INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. XX (20XX) 1–30
DOI: xx.xxxx/itor.xxxxx

Clustering nodes in Internet of Things environments with time and energy constraints: a multiobjective optimization formulation for the gateway placement problem in LoRaWAN®networks

Javier Marenco[a,*], Matias Micheletto[b], Rodrigo Santos[c]

[a]*Escuela de Negocios, Universidad Torcuato Di Tella, Av Figueroa Alcorta 7350 (1428) Buenos Aires, Argentina*
[b]*IIDEPyS GSJ, CONICET-UNPSJB (9000) Comodoro Rivadavia, Argentina*
[c]*DIEC UNS, ICIC CONICET-UNS (8000) Bahía Blanca, Argentina*
E-mail: javier.marenco@utdt.edu [J. Marenco]; matias.micheletto@uns.edu.ar [M. Micheletto]; ierms@criba.edu.ar [R. Santos]

Received DD MMMM YYYY; received in revised form DD MMMM YYYY; accepted DD MMMM YYYY

Abstract

Internet of Things (IoT) has become the leading technology to support smart-environments, with a growing set of applications being devised to provide intelligence in different scenarios. Precision agriculture, Industry 4.0, sustainable smart-cities, and smart-homes are just some of the more active areas in which the IoT is gaining space. The use of hierarchical networks in which the last mile is used to interconnect end-devices with gateways in charge of routing messages to and from the Internet is the most accepted network architecture. When messages have time constraints and nodes operate on batteries, the network organization should consider the cost of gateways, the total energy demand, the bandwidth utilization, and the satisfaction of time constraints. Planning the operation of such a network gives rise to a multi-objective optimization problem, which is considered in this work. This paper presents three main contributions: i) a proof of the NP-hardness of this problem; ii) an integer linear programming formulation for this problem; and iii) a custom heuristic that scales as the size of the instances grows.

Keywords: Internet of Things, multiobjective optimization, integer programming, heuristics

1. Introduction

During the last years, Internet of Things (IoT) has become the communication model for smart-environments, including smart-cities, smart-traffic, and smart-infrastructure, among others. IoT includes in its operation the deployment of sensors, actuators, and service provider devices that can be accessed from the Internet by users (human or applications). These devices are identified by an Internet Protocol

*Author to whom all correspondence should be addressed (e-mail: javier.marenco@utdt.edu).

(IP) address and/or by a semantic description of their properties and methods, or by a combination of both (see, e.g., Elsaleh et al. (2020); Santos et al. (2023)). Although the IoT paradigm is oriented to address each device individually, this is not always possible as many of these devices are small embedded systems with limited memory and processing power. Also, many of them operate on batteries and are deployed in places where it is difficult to replace them, thus imposing the need to manage the energy in a conscious way. This set of constraints prevents the deployment of the full Internet stack on these devices, e.g., to handle the TCP/IP protocol suite.

Instead, IoT proposes a simplified layered model. As Figure 1 shows, there are three layers: perception, networking, and application. In the first layer, the sensor/actuator devices interact with the environment. The networking layer links the sensors with the applications. In the third layer, different applications are developed using the services provided by the previous two layers. The network layer is usually implemented with two hierarchical levels. In the lowest level, the devices are grouped around a cluster head, router, or gateway. These devices are connected at the same time to the Internet through a wire, satellite link, or some other radio protocol.

The proposed organization is quite versatile as these technologies may be used in different scenarios. For example, in environmental monitoring where sensors are deployed in the field, usually far away from urban centers and with limited connectivity to Internet, Low Power Wide Area Networks (LPWANs) as LoRa and Sigfox are a good choice. Bluetooth is oriented to personal area networks, but it is also used in other scenarios like smart-homes. WiFi (IEEE 802.11) implementations are present in smart-buildings like offices, hospitals, and schools, among others. The most accepted technology for the deployment of IoT networks is LoRa, as it allows to connect devices on a point-to-point paradigm and also enables the construction of the whole perception layer connected through a gateway to the cloud using LoRaWAN®. We consider LoRaWAN® (to be defined precisely in Section **??**) in this work.

IoT applications have several non-functional requirements from the networking devices, like the delivery of messages in real time and energy-aware operation. In the first case, messages should be sent before a certain deadline (Santos et al., 2008). Real-time scheduling requires a deterministic approach, and several algorithms and policies have been proposed for this task. For example, in Micheletto et al. (2023) the authors propose the use of the so-called non-preemptive earliest deadline first (NP-EDF) policy (Park, 2007).

There are several optimization problems related to the implementation of IoT networks. A first issue to be considered involves the infrastructure deployment in order to collect/send information from/to sensors and actuators. This problem is known as the *gateway placement problem* (GPP). A second task is related to the energy demand of the end-devices (sensors and actuators). Minimizing the energy demand of the whole system is necessary in order to prolong its lifetime and to reduce maintenance costs. A third issue is given by the minimization of the electromagnetic spectrum use by the system. As many different applications are being instrumented in order to achieve smart-environments, it is important to leave unused as much bandwidth as possible. Finally, the scheduling of the transmission of messages over time must be considered. In this paper, a multi-objective combinatorial optimization problem tackling the first three issues is presented, while message scheduling is implemented with NP-EDF.

Contribution. To the best of the authors' knowledge, this paper makes three contributions to the state of the art. First, the *multi-objective gateway placement problem* (MGPP) is presented and proved to be NP-hard. Although in the literature several authors have stated this assertion for similar problems, we believe that this fact has not been formally proved so far. Second, an integer linear programming formulation

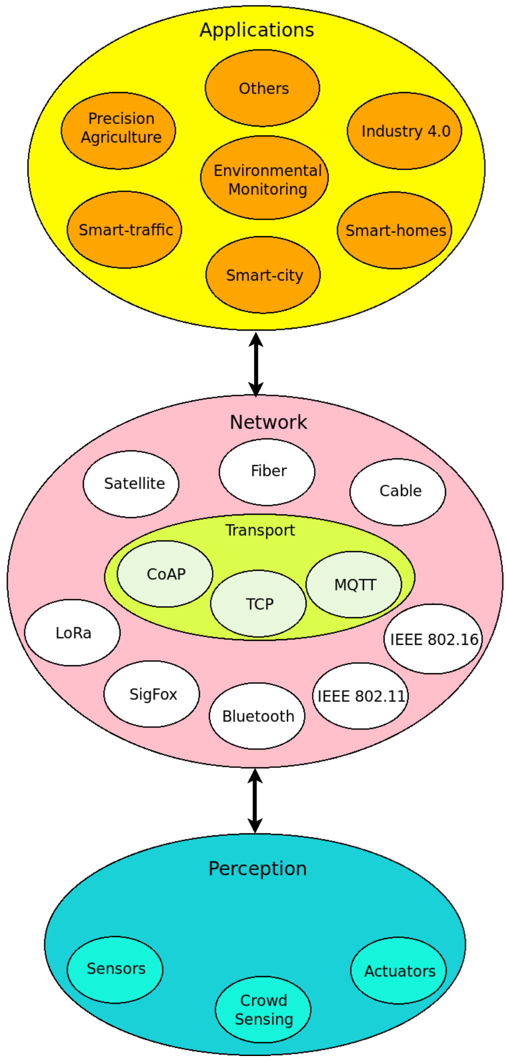

Fig. 1. IoT Layer model.

for this problem is presented, and several improvements to this formulation are identified, allowing to solve the formulation with smaller running times and with a smaller memory footprint. Finally, a greedy heuristic is proposed and compared to the reinforced model in a wide variety of instances that reflect different situations for IoT implementations. Although the results presented are oriented to LoRaWAN®, they can be extended to similar technologies.

Organization. The remainder of this paper is organized as follows. In Section 2 a brief description of LoRa and LoRaWAN® is provided. Section 3 discusses previous work on related problems. In Section 4 the multi-objective gateway placement problem considered in this work is presented and its computational complexity is identified. In Section 5 the integer linear programming formulation and its reinforcements are developed. In Section 6 we evaluate the performance of these formulations with a commercial solver. In Section 7 a greedy heuristic is proposed, and Section 8 reports computational experiments with synthetic instances. Finally, in Section 9 conclusions are drawn. In the Appendix, some guides to adapt the model to other technologies are provided.

2. LoRa and LoRaWAN®description

IoT networks are commonly built on two hierarchical levels. The lowest level is composed by *end-devices*, namely sensors and actuators, and the highest level is composed by *gateways*. The end-devices perform the intended tasks and interact with the gateways, which concentrate the information and upload/download it to a network server, usually located in the cloud. The gateways may implement some data protocols like MQTT or CoAP in order to handle the information flow (Finochietto et al., 2019). The amount of end-devices within an IoT application network may be large (up to several thousands) and with different deployment patterns like high density for industrial and urban or sparse density for precision agriculture and environmental monitoring.

LoRa stands for Long Range, it is a technology based on spread spectrum modulation derived from chirp spread spectrum (CSS) (Vangelista, 2017). It has been acquired by SemTech and became a "de facto" wireless platform for IoT. LoRa defines the physical layer interface within the traditional ISO/OSI model, and can be used in a point to point communication between two devices. In the link layer, SemTech proposes the so-called LoRAWAN® protocol. This protocol is an official standard of the International Telecommunication Union (ITU), ITU-T Y.4480 (ITU, 2021), and defines an architecture based on gateways and end-devices and the way in which they access the common channel. The protocol is supported by the LoRa-Alliance®, which is an open, nonprofit association that has become one of the largest and fastest-growing alliances in the technology sector since its inception in 2015 (SemTech, 2024). The protocol proposes a radio medium access control mechanism based on the Aloha (Abramson, 1970) protocol. As the used radio frequency falls within the unlicensed spectrum, a duty-cycle (DC) associated to each device, or *bandwidth limitation*, is enforced so the the electromagnetic spectrum is not saturated. This limit varies from region to region, and in this work the most used limit of 1% is adopted.

Within LoRaWAN®, end-devices transmit their messages to the gateways within transmission range at a specific *spreading factor* (from SF_7 to SF_{12}) and in one of the 16 available *channels*. By increasing the spreading factor by one unit, the transmission time is almost doubled and the transmission range is also duplicated. In this way, an end-device operating at SF_7 uses half the time of an end-device transmitting at SF_8, but the latter doubles the transmission distance of the former. By doubling the transmission time, the energy demand of that node is also doubled. Both the spreading factors and the channels are orthogonal, thus providing 96 different combinations for simultaneous message transmissions.

LoRaWAN® introduces a rather long overhead of 30 bytes in each message, which affects the real-time communication. When using SF_7 the MAC header will require 45 ms. Without sending payload bytes, the 1% duty-cycle imposes the minimum period to be 4.5 s. This limits the real-time operation of the protocol, as messages requiring shorter update periods are not feasible. The transmission time of a message is a function of the message length, the used spreading factor, and the radio configuration.

The protocol is not oriented to real-time messages as it is based on Aloha. In Micheletto et al. (2023), the authors proposed the use of non-preemptive earliest deadline first (NP-EDF) based on a time divi-

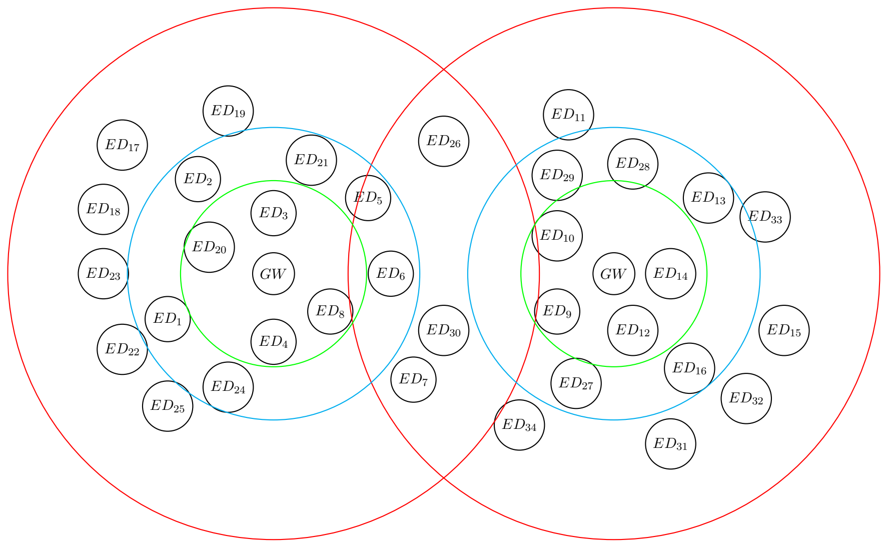

Fig. 2. An example deployment. The two nodes labelled "GW" are the gateways, and the remaining nodes are the end-devices.

sion multiple access mechanism. In this way, the network server with the information obtained through the gateways can provide a time schedule for the end-devices linked to a particular gateway using a predefined spreading factor and channel. This way, collisions of messages are avoided.

Figure 2 shows an example, in which 32 end-devices and two gateways are deployed. Around each gateway three circles with red, green, and cyan colors mark the transmission range associated to three different spreading factors. As can be seen, some of the end-devices may be allocated to any of the two gateways while the others can only use one of them. For example, ED_5, ED_6, ED_7, ED_{26}, and ED_{30} impact with different spreading factors on both gateways. These should be allocated to only one of them and for this the energy demand, span transmission time length, and real-time deadlines satisfaction should be considered. The process should decide the clustering or allocation of end-devices to the gateways minimizing the number of gateways, guaranteeing the deadlines compliance to all messages, and minimizing the energy consumption of the devices involved while interfering as less as possible with other possible IoT networking applications.

The gateways may listen simultaneously to the six spreading factors, and in each spreading factor the utilization factor of the gateway should be less than 1% in order to comply with the message deadlines. As NP-EDF is used, the utilization factor computation is made according to Guan et al. (2008):

$$U_k = \sum_i \frac{C_{ik}}{T_i - C_{ik}}$$

where U_k is the utilization factor at the spreading factor SF_k and the sum is performed over all the end-devices associated to that gateway at the specified spreading factor (Micheletto et al. (2023)).

We assume in this work that all the end-devices use the same transmission power, and that the transmission distance depends on the selected spreading factor and the characteristics of the area in which the

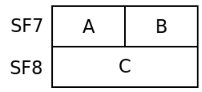

Fig. 3. Time-span optimization

network is deployed. In an open field, like in the case of precision agriculture applications or environmental monitoring, LoRa radios can reach up to ten kilometers using SF_{12}, but in urban environments this distance can be reduced to just two kilometers. Since transmission distances are approximately doubled when the spreading factor is incremented, then in an open field then the reach for SF_7 would be 300 meters and in urban areas just 60 meters. By using the same transmission power, there is more energy demand when the spreading factor is larger. As time is approximately doubled with each increment in the spreading factor, this means the energy demand is also doubled.

Finally, when considering the time restrictions, in order to obtain the smallest span time transmission window, messages should be transmitted in parallel using different spreading factors. For example two end-devices may use SF_7 for transmitting a message each one while a third end-device uses SF_8 for transmitting a third message. In this way three messages are transmitted in what would be the time needed to transmit just two.

This is shown in Figure 3 where nodes A, B and C transmit three messages using just the time needed to transmit two by taking advantage of the orthogonality of the spreading factors.

This discussion implies that there is clearly a trade-off between the time span used to transmit messages, the energy demanded from all the devices, and the location and amount of gateways to be deployed in the field in order to satisfy the delivery of messages in real-time. This leads to a nontrivial multi-objective combinatorial optimization problem when planning the deployment of such a network, and this observation is the key motivation for this work.

3. Related work

There are many papers in the literature related to the gateway placement problem described in this work, although none of them consider the minimization of gateways, energy, and time span at the same time. There are several studies about LoRa and LoRaWAN® and modifications proposed to the medium access control protocol and the management of bandwidth allocation. However, to the best of the authors knowledge, in none of these papers the optimization approach is performed with more than one objective and a real-time scheduling approach.

3.1. Mathematical programming formulations and computational complexity

In Lin (2013), the authors model the gateway placement problem and the association of end-devices to them in a mobile setting, thus generating a dynamic network configuration. The paper is not oriented to the LoRa technology and the authors do not explore the computational complexity of the studied problem. The problem has a single objective function, and the authors propose a particle swarm heuristic algorithm for this problem.

Similarly, in Capone et al. (2009), the authors analyze the assignment of end-devices to gateways but in this case when the network configuration is fixed as it represents a wireless sensor network. The computational complexity of the problem is mentioned but it is not formally proved. The technology used for the communication is not LoRa. Again, the studied problem has only one objective function (minimization of the number of gateways), and the authors propose a greedy heuristic for this problem.

In He et al. (2008) the authors reduce the gateway placement problem to the facilities location problem, although the approach is valid to show the complexity it does not formally prove it. In Luo et al. (2010) and Talay (2007), the authors also cast the gateway placement problem as a faciity location ptoblem and propose different heuristics. Like in the previous papers, the technology used is not LoRa and the minimization procedure only involves one objective function.

3.2. LoRaWAN® technology gateway and time scheduling analysis

In Shanmuga Sundaram et al. (2020) the main challenges for the LoRaWAN® technology in the next years are presented and, among them, energy consumption, resource allocation, and link coordination are identified as the most important ones.

In Finochietto et al. (2023) the authors propose a simple scheduling algorithm to guarantee that the messages' deadlines are met. The algorithm uses for all messages the same period and spreading factor. This paper does not provide insights into the end-device/gateway allocation problem nor on the energy or bandwidth minimization aspects.

In Micheletto et al. (2023), the authors model the gateway placement problem for LoRa networks and use some heuristics to solve it. The optimization is mono-objective and the placement of the gateways is not considered to be part of the problem to be solved. The result obtained in the optimization process may not be applicable due to physical location issues, among other considerations. Being mono-objective, neither the energy nor the time span are considered in the process.

In Marenco et al. (2023) the authors present a work in progress paper proposing the main ideas that are developed in this paper. An integer linear programming model with three terms related to the three objectives is presented. In the current work the integer linear programming model is modified with the introduction of reinforcing constraints that provide an important advantage over the original model, besides a formal proof of the computational complexity of the problem and a heuristic for this problem are presented.

4. The multi-objective gateway placement problem

In this section the multi-objective optimization for the gateway placement problem is presented and its computational complexity is studied.

4.1. Assumptions and definitions

We define $\mathbf{E} = \{1, \dots, n\}$ to be the set of the available end-devices, and $\mathbf{G} = \{1, \dots, m\}$ to be the set of the available gateways. The transmission range of the devices is governed by the used spreading factor, so $\mathbf{R} = \{7, \dots, 12\}$ represents the set of available spreading factors, from SF_7 to SF_{12}. There are 16 possible channels within LoRaWAN® to be used by gateways and end-devices, and we define $\mathbf{Ch} = \{0, \dots, 15\}$ to be the set of available channels.

In this work the following assumptions, which are similar to those presented in Micheletto et al. (2023), are performed.

- Time is considered to be discrete and the time unit is the so-called *slot*. Events are synchronized with the beginning of the slots.
- When changing from SF_k to SF_{k+1}, the number of slots required to transmit the message is duplicated, for $k = 7, \dots, 11$.
- The transmission range is doubled when the spreading factor is incremented by one unit, i.e., the transmission range for SF_{k+1} is twice the transmission range for SF_k, for $k = 7, \dots, 11$.
- For $i = 1, \dots, n$, the end-device i transmits a sequence of periodic messages characterized by a *message length* $Z_i \in \mathbb{Z}_+$ in bytes, a *period* $T_i \in \mathbb{Z}_+$, and a *deadline* $D_i \in \mathbb{Z}_+$. We assume $T_i = D_i$ and $Z_i = Z$ for every end-device $i = 1, \dots, n$. All time units are expressed in slots.
- The time required to transmit one message with spreading factor SF_7 is one time unit, i.e., one slot. This implies that we need $C_k := 2^{k-7}$ slots in order to transmit a message with spreading factor SF_k, for $k = 7, \dots, 12$.
- All end-devices use the same transmission power.
- Non-preemptive scheduling is adopted transmissions by the end-devices.

The end-device i must send a message every T_i slots, for $i = 1, \dots, n$. All messages have the same length Z in bytes, but the transmission time depends on the used spreading factor. The *utilization factor* that the end-device i places on the associated gateway when connected at spreading factor SF_k is given by

$$u_{ik} = \frac{2^{k-7}}{T_i - 2^{k-7}}. \tag{1}$$

As all messages transmitted to a gateway should be received on time, each gateway can schedule up to a maximum capacity for each spreading factor equal to 1%, in compliance with existing conventions. When an end-device is listened by more than one gateway in a certain spreading factor, its transmission impacts on all of them. If the capacity of the gateway is exceeded (i.e., it is over 1%), then a different gateway should be used in a different channel to avoid interference.

Time scheduling is left to the NP-EDF policy, which consists in ordering the transmission of messages according to the relative deadline of each message. When an end-device finishes its transmission, the next one to use the channel is selected based on the time left until the deadline. The end-device with the earliest deadline is scheduled first. The algorithm is conservative, hence no empty slots are left if there are messages ready to be transmitted. While this is useful, it may produce a priority inversion. Considering this fact, the bandwidth requirement or utilization factor of the end-devices is incremented by reducing the period of the messages by the maximum transmission time, as stated in (1).

We define $\bar{s} = \text{lcm}\{T_1, \ldots, T_n\}$ (i.e., the least common multiple of the periods of all end-devices) to be the so-called *hyper-period* of the system, after which all messages repeat in the same order.

With all the previous definitions and assumptions, we can now define the multi-objective combinatorial optimization problem that we are interested in tackling, and which takes into account all the stated considerations.

Definition 1 (multi-objective gateway placement problem, MGPP). We are given the set $\mathbf{E} = \{1, \ldots, n\}$ of end-devices, the set $\mathbf{G} = \{1, \ldots, m\}$ of gateways, the set $\mathbf{R} = \{7, \ldots, 12\}$ of spreading factors, and the set $\mathbf{Ch} = \{0, \ldots, 15\}$ of channels. For $i = 1, \ldots, n$ and $j = 1 \ldots, m$, the parameter $\min_{ij} \in \mathbf{R} \cup \{\infty\}$ specifies the minimum spreading factor allowing the end-device i to contact the gateway j, or ∞ if such a transmission is not possible for any spreading factor. The *multi-objective gateway placement problem* consists in determining

(a) a set $D \subseteq \mathbf{G}$ of gateways to be deployed,
(b) a function $g : \mathbf{E} \to D$ assigning a deployed gateway to each end-device,
(c) a function $c : D \to \mathbf{Ch}$ assigning a channel to each deployed gateway, and
(d) a function $s : \mathbf{E} \to \mathbf{R}$ assigning a spreading factor to each end-device,

in such a way that

(i) for $i = 1, \ldots, n$, the gateway assigned to the end-device i is located within transmission range of the assigned spreading factor, i.e., $s(i) \geq \min_{i,g(i)}$,
(ii) no two gateways use the same channel if there exists an end-device within transmission range for both gateways, and
(iii) the sum of the utilization factors of all the end-devices assigned to each gateway does not exceed 1%, i.e., $\sum_{i \in g^{-1}(j)} u_{i,s(i)} \leq 1$ for $j = 1, \ldots, m$.

The following three objective functions for this multi-objective problem are considered:

1. Minimizing the number $|D|$ of deployed gateways.
2. Minimizing the total power for all transmissions (i.e., $\sum_{i=1}^{n} 2^{s(i)}$).
3. Minimizing the total transmission time (given by $\max_{j \in \mathbf{G},\, k \in \mathbf{R}} \sum_{i \in g^{-1}(j) : s(i) = k} u_{i,s(i)}$).

This definition involves the simultaneous consideration of three objective functions. In practice, we may optimize a weighted sum of these objectives or we may build the Pareto front for these objectives for a particular instance. In any case, the obtained solutions must be evaluated by an expert in the technology, either in order to better calibrate the weight assigned to each objective, or to evaluate the preferred solutions within the Pareto front.

4.2. *Computational complexity*

We explore in this section the computational complexity of MGPP, by showing that minimizing the number of deployed gateways (i.e., the first objective function) is an NP-hard problem.

Theorem 1. *The multi-objective gateway placement problem minimizing the number of deployed gateways is NP-hard.*

Proof. It is straightforward to check that the decision version of MGPP belongs to NP, since we can non-deterministically construct a solution and check in polynomial time whether the number of deployed gateways falls below the given threshold or not. In order to complete the proof, we provide a reduction from a reduced version of the *capacitated facility location problem* (CFLP, see Silva and De la Figuera (2007)), defined to be the restriction of CFLP given by the following constraints:

- association costs from clients to facilities are null, i.e., $c_{ij} = 0$ for every $i \in I$ and every $j \in J$,
- opening costs for the facilities are constant and equal, i.e., $f_j = 1$ for every $j \in J$, and
- all facilities have the same capacity.

This problem is called the *reduced CFLP* (RCFLP). We first prove that the decision version of RCFLP is NP-complete and, in a second step, we show that this problem can be reduced to the decision version of MGPP.

The decision version of RCFLP clearly belongs to NP, and we now provide a reduction from the *set partitioning problem* (SPP), which is NP-complete, to this problem. Given an instance $\mathbf{A} = \{A_1, \ldots, A_m\}$ of SPP, an instance of RCFLP is constructed as follows. We set $\mathbf{I} := \{1, \ldots, m\}$ to be the set of customers, and we take $\mathbf{J} := \{1, 2, \ldots, n\}$ to be the set of facilities, with n large enough (e.g., $n = m$). We also set $f_j := 1$ for all $j \in J$ and the demand of the customer i is defined to be $a_i := A_i$ for $i \in \mathbf{I}$. Finally, we take $b_j := \frac{1}{2}\sum_{i=1}^{m} A_i$. This instance of CFLP has optimal value equal to 2 (namely, exactly two facilities are opened) if and only if A can be partitioned into two subsets that sum up the same number. Indeed, any assignment of customers to k facilities corresponds to a partition of $\mathbf{A}$ into k sets. Since the capacity of each facility is set to $\frac{1}{2}\sum_{i=1}^{m} A_i$, then opening exactly two facilities corresponds to partitioning $\mathbf{A}$ into two sets with equal sum.

We now reduce the decision version of RCFLP to the decision version of MGPP minimizing the number of deployed gateways. An instance of RCFLP is given by a set $\mathbf{I}$ of customers, a set $\mathbf{J}$ of potential facilities, a capacity b for each facility, a demand a_i for each customer $i \in \mathbf{I}$, and a bound $\varphi \in \mathbb{R}$. The problem consists in determining whether there is a feasible solution with objective value φ or less. Given such an instance, we construct an instance of the decision version of MGPP as follows. We take $\mathbf{E} := \mathbf{I}$ as the set of end-devices, i.e., equal to the set of customers in RCFLP. Similarly, we take $\mathbf{G} := \mathbf{J}$ to be the set of gateways, i.e., equal to the set of facilities. We take $\mathbf{Ch} = \{1\}$ (i.e., only one channel) and $\mathbf{R} = \{SF_7\}$ (i.e., only one spreading factor), and we assume that all end-devices are within transmission distance to the possible gateways. We set $T_i := 1/(a_i b_i) + 1/b_i$ for each end-device $i \in \mathbf{E}$, so $u_{ik} = a_i/b$ for the single spreading factor $k \in \mathbf{R}$. In this setting, there exists a feasible solution of RCFLP with objective function φ or less if and only if MGPP admits a feasible solution with objective function φ or less. This assertion stems from the fact that the constructed instance of MGPP is just an alternative representation of the instance of RCFLP, given that every end-device can be assigned to any gateway.

This shows that the decision version of MGPP is NP-complete, hence MGPP is NP-hard. □

5. An integer linear programming formulation

We introduce in this section an integer linear programming (ILP) formulation for the multi-objective gateway placement problem, combining the three objectives in the objective function with appropriate penalties. End-devices are assigned to gateways with a certain spreading factor and channel, and the transmissions are ordered according to the NP-EDF policy.

We say that a gateway and an end-device *listen to each other* if they operate in the same channel and are within the transmission range of the assigned spreading factor. For each $j \in \mathbf{G}$ and each $c \in \mathbf{Ch}$, we introduce the binary variable w_{jc} in such a way that $w_{jc} = 1$ if the gateway j is deployed using channel c. For each $i \in \mathbf{E}$, each $j \in \mathbf{G}$, and each $k \in \mathbf{R}$, we introduce the binary variable x_{ijk} in such a way that $x_{ijk} = 1$ when the end-device i and the gateway j listen to each other using the spreading factor SF_k. Finally, the continuous variable z represents an upper bound to the total transmission time. In this setting, we can provide the following ILP formulation for MGPP.

$$\min \ \alpha \sum_{j\in\mathbf{G}} \sum_{c\in\mathbf{Ch}} w_{jc} + \beta \sum_{i\in\mathbf{E}} \sum_{j\in\mathbf{G}} \sum_{k\in\mathbf{R}} P\ 2^{k-7}\ x_{ijk} + \gamma\ z \tag{2}$$

$$\begin{aligned}
\sum_{c\in\mathbf{Ch}} w_{jc} &\leq 1 && \forall j \in \mathbf{G} && (3)\\
\sum_{j\in\mathbf{G}} \sum_{k\in\mathbf{R}} x_{ijk} &= 1 && \forall i \in \mathbf{E} && (4)\\
x_{ijk} &\leq \sum_{c\in\mathbf{Ch}} w_{jc} && \forall i \in \mathbf{E},\ \forall j \in \mathbf{G},\ \forall k \in \mathbf{R} && (5)\\
\sum_{i\in\mathbf{E}} u_{ik}\ x_{ijk} &\leq 1 + M(1 - \sum_{c\in\mathbf{Ch}} w_{jc}) && \forall j \in \mathbf{G},\ \forall k \in \mathbf{R} && (6)\\
w_{jc} + w_{j'c} &\leq 2 - \sum_{k\in\mathbf{R}_{ijj'}} (x_{ijk} + x_{ij'k}) && \forall i \in \mathbf{E},\ \forall j, j' \in \mathbf{G}, j \neq j',\ \forall c \in \mathbf{Ch} && (7)\\
x_{ijk} &= 0 && \forall i \in \mathbf{E},\ \forall j \in \mathbf{G},\ \forall k \in \mathbf{R} \backslash \mathbf{R}_{ij} && (8)\\
z &\geq \sum_{i\in\mathbf{E}} u_{ik}\ x_{ijk} && \forall j \in \mathbf{G},\ \forall k \in \mathbf{R} && (9)\\
w_{jc} &\in \{0, 1\} && \forall j \in \mathbf{G},\ \forall c \in \mathbf{Ch} && (10)\\
x_{ijk} &\in \{0, 1\} && \forall i \in \mathbf{E},\ \forall j \in \mathbf{G},\ \forall k \in \mathbf{R} && (11)
\end{aligned}$$

The objective function (2) contains three terms, one for each objective. The first term represents the amount of gateways, the second term represents the energy consumed by the end-devices, and the third term represents the transmission time or busy period of gateways. Each term is weighted with real non-negative parameters α, β, and γ, respectively. Constraint (3) indicates that each gateway should use just one channel and in the model an inactive gateway is not assigned to any channel. Constraint (4) indicates that each end-device is listened by exactly one gateway in a given spreading factor. Constraint (5), links the previous variables, allocating an end-device to a gateway only if the given gateway is active. Constraint (6) limits the utilization factor for each active gateway in the corresponding spreading factor, forcing that these do not exceed 1 in each spreading factor. Constraints (7) specify that two end-devices should not use the same channel if they are connected to gateways that are within transmission range.

For these constraints we define $\mathbf{R}_{ijj'} := \{k \in \mathbf{R} : k \geq \min_{ij}$ and $k \geq \min_{ij'}\}$ to be the spreading factor allowing i to listen both to j and j'. Constraints (8) limit the spreading factor with which each end-device is linked to its gateway, and for these constraints we define $\mathbf{R}_{ij} := \{k \in \mathbf{R} : k \geq \min_{ij}$ and $P_i \geq 2^{k-7}\}$ to be the set of spreading factors with which the end-device i can communicate with gateway j. Finally, constraints (9) link the variable z with the assignment variables x; and constraints (10)-(11) specify the variable domains.

We can replace some of the constraints in the formulation (2)-(11) by stronger inequalities (i.e., constraints generating a smaller linear relaxation). Constraints (5) can be replaced by the stronger inequalities

$$\sum_{k \in \mathbf{R}} x_{ijk} \leq w_{jc} \qquad \forall i \in \mathbf{E},\ \forall c \in \mathbf{C}, \tag{12}$$

since each end-device is assigned to a gateway with exactly one spreading factor. The capacity constraints (6) can be simply stated as

$$\sum_{i \in \mathbf{E}} u_{ik}\, x_{ijk} \leq 1 \qquad \forall j \in \mathbf{G},\ \forall k \in \mathbf{R}, \tag{13}$$

given that $\sum_{c \in \mathbf{Ch}} w_{jc} = 1$ if the gateway j is used, for $j \in \mathbf{G}$. Finally, constraints (7) can be replaced by

$$\sum_{j \in \mathbf{G} : k \in \mathbf{R}_{ij}} w_{jc} \leq v_{ik} - (v_{ik} - 1) \sum_{j \in \mathbf{G}} x_{ijk} \qquad \forall i \in \mathbf{E},\ \forall k \in \mathbf{R},\ \forall c \in \mathbf{Ch}. \tag{14}$$

In this expression, $v_{ik} := |\{j \in \mathbf{G} : k \in \mathbf{R}_{ij}\}|$ is the number of gateways visible from the end-device i with spreading factor SF_k. This alternative constraint expresses the fact that no additional gateway can use the channel c if the spreading factor SF_k allows the end-device i to reach the gateway and i is using the channel c, for $i \in \mathbf{E}$, $k \in \mathbf{R}$, and $c \in \mathbf{Ch}$. The formulation (2)-(5), (12)-(14), and (8)-(11), is called the *reinforced model*.

6. Evaluating the integer linear programming formulations

The models proposed in Section 5 were coded in the open source ZIMPL modeling language (Koch, 2004) and solved using CPLEX 12.9 on a server with 2 x Intel® Xeon® Silver 4210R CPU (10 cores/20 threads) @ 2.40 GHz, 768 RAM DDR4-2400 ECC LRDIMM, using a virtual machine with 10 cores, 512 GB of RAM, and the Debian 11 Linux/GNU distribution.

The objective function (2) has three terms, representing the number of gateways, energy demand in the end-devices, and the channel occupancy, respectively. These objectives are conflicting among each other and, in general, minimizing one of them may generate a sub-optimal value for the other two objectives. Properly tuning the parameters α, β, and γ is important, as they can equalize the way in which each term contributes to the objective function. A system designer may decide to benefit one term or the other based on the requirements. For example, energy may not be an issue in small networks where end-devices are connected to the power utility, or it may be the case that gateways are already available and can be used with no additional cost. Similar considerations can be provided for the time span.

The behavior of the exact method was evaluated for different instances with CPLEX. The memory footprint for the ILP model computed with the original and the reinforced models are shown in Table 1. The table shows the case of 100 and 500 end-devices deployed in an area of $100[\mathrm{m}] \times 100[\mathrm{m}]$.

		Memory footprint (MB)		Solving time (sec.)	
$\lvert\mathbf{E}\rvert$	$\lvert\mathbf{G}\rvert$	Original	Reinforced	Original	Reinforced
50	30	79.2	6.9	72.38	1.81
100	30	161.0	13.9	212.51	9.25
150	30	245.5	21.0	407.30	13.53
200	30	326.7	28.1	415.26	10.86
500	30	836.00	71.0	>3600.00	76.24

Table 1
Memory footprint and running time to solve different instances to optimality with the original model and the reinforced model.

7. A heuristic for the MGPP

In this section we present a heuristic for the MGPP, based on an approach similar to the coverage in state machines minimization in digital circuits design.

A gateway $j \in \mathbf{G}$ is said to be *essential* if there is at least one end-device that can only connect to the gateway j. Similarly, if an end-device can only connect to a single gateway, we say that the end-device is *essential*. The first step in the heuristic is to determine if there are essential gateways and in that case to allocate the subset of essential end-devices to the corresponding essential gateways. After this step, the remaining end-devices are allocated to gateways, considering first the already used gateways (initially, the essential gateways) and randomly selecting the other gateways in case the essential gateways are not within the transmission range. In any case, if an end-device can be allocated to more than one gateway included in the solution, the one with the lowest spreading factor is used subject to the restrictions of duty-cycle and utilization factor for each spreading factor. When a solution improves the objective function value it is taken as the new best one replacing the old one. The procedure is repeated a predefined number of times.

Algorithm 1 presents the proposed heuristic. The first call is to search for the essential gateways and it is presented in Algorithm 2. Once a gateway is found to be essential, it is stored in the list of necessary gateways for the system to be feasible. Essential end-devices are allocated with the lowest possible spreading factor compatible with the period and the duty-cycle restriction to the (only) associated gateway. In Algorithm 3 the allocation of end-devices that can be connected to more than one gateway is performed. The non-essential gateways are randomly ordered and the list of end-devices not yet allocated is traversed. The end-device is allocated to the first gateway to which it can connect with the lowest possible spreading factor. When all the end-devices has been allocated to objective function is computed. In case it is lower than the previous solution, it is stored. The process is repeated max_iter times or if there is no decrement or the time limit is reached. Finally, in Algorithm 4 a reallocation procedure is

ED_i	A	B	C	D	T
1	7	8	9	10	1600
2	8	7	7	10	1600
3	8	9	7	11	1600
4	10	8	10	9	1600
5	7	10	7	8	1600
6	9	10	10	10	1600
7	8	9	8	9	1600
8	10	7	10	10	1600
9	11	9	9	10	1600

Table 2
An instance with nine end-devices and four non-essential gateways.

performed. Basically it is evaluated the possibility of moving an end-device from one gateway to another if by doing so, the spreading factor is reduced or eventually a gateway can be emptied. The idea behind this is to reduce as much as possible the energy demand to the end-devices. This heuristic has a computational complexity of $O(n \times m \times max\_iter)$.

Within the heuristic, the instance is represented by a matrix with end-devices in the rows and gateways in the columns. The last column contains the periods of the messages associated to each end-device. Each cell has a number that represents the lowest spreading factor with which the end-device can reach the gateway. In case they are not within transmission range for any spreading-factor or the time needed violates the duty-cycle constraint (1%), the cell takes a value of 100.

Example. Consider a system with nine end-devices and four possible gateways as described in Table 2. As can be seen there are not essential gateways. It is assumed that the message transmission length for SF_7 is one time unit and for SF_{12} is thirty two. As all the periods are equal to 1600 slots, the highest spreading factor that can be used is 11. In Table 3 each row represents the maximum spreading factor used by the end-devices for the allocation to gateways. In the case of SF_7 the end-devices 4, 6 and 9 can not be allocated to any gateway and in the case of SF_8 the end-devices 6 and 9 can not be allocated either. With SF_9 there is full coverage but it is needed to use at least two gateways, A and B. There are other possibilities using more gateways but for simplicity it is just listed one. In the case of SF_{10} there is full coverage with just one gateway, B or C, or by combining two or more. Finally, with SF_{11} all the end-devices can be allocated to any gateway. The heuristic shuffles with a uniform random function the list of gateways and allocates the end-devices. The best solution after several iterations is chosen.

8. Experimental evaluation

The proposed heuristic was evaluated with different instances and compared to the exact solution provided by the reinforced model solved with CPLEX in the ICIC server when it was possible (see Section 6 for its characteristics). In what follows, the performed experiments are explained and the results are analised.

SF	Partition	Coverage	Chosen
7	C={2,3,5}, A={1,5}, B={2,8}	NO	
8	A={1,2,3,5,7}, B={1,2,4,8}, C={2,3,5,7},D={5},	NO	
9	A={1,2,3,5,6,7}, B={1,2,3,4,7,8,9}, C={1,2,3,5,9}, D={4,5,7}	YES	A={1,3,5,6,7}, B={2,4,8,9}
10	A={1,2,3,4,5,6,7,8}, B={1,2,3,4,5,6,7,8,9}, C={1,2,3,4,5,6,7,8,9}, D={1,2,4,5,6,7,8,9}	YES	B={1,2,3,4,5,6,7,8,9}
11	A={1,2,3,4,5,6,7,8,9}, B={1,2,3,4,5,6,7,8,9}, C={1,2,3,4,5,6,7,8,9}, D={1,2,3,4,5,6,7,8,9}	YES	C={1,2,3,4,5,6,7,8,9}

Table 3
Coverage computation

Algorithm 1: Main heuristic procedure

Input: Optimization parameters, network model
Output: A feasible solution

```
1 begin_t ← now() ;                      // Initialize timer
2 allocateEssentialNodes() ;             // Allocate mandatory links (essential nodes)
3 allocateNonEssentialNodes() ;          // Main allocation stage
4 improveByNodeReallocation() ;          // Move nodes between gateways lowering SF
5 evalSolution() ;                       // Compute objectives values of resulting allocation
6 finish_t ← now() − begin_t ;           // Compute execution time
```

8.1. Instance benchmark

Three kind of instances were generated in order to represent different situations that may arise in the IoT realm. The first set of instances is composed by a set of end-devices and gateways deployed in a reduced square area with a side length of 100 m (hence the maximum distance between opposite corners is 142 m). Considering the distances that are reachable with the LoRa technology, all the end-devices are able to reach at least a gateway within the region. The second set of instances was generated for the same map but shortening the transmission distance for the different spreading factors (thus generating essential gateways). The third set of instances was generated for large amounts of end-devices deployed in bigger square areas, with side lengths of 500 m and 1000 m. For these maps, only a few instances

Algorithm 2: allocateEssentialNodes()

```
Input: Network model
Output: Allocation of mandatory links
init(allocation) ;                                  // Initialize allocation's data structure
essGWSet ← [] ;                                     // List of essential gateways
nEssGWSet ← [] ;                                    // List of non-essential gateways
nEssED ← [] ;                                       // List of non-essential end-devices
for i = 1, . . . , n do
    gws-of-ed ← getReachableGateways(i) ;           // Get all gateways that can be connected to the end-device i
    if gws-of-e.size() == 1 then
        essGWSet.insert(gws-of-ed[0]) ;             // Add essential gateway to list
        allocation.connect(i, gws-of-ed[0], minSF) ; // Allocate essential link
    else
        nEssED.insert(i)
for j = 1, . . . , m do
    if essGWSet.contains(j) == false then
        nEssGWSet.insert(j) ;                       // Save non essential gateways in another list
```

were evaluated with the exact method as the other ones demanded more than 512GB of RAM and the Linux kernel killed the CPLEX process due to memory issues.

For each set of instances, three types of time requirements were generated: hard, medium, and soft. In the first group, the period of the messages associated to the end-devices were short preventing the use of all the spreading factors. For the medium and soft cases the periods were larger so it was possible to use all the available spreading factors.

End-devices and gateways were deployed within the map by two different random distributions. In the first one a *uniform distribution* was chosen in order to cover the whole map. In the second one a so-called *clouds distribution* was used. The clouds distribution deployed the end-devices and gateways around different points generating an area with a high density of nodes while other parts of the map have just a few nodes. Both cases have interest for the IoT paradigm. The first one is associated with the deployment of sensors in open areas like national parks, or precision agriculture. The second one is associated with urban environments where nodes are gathered in buildings or specific areas.

The evaluation was carried out on 24 different sets of instances. For each one, solutions were calculated for a wide variety of number of devices, and in each case the average of five cases was computed. In total 1170 runs were made. The results are presented in Tables 4 to 27.

Algorithm 3: allocateNonEssentialNodes()

```
Input: iterMax, timeout
Output: GW number, positions and ED allocation
minCost ← ∞ ;                                        // Best cost, initially infinite
for sf = 7, ..., 12 do
    coverage = getCoverage(sf) ;                     // Compute coverage of current SF
    if coverage == 100% then
        while iter < iterMax and time < timeOut do
            shuffle(essGWSet) ;       // If non empty, shuffle list of essential gateways
            shuffle(nEssGWSet);                  // Shuffle list of non essential gateways
            gwList ← concat(essGWSet, nEssGWSet) ;      // Concatenate list of gateways
            for i = 1, ..., n do
                for j = 1, ..., m do
                    gw = gwList[j] ;                           // Current gateway index
                    connected ← allocation.connect(i, gw, sf);  // Try to connect node to gateway
                      with minimum possible SF, from 7 up to sf
                    if connected then
                        break ;                           // Continue with following node
            if allocation.connectedCount == n then
                cost = evalAllocation(allocation) ;          // Compute allocation cost
                if cost < minCost then
                    minCost ← cost ;                                // Update best cost
```

8.2. *Experimentation plan*

Following the combinations mentioned above, solutions were computed with both the exact method (CPLEX) and the heuristic proposed here. Not all the instances could be solved by the two approaches as the size of the exact model overpassed the 512 GB RAM memory available in the server and the processes were killed by the operating system or the time demanded by the solver was over 5400 seconds which was taken as time-out limit.

8.2.1. *Non-essential gateways*

In these instances all end-devices could connect to more than one gateway so there were no essential gateways. Six different cases with 20, 50, 100, 150, 200, and 500 end-devices were evaluated. When the deployment was made using the uniform distribution, in all cases, even when using the hard time requirements just one gateway was enough to find a solution. As can be seen in Tables 4-9. Both methods provided similar results with the greedy heuristic being in all cases within reasonable distance of the optimal solution. It is important to notice that the time demanded by the heuristic was two or three orders of magnitude smaller than the running time to optimality used by CPLEX. When the end-devices

Algorithm 4: improveByNodeReallocation()

```
Input: Second stage allocation
Output: Improved allocation
sortGWsByCoverage() ;                    // Sort gateways in ascending order of nodes connected
for j = 1, ..., m do
    for i = 1, ..., n do
        gwsOfEd ← getGatewaysInRange(i) ;          // List of gateways reachable by node i
        for j' ∈ gwsOfEd do
            if allocation.has(j') and j' ≠ j then
                allocation.move(i, j') ;    // Move node if j' has available frequency use and the
                  connection has lower spreading factor [estos chequeos se hacen en la función
                  ``has'' del ``if''? si no, habría que poner estos chequeos explicitamente en el
                  pseudocodigo]
```

were deployed following the clouds distribution, the solutions obtained with the heuristic were again close to the optimal solution, with a gap below 10%.

8.2.2. Essential gateways

For the case of the smallest map the transmission distance for each spreading factor was shortened significantly by about one order of magnitude to generate the essentials condition, thus allowing a comparison of the heuristic and the exact method. Tables 10-15. As can be seen, the objective function value is quite close to the optimal one while there are differences in the energy and utilization factors terms.The amount of gateways is similar and this is probably due to the fact that the heuristic search is based mainly in reducing the number of gateways. Like before, both uniform and clouds distribution have similar results and although the CPLEX solved the problems in less time it is still at least two orders of magnitude higher than the heuristic-

8.2.3. Larger map areas

When the area in which the end-devices and gateways was enlarged to 500 m and 1000 m, the amount of end-devices that should be deployed to generate a problem with enough complexity increased considerably and almost none of the systems could be solved with the exact method. In these maps, it was not necessary to reduce the transmission distance as essential gateways appear naturally. In Tables 16-27, the results for the larger maps instances are presented, showing that the behavior is in line with the solutions found when the transmission distance was reduced in the map 100x100m case.

8.3. Discussion

In the first group of instances, small maps 100x100m with end-devices using the normal transmission range for the different spreading factors, the greedy heuristic and the exact method produced almost the

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	3.29	1.0	24.6	0.028	0.003	3.196	1.0	24.0	0.026	2.2
50	3.29	1.0	62.8	0.063	0.012	3.174	1.0	60.4	0.063	16.0
100	3.34	1.0	130.6	0.133	0.024	3.236	1.0	122.2	0.127	53.5
150	3.29	1.0	197.4	0.187	0.038	3.188	1.0	182.8	0.185	235.6
200	3.35	1.0	258.4	0.273	0.051	3.236	1.0	243.8	0.261	911.1
500	3.32	1.0	661.0	0.641	0.192	1.698	19.0	1242.2	0.1	79.7

Table 4
Uniform distribution, 100x100 map, hard real time constraints without essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	2.30	1.0	20.8	0.007	0.003	2.27	1.0	20.4	0.007	0.78
50	2.31	1.0	51.6	0.018	0.013	2.31	1.0	51.4	0.018	3.83
100	2.34	1.0	107.6	0.034	0.025	2.33	1.0	106.8	0.034	25.53
150	2.33	1.0	161.0	0.049	0.040	2.33	1.0	160.8	0.049	38.66
200	2.32	1.0	210.0	0.071	0.055	2.31	1.0	206.8	0.072	70.83
500	2.32	1.0	519.2	0.183	0.199	2.32	1.0	517.2	0.183	641.50

Table 5
Uniform distribution, 100x100 map, medium real time constraints without essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	2.18	1.0	20.6	0.004	0.003	2.18	1.0	20.6	0.004	0.63
50	2.17	1.0	51.2	0.009	0.012	2.17	1.0	51.2	0.009	3.20
100	2.19	1.0	105.4	0.018	0.025	2.18	1.0	104.0	0.020	15.90
150	2.19	1.0	157.0	0.027	0.040	2.18	1.0	156.0	0.027	38.63
200	2.17	1.0	206.0	0.036	0.053	2.17	1.0	205.2	0.036	40.54
500	2.15	1.0	500.0	0.094	0.193	2.15	1.0	500.0	0.094	1175.99

Table 6
Uniform distribution, 100x100 map, soft real time constraints without essential gateways

same results in all the objectives and in the general objective function. This is due to the fact that the problem is reduced to find the best gateway, as all the end-devices are within transmission distance to any possible gateway deployed in the area. The way in which the heuristic searches the space of solutions is in this scenario very fast. As the time requirements are relaxed the heuristic solution is closer to the exact solution for all the cases.

In the second group of instances, small maps 100x100m but with restricted transmission ranges as to generate essential gateaways, the search space is more complex and the heuristic solutions are not as close to the exact one as in the previous case. While the number of gateways necessary to allocate and schedule all the messages is similar there are important differences in the energy and utilization

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	3.45	1.0	23.2	0.033	0.003	3.27	1.0	24.2	0.027	1.35
50	3.36	1.0	56.0	0.079	0.013	3.16	1.0	60.0	0.061	9.56
100	3.37	1.0	116.6	0.154	0.025	3.18	1.0	119.8	0.125	35.71
150	3.39	1.0	178.0	0.229	0.040	3.21	1.0	180.6	0.193	240.64
200	3.46	1.0	215.2	0.356	0.051	3.21	1.0	241.0	0.259	350.17
500	3.39	1.0	603.8	0.758	0.193	1.41	12.6	1041.6	0.157	5411.60

Table 7
Cloud distribution, 100x100 map, hard real time constraints without essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	2.28	1.0	20.0	0.007	0.003	2.28	1.0	20.0	0.007	1.18
50	2.31	1.0	50.4	0.019	0.013	2.31	1.0	50.4	0.019	8.05
100	2.29	1.0	100.0	0.037	0.026	2.29	1.0	100.0	0.037	38.82
150	2.29	1.0	150.0	0.055	0.042	2.29	1.0	150.0	0.055	73.29
200	2.29	1.0	200.0	0.074	0.056	2.29	1.0	200.0	0.074	129.02
500	2.29	1.0	500.0	0.189	0.203	2.29	1.0	500.0	0.189	1655.34

Table 8
Cloud distribution, 100x100 map, medium real time constraints without essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	2.14	1.0	20.0	0.003	0.004	2.14	1.0	20.0	0.003	1.24
50	2.15	1.0	50.0	0.009	0.013	2.15	1.0	50.0	0.029	6.89
100	2.15	1.0	100.2	0.019	0.026	2.15	1.0	100.2	0.019	31.44
150	2.15	1.0	150.0	0.029	0.041	2.15	1.0	150.0	0.029	64.00
200	2.15	1.0	200.0	0.038	0.055	2.15	1.0	200.0	0.038	100.58
500	2.15	1.0	500.0	0.095	0.195	2.15	1.0	500.0	0.095	1249.07

Table 9
Cloud distribution, 100x100 map, soft real time constraints without essential gateways

factor (time span) terms. The way in which the heuristic searches the space is greedy and allocates the end-devices to the first possible gateway. In the last step of the heuristic, a reallocation procedure is performed but this may be improved with other techniques that are out of the purpose of this work. With these limitations, the heuristic solutions are acceptable.

The third group of instances have thousands of end-devices deployed in larger areas. The exact method is not feasible even with a server that has 512GB of RAM which is beyond the standard for any application at the moment. The smaller systems with 1000 to 3000 end-devices in the 500x500 m map, and 4000 end-devices in the 1000x1000 m map, respectively were solved by the CPLEX solver. In the first case, the number of gateways is similar in the heuristic and the exact solutions, the energy required

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	17.05	8.2	158.2	0.024	0.001	15.39	7.6	137.0	0.024	0.004
50	25.10	17.0	364.8	0.051	0.002	21.93	16.2	262.4	0.031	0.142
100	27.59	21.4	564.0	0.070	0.007	25.65	20.2	512.8	0.041	0.356
150	32.23	24.6	1052.0	0.112	0.009	28.61	23.4	747.4	0.039	0.618
200	36.06	27.8	1515.6	0.175	0.011	32.62	27.6	957.8	0.058	0.778
500	53.41	46.8	3081.6	0.290	0.057	50.15	46.2	1901.0	0.094	3.984

Table 10
Uniform distribution, 100x100 map, hard real time constraints with essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	16.85	9.2	144.0	0.011	0.001	13.81	6.8	132.4	0.010	0.138
50	16.60	5.2	533.2	0.048	0.004	14.21	5.4	421.0	0.025	0.514
100	16.93	5.2	1092.8	0.103	0.006	14.76	5.4	900.8	0.046	1.118
150	17.71	5.8	1637.4	0.181	0.011	15.15	5.8	1345.6	0.065	3.306
200	17.85	6.2	2153.8	0.224	0.014	15.34	5.6	1883.6	0.085	4.514
500	17.47	5.8	5429.2	0.532	0.069	15.53	5.6	4796.2	0.220	82.506

Table 11
Uniform distribution, 100x100 map, medium real time constraints with essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	15.61	8.4	138.8	0.007	0.001	13.69	5.0	167.2	0.009	0.112
50	16.63	4.8	564.4	0.035	0.003	14.02	5.8	402.2	0.011	0.610
100	16.64	5.2	1101.6	0.055	0.007	14.74	5.2	934.8	0.025	1.598
150	16.50	5.4	1605.0	0.067	0.011	15.19	5.6	1407.6	0.03	3.738
200	17.04	6.4	2057.2	0.090	0.015	15.24	5.2	1972.2	0.045	5.712
500	11.15	3.4	3696.6	0.220	0.076	9.51	3.6	2872.8	0.122	104.752

Table 12
Uniform distribution, 100x100 map, soft real time constraints with essential gateways

is a bit higher in the heuristic case, and the utilization factor (time span) used by the heuristic is also higher. For the second case, the CPLEX solution required more gateways but the demanded energy and the utilization factor were significantly smaller. Finally, larger systems with more than 4000 end-devices were not solved by the CPLEX as the amount of memory demanded surpassed the 512GB limit. In those instances only the solutions found by the heuristic are shown.

What can be seen is that the differences between the heuristic and exact solutions when it was possible to compare follow a certain pattern. In this way, it is possible to consider that when the CPLEX is unable to solve the problems given the actual hardware limitations, the performance of the heuristic can be assumed to be reasonably good. Further improves can come from the use of artificial intelligence

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	11.05	4.8	103.0	0.028	0.001	10.20	4.8	93.8	0.018	0.062
50	16.96	11.8	220.2	0.048	0.004	14.46	10.6	171.4	0.028	0.244
100	22.43	16.6	504.6	0.101	0.009	18.26	14.8	319.8	0.035	0.602
150	24.37	18.6	749.6	0.143	0.014	20.67	17.4	465.6	0.037	0.904
200	30.02	24.2	1023.8	0.18	0.020	25.44	22.6	537.4	0.040	1.358
500	37.66	32.0	2434.8	0.505	0.099	31.76	29.2	1230.0	0.067	10.364

Table 13
Cloud distribution, 100x100 map, hard real time constraints with essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	10.26	4.0	115.2	0.038	0.001	8.51	3.6	91.2	0.010	0.188
50	11.3	4.8	294.6	0.039	0.005	9.30	3.4	277.0	0.023	0.908
100	11.20	4.6	606.6	0.069	0.010	9.26	3.8	518.6	0.036	2.560
150	12.12	3.4	1184.0	0.150	0.015	9.48	3.6	829.8	0.098	7.112
200	32.83	2.8	1508.4	0.190	0.019	9.44	3.4	1138.6	0.089	9.906
500	10.99	2.8	3743.0	0.435	0.086	9.00	3.0	2868.4	0.238	159.562

Table 14
Cloud distribution, 100x100 map, medium real time constraints with essential gateways

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
20	9.53	4.0	106.2	0.006	0.001	8.49	3.4	99.0	0.005	0.194
50	10.54	4.0	313.6	0.017	0.004	8.96	3.8	251.2	0.026	0.990
100	12.06	6.0	583.8	0.029	0.025	9.86	4.2	567.8	0.023	2.842
150	11.71	4.2	1070.2	0.065	0.014	9.63	4.0	815.6	0.032	5.002
200	11.83	5.0	1309.0	0.073	0.021	9.45	3.6	1134.2	0.047	4.963
500	11.13	3.4	3696.6	0.220	0.076	9.51	3.6	2872.8	0.108	104.752

Table 15
Cloud distribution, 100x100 map, soft real time constraints with essential gateways

techniques to provide a better distribution of the end-devices to reduce the energy and utilization factor (time span) demand.

9. Conclusions

In this paper the multiobjective gateway placement problem was presented, modeled and solved both with an exact method solver (CPLEX) and a custom heuristic. It has been prooved that the problem has NP-Hard characteristics and that because of that as soons as the size of the system to solve increases the

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
1000	2.30	17.0	1583.6	0.488	0.46	1.70	19.0	1242.2	0.098	79.68
2000	1.13	15.0	3477.6	0.894	1.69	0.91	14.6	2723.2	0.196	5359.14
3000	1.47	17.8	4851.2	0.913	4.23					
4000	1.76	20.0	6054.4	0.961	7.96					
5000	2.00	20.2	7312.8	0.855	13.10					
6000	2.22	24.8	8089.8	0.800	22.04					
7000	2.72	18.8	10939.8	1.000	28.99					
8000	2.86	24.0	11207.4	0.919	41.81					
9000	3.28	22.4	13470.4	1.000	49.55					
10000	3.51	22.8	14487.6	1.000	69.02					
11000	3.81	21.8	15951.6	1.000	88.51					
12500	2.48	24.2	19155.6	1.000	110.85					
14000	2.61	33.8	18600.4	1.000	171.14					
15500	2.96	28.6	23105.8	1.000	209.02					
17000	3.10	31.0	24018.2	1.000	279.48					
18500	3.32	34.2	25557.6	1.000	346.41					
20000	3.61	33.4	28648.8	1.000	409.10					

Table 16
Uniform distribution, 500x500 map, hard real time constraints

ED	Greedy Heuristic				
	OF	G	E	U	Time
4000	12.296	34.8	10818.8	1.0	5.694
6000	4.038	41.8	15244.0	1.0	15.114
8000	2.94	47.6	19118.8	1.0	32.23
10000	2.212	51.4	22889.8	1.0	57.724
15000	2.824	52.8	34568.4	1.0	156.566
20000	3.344	62.0	44979.4	1.0	318.622
25000	3.68	70.6	44530.2	1.0	575.894
30000	3.944	74.6	48310.6	1.0	1007.73
35000	4.474	76.6	58039.6	1.0	1404.254
40000	5.21	81.0	71065.0	1.0	1542.424

Table 17
Uniform distribution, 1000x1000 map, hard real time constraints

exact solution can not be computed. The heuristic proposed is based on the state minimization problem in the digital circuits design of state machines. Thus, the algorithm is pointed to reduced the number of partitions (gateways) to obtain full coverage. When the distances are short enough as to provide coverage to all the end-devices in an area, both the exact and heuristic methods provide almost the same result, but the heuristic does that in at least two orders of magnitude faster. When there are essential gateways,

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
1000	2.184	12.4	1821.8	0.148	0.32	1.62	19.8	1209.0	0.021	148.67
2000	1.046	11.6	3808.8	0.327	1.074	0.85	13.8	2781.2	0.051	3479.994
3000	1.368	12.8	17473.6	0.3	2.468					
4000	1.684	13.4	6774.0	0.386	4.432	8.082	40.0	7238.4	0.059	1066.566
5000	1.96	14.4	8074.6	0.368	6.998					
6000	2.15	16.2	8841.8	0.368	10.892					
7000	2.46	17.2	10213.8	0.471	16.146					
8000	2.884	15.2	12454.2	0.579	20.398					
9000	3.154	16.0	13732.0	0.59	28.884					
10000	3.36	17.0	14661.8	0.56	35.568					
11000	3.826	15.6	17022.6	0.714	40.892					
12500	2.428	14.8	20765.6	0.722	55.348					
14000	2.496	16.0	21217.2	0.748	76.652					
15500	2.63	20.2	21726.6	0.683	101.784					
17000	2.792	20.2	23384.4	0.647	128.882					
18500	2.842	22.8	23457.0	0.537	165.82					
20000	3.28	20.2	28209.4	0.705	189.328					

Table 18
Uniform distribution, 500x500 map, medium real time constraints

ED	Greedy Heuristic				
	OF	G	E	U	Time
4000	13.188	14.2	12408.6	0.635	3.94
6000	3.598	21.8	14433.8	0.497	10.936
8000	2.494	19.8	20517.0	0.598	20.58
10000	1.664	22.8	23645.6	0.655	36.512
15000	2.256	25.0	34581.8	0.667	104.158
20000	2.766	30.8	42446.4	0.73	201.914
25000	3.262	30.4	52448.8	0.798	364.462
30000	3.698	32.4	61158.2	0.796	591.218
35000	4.108	37.4	66499.4	0.886	839.17
40000	4.636	39.2	76327.0	0.944	1114.922

Table 19
Uniform distribution, 1000x1000 map, medium real time constraints

the heuristic performance is not as good as in the previous case. However, it is still much faster than the exact method.

In the future, the heuristic approach can be improved with artificial intelligence techniques that after obtaining the minimum number of gateways can provide a better allocation of end-devices as to reduce the energy and utilization factor (time span) terms further.

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
1000	2.184	12.4	1821.8	0.148	0.32	1.62	19.8	1209.0	0.021	148.67
2000	1.046	11.6	3808.8	0.327	1.074	0.85	13.8	2781.2	0.051	3479.994
3000	1.368	12.8	17473.6	0.3	2.468					
4000	1.684	13.4	6774.0	0.386	4.432					
5000	1.96	14.4	8074.6	0.368	6.998					
6000	2.15	16.2	8841.8	0.368	10.892					
7000	2.46	17.2	10213.8	0.471	16.146					
8000	2.884	15.2	12454.2	0.579	20.398					
9000	3.154	16.0	13732.0	0.59	28.884					
10000	3.36	17.0	14661.8	0.56	35.568					
11000	3.826	15.6	17022.6	0.714	40.892					
12500	2.428	14.8	20765.6	0.722	55.348					
14000	2.496	16.0	21217.2	0.748	76.652					
15500	2.63	20.2	21726.6	0.683	101.784					
17000	2.792	20.2	23384.4	0.647	128.882					
18500	2.842	22.8	23457.0	0.537	165.82					
20000	3.28	20.2	28209.4	0.705	189.328					

Table 20
Uniform distribution, 500x500 map, soft real time constraints

ED	Greedy Heuristic				
	OF	G	E	U	Time
4000	13.19	14.2	12408.6	0.635	3.94
6000	3.608	21.8	14433.8	0.497	10.94
8000	2.49	19.8	20517.0	0.598	20.58
10000	1.66	22.8	23645.6	0.655	36.51
15000	2.26	25.0	34581.8	0.667	104.16
20000	2.77	30.8	42446.4	0.730	201.91
25000	3.26	30.4	52448.8	0.798	364.46
30000	3.70	32.4	61158.2	0.796	591.22
35000	4.11	37.4	66499.4	0.886	839.17
40000	4.64	39.2	76327.0	0.944	1114.92

Table 21
Uniform distribution, 1000x1000 map, soft real time constraints

Appendix Extending the GPP problem to other technologies

The gateway placement problem is common to all the wireless networks. In the model described in Section 5, the optimization problem was specified for the LoRaWAN®technology. However, in the case the system uses other technologies, the problem remains similar.

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
1000	1.73	8.2	1198.2	0.47	0.431	5.83	32.8	4899.4	0.351	500.312
2000	0.77	8.0	2354.8	0.82	1.652	2.38	40.3	7192.0	0.341	5401.333
3000	1.05	10.8	3388.2	0.98	3.546					
4000	1.34	11.6	4754.2	1.00	6.94					
5000	1.57	14.6	5628.6	1.00	12.6					
6000	1.79	14.2	6746.4	1.00	20.528					
7000	2.05	18.6	7598.4	1.00	30.55					
8000	2.28	18.2	8806.6	1.00	45.418					
9000	2.70	17.0	11040.8	1.00	58.038					
10000	2.92	20.0	11813.0	1.00	82.978					
11000	3.21	21.0	13202.6	1.00	97.456					
12500	1.99	23.8	14528.8	1.00	148.252					
14000	2.12	22.4	17980.8	1.00	187.136					
15500	2.60	23.8	20439.2	1.00	248.806					
17000	2.69	28.8	22425.4	1.00	367.752					
18500	2.95	31.0	22538.2	1.00	455.492					
20000	3.30	30.2	26152.8	1.00	502.0					

Table 22
Cloud distribution, 500x500 map, hard real time constraints without essential gateways

ED	Greedy Heuristic				
	OF	G	E	U	Time
4000	10.50	24.2	9239.0	1.0	6.27
6000	3.16	25.0	12525.4	1.0	18.09
8000	1.89	24.4	13254.6	1.0	41.08
10000	1.60	34.6	17361.0	1.0	66.61
15000	1.92	38.2	22473.2	1.0	189.01
20000	2.53	40.8	33466.0	1.0	405.76
25000	2.92	48.8	38159.6	1.0	744.12
30000	3.57	50.0	50572.6	1.0	1158.33
35000	4.10	55.0	59171.0	1.0	1634.93
40000	5.03	56.0	77403.8	1.0	2574.48

Table 23
Cloud distribution, 1000x1000 map, hard real time constraints without essential gateways

9.1. IEEE 802.11

This technology is the most used for office and home Internet deployments. It is common however, to have interference problems with the different access-points and amount of devices connected to them. Like in the case of LoRaWAN®the different versions of IEEE 802.11 operate with a set of channels

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
1000	1.40	7.8	1171.0	0.09	0.36	1.28	9.4	1111.0	0.042	2588.92
2000	0.66	8.2	2350.8	0.19	1.12	0.64	5.0	2461.0	0.135	5414.42
3000	0.94	8.8	3626.8	0.27	2.42					
4000	1.16	10.2	4503.6	0.36	4.35					
5000	1.50	11.2	6016.4	0.51	6.37					
6000	1.67	11.0	6767.6	0.60	9.77					
7000	1.94	10.2	8211.2	0.67	13.12					
8000	2.23	10.8	9482.8	0.75	16.43					
9000	2.49	12.4	10595.8	0.79	22.66					
10000	2.71	12.2	9530.4	0.82	28.11					
11000	2.88	12.4	12454.2	0.92	36.35					
12500	1.85	9.6	15611.0	0.98	48.45					
14000	1.95	11.2	16494.4	0.96	64.64					
15500	2.09	12.0	17724.6	1.00	82.61					
17000	2.30	11.0	20024.4	1.00	98.25					
18500	2.49	10.6	21976.6	1.00	123.74					
20000	2.60	13.0	22680.6	1.00	144.87					

Table 24
Cloud distribution, 500x500 map, medium real time constraints without essential gateways

ED	Greedy Heuristic				
	OF	G	E	U	Time
4000	8.17	10.4	7654.0	0.396	3.518
6000	2.33	12.2	10025.6	0.492	9.350
8000	1.66	13.4	13304.8	0.575	17.645
10000	1.17	15.8	16541.0	0.699	31.948
15000	1.63	15.6	25845.6	0.789	78.114
20000	1.92	19.2	30143.0	0.852	154.082
25000	2.32	22.2	37176.2	0.946	280.842
30000	2.67	24.0	43111.8	0.990	441.702
35000	2.93	23.6	48357.8	1.000	682.428
40000	3.58	23.6	61409.8	1.000	877.996

Table 25
Cloud distribution, 1000x1000 map, medium real time constraints without essential gateways

and codification schemes that allow for multiple messages transmissions (MIMO). To adapt the model to this technology, the number of orthogonal channels available should be computed and the amount of codification schemes can be used to replace the spreading factor variable. The duty-cycle restriction can be removed or modified to the standard specification. The restrictions related to transmission range are unified as in this case, all the codification schemes have the achieve the same distance.

ED	Greedy Heuristic					CPLEX				
	OF	G	E	U	Time	OF	G	E	U	Time
1000	1.40	7.8	1171.0	0.092	0.356	1.276	9.4	1111.0	0.042	2588.924
2000	0.66	8.2	2350.8	0.194	1.124	0.64	5.0	2461.0	0.135	5414.42
3000	0.94	8.8	3626.8	0.270	2.422					
4000	1.16	10.2	4503.6	0.360	4.348					
5000	1.50	11.2	6016.4	0.513	6.372					
6000	1.67	11.0	6767.6	0.605	9.776					
7000	1.94	10.2	8211.2	0.68	13.118					
8000	2.23	10.8	9482.8	0.753	16.426					
9000	2.49	12.4	10595.8	0.79	22.658					
10000	2.71	12.2	9530.4	0.824	28.108					
11000	2.88	12.4	12454.2	0.919	36.354					
12500	1.85	9.6	15611.0	0.982	48.454					
14000	1.95	11.2	16494.4	0.958	64.636					
15500	2.09	12.0	17724.6	1.0	82.606					
17000	2.34	10.5	20538.0	1.0	98.252					
18500	2.49	10.6	21976.6	1.0	123.742					
20000	2.60	13.0	22680.6	1.0	144.874					

Table 26
Cloud distribution, 500x500 map, soft real time constraints without essential gateways

ED	Greedy Heuristic				
	OF	G	E	U	Time
4000	8.172	10.4	7654.0	0.396	3.518
6000	2.326	12.2	10025.6	0.492	9.35
8000	1.662	13.4	13304.8	0.575	17.645
10000	1.166	15.8	16541.0	0.699	31.948
15000	1.634	15.6	25845.6	0.789	78.114
20000	1.922	19.2	30143.0	0.852	154.082
25000	2.318	22.2	37176.2	0.946	280.842
30000	2.674	24.0	43111.8	0.99	441.702
35000	2.928	23.6	48357.8	1.0	682.428
40000	3.58	23.6	61409.8	1.0	877.996

Table 27
Cloud distribution, 1000x1000 map, soft real time constraints without essential gateways

9.2. Bluetooth

This technology was first conceived as a peer-to-peer communication standard. However, the impulse of IoT applications has made it evolve and today it is possible to connect up to seven devices to a unique gateway. The gateway placement problem is again present in this case. Further adaptation of the restrictions may be necessary as duty-cycle and distances differ.

9.3. IEEE 802.16

This technology, commonly known as WiMax, allows the wireless transmission of messages in long distances. The difference with LoRa is that the power used is significantly higher. Like in the case of IEEE 802.11 there are several orthogonal channels and codification schemes that allow simultaneous transmissions. The gateway placement problem can be solved again adapting the number of channels and codification techniques to replace the spreading-factor constraints.

References

Abramson, N., 1970. The aloha system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pp. 281–285.

Capone, A., Cesana, M., De Donno, D., Filippini, I., 2009. Optimal placement of multiple interconnected gateways in heterogeneous wireless sensor networks. In Fratta, L., Schulzrinne, H., Takahashi, Y. and Spaniol, O. (eds), *NETWORKING 2009*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 442–455.

Elsaleh, T., Enshaeifar, S., Rezvani, R., Acton, S.T., Janeiko, V., Bermudez-Edo, M., 2020. Iot-stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services. *Sensors* 20, 4.

Finochietto, J.M., Dieng, O., Mosse, D., Santos, R.M., 2023. Adding empirical real-time guarantees to lorawan. In *ACM International Conference Proceeding Series*, p. 99 – 107.

Finochietto, M., Eggly, G.M., Santos, R., Orozco, J., Ochoa, S.F., Meseguer, R., 2019. A role-based software architecture to support mobile service computing in iot scenarios. *Sensors* 19, 21, 4801.

Guan, N., Yi, W., Gu, Z., Deng, Q., Yu, G., 2008. New schedulability test conditions for non-preemptive scheduling on multiprocessor platforms. In *2008 Real-Time Systems Symposium*, pp. 137–146.

He, B., Xie, B., Agrawal, D.P., 2008. Optimizing deployment of internet gateway in wireless mesh networks. *Computer Communications* 31, 7, 1259–1275. Special Issue: Resource Management and routing in Wireless Mesh Networks.

ITU, 2021. Recommendation itu-t y.4480 “low power protocol for wide area wireless networks”. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=14818.

Koch, T., 2004. Rapid Mathematical Programming. Ph.D. thesis, Technische Universität Berlin. ZIB-Report 04-58.

Lin, C.C., 2013. Dynamic router node placement in wireless mesh networks: A pso approach with constriction coefficient and its convergence analysis. *Information Sciences* 232, 294–308.

Luo, J., Wu, W., Yang, M., 2010. Interference-aware gateway placement for wireless mesh networks with fault tolerance assurance. In *2010 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, pp. 2373–2380.

Marenco, J., Micheletto, M.J., Santos, R.M., 2023. Multiobjective formulation for last-mile optimization in wireless networks. In *2023 XLIX Latin American Computer Conference (CLEI)*, pp. 1–5.

Micheletto, M., Zabala, P., Ochoa, S.F., Meseguer, R., Santos, R., 2023. Determining real-time communication feasibility in iot systems supported by lorawan. *Sensors* 23, 9.

Park, M., 2007. Non-preemptive fixed priority scheduling of hard real-time periodic tasks. In *Proceedings of the 7th International Conference on Computational Science, Part IV: ICCS 2007*, Springer-Verlag, Berlin, Heidelberg, p. 881–888.

Santos, R., Lipari, G., Santos, J., 2008. Improving the schedulability of soft real-time open dynamic systems: The inheritor is actually a debtor. *Journal of Systems and Software* 81, 7, 1093 – 1104.

Santos , R., Eggly, G., Gutierrez, J., Chesñevar , C.I., 2023. Extending the iot-stream model with a taxonomy for sensors in sustainable smart cities. *Sustainability* 15, 8.

SemTech, 2024. Lora alliance. https://lora-alliance.org/.

Shanmuga Sundaram, J.P., Du, W., Zhao, Z., 2020. A survey on lora networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials* 22, 1, 371–388.

Silva, F.J.F., De la Figuera, D., 2007. A capacitated facility location problem with constrained backlogging probabilities. *International journal of production research* 45, 21, 5117–5134.

Talay, A.C., 2007. A gateway access-point selection problem and traffic balancing in wireless mesh networks. In *Applications of Evolutionary Computing: EvoWorkshops 2007: EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog. Proceedings*, Springer, pp. 161–168.

Vangelista, L., 2017. Frequency shift chirp modulation: The lora modulation. *IEEE Signal Processing Letters* 24, 12, 1818–1821.