

[15] B. Khurshid, S. Maqsood, M. Omair, B. Sarkar, I. Ahmad, and K. Muhammad, "An improved evolution strategy hybridization with simulated annealing for permutation flow shop scheduling problems," *IEEE Access*, vol. 9, pp. 94505–94522, 2021.

[16] Y. Zhou, W. Xu, Z.-H. Fu, and M. Zhou, "Multi-neighborhood simulated annealing-based iterated local search for colored traveling salesman problems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16072–16082, Sep. 2022.

[17] P. Liu, K. An, J. Lei, G. Zheng, Y. Sun, and W. Liu, "SCMA-based multiaccess edge computing in IoT systems: An energy-efficiency and latency tradeoff," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 4849–4862, Apr. 2022.

[18] R. Deng, B. Di, H. Zhang, L. Kuang, and L. Song, "Ultra-dense LEO satellite constellations: How many LEO satellites do we need?" *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4843–4857, Aug. 2021.

[19] P. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, A. Bandi, and B. Ottersten, "A RAN resource slicing mechanism for multiplexing of eMBB and URLLC services in OFDMA based 5G wireless networks," *IEEE Access*, vol. 8, pp. 45674–45688, 2020.

[20] S. Tabbane, "IoT network planning," Int. Telecommun. Union, Geneva, Switzerland, Rep. ST_V8_14122016, 2016. [Online]. Available: https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/SiteAssets/Pages/Events/2016/Dec-2016-IoT/IoTtraining/IoT network planning ST 15122016.pdf

[21] I. Maity, S. Misra, and C. Mandal, "ETHoS: Energy-aware traffic engineering for sustainable hybrid SDN," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 4, pp. 875–886, Oct.–Dec. 2022.

[22] M. Grant and S. Boyd. "CVX: MATLAB software for disciplined convex programming, version 2.1." cvxr.com, Mar. 2014. [Online]. Available: http://cvxr.com/cvx