Topology Design for Cellular IoT: From ILP to ML Perspective

Nakrop Jinaporn[†], Non-member

ABSTRACT

Due to the emerging deployment of cellular IoT, a network topology design appears to be one of the greatest challenges faced by mobile network operators, that is, both the capacity maximization and the overall network cost minimization have been considered as the objective of network planning. In this article, the topology design for cellular IoT is divided into two subproblems: gateway location and gateway connection problems. They are formulated as the integer linear programming problem. For the former subproblem, the best gateway locations and the optimal network cost can be obtained by the optimization approach to form multiple local networks. For the latter subproblem, a connection of selected gateways with the minimum connection cost can be presented by the Kruskal algorithm to form a backbone-like network. This results in a two-layered network with the minimum network cost. According to the results, a significant reduction in the network cost could be obtained with the optimal setting of system parameters. In addition to the optimization approach, the gateway location problem is examined by means of clustering algorithms. The gateway placement can be obtained by K-medoids clustering with the low time complexity.

Keywords: Topology Design, Cellular IoT, ILP, Clustering

1. INTRODUCTION

According to [1], key requirements and application examples are defined for fifth generation (5G). For example, a data rate of at least 1 Gbps is required to support virtual reality (VR) applications. 300,000 connected devices per access point (AP) are expected for massive deployment of sensors. The data transmission with 99.999 percent reliability is required for teleprotection in smart grid networks. This results in the greatest challenges of mobile and wireless networks beyond 2020. The promising solutions are presented such as direct device-to-device (D2D) communications, massive machine communications (MMC), ultra-dense networks (UDNs) and ultra-reliable communications (URC). To develop these solutions, key technology components are also described in [1].

5G use cases can be divided into three categories [2]: enhanced mobile broadband (eMBB), massive machine type communication (mMTC) and ultra-reliable low latency communication (URLLC). First, the mobile broadband with improved data rates and seamless user experience is the focus of interest in eMBB. Second, reliability and latency are the main requirement of URLLC. Self driving car, mission critical application and industrial automation are the example of this use case. Third, a connection of 1 million low-cost and long-life devices per km^2 is expected in mMTC. Some examples are precision agriculture and smart city using Internet of Things (IoT) devices.

Existing wireless technologies, including Zigbee, Bluetooth, WiFi, long range radio (LoRa) and cellular IoT, can be used to support connected IoT devices [3]. Compared to other technologies, massive access, wide coverage area, quality of service (QoS) provisioning and tight coordination can be provided by cellular IoT [4]. Moreover, it can reuse the existing cellular infrastructure. Therefore, cellular IoT is the main focus of attention in this article.

Although cellular IoT could get benefit from cellular technologies, there are still challenges in the real-world deployment of cellular IoT, i.e. both operational and planning levels. The cellular operators may deal with massive IoT devices with sporadic traffic, small payload, low power, ubiquitous distribution, limited capability, stringent latency constraint and heterogeneous QoS requirement [3]. Besides, the conventional scheme of cellular systems may be infeasible to support cellular IoT. This leads to a lot of challenges in practice such as energy efficiency, signaling overhead and network congestion [4].

In this article, a topology design problem in cellular IoT is the main focus of interest. The objective is to connect the gateways and IoT devices to form a two-layered network architecture while the overall network cost is minimized. In addition to the optimization approach, machine learning (ML)-based gateway placement is examined. The main contributions include the following.

Problem formulation: A network topology design, including gateway location and gateway connection problems, is formulated as the integer linear programming (ILP) problem to minimize the network cost.

Manuscript received on April 23, 2023; revised on May 29, 2023; accepted on July 13, 2023. This paper was recommended by Associate Editor Kampol Woradit.

The author is with Department of Electrical and Electronic Engineering, Faculty of Engineering, Ubon Ratchathani University, Thailand.

[†]Corresponding author: nakrop.j@ubu.ac.th

Digital Object Identifier: 10.37936/ecti-eec.2024221.252931

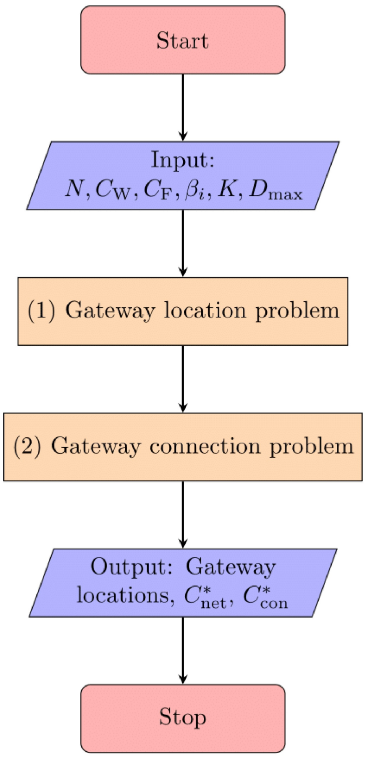

***Fig. 1**: Block diagram of the topology design.*

ILP-based topology design: The gateway location problem is first modeled by the optimization approach to evaluate the selection of gateway locations. Subsequently, the connection of selected gateways is formed by means of the Kruskal algorithm. The best locations and connection of gateways with the minimum network cost will be then obtained. The whole process of the topology design in this study is shown in Fig. 1. Let us describe the inputs and outputs later in Sec. 3-4.

ML-based gateway placement: The same gateway location problem is solved by clustering algorithms (i.e. K-means and K-medoids clustering). The gateway locations (i.e. cluster heads) and the network cost are then obtained. We aim to introduce ML into the topology design problem.

The remaining sections are listed as follows: Related work is discussed in Sec. 2. A system model and problem formulation are described in Sec. 3. ILP-based topology design is presented in Sec. 4. ML-based gateway placement is introduced in Sec. 5. Key points and research challenges are identified in Sec. 6.

2. RELATED WORK

To support the use case of mMTC, the two main cellular IoT standards, including LTE-MTC (LTE-M) and Narrowband IoT (NB-IoT), are introduced by the Third Generation Partnership Project (3GPP) in LTE Release 13. LTE-M is developed to support mobile and high data rate IoT applications such as vehicle tracking and video surveillance while NB-IoT is suitable for fixed and low data rate IoT applications such as smart metering and home automation [5]. Both cellular IoT technologies could face with the various challenges mentioned in Sec. 1. Out of these challenges, due to a large number of IoT devices and a variety of network architectures, a topology design has become another important issue resulting in the deployment of cellular IoT networks.

According to [6], a LTE-A network architecture for machine-to-machine (M2M) communications is presented. The LTE-A network includes the core network (CN) and the radio access network (RAN). The CN offers multiple functionalities such as user equipment (UE) management and Internet Protocol (IP) data traffic transmission. The RAN consisting of base stations (eNBs) provides the radio access (i.e. user and control plane protocols) for UEs. Either a traditional cellular device or a machine type communications (MTC) device can be considered as the UE. The number of MTC devices is connected to the eNBs either directly or via the gateway. The gateway could offer local control and functionalities for M2M communications such as data routing and processing. The eNB-to-gateway, gateway-to-gateway, and MTC-to-MTC links can be established based on a variety of wireless standards such as LTE-A and WiFi. The similar architecture has been applied to deploy cellular IoT networks [7, 8]. Besides, the comprehensive details of the network architecture for cellular IoT (i.e. LTE-M and NB-IoT) are provided in [9, 10].

Since the existing cellular infrastructure can be reused to develop cellular IoT networks, the process of cellular network planning should be examined, resulting in a topology design in cellular IoT. The cellular network planning has been divided into three steps [11]. First, the pre-planning (or dimensioning) is to determine the approximate number of eNBs within the coverage area (i.e. a region of interest). Second, the detailed planning is to deal with the placement of these eNBs. Third, the post-planning (or optimization) is to analyze the network performance and adjust network parameters in order to improve the network operation. The objectives of cellular network planning could be defined in connection with the overall network cost, capacity, coverage, power consumption and handover zone. Traffic models, potential site locations, eNB models and propagation models being the input of planning process are required while the output (or goal) may comprise the optimal number of eNBs, best eNB locations, optimal eNB types, parameter configuration, frequency reuse pattern and capacity dimensioning. Both infrastructure networks and ad hoc networks have been widely deployed based on the planning process. Clearly, the same process is still essential for a topology design problem in cellular IoT.

In [12-14], the network cost consists of capital expenditure (CAPEX) and operational expenditure (OPEX) costs. The CAPEX is the money that an operator spends on the initial setup of the network such as site building, base station equipment, site equipment and transmission network building. The OPEX is the expense of operating and maintaining the network such as operation OPEX, site rent, energy consumption and transmission network OPEX. Obviously, this results in the total cost of ownership (TCO) which could be considered in connection with the topology design problem.

To consider a topology design, the main question is on how network elements are connected to form a connected network topology. For example, a two-layered network architecture typically consists of a backbone network and multiple local networks. In [15], the topology design is divided into two subproblems: location and connection problems. In the context of cellular IoT networks, the location problem is to select gateways from the given node locations and assign IoT device(s) to each individual gateway to form multiple local networks. Subsequently, the connection problem is to connect these gateways to form a backbone network. Breaking the topology design problem into subproblems, the problem can be solved within the available time and heuristics can be developed step-by-step.

To the best of our knowledge, the topology design for wireline communications is in a majority in the literature. In [16], the network topology design, including concentrator location problem (concentrator node selection and non-concentrator node assignment) and concentrator connection problem (backbone network formation), is examined for an optical network. Routing and wavelength assignment (RWA) and optical amplifier placement are then included in the topology design problem in [17]. In smart grid communication networks, the phasor measurement unit and communication link placement problems are jointly considered by means of mixed integer quadratic constraint programming (MIQCP) in [18].

Out of the optimization approach, clustering being the most common algorithm in unsupervised learning is introduced to solve the gateway location problem. To minimize the distance between data aggregation points (DAPs) and smart meters (SMs), the DAPs placement for advanced metering infrastructure (AMI) in residential grids using K-means clustering is proposed in [19] while the similar problem is examined by means of K-medoids clustering in [20].

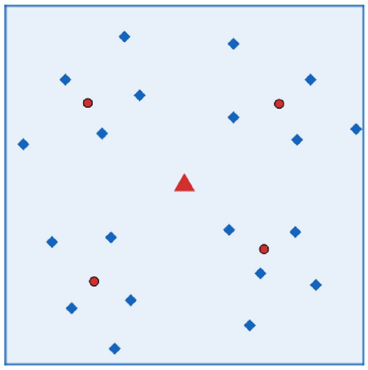

Fig. 2: *A cell layout with a single eNB (a red triangle), gateways (red dots) and IoT devices (blue diamonds).*

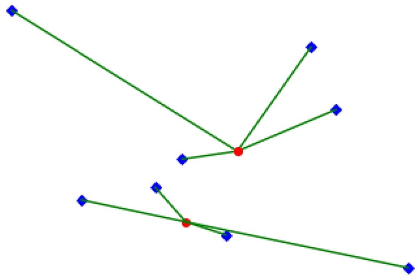

Fig. 3: *Multiple local networks with the number of gateways (red dots) and IoT devices (blue diamonds).*

3. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the system model and problem formulation of a topology design in cellular IoT are discussed.

3.1 System Model

A cell layout, including a single eNB, gateways and IoT devices, as shown in Fig. 2 is considered. The eNB has already been installed over the area. In this way, the existing cellular infrastructure could be reused to support the deployment of cellular IoT in the area. In our system model, the similar architecture, including IoT devices and gateways, presented in [6] is used for the modeling. Let us assume that each IoT device has to connect to the eNB (i.e. the cellular network) via a single gateway as illustrated in Fig. 3. Either wired or wireless technology can be used to make a connection between the IoT device and the gateway.

Based on the network cost model in [12], the number of installed gateways has an impact on the network cost in terms of equipment cost. Moreover, the link between node elements results in the connection (or transmission) cost. Thus, a topology design which can minimize the network cost (i.e. the CAPEX) is the main focus of attention in this article. Note that, only equipment and connection costs are considered for the sake of our study. Other items of CAPEX and OPEX costs in practice could be included in the network cost model.

3.2 Problem Formulation

The topology design is divided into two subproblems as mentioned in [15]: gateway location and gateway connection problems. Related work in [21, 22] results in the ILP formulation of each subproblem and an algorithm for the gateway connection problem. Note that, the interference and channel effects are not considered here.

3.2.1 Gateway Location Problem

To consider the gateway location problem, the network information such as possible site locations (or nodes) is required. This problem can be formulated as an ILP problem as follows:

Given Information
N: Number of nodes
α_{ij}: Connection cost between node i and node j
β_i: Equipment cost of gateway i
K: Maximum number of IoT devices attached to a gateway
D_{ij}: Separation distance between node i and node j
D_{max}: Maximum distance between each IoT device and its gateway

Variables
$y_i \in \{0, 1\}$: Set to 1 if selected node i is a gateway. Otherwise, set to 0
$x_{ij} \in \{0, 1\}$: Set to 1 if node i is assigned to gateway j. Otherwise, set to 0. Let $x_{ii} = 1$ if selected node i is a gateway

Constraints
Each end device is attached to a single gateway

$$\forall i \in \{1, \ldots, N\}, \sum_{j=1}^{N} x_{ij} = 1 \tag{1}$$

Gateway capacity

$$\forall j \in \{1, \ldots, N\}, \sum_{i=1}^{N} x_{ij} \leq K y_j \tag{2}$$

Gateway-end device range

$$\forall i, j \in \{1, \ldots, N\}, D_{ij} x_{ij} \leq D_{\max} \tag{3}$$

Integer constraints

$$\forall i, j \in \{1, \ldots, N\}, x_{ij} \in \{0, 1\} \tag{4}$$

$$\forall i \in \{1, \ldots, N\}, y_i \in \{0, 1\} \tag{5}$$

Objective
Minimizing the network cost (i.e. connection and equipment costs) of the network topology

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij} x_{ij} + \sum_{i=1}^{N} \beta_i y_i \tag{6}$$

The connection cost between node i and node j can be defined as follows:

$$\alpha_{ij} = C_{\mathrm{W}} + C_{\mathrm{F}} D_{ij} \tag{7}$$

Denote the fixed cost of wireless and wired connection between any two node elements by C_W and C_F, respectively. Let us assume that both wireless and wired connections coexist (i.e. the extreme case).

3.2.2 Gateway Connection Problem

The connection of gateways (i.e. minimum spanning tree) is considered here. The connection problem can be formulated as an ILP problem as follows:
Given Information
$\mathcal{N}$: Set of gateway nodes ($|\mathcal{N}| = n$)
$\mathcal{E}$: Set of gateway edges ($|\mathcal{E}| = m$) where every gateway edge $(i, j) \in \mathcal{E}$
γ_{ij}: Cost of a directed link from gateway i to gateway j (i.e. cost of gateway edge (i, j)). Let $\gamma_{ij} = 0$ for $i = j$
Variables
$z_{ij} \in \{0, 1\}$: Set to 1 if gateway i is connected to gateway j (i.e. the gateway edge (i, j) is included in the tree $\mathcal{T}$). Otherwise, set to 0
Constraints
The tree $\mathcal{T}$ has n - 1 edges

$$\sum_{(i,\, j) \in \mathcal{E}} z_{ij} = n - 1 \tag{8}$$

Sub-tour elimination

$$\forall \mathcal{S} \subset \mathcal{N},\ \mathcal{S} \neq \emptyset,\ \mathcal{N}, \quad \sum_{(i,\, j)\, \in\, \mathcal{E} | i,\, j\, \in\, \mathcal{S}} z_{ij} \leq |\mathcal{S}| - 1 \tag{9}$$

Integer constraints

$$\forall (i, j) \in \mathcal{E},\ z_{ij} \in \{0, 1\} \tag{10}$$

Objective
Minimizing the total cost (i.e. connection cost) of the gateway network

$$\min \sum_{(i,\, j) \in \mathcal{E}} \gamma_{ij}\, z_{ij} \tag{11}$$

Clearly, the number of constraints becomes exponential (i.e. 2^n - 1) resulting in the high computational complexity. Therefore, instead of optimization approach, the Kruskal algorithm is used to find the gateway network (i.e. a minimum spanning tree) for the sake of our study. A pseudocode of the Kruskal algorithm can be presented as Alg. 1.

Algorithm 1 Kruskal algorithm

```
1: Sort all the edges (i, j) in non-decreasing order of γ_ij
2: Select the smallest edge from the sorted list
   obtained in line 1
3: if No cycle is formed then
4:         The edge is included in the tree T
5: else
6:         The edge is discarded
7: end if
8: Repeat lines 2-7 until there are n - 1 edges in the
   tree T
```

4. ILP-BASED TOPOLOGY DESIGN

In this section, the subproblems in Sec. 3 are modeled and solved by Python with PuLP [23]. System parameters for modeling are provided. The optimal solutions to the two subproblems are then investigated. Note that, the performance evaluation is performed on a 3.7 GHz AMD Ryzen 7 CPU with 16 GB of RAM.

***Table 1**: System Parameters of Gateway Location Problem.*

Parameter	Value
N	20
C_W (in \$US per link)	10
C_F (in \$US per meter)	1
β_i (in \$US per gateway)	50-500
K	5
D_{max} (in m)	50-400

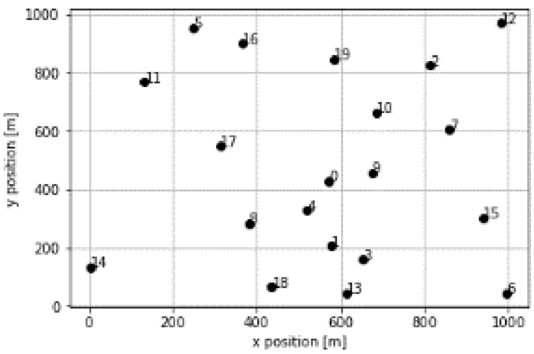

***Fig. 4**: All potential node locations within the region of interest.*

4.1 System Parameters

Tab. 1 shows system parameters to model the gateway location problem. Note that, these parameters are selected based on the cell layout in Fig. 2 and approximate CAPEX costs. N potential node locations are randomly generated using the uniform distribution over a square area of 1000 m by 1000 m as shown in Fig. 4. For the gateway connection problem, the locations of selected gateways from the first subproblem are the main input into the Kruskal algorithm. Assume that the edge cost is equal to the distance between gateways i and j, i.e. $\gamma_{ij} = D_{ij}$.

4.2 Numerical Results

Numerical results of two subproblems are presented and discussed here.

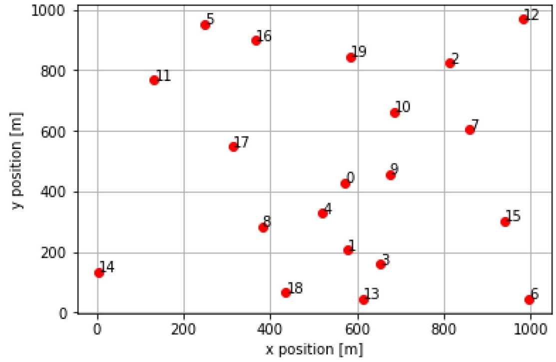

***Fig. 5**: The best gateway locations with β_i = \$300 and D_{max} = 50 m.*

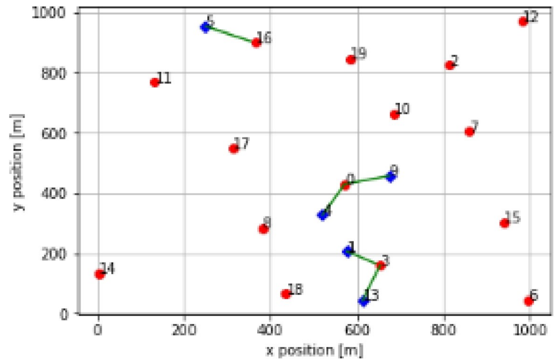

***Fig. 6**: The best gateway locations with β_i = \$300 and D_{max} = 150 m.*

4.2.1 Gateway Location

Denote the optimal network cost from (6) by C^*_{net}. The impact of D_{max} and β_i on C^*_{net} is examined. Figs. 5-7 show the best locations of gateways for D_{max} = 50, 150 and 300 (in m) where β_i = \$300, respectively. The selected gateways are shown as red dots. There are no local networks with D_{max} = 50 m as illustrated in Fig. 5 while multiple local networks covering the area could be formed with D_{max} = 300 m as shown in Fig. 7. The gateway near the bottom left-hand corner of the area (i.e. node number 14) is not connected to any IoT device since it is far from other node locations resulting in the high connection cost.

The optimal network costs and the runtime values with different D_{max} are listed in Tab. 2. According to the results, when D_{max} increases in value, there are a significant decrease in C^*_{net} and a gradual increase in the runtime. Compared to the case of D_{max} = 50 m, there is around a 25% reduction in C^*_{net} for D_{max} = 300 m. Clearly, the D_{max} of 300 m should be the best setting in this situation. The network cost could not be further minimized by the higher values of D_{max} Compared to the case of D_{max} = 300 m, the same C^*_{net} is obtained by D_{max} = 400 m. However, this results in the higher runtime as shown in Tab. 2.

The best locations of gateways for β_i = 100, 250 and

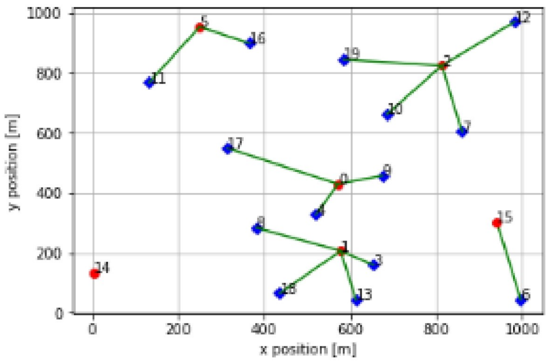

***Fig. 7**: The best gateway locations with β_i = \$300 and D_{max} = 300 m.*

***Table 2**: Optimal Network Costs of Different D_{max}.*

$D_{\max}$ (in m)	C^*_{net} (in \$US)	Runtime (in sec)
50	6200	0.32
100	5988.24	0.35
150	5259.71	0.35
200	5099.50	0.30
250	4717.16	0.47
300	4665.08	3.66
350	4665.08	4.81
400	4665.08	12.11

500 (in \$US) are shown in Fig. 8-10 where D_{max} = 300 m, respectively. A higher β_i results in a smaller number of gateways. In other words, a larger number of local networks could be formed by a high value of β_i. For example, there is only a single local network for β_i = \$100 in Fig. 8 while multiple local networks are formed over the area for β_i = \$500 in Fig. 10.

The optimal network costs and the runtime values with different β_i are presented in Tab. 3. Clearly, a higher β_i results in a higher C^*_{net}. There is a significant increase in the runtime for β_i = \$50-250, after which point it gradually reduces with a larger β_i Although the same local networks covering the area could be obtained as shown in Fig. 7 and Fig. 10, compared to the case of β_i = \$500, there is approximately a 20% decrease in C^*_{net} for β_i = \$300.

According to the results, therefore, both D_{max} and β_i have a significant impact on not only C^*_{net} but also the network topology (i.e. local networks). This could result in the design trade-offs in practice.

4.2.2 Gateway Connection

The best gateway positions (i.e. six gateways) as shown in Fig. 7 and Fig. 10 are selected to form a minimum spanning tree. Subsequently, the gateway network as illustrated in Fig. 11 is obtained with the time complexity $O(m \log m)$ (or equivalently $O(m \log n)$) by means of the Kruskal algorithm mentioned in Sec. 3.

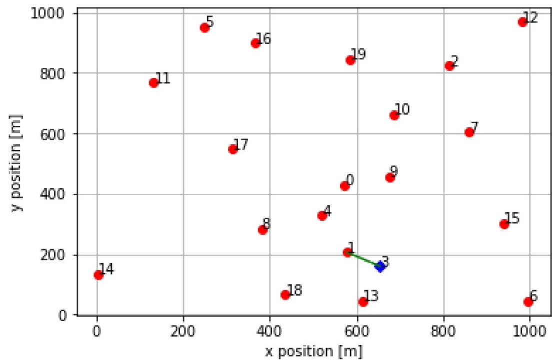

***Fig. 8**: The best gateway locations with D_{max} = 300 m and β_i = \$100.*

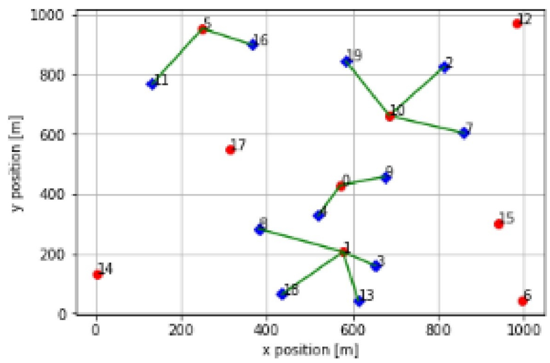

***Fig. 9**: The best gateway locations with D_{max} = 300 m and β_i = \$250.*

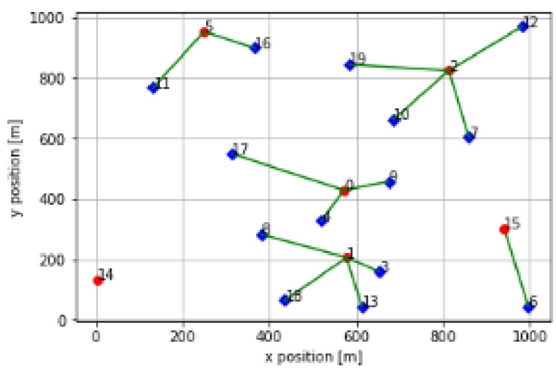

***Fig. 10**: The best gateway locations with D_{max} = 300 m and β_i = \$500.*

Recall that, m is the number of edges and n is the number of nodes (or vertices).

The cost of each gateway edge is listed in Tab. 4. Thus, the total (minimum) cost of gateway connection C^*_{con} is around 2218 m. Note that, the edge cost is equal to the distance between gateways i and j for the sake of our study. The link distance has a huge impact on the transmission delay which is one of the greatest challenges faced by mobile network operators. Other factors may be included in the gateway connection problem in practice such as the network survivability.

Although breaking a problem into subproblems may

***Table 3**: Optimal Network Costs of Different β_i.*

β_i (in \$US)	C^*_{net} (in \$US)	Runtime (in sec)
50	1200.00	0.26
100	2188.24	0.55
150	3009.71	3.22
200	3743.61	16.46
250	4281.28	13.76
300	4665.08	3.57
350	4965.08	1.17
400	5265.08	0.82
450	5565.08	0.50
500	5865.08	0.50

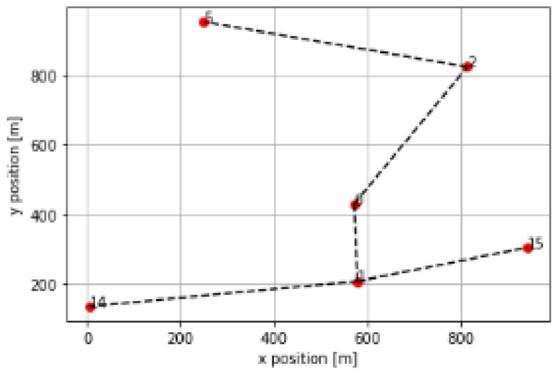

***Fig. 11**: A network of gateways.*

***Table 4**: Network Costs of Gateway Connection Problem.*

Gateway Edge (i, j)	Cost (γ_{ij})
(0, 1)	222.89
(1, 15)	375.59
(0, 2)	462.94
(2, 5)	557.96
(1, 14)	578.55

result in some loss of optimality, the topology design problem considered in this work is solvable with the fair time complexity. Furthermore, this work could be modified to offer a planning tool for not only cellular IoT but also several practical networks where other CAPEX/OPEX costs and constraints may be added in the topology design problem.

5. ML-BASED GATEWAY PLACEMENT

In this section, we aim to provide a fascinating insight into ML-based topology design. Unlike Sec. 4, the gateway location problem is solved by clustering algorithms instead. Because of the easy implementation, K-means and K-medoids clustering being the most common algorithm are selected to model the gateway placement

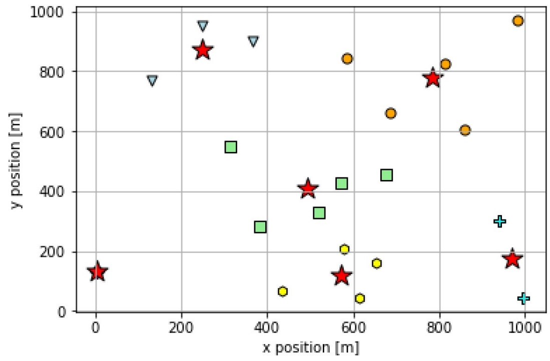

***Fig. 12**: K-means clustering-based gateway placement (N = 20 and k = 6).*

by means of Python with scikit-learn [24]. Denote the number of clusters by k. According to [25], a pseudocode of K-means and K-medoids clustering can be shown as Alg. 2. Note that, K-medoids clustering updates centroids with the partitioning around medoids (PAM) algorithm compared to K-means clustering. The previous centroid and non-centroid points are swapped. The point with a minimum loss (i.e. minimum dissimilarity to all other points in the cluster) is then considered as the new centroid.

Algorithm 2 K-means and K-medoids clustering

1: Initialization: Randomly select k centroids from sample points
2: Assignment: Find the Euclidean distance between each sample point and all centroids. Assign each sample point to the nearest centroid
3: **if** K-means is used **then**
4: Update: Move the centroids to the center of the assigned sample points from line 2
5: **else if** K-medoids is used **then**
6: Update: Update centroids with the PAM algorithm
7: **end if**
8: Repeat: Repeat lines 2-7 until the cluster assignments do not change

With the same node locations (i.e. $N = 20$) and the same number of clusters (i.e. $k = 6$) from the ILP-based gateway placement in Sec. 4, Fig. 12 and Fig. 13 show the gateway placement using K-means and K-medoids clustering, respectively. There are six clusters (i.e. green squares, orange circles, blue triangles, cyan pluses, magenta diamonds and yellow hexagons) with the six centroids (i.e. red stars) over the cell area, that is, the gateways should be placed in the centroid locations. In this way, multiple local networks can be formed by clustering algorithms.

Compared with the ILP-based gateway placement in Fig. 10, although K-means clustering can offer the similar local networks as shown in Fig. 12, all gateways (i.e. the final centroids) are not placed in the actual node locations (i.e. generated node locations in Fig. 4). For K-medoids

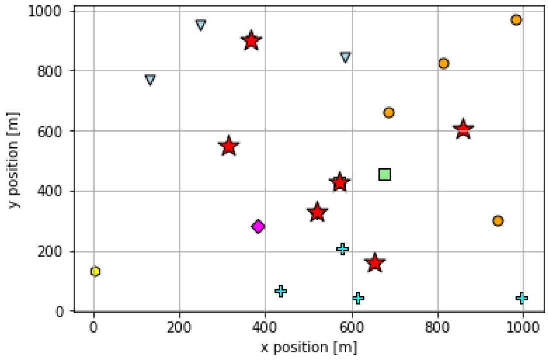

***Fig. 13**: K-medoids clustering-based gateway placement (N = 20 and k = 6).*

clustering, the gateway placement is obtained where all gateways are dropped in the actual node locations as illustrated in Fig. 13. Then, K-medoids clustering appears to be feasible for solving the gateway location problem in this study. The network cost C_{net} can be computed by the objective function in (6).

Let us consider the gateway placement based on K-medoids clustering with larger network sizes (i.e. N) in Fig. 14-17. Tab. 5 shows the network costs and the runtime values based on K-medoids clustering with different values of N and β_i = \$300. The same fixed costs (i.e. C_W and C_F) in Tab. 1 are also used. Note that, the gateway capacity K and the gateway-end device range D_{max} constraints are not considered here.

According to the results, although the number of nodes increases significantly, the gateway location problem can be solved with the low time complexity. Unlike the optimization approach, the computation time does not increase significantly as the network size increases. In other words, the time complexity in Tab. 5 may be considered as $O(1)$. The optimal number of clusters in K-medoids clustering could be determined by several methods such as elbow and silhouette methods. Note that, the quality of clustering is beyond the scope of this article, but a good summary with illustrative examples can be found in [25].

To evaluate the computational efficiency between ILP and K-medoids clustering, Tab. 6 provides the cost comparison between them for N = 20 resulting from multiple scenarios of potential gateway locations. The same parameters (i.e. β_i = \$300, C_W = 10 and C_F = 1) are used for both approaches while K = 5 and D_{max} = 300 are considered for only ILP approach. Obviously, ILP outperforms K-medoids clustering in terms of the network cost. However, the time complexity could be the issue of ILP if a large-scale problem is considered.

Therefore, K-medoids clustering may collaborate with the optimization approach to solve the gateway location problem in practice. To form a backbone network, the Kruskal algorithm in Alg. 1 can be used in connection with the K-medoids-based gateway placement.

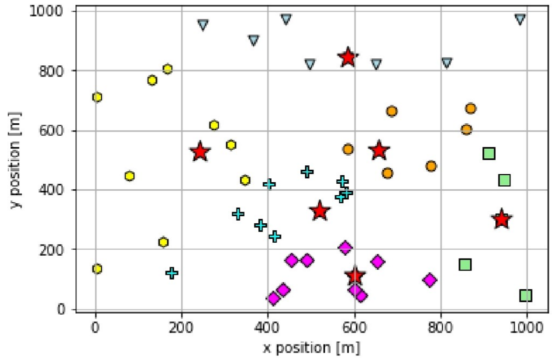

***Fig. 14**: K-medoids clustering-based gateway placement (N = 50 and k = 6).*

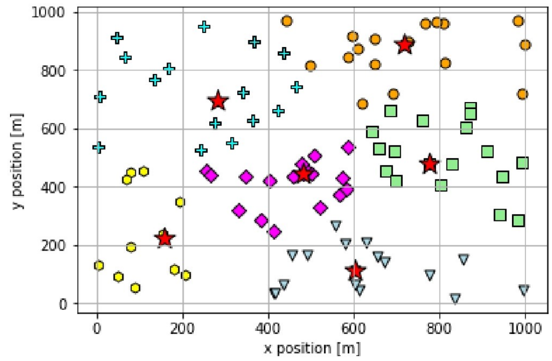

***Fig. 15**: K-medoids clustering-based gateway placement (N = 100 and k = 6).*

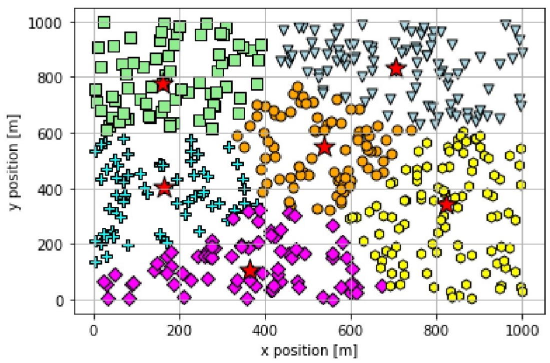

***Fig. 16**: K-medoids clustering-based gateway placement (N = 500 and k = 6).*

***Table 5**: Network Costs of K-Medoids Clustering-based Gateway Placement for different N*

N	C_{net} (in \$US)	Runtime (in sec)
20	5248.57	0.12
50	10148.04	0.11
100	17600.31	0.12
500	88227.30	0.13
1000	176661.04	0.16

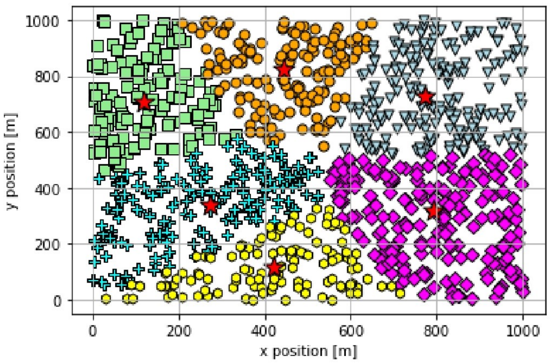

Fig. 17: *K-medoids clustering-based gateway placement (N = 1000 and k = 6).*

Table 6: *Network Costs between ILP and K-Medoids Clustering-based Gateway Placement (N = 20).*

Iteration	C^*_{net} (in $US)	C_{net} (in $US)
1	4373.98	4385.55
2	3987.50	4462.31
3	3856.66	4075.05
4	3913.33	4616.25
5	4011.82	5518.87
6	3998.43	4221.85
7	3903.91	4771.56
8	3820.86	3824.23
9	3954.28	4365.97
10	4665.08	5248.57
Average Cost	4048.59	4549.02

6. CONCLUSION

In this article, the topology design in cellular IoT was divided into two subproblems: gateway location and gateway connection problems. Both subproblems were formulated as the ILP problem to minimize the total network cost. First, the gateway location problem was solved by means of the optimization approach. According to the results, the best locations of gateways and the optimal network cost were obtained. Using suitable system parameters (i.e. D_{max} and β_i), this could result in both the topology design (i.e. local networks) and a significant decrease in C^*_{net}. Second, the gateway connection problem was solved by the Kruskal algorithm. A connection of gateways (i.e. a minimum spanning tree) was obtained with the minimum gateway connection cost C^*_{con}. In addition to the optimization approach, the gateway location problem was solved by clustering algorithms. K-medoids clustering could offer the gateway placement with the low time complexity. The cost comparison between ILP and K-medoids clustering-based gateway placement was then presented. For future work, to consider the real-world topology design problem, massive network elements, interference, channel effects and other environmental factors should be included in the problem formulation. The mathematical model could be then solved by means of optimization heuristics and machine learning.

REFERENCES

[1] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus, and M. Fallgren, “Scenarios for 5G mobile and wireless communications: the vision of the METIS project,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26-35, May 2014.

[2] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, “5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201-1221, Jun 2017.

[3] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, “Massive Access for 5G and Beyond,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 615-637, Mar 2021.

[4] S. K. Sharma and X. Wang, “Toward Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 426-471, 2020.

[5] A. Hoglund, X. Lin, O. Liberg, A. Behravan, E. A. Yavuz, M. Van Der Zee, Y. Sui, T. Tirronen, A. Ratilainen, and D. Eriksson, “Overview of 3GPP Release 14 Enhanced NB-IoT,” *IEEE Network*, vol. 31, no. 6, pp. 16-22, Nov 2017.

[6] M. Hasan, E. Hossain, and D. Niyato, “Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches,” *IEEE Communications Magazine*, vol. 51, no. 6, pp. 86-93, Jun 2013.

[7] M. Chen, Y. Miao, Y. Hao, and K. Hwang, “Narrow Band Internet of Things,” *IEEE Access*, vol. 5, pp. 20 557-20 577, 2017.

[8] J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, “Narrowband Internet of Things: Implementations and Applications,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2309-2314, Dec 2017.

[9] GSMA, “LTE-M Deployment Guide to Basic Feature Set Requirements,” Jun 2019. [Online]. Available: https://www.gsma.com/iot/wpcontent/uploads/2019/08/201906-GSMA-LTEM-Deployment-Guide-v3.pdf

[10] GSMA, “NB-IoT Deployment Guide to Basic Feature Set Requirements,” Jun 2019. [Online]. Available: https://www.gsma.com/iot/wpcontent/uploads/2019/07/201906-GSMA-NBIoT-Deployment-Guide-v3.pdf

[11] A. Taufique, M. Jaber, A. Imran, Z. Dawy, and E. Yacoub, “Planning Wireless Cellular Networks of

Future: Outlook, Challenges and Opportunities," *IEEE Access*, vol. 5, pp. 4821-4845, 2017.

[12] 5G-PPP, "Deliverable D1.2 Quantitative technoeconomic feasibility assessment," Sep 2017. [Online]. Available: https://metis-ii.5g-ppp.eu/wpcontent/uploads/deliverables/METISII_D1.2_V1.1.pdf

[13] M. I. Hossain and J. I. Markendahl, "Comparison of LPWAN Technologies: Cost Structure and Scalability," *Wireless Personal Communications*, vol. 121, no. 1, pp. 887-903, Nov 2021.

[14] R. I. Rony, E. Lopez-Aguilera, and E. GarciaVillegas, "Cost Analysis of 5G Fronthaul Networks Through Functional Splits at the PHY Layer in a Capacity and Cost Limited Scenario," *IEEE Access*, vol. 9, pp. 8733-8750, 2021.

[15] P. Saengudomlert, *Optimization for Communications and Networks*. CRC Press, 2011.

[16] A. Sripetch and P. Saengudomlert, "Topology Design of Optical Networks Based on Existing Power Grids," in *Fifth Annual Conference on Communication Networks and Services Research (CNSR '07)*, 2007, pp. 35-40.

[17] A. Sripetch and P. Saengudomlert, "Optimization for Optical Network Designs Based on Existing Power Grids," *IEICE Transactions on Communications*, vol. E91-B, no. 3, pp. 689-699, Mar 2008.

[18] X. Zhu, M. H. F. Wen, V. O. K. Li, and K.-C. Leung, "Optimal PMU-Communication Link Placement for Smart Grid Wide Area Measurement Systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4446-4456, Jul 2019.

[19] G. Wang, Y. Zhao, Y. Ying, J. Huang, and R. M. Winter, "Data Aggregation Point Placement Problem in Neighborhood Area Networks of Smart Grid," *Mobile Networks and Applications*, vol. 23, no. 4, pp. 696-708, Aug. 2018.

[20] J. L. Gallardo, M. A. Ahmed, and N. Jara, "Clustering Algorithm-Based Network Planning for Advanced Metering Infrastructure in Smart Grid," *IEEE Access*, vol. 9, pp. 48992-49006, 2021.

[21] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Pearson, 1992.

[22] R. Rardin, *Optimization in Operations Research*, 2nd ed. Pearson, 2016.

[23] Pulp, "Optimization with PuLP - PuLP 2.6.0 documentation." [Online]. Available: https://coin-or.github.io/pulp/index.htm

[24] S. learn developers, "scikit-learn Machine Learning in Python." [Online]. Available: https://scikit-learn.org/stable/

[25] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2nd ed. Packt Publishing, 2017.

Nakrop Jinaporn is an assistant professor (communications engineering) in the Department of Electrical and Electronic Engineering at Ubon Ratchathani University, Thailand. He graduated with a B.Eng. (Electrical Engineering) from Ubon Ratchathani University in 2008. He obtained an M.Eng. (Telecommunications) from Asian Institute of Technology (AIT), Thailand, in 2010 and a Ph.D. (Communications) from the University of Bristol, UK, in 2020. His research interest is wireless resource allocation and machine learning for 5G and beyond.