IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Optimization of Resource Management for NFV-Enabled IoT Systems in Edge Cloud Computing

TUAN-MINH PHAM[1,2], THI-THUY-LIEN NGUYEN[3,4]
[1]Faculty of Computer Science, Phenikaa University, Hanoi 12116, Vietnam
[2]Phenikaa Institute for Advanced Study (PIAS), Phenikaa University, Hanoi 12116, Vietnam
[3]VNU-University of Engineering and Technology, Hanoi 100000, Vietnam
[4]Hanoi National University of Education, Hanoi 100000, Vietnam

Corresponding author: Tuan-Minh Pham (e-mail: minh.phamtuan@phenikaa-uni.edu.vn).

ABSTRACT The Internet of Things (IoT) has been envisioned as an enabler of the digital transformation that can enhance different features of people's daily lives, such as healthcare, home automation, and smart transportation. The vast amount of data generated by a massive number of devices in an IoT system could lead to a severe performance problem. Edge cloud computing and network function virtualization (NFV) technologies are potential approaches to improve the efficiency of resource use and the flexibility of responsive services in an IoT system. In this paper, we consider the joint optimization problem of gateway placement and multihop routing in the IoT layer, the problem of service placement in the edge and cloud layers of an NFV-enabled IoT system in edge cloud computing (NIoT). We propose three optimization models (i.e., GMO, SP1O, SP2O) that allow an IoT service provider to find the optimal deployment of gateways, the optimal resource allocation for service functions, and the optimal routing according to a cost function with a performance constraint in a NIoT system. We then develop three approximation algorithms (i.e., GMA, SP1A, SP2A) for tackling the problems in a large-scale NIoT system. The evaluation results under a set of scenarios with various topologies and parameters show that the approximation algorithms can obtain results close to the optimal solution with a significant reduction in computation time. We also derive new insights into the strategy for an IoT provider to optimize its objectives. Specifically, the results suggest that an IoT provider should select an appropriate service placement strategy with regard to a charging agreement with an NFV infrastructure provider, and only deploy service functions with a strict delay requirement on the edge of networks for optimizing its cost.

INDEX TERMS NIoT, resource management, optimization, NFV-enabled IoT systems, edge cloud computing

I. INTRODUCTION

THE Internet of Things (IoT) as the interconnection of a set of things (e.g., humans, actuators, sensors) over the Internet has been envisioned as an enabler of the digital transformation that can enhance different features of people's daily lives such as healthcare, home automation, and smart transportation. For example, an IoT based smart transportation system, which uses data collected from numerous sensors and processed by several service functions deployed in the cloud, can resolve many problems such as traffic congestion, traffic accident prediction, and the scarcity of car parking facilities. With an estimated number of 41.6 billion devices interconnected by 2025, the enormous amount of data created by those devices needs to be transmitted, stored, and processed in a specific time requirement for providing responsive applications [1]. Hence, the design of an IoT system with the efficiency of resource use and the flexibility of responsive services is strongly desired.

Edge cloud computing and network function virtualization (NFV) technologies are potential approaches to improve resource use efficiency and highly flexible services in an IoT system by moving computing resources to the edge of networks close to IoT nodes [2]–[4]. Further, the adoption of NFV can provide a high degree of dynamic elasticity of IoT services due to the high versatility in the location and position of a particular service function composing an IoT service. This paper aims to develop optimization models and algorithms to provide efficient usage of resources and energy for an NFV-enabled IoT system in edge cloud computing (NIoT).

More specifically, we consider a NIoT system composed of three layers: the IoT layer, the edge layer, and the cloud layer. Data generated by sensors at the IoT layer is routed through an IoT gateway to the edge and cloud layers for being processed by service functions. We take into account the support of multihop routing at the IoT layer for efficient data communication. In such a NIoT system, the challenging questions are the following: What is the optimal location of gateway nodes? What is a routing solution with a performance guarantee in a NIoT system with the support of multihop communication? What is the optimal location of service functions at the edge and cloud layers to minimize the computing and energy cost? We aim at addressing those questions as an essential part of designing a high-performance, flexible, and responsive NIoT system.

A detailed discussion of the literature on the use of NFV for IoT in edge cloud computing has been provided in Section II. As discussed in Section II, much of the existing work has investigated the integration of IoT and edge cloud computing [2], [4]–[10]. Some works have considered the performance of an IoT system based on NFV and edge cloud computing [3], [11]. However, none of these works have addressed the optimization problem of resource management, taking into account multihop routing and service functions chaining for the energy efficiency, efficient resource use, and high flexibility of a NIoT system.

The main contributions of the paper are as follows:

- We introduce two optimization problems of resource management for NFV-enabled IoT systems in edge cloud computing: the joint optimization problem of gateway placement and multihop routing at the IoT layer, the problem of service placement at the edge and cloud layers. Our proposed optimization models (i.e., GMO, SP1O, SP2O) allow us to determine the optimal location of gateways, optimal routing, and optimal service placement according to a cost function with a performance guarantee represented by the maximum number of relays.
- We propose approximation algorithms for tackling the problems in a large-scale system. The approximation solutions for the gateway placement, routing, and service placement are very close to the optimal solutions.
- The evaluation results present some useful insights into the optimization of computing and energy costs related to IoT providers' deployment strategy. Specifically, a charging agreement with an NFV infrastructure provider has a significant impact on the IoT provider's optimization objective. An IoT provider should only deploy service functions with a strict delay requirement on the edge of networks for minimizing its cost.

The rest of this paper is organized as follows. Section II presents an overview of related works. In Section III, we describe the evolution of IoT systems in resource management from a physically isolated system to an NFV-enabled IoT system in edge cloud computing. In Section IV, we present the details of an IoT system based on NFV in edge cloud computing and define the optimization problems of gateway placement, routing, and service placement in the system. In Section V and VI, we propose three mixed-integer linear programming (MILP) models to obtain the optimal solutions for the problems previously described. Section VII presents our proposed approximation algorithms for addressing the problems in a large-scale NIoT system. Section VIII shows the evaluation results for the optimization models and approximation algorithms. Finally, we conclude the paper in Section IX.

II. RELATED WORK

Massive data generated by multiple sensors need more processing in remote server applications for a wide variety of intelligent functions. An IoT system can gain the practically infinite resources from the cloud to compensate its small storage and limited processing capability when IoT functions are implemented on the cloud. Resource management for such a Cloud-IoT system has been studied extensively. For example, Mitton et al. propose an infrastructure design of a Cloud-IoT system for smart cities [12]. He et al. present a cloud platform of IoT-based vehicular data for intelligent parking and a vehicular data mining service [13]. In [14], Botta et al. provide a survey of researches on the integration of Cloud computing and IoT. While these proposals produce a performance advantage in completion times and energy costs, they cannot obtain the minimum energy consumption and responsive time.

When we explore new IoT applications with big data and real-time requirements, the virtues of proximity become more critical. The edge computing paradigm provides a promising solution to enhancing service quality and energy consumption by offloading computation tasks to multiple edge nodes close to consumers. Several recent studies have been dedicated to resource management problems in edge cloud computing for IoT by investigating various critical problems. For instance, Lan et al. propose an IoT access framework focused on edge computing that allows the exposure of massive devices and resource capacity as a single unified interface [2]. Xu et al. propose a computation offloading method for dynamic task scheduling in an IoT system based on cloud edge computing to improve the completion time and save the energy consumption for mobile devices [5]. Kherraf et al. study optimization models and algorithms for resource allocation and workload assignment in IoT networks concentrated on mobile edge computing (MEC) [6]. Mehrabi et al. show that device-to-device (D2D) communication can be exploited in MEC for computation offloading and content caching [15]. However, it requires an appropriate amount of resources available at end nodes. In another direction, some authors use machine learning techniques to improve throughput and reduce the amount of transmitted data in an

IoT system based on edge cloud computing [7], [8].

Recently, Zhao et al. propose an approximation algorithm for the placement problem of IoT services, which concerns the decision of where to place multiple IoT functions in edge cloud computing according to their requirements of service quality [9]. In [10], the authors investigate an optimization model for addressing the service placement problem. However, as the model is nonlinear, it is time-consuming to find the optimal solution. These studies propose various approaches for addressing different resource management problems in an IoT system based on edge cloud computing. While edge cloud computing enables responsive functions in an IoT system by the virtues of proximity, it is not able to provide a true service overlay, which can be supported by NFV due to the capacity of chaining service functions.

NFV is a network architecture paradigm in which a communication service can be created by chaining various blocks of network functions (e.g., middle-box functions) scattered over numerous data centers. Researchers have recently considered many problems in NFV, including resource allocation, service function chaining (SFC), and routing optimization [16]–[20]. Within the research literature, various topics have also been explored, highlighting how future IoT networks should use NFV. For example, Wang et al. suggest NFV with multiflow transmissions in an IoT environment to establish a network slice [11]. The same goals refer to Mouradian et al. [3]. The aim, however, is to design the distributed IoT gateway for on-the-fly disaster management with NFV and SDN technologies. Differently, Fu et al. build an NFV controlled IoT platform, which separates large VNFs into simple VNF components and uses machine learning for robust SFC integration [21].

However, no existing research has focused on the optimization problem of resource management for NFV-enabled IoT systems in edge cloud computing, which takes into account the feature of service functions chaining for resource use efficiency and flexibility of responsive services thanks to virtualization techniques in NFV and edge cloud computing. This paper is an extended version of our work presented at the 6th NAFOSTED Conference on Information and Computer Science (NICS 2019) [4]. In [4], we consider the resource management at the IoT layer for delivering data from sensors to IoT gateways. In this work, we provide novel results of optimization models and algorithms for resource management in a NIoT system, taking into consideration optimal resource allocation at both the edge and the cloud layers, and the service function chaining for optimizing computing cost and energy cost under various performance and resource constraints.

III. EVOLUTION OF IOT SYSTEMS IN RESOURCE MANAGEMENT

IoT is generally characterized by real-world small things, widely distributed, with limited storage and processing capacity. Due to the availability of virtually unlimited storage and processing capabilities at low cost in the cloud, many IoT service providers widely adopted a cloud computing model for delivering IoT services over the Internet. In this section, we start by discussing resource management in the cloud for IoT systems, including advantages, architectures, and issues. We then analyze the characteristics of NFV, which support edge cloud computing in IoT systems.

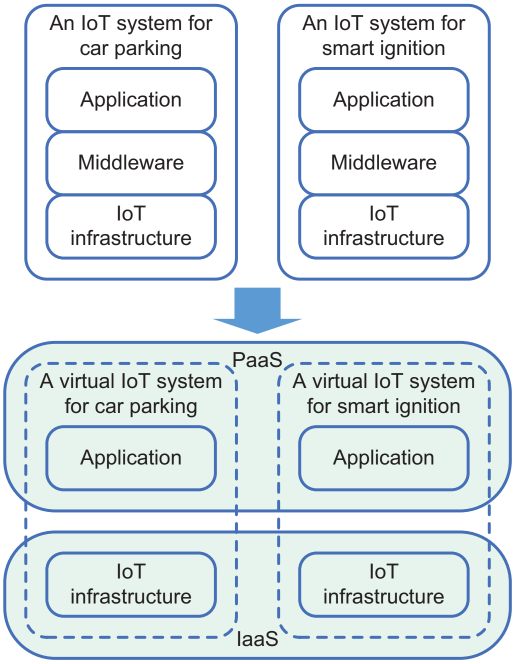


FIGURE 1: Evolution of IoT systems from a physically isolated system to a Cloud-IoT system.

A. THE DEVELOPMENT OF IOT SYSTEMS FROM PHYSICALLY ISOLATED SYSTEMS TO CLOUD COMPUTING

IoT services have been offered in single domains, such as car parking systems, smart ignition systems, and smart home [22]. Domain-specific or project-specific specifications define the implementation of all components in these systems, from sensors and actuators to smart service modules. While this service delivery model based on single domains has guided the development of providing IoT services over the past several years, it leads to many geographically separated vertical structures in which hardware, networks, and application logics are tied directly. Cloud can offer an efficient resource management system for IoT infrastructure as virtualized cloud resources can be rented on-demand and delivered as general utilities.

The cloud infrastructure systems are usually available for users in one of three service models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In particular, IaaS offers computing resources as a service. PaaS contains operating systems and application systems as well as other elements of the system (e.g., database and file system). SaaS means that the provider offers the software on the common platforms.

IoT service providers may implement a domain-independent PaaS framework that provides essential cloud infrastructure for IoT services. In such a PaaS framework, an IoT service in various application domains can be allocated without a constraint on specific application logics (Fig. 1).

B. NFV SUPPORT FOR EDGE CLOUD COMPUTING IN IOT SYSTEMS

When we explore new IoT applications with big data and real-time requirements, the virtues of proximity become critical. Edge computing, where resources are placed at the network edge close to service consumers, has attracted much attention in recent years. Edge cloud computing offers highly responsive, scalable, and reliable services for IoT. Specifically, the physical proximity of IoT services to service consumers first allows the achievement of reduced end-to-end delay and low bandwidth in cloud-based applications. It is valuable for high responsive services such as smart transportation, healthcare monitoring, and quality control in factory automation that offload computation to the edge. Second, when the raw data is analyzed at the edge, the extracted information required to be transmitted to the cloud is significantly lower. Third, a backup service at the edge will adequately cover a failure if a cloud service becomes inaccessible due to network failure or server collapse.

The use of edge cloud computing with NFV is a potential approach to flexible, efficient, and responsive IoT services. While edge computing allows the achievement of highly responsive services, NFV supports a high degree of dynamic elasticity of IoT services due to the high versatility in the location and position of a particular service function composing the IoT service. By the support of SFC in NFV, we can create a new service and update an existing service at rapid rates. Those services can be allocated resources on the fly in an automated fashion. In addition to the use of establishing a service path from the service chain, an essential feature of SFC is that it gives the provider a flexible approach for adding missing functionality to the highly integrated solution set.

This work enforces the added values of NFV technology in edge cloud computing for highly responsive IoT services, and scalability and flexibility of service composition. In particular, the Network Functions Virtualization Infrastructure as a Service (NFVIaaS) can support generic IaaS computing loads, including cloud-based applications (IoT applications) and network functions. It also allows us to establish connectivity dynamically (e.g., NaaS) among virtual functions for creating a new SFC. The services provided by the NFVIaaS should be available across providers for cost-efficiency. Figure 2 shows an example of end-end IoT services in an NFV-enabled IoT system across providers. In the figure, Provider 2 runs IoT functions on the NFVI of Provider 1 by a contractual service agreement between them. Provider 2 can combine its instances running on its own NFV infrastructure and its instances running on Provider 1's NFV Infrastructure into an SFC to create an end-to-end service.

Our work considers the problem of resource management in an edge cloud computing model in NFV-enabled IoT systems composed of three layers: the cloud layer, the edge layer, and the IoT layer (Fig. 3). We aim to design models and algorithms for coordinating the resources and networks needed to set up cloud-based services and applications, which can be located at the edge and cloud layers. The optimization models and algorithms can be deployed as a component of the NFV Orchestrator, a functional block of NFV MANO developed by ETSI for the management and orchestration of all virtual resources.

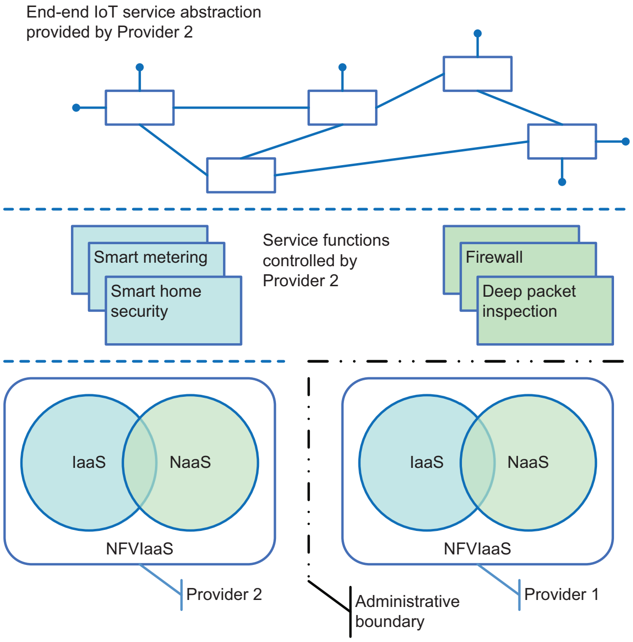


FIGURE 2: End-end IoT services in an NFV-enabled IoT system across providers.

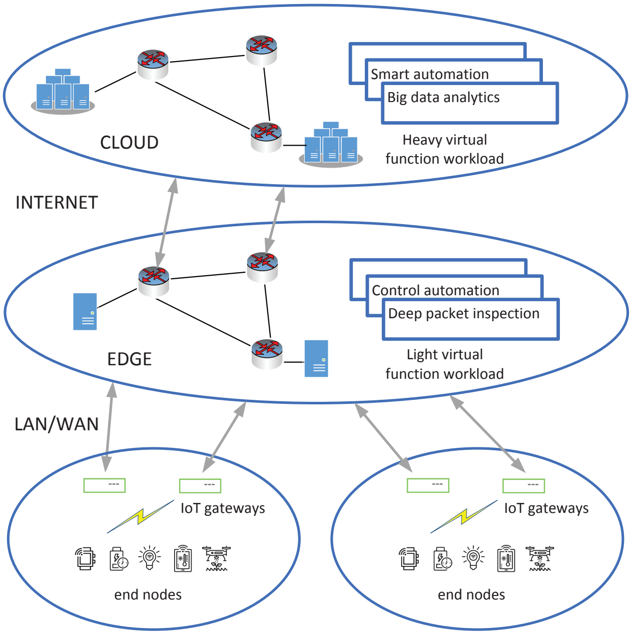


FIGURE 3: An edge cloud computing model in NFV-enabled IoT systems.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3026711, IEEE Access

TABLE 1: Summary of notations

Input parameters	
$G(V,E)$	A NIoT system where V is a set of nodes and E is a set of directed links. $V = V_S \cup V_K \cup V_Q$ where V_S is a set of IoT devices, V_K is a set of edge nodes, and V_Q is a set of cloud nodes.
η	The maximum relays used in multihop routing
ς_i	The cost for deploying node $i \in V_S$ as an IoT gateway
e_{ij}	If node $i \in V_S$ and node $j \in V_S$ have a direct link, $e_{ij} = 1$, otherwise $e_{ij} = 0$
e_{kq}	The bandwidth capacity of link (k, q), $(k \in V_K, q \in V_Q)$
F_K	A set of service functions deployed at edge nodes
F_Q	A set of service functions deployed at cloud nodes
ω_f	The number of computing resources required for providing function f for one unit of traffic
τ_{fi}	The cost of providing function f deployed at node i for one unit of traffic
c_k	The computing capacity of edge node k, $k \in V_K$
c_q	The computing capacity of cloud node q, $q \in V_Q$
b_g	The total traffic passing IoT gateway $g \in V_S$
Output variables	
x_i	A binary variable that represents a solution for the deployment of IoT gateways. If node i is an IoT gateway, $x_i = 1$, otherwise $x_i = 0$
z_{vg}	A binary variable that represents a solution for the gateway selection of end node v at the IoT layer. If data generated by v are routed to gateway g, $z_{vg} = 1$, otherwise $z_{vg} = 0$.
y_{ij}^{vg}	A binary variable that represents a routing solution at the IoT layer. If a link from node i to node j is used for the data flow from node v to node g, $y_{ij}^{vg} = 1$, otherwise $y_{ij}^{vg} = 0$.
r_{gkq}	A binary variable that represents a solution to service placement in cloud edge computing. If a data flow generated by node g is processed by node $k \in K$ and node $q \in Q$, $r_{gkq} = 1$, otherwise $r_{gkq} = 0$.
u_k	A binary variable that represent the state of power consumption of an edge node. If edge node k is active, $u_k = 1$, otherwise $u_k = 0$.
u_q	A binary variable that represent the state of power consumption of a cloud node. If cloud node q is active, $u_q = 1$, otherwise $u_q = 0$.

IV. SYSTEM DESCRIPTION

In this section, we formally describe a NIoT system and state the research problem. In a NIoT system, edge nodes and cloud nodes are NFVI nodes deployed at the edge and the cloud layers, respectively (Fig. 3). IoT nodes are nodes attached to the IoT layer. We classify IoT nodes into two types: end nodes and IoT gateways. End nodes are devices with capacity limitations such as sensors and actuators typically fitted with simple functions, i.e., collecting and delivering data to their gateways. IoT gateways, called gateways for short, are responsible for gathering data from end nodes and maintaining a stable link to several service functions deployed in the edge and cloud layers.

We represent a NIoT system by a directed graph $G(V, E)$. $V = V_S \cup V_K \cup V_Q$ is a set of nodes in the NIoT system, where V_S is a set of IoT devices, V_K is a set of edge nodes, and V_Q is a set of cloud nodes. $E = \{e_{ij}\}$ $(i, j \in V)$ is a set of links in the NIoT system. Network congestion rarely happens at the IoT layer but on the path from a gateway to the cloud. It comes from the fact that an amount of data increase significantly after the data are gathered at a gateway. Hence, we only consider the bandwidth capacity of a link among nodes among a gateway, an edge node, and a cloud node. At the IoT layer, if node i and node j have a direct link, $e_{ij} = 1$, otherwise $e_{ij} = 0$. In a NIoT system with a massive number of sensors, it is crucial to consider the hop-by-hop communication at the IoT layer for efficient data transmission. Since we focus on optimizing resource management at the software level rather than the physical level, our system model does involve several physical factors of IoT, such as wireless low-power technologies and data transmission at the IoT embedded device's hardware level. We denote the maximum relays used in multihop routing at the IoT layer by η. Let F_K and F_Q denote a set of service functions deployed at the edge and cloud layers, respectively. We define c_k to be the computing capacity of edge node k, $k \in V_K$. c_q is the computing capacity of cloud node q, $q \in V_Q$. We summarize the main mathematical notations in Table 1.

In a NIoT system, data are collected from end nodes to IoT gateways. The data then are routed to the edge layer and the cloud layer, depending on services requested from customers. We consider the resource management problems in the planning and operating stages. In the first stage, a provider wants to optimize the gateways' location for minimizing the deployment cost while fulfilling system requirements. In the second stage, a provider aims at optimizing the service placement for minimizing the operating cost while satisfying customer requests.

In the first optimization problem, we assume that data generated by an end node are required to be routed to a gateway. Given the support of multihop routing at the IoT layer, data generated by end nodes might pass across multiple relay nodes (i.e., end nodes) before entering a gateway. We suppose that the delay (i.e., the routing performance) is represented by a number of end nodes used as a relay along a routing path from an end node to a gateway. The deployment cost of a gateway node depends on where it is located. We denote by ς_i the cost for deploying node $i \in V_S$ as an IoT gateway. We state the joint optimization of gateway placement and routing as follows.

Problem 1 (Gateway placement and multihop routing (GM)): Given a set of nodes V_S, a set of links among these nodes, and the maximum delay η, find a solution of gateway placement and routing, satisfying constraints on routing and delay in order to minimize the deployment cost.

In the second optimization problem, we assume that b_g is the total traffic passing IoT gateway $g \in V_S$. Let ω_f be the number of computing resources required for providing function f for one unit of traffic. We denote by τ_{fi} the cost for providing function f deployed at node i for one unit of traffic. We consider two optimization problems of service placement with two different objectives: minimization of computing cost (SP1), minimization of energy cost (SP2). The problems are stated as follows.

Problem 2 (Service placement (SP1, SP2)): Given $G = (V, E)$ and a set of service functions deployed at the edge and cloud layers, find a solution of service placement, satisfying constraints on system capacity and services requested in order to minimize the computing cost (SP1) and the energy cost (SP2).

We will develop our solution for solving the GM, SP1, and SP2 problems in the next section.

V. OPTIMIZATION MODEL FOR GATEWAY PLACEMENT AND MULTIHOP ROUTING

We formulate the GM problem as a MILP model, called GMO (i.e., the GM optimization model) that enables us to achieve the optimal gateway placement and routing for minimizing the deployment cost in a NIoT system with the support of multihop routing. The variables are as follows:

- x_i is a binary variable that represents a solution for the deployment of IoT gateways. If node i is an IoT gateway, $x_i = 1$, otherwise $x_i = 0$.
- z_{vg} is a binary variable that represents a solution for the gateway selection of node v at the IoT layer. If data generated by v are routed to gateway g, $z_{vg} = 1$, otherwise $z_{vg} = 0$.
- y_{ij}^{vg} is a binary variable that represents a routing solution at the IoT layer. If a link from node i to node j is used for the data flow from node v to node g, $y_{ij}^{vg} = 1$, otherwise $y_{ij}^{vg} = 0$.

Definition 1 (Deployment cost): The formula for computing the deployment cost is given by:

$$\Psi_{GM} = \sum\nolimits_{i \in V_S} \varsigma_i x_i. \tag{1}$$

The GMO model is as follows:

Minimize $\quad \Psi_{GM}$

Subject to:

$$y_{ij}^{vg} \leqslant z_{vg}, \quad \forall v, g, i, j \in V_S \tag{2}$$

$$y_{ij}^{vg} \leqslant e_{ij}, \quad \forall v, g, i, j \in V_S \tag{3}$$

$$z_{vg} \leqslant 1 - x_v, \quad \forall v, g \in V_S \tag{4}$$

$$z_{vg} \leqslant x_g, \quad \forall v, g \in V_S \tag{5}$$

$$\sum\nolimits_{g \in V_S} z^{vg} = 1 - x_v, \quad \forall v \in V_S \tag{6}$$

$$\sum\nolimits_{i \in V_S} y_{ig}^{vg} \leqslant 1, \quad \forall v, g \in V_S \tag{7}$$

$$\sum\nolimits_{i \in V_S} y_{ig}^{vg} \geqslant z_{vg}, \quad \forall v, g \in V_S \tag{8}$$

$$\sum\nolimits_{i \in V_S} y_{vi}^{vg} \leqslant 1, \quad \forall v, g \in V_S \tag{9}$$

$$\sum\nolimits_{i \in V_S} y_{vi}^{vg} \geqslant z_{vg}, \quad \forall v, g \in V_S \tag{10}$$

$$\sum\nolimits_{j \in V_S} y_{ij}^{vg} = \sum\nolimits_{j \in V_S} y_{ji}^{vg}, \quad \forall v, g, i \in V_S, i \neq v, i \neq g \tag{11}$$

$$\sum\nolimits_{i,j \in V_S} y_{ij}^{vg} \leqslant \eta, \quad \forall v, g \in V_S. \tag{12}$$

We aim at optimizing the cost of gateway deployment while satisfying a requirement of routing performance represented by a maximum number of relays from an end node to its gateway. Conditions (2) and (3) assure that link (i, j) belongs to path from s to d (i.e., $y_{ij}^{sd} = 1$) only if data generated by end node s is routed through IoT gateway d (i.e., $z_{sd} = 1$) and link (i, j) exists (i.e., $e_{ij} = 1$). Conditions (4) and (5) guarantee that data is routed from v to g only if v is an end node (i.e., $x_v = 0$) and g is an IoT gateway (i.e., $x_g = 1$). Condition (6) assures that an end node sends data to one gateway. Conditions (7), (8), (9) and (10) ensure that the number of paths routing data from an IoT sensor to an IoT gateway is one. The constraint on a flow conservation guarantee for each routing path is given by (11). Condition (12) is the delay constraint represented as the maximum number of relays used to send data from an end node to a gateway. The MILP model's output is the optimal solution for gateway placement and multihop routing represented by x_i and y_{ij}^{sd}.

VI. OPTIMIZATION MODEL FOR SERVICE PLACEMENT

A. MINIMIZATION OF THE COMPUTING COST

We formulate the SP1 problem as a MILP model, namely SP1O (i.e., the SP1 optimization model), to find the optimal solution to service placement in edge cloud computing with the objective of minimizing the computing cost. We represent a solution to the problem by binary variable r_{gkq}. If a data flow generated by IoT gateway g is processed by node k in the edge and node q in the cloud, $r_{gkq} = 1$, otherwise $r_{gkq} = 0$.

A number of computing resources required for providing function f at the edge layer when data traffic is routed from gateways to edge node k are given by:

$$\psi_{fk} = \omega_f \sum\nolimits_{g \in N, q \in Q} b_g r_{gkq}. \tag{13}$$

A number of computing resources required for providing function f at the cloud layer when data traffic is routed from gateways to cloud node q are given by:

$$\psi_{fq} = \omega_f \sum\nolimits_{g \in N, k \in K} b_g r_{gkq}. \tag{14}$$

Definition 2 (Computing cost): The formula for calculating the computing cost is given by:

$$\Psi_{SP1} = \sum\nolimits_{k \in V_K, f \in J_K} \tau_{fk} \psi_{fk} + \sum\nolimits_{q \in V_Q, f \in J_Q} \tau_{fq} \psi_{fq}. \tag{15}$$

The SP1O model is as follows:

Minimize Ψ_{SP1}

Subject to:

$$\sum_{k\in V_K, q\in V_Q} r_{gkq} = 1, \quad \forall g \in V_S \tag{16}$$

$$\sum_{g\in V_S} b_g r_{gkq} \leqslant e_{kq}, \quad \forall k \in V_K, q \in V_Q \tag{17}$$

$$\sum_{f\in J_K} \psi_{fk} \leqslant c_k, \quad \forall k \in V_K \tag{18}$$

$$\sum_{f\in J_Q} \psi_{fq} \leqslant c_q, \quad \forall q \in V_Q. \tag{19}$$

The objective of SP1O is to minimize the usage cost of computing resources for realizing service requirements. Condition (16) assures that one edge node and one cloud node are selected for processing data traffic collected at an IoT gateway. Condition (17) guarantees that data traffic routed through a link between edge node k and a cloud node q is not more than link capacity e_{kq}. Conditions (18) and (19) present the constraints on the computing capacity of an edge node and a cloud node.

B. MINIMIZATION OF THE ENERGY COST

We further develop the SP1O model for finding the optimal solution to service placement in edge cloud computing with the objective of minimizing the energy cost, called SP2O (i.e., the SP2 optimization model). A solution to the SP2 problem is represented by a binary variable r_{gkq}, which was explained in the SP1O model. The energy usage depends on the number of active nodes of the edge and cloud layers. If a node of the edge and cloud layers provides a service function for a data flow from the IoT layer, its state is active, otherwise its state is inactive. We ignore the power consumption in the inactive state as it is a negligible quantity, compared with that in the active state. The objective of minimizing the energy cost can be represented as the number of active nodes in the edge and cloud layers. To represent the state of a node in the edge and cloud layers, we introduce binary variables u_k and u_q, respectively. If edge node k is active, $u_k = 1$, otherwise $u_k = 0$. If cloud node q is active, $u_q = 1$, otherwise $u_q = 0$. To describe constraints on uq and u_k, we define θ as a large integer number.

Definition 3 (Energy cost): The formula for computing the energy cost is given by:

$$\Psi_{SP2} = \sum_{k\in V_K} u_k + \sum_{q\in V_Q} u_q. \tag{20}$$

The SP2O model is as follows:

Minimize Ψ_{SP2}

Subject to:

Condition (16), (17), (18), (19)

$$\sum_{g\in V_S, q\in V_Q} r_{gkq} \leqslant \theta u_k, \quad \forall k \in V_K \tag{21}$$

$$\sum_{g\in V_S, q\in V_Q} r_{gkq} \geqslant u_k, \quad \forall k \in V_K \tag{22}$$

$$\sum_{g\in V_S, k\in V_K} r_{gkq} \leqslant \theta u_q, \quad \forall q \in V_Q \tag{23}$$

$$\sum_{g\in V_S, k\in V_K} r_{gkq} \geqslant u_q, \quad \forall q \in V_Q. \tag{24}$$

In the SP2O model, the constraints on the fulfilment of service requirement and the system capacity are similar to those used in the SP1O model (i.e., Eq. (16), (17), (18), (19)). Conditions (21) and (22) assure that edge node k will be in the active state if k is selected for processing data traffic collected at any gateway, otherwise k will be in the inactive state. Conditions (23) and (24) guarantee that cloud node q will be in the active state if q is selected for processing data traffic collected at any gateway, otherwise q will be in the inactive state.

VII. APPROXIMATION ALGORITHMS

In the previous section, we develop three MILP models for finding the optimal location of IoT gateways, the optimal routing, and the optimal placement of service functions in a NIoT system. However, the MILP solvers often fail to solve a large model with hundreds of gateways. For example, for a scenario with 300 IoT devices, GMO has tens of billions of variables, which is too large for CPLEX to handle. Hence, we propose an approximation algorithm for a large-scale NIoT system. We start by describing the primary steps of the algorithm. We then present some adaptations for solving the two optimization problems of resource management in a NIoT system.

A. PRIMARY STEPS

The key concept of the algorithm approach is based on the Simulated Annealing (SA) with the development of a neighborhood function and a solution representation for the resource management problems in a NIoT system. SA is a heuristic approach to a search of the global optimum for the optimization problem whose solution set contains a local optimum [23]. It considers a worse scenario with a certain probability in the searching procedure for the optimal solution. This approach has the advantage of being simple and effective due to the capacity for escaping from local optimum.

The algorithm starts with a temperature parameter T decreasing after some steps in the searching procedure by a cooling function $C(T)$. It loops until T is less than a stop temperature T_f. We denote by M the number of iterations for each T. Let S be an initial solution. The details of the search procedure are as follows:

- Step 1: Initialize a set of algorithm parameters including T, T_f, and a counter variable $n = 1$ that represents a number of iterations for T. Find an initial solution S.
- Step 2: Generate a neighborhood solution $S^{'}$ from the current solution S. If the objective function value of the neighborhood solution $\Phi(S')$ is less than that of the current solution $\Phi(S)$, move to a better solution $S \leftarrow S'$ and go to Step 4. Otherwise, go to Step 3.
- Step 3: Let $\Delta = \Phi(S') - \Phi(S)$. ε is a random number uniformly distributed on the interval (0,1). If $\varepsilon < \exp(-\Delta/T)$, move to a new solution $S \leftarrow S^{'}$ and go to Step 4.
- Step 4: Increase a number of iterations for T, $n \leftarrow n+1$. If $n > M$, go to Step 5. Otherwise, continue the loop from Step 2.
- Step 5: Use $C(T)$ to update the current temperature: $T \leftarrow C(T)$. If $T \geq T_f$, $n \leftarrow 1$ and go to Step 2. Otherwise, finish the searching procedure.

B. ALGORITHM FOR NEIGHBORHOOD GENERATION OF GATEWAY PLACEMENT AND MULTIHOP ROUTING

We represent a solution of the gateway placement in a NIoT system at the IoT layer by a set of integers (i.e., $S_{GM} = \{g \in V_S\}$). For example, $S_{GM} = \{1, 2, 3\}$ means the positions of gateways are 1, 2, and 3 at the IoT layer, the positions of end nodes are $V_S \backslash S_{GM}$. We propose a neighborhood function, namely GMA-N, to move from a solution S_{GM} to a neighborhood solution $S^{'}_{GM}$. The SA algorithm for finding an approximation solution of the problem of gateway placement and multihop routing, called GMA, follows the necessary steps presented in Section VII-A and uses our proposed neighborhood function GMA-N.

In GMA-N, we propose three moving operators that allow us to change the current solution to a neighborhood solution:

- Add$(v, S^{'}_{GM})$: The procedure adds a new gateway to the current solution by inserting a new integer $v \in V_S \backslash S^{'}_{GM}$ into $S^{'}_{GM}$.
- Remove$(g, S^{'}_{GM})$: The procedure removes a gateway in the current solution by deleting one integer $g \in S^{'}_{GM}$ from S_{GM}.
- Exchange$(g, v, S^{'}_{GM})$: The procedure moves one gateway to a new location by changing one integer $g \in S^{'}_{GM}$ to another value that is not in S_{GM} (i.e., $v \in V_S \backslash S^{'}_{GM}$).

The details of the GMA-N algorithm for neighborhood generation of gateway placement and multihop routing are presented in Algorithm 1. The rules for selecting a moving operator are as follows. If the number of gateways is one, Add() or Exchange() is allowed to operate with a probability depending on a probability parameter γ (i.e., line 7-10). If all IoT nodes are selected as gateways (i.e., $\left|S^{'}_{GM}\right| = |V_S|$), Remove() is selected (i.e., line 11-13). If the number of gateway is more than one and less than the number of IoT nodes (i.e., $1 < \left|S^{'}_{GM}\right| < |V_S|$, one of three operators is selected with a probability depending on probability parameters α and β (i.e., line 14-18). Note that $S^{'}_{GM}$ is a feasible solution if there exists a routing solution for delivering data from all end nodes to gateways. We use parameters α, β, γ to control the priority of moving operators in generating a neighborhood solution.

Algorithm 1 Algorithm for neighborhood generation of gateway placement and multihop routing

```
 1: function GMA-N($S_{GM}$)
 2:     $S^{'}_{GM} \leftarrow S_{GM}$
 3:     $\Sigma_{GM} \leftarrow \emptyset$
 4:     $\Pi_{GM} \leftarrow$ all pairs of gateways and end nodes in
            $S_{GM} \times \{V_S \backslash S_{GM}\}$
 5:     for all $(g, v) \in \Pi_{GM}$ do
 6:         $\varepsilon \leftarrow$ a random number in (0,1)
 7:         if $\left|S^{'}_{GM}\right| = 1$ then
 8:             $S^{'}_{GM} \leftarrow$ EXCHANGE$(g, v, S^{'}_{GM})$ if $\varepsilon < \gamma$
 9:             $S^{'}_{GM} \leftarrow$ ADD$(v, S^{'}_{GM})$ if $\varepsilon \geq \gamma$
10:         end if
11:         if $\left|S^{'}_{GM}\right| = |V_S|$ then
12:             $S^{'}_{GM} \leftarrow$ REMOVE$(g, S^{'}_{GM})$
13:         end if
14:         if $1 \leq \left|S^{'}_{GM}\right| \leq |V_S|$ then
15:             $S^{'}_{GM} \leftarrow$ EXCHANGE$(g, v, S^{'}_{GM})$ if $\varepsilon < \alpha$
16:             $S^{'}_{GM} \leftarrow$ ADD$(v, S^{'}_{GM})$ if $\varepsilon \in [\alpha, \beta]$
17:             $S^{'}_{GM} \leftarrow$ REMOVE$(g, S^{'}_{GM})$ if $\varepsilon > \beta$
18:         end if
19:         if $S^{'}_{GM}$ is feasible and $S' \notin \Sigma_{GM}$ then
20:             $\Sigma_{GM} \leftarrow \Sigma_{GM} \cup S^{'}_{GM}$
21:             return $S^{'}_{GM}$
22:         else
23:             $S^{'}_{GM} \leftarrow S_{GM}$
24:         end if
25:     end for
26: end function
```

C. ALGORITHM FOR NEIGHBORHOOD GENERATION OF SERVICE PLACEMENT

We represent a solution of service placement of a NIoT system as a list of tuples composed of an edge node and a cloud node, which is denoted by $S_{SP} = ((k_i, q_i) : i = 1 \ldots |S_{GM}|, k_i \in V_K, q_i \in V_Q)$. The solution shows that a set of virtual service functions for gateway i with a light workload is deployed at an edge node k_i and that with a heavy workload is deployed at a cloud node q_i.

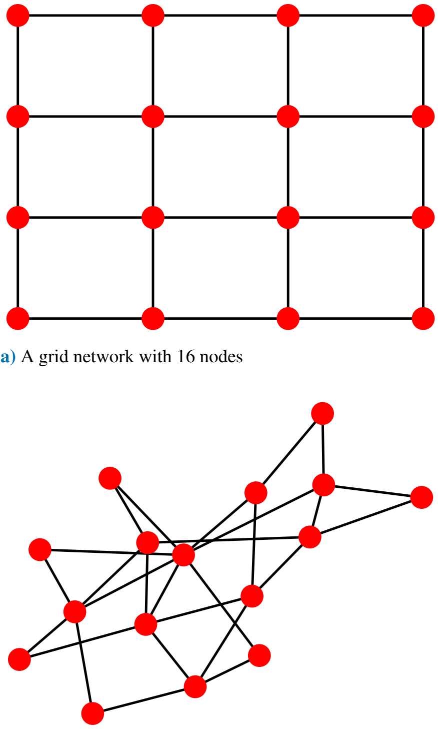
We propose the SA algorithms for the SP1 problem and the SP2 problem, called SP1A and SP2A, respectively, which use a similar neighborhood function, namely SPA-N. The difference between SP1A and SP2A is in Step 2 and Step 3 presented in Section VII-A when we compute the objective value. In particular, the SP1A algorithm uses Eq. (15) and the SP2A algorithm uses Eq. (20). The detail of the

Algorithm 2 Algorithm for neighborhood generation of service placement

```
function SPA-N($S_{SP}$)
    $S'_{SP} \leftarrow S_{SP}$
    $\Sigma_{SP} \leftarrow \emptyset$
    $\Pi_{SP} \leftarrow$ all pairs of edge nodes and cloud nodes in
        $V_K \times V_Q$
    for all $(k, q) \in \Pi_{SP}$ do
        $i \leftarrow$ a random number in $[1, |S_{SP}|]$
        $S'_{SP} \leftarrow$ REPLACE$(k, q, i, S'_{SP})$
        if $S'_{SP}$ is feasible and $S'_{SP} \notin \Sigma_{SP}$ then
            $\Sigma_{SP} \leftarrow \Sigma_{SP} \cup S'_{SP}$
            return $S'_{SP}$
        else
            $S'_{SP} \leftarrow S_{SP}$
        end if
    end for
end function
```

TABLE 2: Evaluation scenarios

Parameters	Grid	Barabasi-Albert
Network size (nodes)	16, 25, 36, 49	16, 25, 36, 49
Maximum relay	1, 2, 3, 4	1, 2, 3, 4



(a) A grid network with 16 nodes

(b) Barabsi-Albert (16 nodes, 4 edges attached from a new node to existing nodes, and 2 nodes initially attached to the network)

FIGURE 4: The grid and Barabasi-Albert topologies.

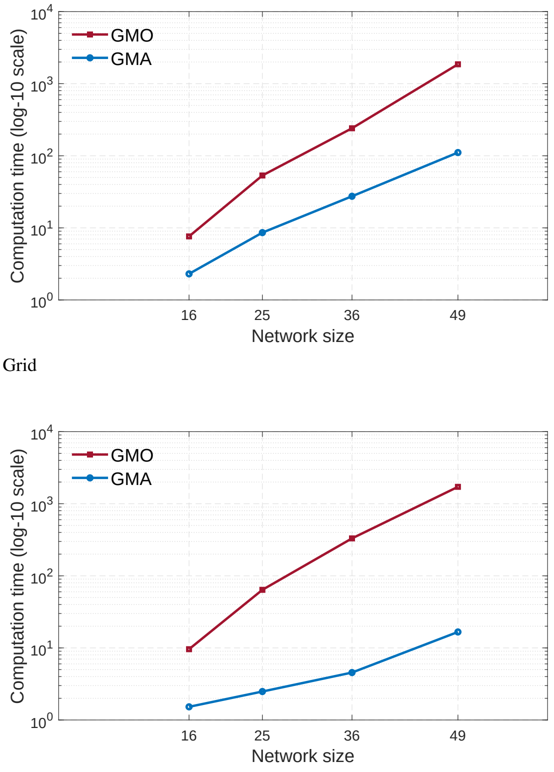
SPA-N algorithm for neighborhood generation of service placement is presented in Algorithm 2. In SPA-N, we denote all pairs of edge nodes and cloud nodes by $\Pi_{SP} = ((k, q) : k \in V_K, q \in V_Q)$. We define an operator for moving from a solution to another solution. The Replace(k, q, i, S_{SP}) operator changes the edge node and cloud node associated with gateway i in solution S_{SP} by edge node k and cloud node q. We use the Replace operator for a random gateway i and each (k, q) of Π_{SP} in succession until we find a feasible solution.

VIII. EVALUATION

This section presents an assessment of our optimization models and algorithms for the two problems of resource management in a NIoT system. We will start with a summary of various evaluation scenarios and several parameter settings for the algorithms. We then evaluate the performance of our proposed solutions in terms of several primary performance metrics. The evaluation also gives new insights into the strategy for an IoT provider to optimize its objectives.

A. SCENARIOS AND PARAMETERS SETTING

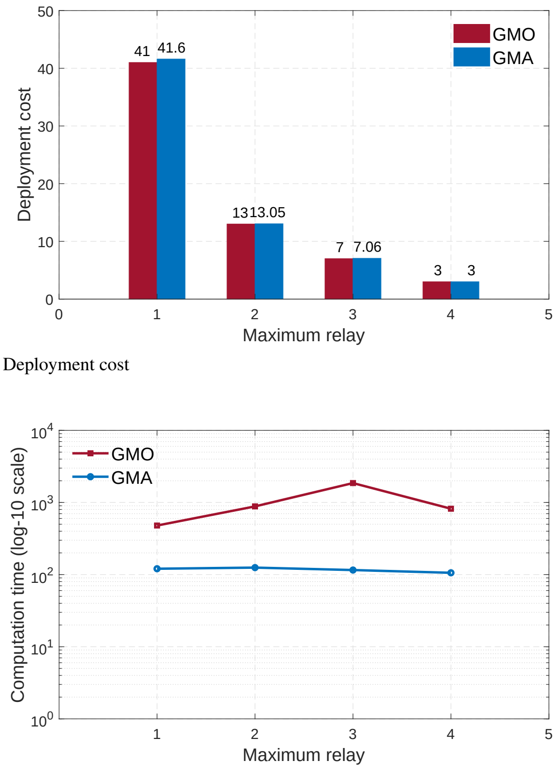
The topologies of the IoT layer used in our evaluation are the grid networks and synthetic topologies based on the Barabasi-Albert model [24], which are illustrated in Fig. 4. The parameters are summarized in Table 2. In the GMA algorithm, we assign $\gamma = 0.5$ as we consider the same probability for the Exchange and Add operators. We choose $\alpha = 0.4$ and $\beta = 0.7$ as we give a higher probability for the Exchange operator when the number of gateways is larger than or equal to one. A set of service functions required to be processed at the edge layer and that provided by the cloud layer is five functions. We define the basic unit of computing resources in our evaluation as 10^3 cycles per second. The computing resource required by a service function for processing one unit of data traffic is a random number uniformly distributed between 1 and 5. The computing resource of an edge node and a cloud node is 50×10^6. The capacity of a link between an edge node and a cloud node is 50 Gbps. The capacity of a link between a gateway and an edge node is 5 Gbps. The cost of processing a traffic unit at an edge node is uniformly distributed between 5 and 10. The cost of processing a traffic unit at a cloud node is uniformly distributed between 1 and 5. The data volume generated by an end node is a random number uniformly distributed between 10 and 1000. We use CPLEX to solve the GMO, SP1O, and SP2O models for finding the optimal results [25]. We carried out experiments in an X86-based PC with a two-core 2.7 GHz Intel processor and 8 GB memory. We evaluate the performance of our proposed solutions in terms of some primary metrics, including the deployment cost, the computing cost, the energy cost (i.e., the number of active nodes in the edge and cloud layers), and the computation time, which are computed as the average value in 50 runs.



(a) Grid

(b) Barabasi-Albert

FIGURE 5: Comparison between the computation time of GMA and that of GMO when varying network size.



(a) Deployment cost

(b) Computation time

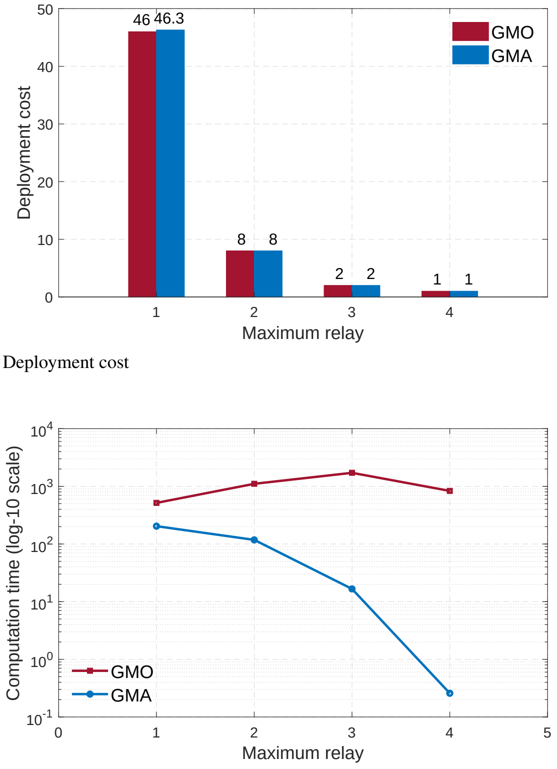
FIGURE 6: Comparison between GMA and GMO when varying the maximum relay with the grid topology.

B. PERFORMANCE EVALUATION OF GATEWAY PLACEMENT AND MULTIHOP ROUTING

We begin by assessing the efficiency of GMA in comparison with the optimal results produced by GMO for the problem of resource management at the IoT layer of a NIoT system. The maximum relay is three nodes for all topologies. We vary the network size between 16 and 49. Fig. 5 depicts the computation time of GMA and GMO. Note that the time is plotted on a log-10 scale. The results show that GMA is significantly faster than GMO in both grid and Barabasi-Albert topologies. More specifically, the ratio between the computation time of GMA and that of GMO increases from 3 times to 17 times when the number of nodes varies from 16 to 49 nodes.

Second, we evaluate the impact of the maximum relay on the deployment cost of gateways at the IoT layer of a NIoT system. We vary the maximum relay between one and four in the grid and Barabasi-Albert topologies with 49 nodes. We plot the deployment cost as a function of the maximum relay in Fig. 6 and Fig. 7. Fig. 6a and Fig. 7a show that the results produced by GMA are very close to the optimal results. We observe that GMA is more efficient when the maximum relay increases. We argue that the higher number of relays would lead to more possibility to improve an approximate solution. Both figures show that the deployment cost decreases when the maximum relay increases. This occurs because a large number of the maximum relay could result in an increase in the number of end nodes connected to one gateway, or a decrease in the number of gateways. Therefore, the deployment cost reduces. Fig. 6b and Fig. 7b plot the computation time in seconds. The results show that GMA's computation time is always significantly lower and more stable than that of GMO. We note that the maximum relay should be selected appropriately because of some performance issues of multihop routing in the IoT layer [26]. We can control the maximum relay used in multihop routing by parameter η.

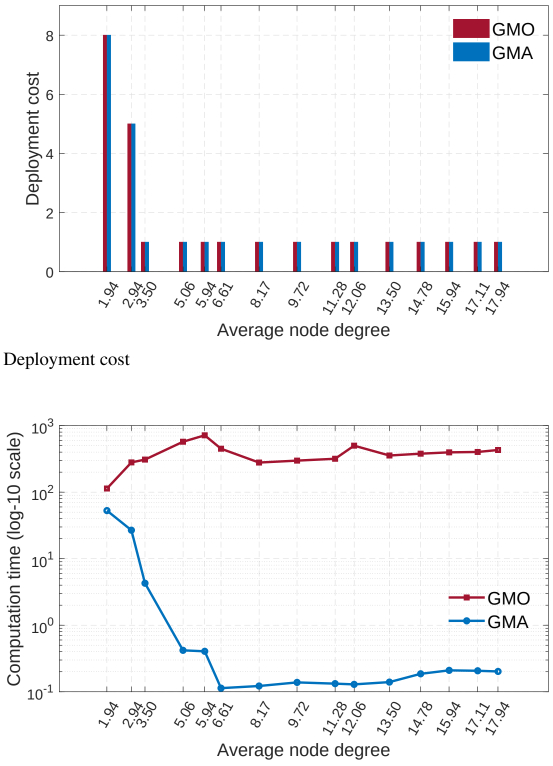
Third, we investigate the impact of the network density represented by the average node degree on the deployment cost. Fig. 8 plots a comparison between GMA and GMO regarding the deployment cost and the computation time in a Barabasi-Albert topology with 36 nodes when the maximum relay is three. The results show that the deployment cost of the solution produced by GMA is very close to the optimal solution, while the computation time of GMA is considerably smaller than that of the optimization model. We also note that the deployment cost rapidly reduces as the average node degree grows. We argue that the improvement of the number of potential connections in a high-density network leads to a decrease in the number of gateways. As a result, the deployment cost decreases. To summarize, the GMA approximation algorithm can find a feasible solution that is very close to the optimal one with significantly reduced time.



(a) Deployment cost

(b) Computation time

FIGURE 7: Comparison between GMA and GMO when varying the maximum relay with the Barabasi-Albert topology.



(a) Deployment cost

(b) Computation time

FIGURE 8: Comparison between GMA and GMO when varying the average node degree with the Barabasi-Albert topology.

Finally, we evaluate GMA in a large scenario with 400 IoT nodes for demonstrating the scalability of GMA. We compare GMA with a greedy algorithm because it is time-consuming for GMO to obtain the optimal solution of gateway placement in a large scenario. In the greedy algorithm, we sort a list of IoT nodes first by their degree in descending order, then by their cost in ascending order. After the list of IoT nodes is sorted, the algorithm selects an IoT node as a gateway, then connects the gateway to other IoT nodes that can deliver their data through the gateway. The process of gateway selection completes when all IoT nodes are linked to a gateway. Fig. 9 depicts the cost of gateway deployment in a large scenario when the maximum relay varies between 1 and 5 hops. The result shows that GMA is capable of finding an approximation solution for gateway placement and routing in a large scenario.

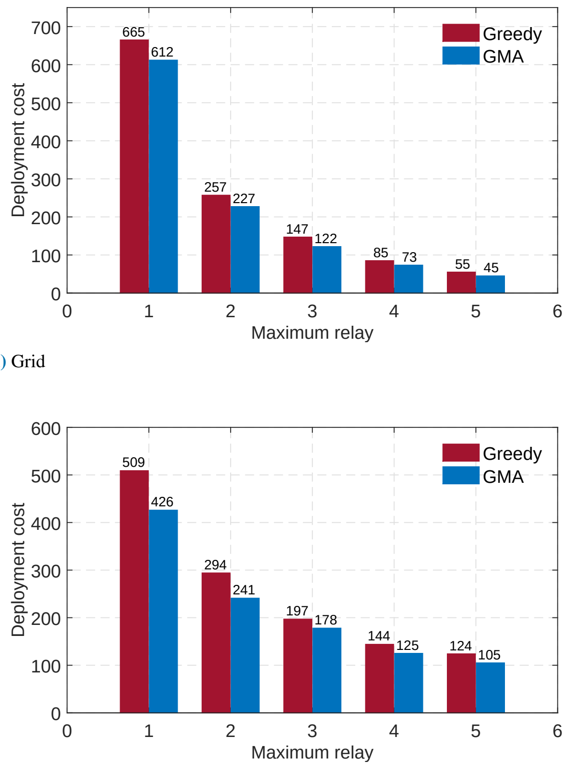
C. PERFORMANCE EVALUATION OF SERVICE PLACEMENT

In the performance evaluation of our proposed solution for service placement, we consider a NIoT system composed of 30 edge nodes and 50 cloud nodes. We vary the number of IoT gateways between 5 and 200 nodes. We compute three metrics, including the computing cost of service placement, the number of active nodes used for fulfilling the service requirements, and the computation time for finding a solution.

We first evaluate the performance of the SP1A and SP2A algorithms in comparison with the SP1O and SP2O optimization models solved by CPLEX. Fig. 10 shows that the computing costs of the approximation solutions produced SP1A and SP2A are very close to those solved by SP1O and SP2O, respectively. We observe a trade-off between the number of active nodes used at the edge and cloud layers (i.e., the energy cost) and the computing cost. For example, as shown in Fig. 10 and Fig. 11, SP1A is better than SP2A in terms of the computing cost while it is worse than SP2A in terms of the energy cost. We can infer that an IoT service provider might need to select an appropriate optimization strategy according to a charging agreement with an NFVIaaS provider. Furthermore, Fig. 12 shows that the computation time of the SP1A and SP2A algorithms are significantly lower than the SP1O and SP2O model solved by CPLEX. In summary, the SP1A and SP2A algorithms are efficient approaches for finding an approximation solution of service placement for a NIoT system.

Next, we study the impact of the location of service functions in edge cloud computing. We consider two cases: all service functions are deployed in the cloud layer (i.e., the Cloud case), and service functions are deployed in both the edge and the cloud layers (i.e., the Edge-Cloud case). In the Cloud case, note that data traffic is still routed from a gateway through an edge node to a cloud node. However, as we do

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3026711, IEEE Access



(a) Grid

(b) Barabasi-Albert

FIGURE 9: The cost of gateway deployment in a large scenario with 400 IoT nodes.

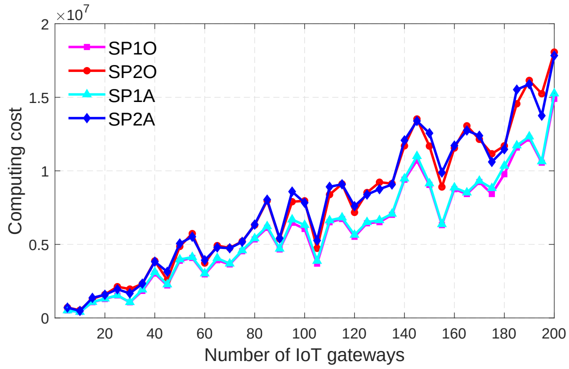


FIGURE 10: Computing cost of a solution produced by SP1A, SP2A, SP1O and SP2O.

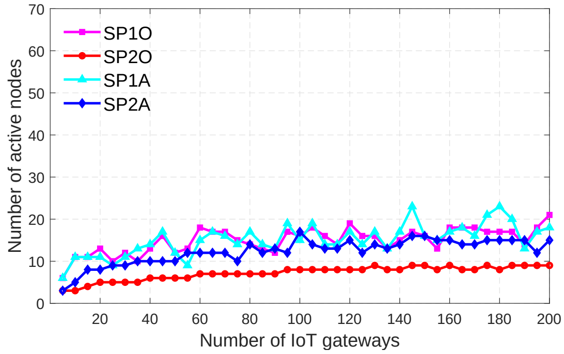


FIGURE 11: Energy cost of a solution produced by SP1A, SP2A, SP1O and SP2O.

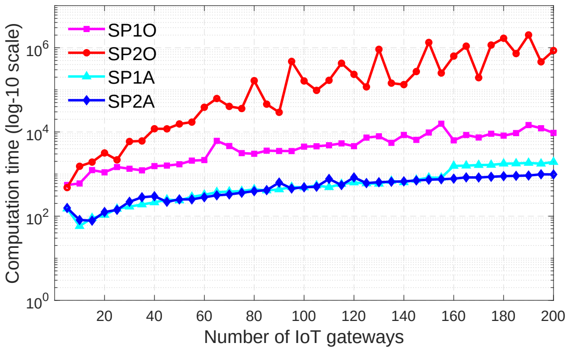


FIGURE 12: Computation time of SP1A, SP2A, SP1O and SP2O.

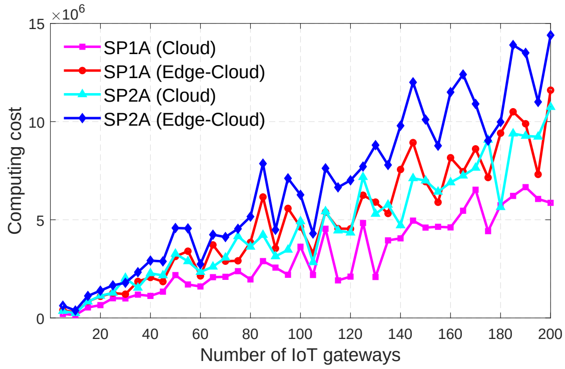


FIGURE 13: Comparison between the computing cost of the Cloud case and that of the Edge-Cloud case.

not deploy any service function in the edge layer, the cost related to edge nodes is not included in the cost functions (i.e., Eq. (15), (20)). Fig. 13, 14, and 15 plot the computing cost, the number of active nodes, and the computation time in the two cases. We observe that the computing cost of the Edge-Cloud case is higher than that of the Cloud case. This was to be expected due to the high resource cost at the edge. In other words, a customer is charged more for responsive service functions. Furthermore, Fig. 14 shows that the number of active nodes in the Cloud case is lower than that in the Edge-Cloud case. Consequently, the Edge-Cloud case requires more energy than the Cloud case. We argue that it is the cost of highly responsive, scalable, and reliable services offering by edge cloud computing. It implies that an IoT service provider should only deploy service functions with a strict delay requirement on the edge for optimizing its cost.

IX. CONCLUSION

We addressed the joint optimization problem of gateway placement and multihop routing in the IoT layer, the problem of service placement in the edge and cloud layers for a NIoT system. We proposed the GMO, SP1O, and SP2O models for obtaining the optimal solutions. An IoT service provider

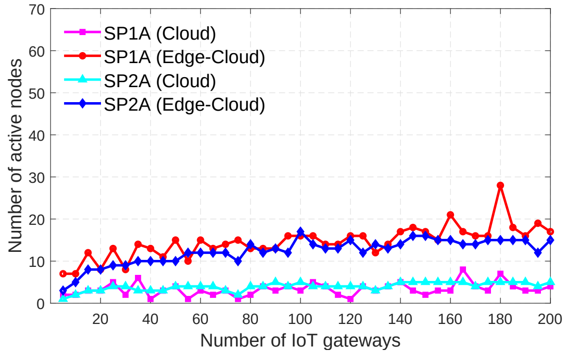


FIGURE 14: Energy cost of the Cloud and Edge-Cloud cases.

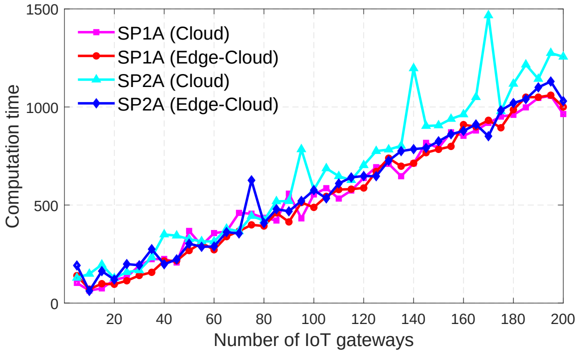


FIGURE 15: Computation time of SP1A and SP2A in the Cloud and Cloud-Edge cases.

can exploit our solution for determining the optimal gateway deployment, the optimal routing, and the optimal resource allocation to service functions in a NIoT system. We then developed the GMA, SP1A, SP2A algorithms for tackling the problems in a large-scale NIoT system. The evaluation results under diverse topologies show that the approximation algorithms can find the results close to the optimal solution with significantly reduced time. We observed that the deployment cost reduces as the maximum number of relays and the network density increase. We can infer from our evaluation that an IoT service provider might need to select an appropriate optimization strategy according to a charging agreement with an NFVIaaS provider. The results also suggest that an IoT service provider should only deploy service functions with a strict delay requirement on the edge for optimizing its cost. Our future work will consider the strict delay requirements of certain IoT services, the support of D2D communication for computation offloading, and the optimization of resource management at the physical level. It would also be of interest to study the collaboration between several IoT service providers for further improving the performance, the flexibility, and efficiency of a highly responsive NIoT system [19], [27].

ACKNOWLEDGMENT

This work was supported under Project no. 102.02-2020.13 granted by the National Foundation for Science and Technology Development (NAFOSTED) affiliated with the Vietnam Ministry of Science and Technology (MOST). The authors would like to thank Hoai-Nam Chu for his initial implementation of models and algorithms for the problem of gateway placement. The authors are sincerely grateful to the anonymous reviewers for many helpful comments and constructive suggestions.

REFERENCES

[1] C. MacGillivray and D. Reinsel, “Worldwide global datasphere iot device and data forecast, 2019-2023,” IDC, Tech. Rep., 2019.

[2] L. Lan, R. Shi, B. Wang, and L. Zhang, “An iot unified access platform for heterogeneity sensing devices based on edge computing,” IEEE Access, vol. 7, pp. 44 199–44 211, 2019.

[3] C. Mouradian, N. T. Jahromi, and R. H. Glitho, “Nfv and sdn-based distributed iot gateway for large-scale disaster management,” IEEE Internet of Things Journal, vol. 5, no. 5, pp. 4119–4131, 2018.

[4] H.-N. Chu and T.-M. Pham, “Joint optimization of gateway placement and multi-hop routing for the internet of things,” in Proceedings of the 6th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, 2019, pp. 88–93.

[5] X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, and L. Qi, “A computation offloading method over big data for iot-enabled cloud-edge computing,” Future Generation Computer Systems, vol. 95, pp. 522 – 533, 2019.

[6] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghrayeb, “Optimized provisioning of edge computing resources with heterogeneous workload in iot networks,” IEEE Transactions on Network and Service Management, vol. 16, no. 2, pp. 459–474, 2019.

[7] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, “An energy efficient iot data compression approach for edge machine learning,” Future Generation Computer Systems, vol. 96, pp. 168 – 175, 2019.

[8] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, “Learning-based context-aware resource allocation for edge-computing-empowered industrial iot,” IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4260–4277, 2020.

[9] L. Zhao, J. Wang, J. Liu, and N. Kato, “Optimal edge resource allocation in iot-based smart cities,” IEEE Network, vol. 33, no. 2, pp. 30–35, 2019.

[10] A. M. Maia, Y. Ghamri-Doudane, D. Vieira, and M. F. de Castro, “Optimized placement of scalable iot services in edge computing,” in Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019, pp. 189–197.

[11] M. Wang, B. Cheng, X. Liu, Y. Yue, B. Li, and J. Chen, “A sdn/nfv-based iot network slicing creation system,” in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 666–668.

[12] N. Mitton, S. Papavassiliou, A. Puliafito, and K. S. Trivedi, “Combining cloud and sensors in a smart city environment,” EURASIP Journal on Wireless Communications and Networking, 2012.

[13] W. He, G. Yan, and L. D. Xu, “Developing vehicular data cloud services in the iot environment,” IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1587–1595, 2014.

[14] A. Botta, W. de Donato, V. Persico, and A. Pescape, “Integration of cloud computing and internet of things: A survey,” Future Generation Computer Systems, vol. 56, pp. 684 – 700, 2016.

[15] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek, “Device-enhanced mec: Multi-access edge computing (mec) aided by end device computation and caching: A survey,” IEEE Access, vol. 7, pp. 166 079–166 108, 2019.

[16] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, “An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures,” IEEE/ACM Trans. Netw., vol. 25, no. 4, pp. 2008–2025, 2017.

[17] T.-M. Pham, T.-T.-L. Nguyen, S. Fdida, and H. T. T. Binh, “Online load balancing for network functions virtualization,” in Proceedings of the 2017 IEEE International Conference on Communications (ICC), May 2017, pp. 1–6.

[18] M. C. Luizelli, D. Raz, and Y. Sa’ar, “Optimizing nfv chain deployment through minimizing the cost of virtual switching,” in Proceedings of the 2018 IEEE Conference on Computer Communications (INFOCOM), April 2018, pp. 2150–2158.

[19] T.-M. Pham and H.-N. Chu, "Multi-provider and multi-domain resource orchestration in network functions virtualization," IEEE Access, vol. 7, pp. 86 920–86 931, 2019.

[20] T.-M. Pham, S. Fdida, T.-T.-L. Nguyen, and H.-N. Chu, "Modeling and analysis of robust service composition for network functions virtualization," Computer Networks, vol. 166, p. 106989, 2020.

[21] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Service function chain embedding for nfv-enabled iot based on deep reinforcement learning," IEEE Communications Magazine, vol. 57, no. 11, pp. 102–108, 2019.

[22] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, no. 7, pp. 1497 – 1516, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870512000674

[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671–680, may 1983.

[24] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," Reviews of Modern Physics, vol. 74, no. 1, pp. 47–97, 06 2001.

[25] IBM ILOG CPLEX Optimizer. Available online: https://www.ibm.com/analytics/cplex-optimizer/ (accessed on 18 July 2020).

[26] A. ElSamadouny, M. Hasna, T. Khattab, K. Abualsaud, and E. Yaacoub, "On the delay of finite buffered multi-hop relay wireless internet of things," in 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), 2019, pp. 1–7.

[27] T.-M. Pham, "Analysis of ISP caching in information-centric networks," in Proceedings of the 11th IEEE-RIVF International Conference on Computing and Communication Technologies, 2015, pp. 151–156.