IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 September 2022, accepted 31 October 2022, date of publication 4 November 2022, date of current version 15 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219427

SURVEY

A Survey on Matching Theory for Distributed Computation Offloading in IoT-Fog-Cloud Systems: Perspectives and Open Issues

HOA TRAN-DANG[1], (Member, IEEE), AND DONG-SEONG KIM[1,2], (Senior Member, IEEE)
[1]Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi-si 39177, South Korea
[2]Industrial Academic Cooperation Foundation, Kumoh National Institute of Technology, Gumi-si 39177, South Korea

Corresponding author: Dong-Seong Kim (dskim@kumoh.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Grand Information Technology Research Center Support Program Supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2020-2020-0-01612; in part by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under Grant 2018R1A6A1A03024003; and in part by the Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant NRF-2020R1I1A1A01073019.

ABSTRACT Fog computing has been widely integrated in the IoT-based systems, creating IoT-Fog-Cloud (IFC) systems to improve the system performances and satisfy the quality of services (QoS) and quality of experience (QoE) requirements for the end users (EUs). This improvement is enabled by computational offloading schemes, which perform the task computation nearby the task generation sources (i.e., IoT devices, EUs) on behalf of remote cloud servers. To realize the benefits of offloading techniques, however, there is a need to incorporate efficient resource allocation frameworks, which can deal effectively with intrinsic properties of computing environment in the IFC systems such as resource heterogeneity of computing devices, various requirements of computation tasks, high task request rates, and so on. While the centralize optimization and non-cooperative game theory based solutions are applicable in a certain number of application scenarios, they fail to be efficient in many of cases, where the global information and control might be unavailable or cost-intensive to achieve it in the large-scale systems. The need of distributed computational offloading algorithms with low computation complexity has motivated a surge of solutions using matching theory. In the present review, we first describe the fundamental concept of this emerging tool enabling the distributed implementation in the computing environment. Then the key solution concepts and algorithmic implementations proposed in the framework of literature are highlighted and discussed. Given the powerful tool of matching theory, its full capability is still unexplored and unexploited in the literature. We thereby discover and discuss existing challenges and corresponding solutions that the matching theory can be applied to resolve them. Furthermore, new problems and open issues for application scenarios of modern IFC systems are also investigated thoroughly.

INDEX TERMS IoT-Fog-cloud systems, matching theory, distributed algorithm, computational offloading.

I. INTRODUCTION

Practically, the Internet of Things (IoT) has become an integral element for realizing smart practical systems such as smart cities [1], smart grids [2], smart factories [3], smart logistics, and supply chain [4], [5]. The fundamental aspect of IoT-based systems is to connect all devices through the Internet protocol to exchange high volume data and process them to create smart services and applications [6], [7]. Owing to limited computation resources, network, storage, and energy, IoT devices are inadequate for executing all computational tasks, especially tasks with huge volumes and complex data structures. Cloud computing is an essential solution to this problem because it provides powerful resources to fulfill tasks efficiently [8], [9]. However, cloud computing-based solutions do not always meet the expected

The associate editor coordinating the review of this manuscript and approving it for publication was Eyhab Al-Masri.

quality of service (QoS) and quality of experience (QoE) requirements for some classes of IoT applications, especially latency-sensitive ones because of the long physical distance between the IoT devices and the remote cloud servers, scarce spectrum resources, and intermittent network connectivity.

This has led to the emergence of fog computing, which extends the cloud computing resources (i.e., computing, storage, and networking) closer to the data generation sources (i.e., IoT devices), thereby allowing for the prescribed QoS requirements of services and application to be met by enabling the fog computing devices (e.g., switches, gateways, and hubs) to process and offload most tasks on behalf of the cloud servers in a distributed manner [10], [11]. Recently, the advance of networking generation (i.e, 5G and beyond) leads to an increasing demand of ubiquitous computing and pervasive service accesses by a numerous number of Internet-connected mobile devices and end users (EUs). Motivated by the necessity of network architecture enhancement, a paradigm of fog radio access networks (F-RANs) has emerged as a promising evolution path for 5G network architecture [12], [13], which along with cloud radio access networks (C-RANs) [14] provide the pervasive computing services. In F-RANs, a fog computing layer is deployed at the edge of networks, allowing a part of the service and application requirements to be responded locally without the need of the centralized cloud computing. Therefore, by taking full advantage of distributed caching and centralized processing, F-RANs provide great flexibility to satisfy QoS requirements of various 5G-based services and applications. Besides providing the cloud like services to EUs, the fog computing potentially improves the performance of fog-based systems such as reduction of service delay [15], and energy saving [16] through efficient computational offloading algorithms [17]. Ultimately, IoT-Fog-Cloud (IFC) systems formed by the integration of IoT, fog, and cloud are able to provide uninterrupted services and applications with significant QoS improvement along the things-to-cloud continuum.

To further realize the above benefits of computing paradigms, the IFC systems require efficient resource allocation and management strategies to perform computational offloading operations [18]. However, there are many factors that challenge the design and development of effective offloading strategies. First, an IFC system consists of heterogeneous computing devices with different storage capacity, computation, and networking characteristics. Except the cloud servers, the IoT devices (e.g., smart phones, tablets) and fog nodes (FNs) (e.g., gateways, switches, and hubs) are resource constrained, thus limiting the capability of processing a large type of computation tasks. For example, some IoT and fog devices can support the process of only one data type such as image, text, video, or audio [19]. In addition, modern applications such as artificial intelligence and machine learning algorithms require to the computation of complex tasks, which typically include multiple types of input data [20]. Second, the diverse task requests also have a significant impact on offloading performance in the IFC systems. For example, some fog devices are unable to process the entire data of heavy tasks owing to a lack of storage and limitation of computational capability. Consequently, more tasks are likely to be queued in more powerful resource fogs causing the over-utilized workload at these nodes. Third, the request rate directly impacts on the queuing state of fogs. Therefore, without an efficient resource allocation policy, a high rate of task request may lead to a high workload imbalance among the fog devices, as the fog nodes with powerful computing resources may receive more task requests.

There are a large number of centralized optimization techniques and algorithms proposed in the literature to provide optimal solutions to the aforementioned resource allocation problems [21], [22]. For instance, offloading multiple tasks of fog nodes (FNs) to multiple neighbor FNs (i.e., helper nodes (HNs)) is modeled as a multi-task multi-helper (MTMH) problem, which aims to allocate the fog computing resources for processing tasks to minimize the average delay of task execution. The multi-objective optimization problem is also investigated to examine the trade-off of performance in terms of energy consumption, delay, and execution cost [23]. The optimization based solutions, however, require a centralized control to gather the global system information, thus incurring a significant overhead and computation complexity of algorithms. This complexity is further amplified by the rapidly increase of density and heterogeneity of IFC computing systems [24] because the centralized optimizations many not able to properly handle the challenges of dense and heterogeneous fog computing environment when dealing with combination integer programming problems [25].

The aforementioned limitations of optimization have lead to a second groups of solutions that apply the non-cooperative game theory to avoid the cost-intensive centralized resource management as well as substantially reduce the complexity of algorithms [26], [27]. Despite their potentials, such approaches pose several limitations. First, classical game theoretical algorithms such as best response require some information regarding actions of other players [28]. Correspondingly, many assumptions are introduced in the game theory-based algorithms to simplify the system models that, in some case, are impractical. Second, most game-theoretic solutions, for example, Nash equilibrium, investigate one-sided stability notions in which equilibrium deviations are evaluated unilaterally per player. In addition, in the IFC systems, the stability must be concerned by both sides of players, i.e., resource providers and resource requesters.

Ultimately, managing resource allocation effectively in such a complex environment of IFC systems leads to a fundamental shift from the traditional centralized mechanism toward distributed approaches. Recently, matching theory has emerged as a promising technique for resource allocation problems in many applications, which can alleviate

the shortcomings of game theory and optimization-based approaches. Alternatively, while the optimization and game theory based solutions are efficient in a some limited scenarios, the matching-based approaches have potential advantages owing to the distributed and low computational complexity algorithm. However, to reap the benefits of matching theory for task offloading and resource allocation in fog environment there is a need of advanced frameworks to handle their intrinsic properties such as heterogeneity of fog computing devices as well as the novel QoS demands of future generation systems. This directly expose new challenges and associated open issues.

In these regards, this paper provides four important contributions as follows.

- An end-to-end model of IFC system and its associated features such as architecture, computation tasks, computation offloading models, and a generic optimization form of task offloading problems are described to highlight the intrinsic properties of fog-based computing systems.
- The fundamental concept of matching theory and its key models are summarized towards the applications for resource allocation problems.
- The paper emphasizes on reviewing and investigating the proposed matching-based distributed algorithms in the existing literature to solve the computation offloading related problems.
- Remaining challenges and open issues are explored and discussed to provide the future directions of researches and development regarding the usage of matching theory in the new problems and application scenarios.

The remainder of this paper is organized as follows. Section II briefly summarizes the key concepts of matching theory including models, classification, and conventional algorithmic solution. Section III reviews the related works that cover existing distributed algorithms without using the matching theory, and the applications of matching theory for resource allocation problems in the wireless networks. Section IV presents the generic models regarding the IFC systems, computational tasks, computation offloading models, and generic optimization problem formulation. Section V discusses and analyzes the matching-based models proposed in the literature to solve the computation offloading problems. Section VI explores the remaining challenges and discusses associated open issues. Section VI concludes the paper.

II. RELATED WORKS

Computation offloading is a pivotal operation in the IoT systems that leverage the edge and fog computing technologies to improve the QoS and QoE. To design efficient offloading algorithms that cope with challenges of fog computing environment and various requirements of services, there are some sort of algorithms and techniques developed in the literature.

In a comprehensive assessment on fog computing architecture and algorithm introduced in [29], the computation offloading process involves three specific problems that are task offloading and load distribution, task scheduling, and resource sharing. The authors evaluate and discuss the proposed algorithms according to the five criteria including heterogeneity, QoS management, scalability, mobility, federation, and interoperability. The algorithms are derived from different approaches such as global optimization, distributed computation, and learning-based methods. However, the most of algorithms do not satisfy all the predefined criteria. For example, the global optimization-based algorithms are no longer to support the scalability requirement owing to its computation complexity in the case of large scale systems.

The work [30] concerns on the optimization models for optimizing the system performance in terms of latency, energy consumption, caching, service placement, and load balancing. Many approaches applied to solve the optimization problems include mix-integer linear programming (MILP), graph theory, game theory, and greedy methods.

Evaluating the distributed computation offloading, the study [18] emphasizes on the locations (i.e., cloud, fog, or local at terminal nodes) where the task offloading is taken place. Accordingly, the algorithms take into account various factors such as task requirements, computation capability of computing nodes, network scale to determine the appropriate locations. The evaluation exposes that the centralized optimization-based algorithms is able to derive the optimal performance of systems, although they suffers from the high computation complexity in the large scale systems. Many distributed and greedy algorithms are investigated, but some of them are efficient in limited application scenarios.

The authors in [31] focus on stochastic-based offloading mechanisms in three major computation environments: mobile cloud computing (MCC), mobile edge computing (MEC), and fog computing (FC). The algorithms are constructed based on the mechanisms following Markov chain, Markov process, and hidden Markov models.

The authors in [32] conduct a comprehensive review of existing literature that applies machine learning for deriving the computation offloading mechanisms in the computing systems. Furthermore, the associated comparative analysis is provided for a comprehensive comparison of stochastic Markov-based offloading mechanisms.

In this work, we conduct a survey of recent offloading algorithms using matching theory that potentially release the shortcomings of aforementioned algorithms by its lightweight and distributed mechanism [33]. To the best of our knowledge, our paper is the first work accessing the state-of-the-art in the matching theory-based algorithms for computational offloading in the IFC systems.

III. MATCHING THEORY FUNDAMENTALS

A. BASIC CONCEPT

Matching theory has been considered as a potential mechanism to solve the resource allocation problems in the context of wireless networks [34], [35], [36] since it can alleviate

some shortcomings of game theory and optimization-based solutions. Basically, matching theory provides mathematically tractable solutions for the combination problem of matching players in two distinct sets, depending on the individual information and preference of each player. Although the main matching models in the literature deal with two sets of agents, it should be noted that there are matching models that are among the agents in one set only (i.e., stable roommate problem [37]), and matching models among three sets of agents [38], [39].

The basic task offloading problems can be interpreted as a matching problem between the set $\mathcal{R}$ of computing resources ($\mathcal{R} = \mathcal{I} \cup \mathcal{F} \cup \mathcal{C}$) and the set $\mathcal{T}$ of computation tasks. Depending on the scenarios, the resources can be of different abstraction levels, representing base stations, time, power, storage, CPU. Due to the limitation of resource, each fog computing device has a quota that defines the maximum number of players (tasks) with which it can be matched. The main goal of matching is to optimally match resources and tasks given their individual, often different, objectives and exchanged information. Each resource (task) builds a ranking of the tasks (resources) using a preference relation. The concept of preference represents the individual view that each resource or task has of the other set, based on the local information. In its basic form, a preference can simply be defined in terms of an objective utility function that quantifies the QoS achieved by a certain resource-task matching. However, a preference is more generic than a utility function in that it can incorporate additional qualitative measures extracted from the information available to tasks and resource subject the dynamic change of fog computing environment. In the most of works in the literature, the utility function is used to construct the preference lists of agents.

The inherent presence of selfishness and rational of agents prevents the systems to derive a global optimal solution. The objective of matching game is to achieve stability instead of optimality, at which there is no incentive incurred to devise the current matched pairs of agents. The concept of stability is defined differently based on the matching models, which are discussed in the following sections.

B. CLASSIFICATION

There are many variants of matching problems [40]. From the preference list point of view, it distinguishes three types of preference lists including complete, ties, and incomplete PL with ties. Meanwhile, for applying in the resource allocation problem of wireless networks, [41] consider canonical, matching with externalities, matching with dynamics model. Regarding the transfer between two sets of matching game, there are two classes of matching problems: matching with transfer and matching without transfer. In the context of resource allocation for computation offloading in the IFC, another features such as incomplete, ties PL, transfer as well as externalities are considered to be variants of these matching classes. Regarding the number of sets of player involving the matching, there are three typical types of matching models including one set (i.e., the roommate matching problem), two sets, and three sets (three dimensional matching). In the followings, we discuss the matching models according to three classes of matching game between plays in the two sets.

1) ONE-TO-ONE MATCHING

The most prominent model of one-to-one matching is marriage model. In this model, there is two distinct sets of agents represented by $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, y_2, \ldots, y_k\}$, respectively. Each agent has a complete preference list over the agents on the other side. Assume that an agent $x \in \mathcal{X}$ has a preference list $\mathcal{P}(x) = \{y_2, y_4, x, y_1, y_3, \ldots\}$. This means that x prefers agent y_2 to y_4 and prefers remaining single (x) over matching with y_1 or y_3. Denote $y_i \succ_x y_j$ to express that an agent x prefers agent y_i to y_j. In particular, as $y_i \succeq_x y_j$ there exists a tie in the preference list of agent x.

The one-to-one matching model is defined as follow:

Definition 1: The outcome of one-to-one matching model is a matching $\mathcal{M}$: $\mathcal{X} \cup \mathcal{Y} \mapsto \mathcal{X} \cup \mathcal{Y}$ such that the three following constraints are satisfied:

- For any $x \in \mathcal{X}$, $\mathcal{M}(x) \in \mathcal{Y} \cup \{x\}$,
- For any $y \in \mathcal{Y}$, $\mathcal{M}(y) \in \mathcal{X} \cup \{y\}$,
- For any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, $x = \mathcal{M}(y)$ if and only if $y = \mathcal{M}(x)$.

In the one-to-one matching model, each agent x can only be matched with one agent y, and x remains unmatched if $\mathcal{M}(x) = x$. The objective of matching is to reach the stable status for all pairs.

Definition 2: A matching $\mathcal{M}$ is pairwise stable if there is no block pair (x, y).

Definition 3: (x, y) is a block pair for a matching $\mathcal{M}$ if three following conditions are satisfied:

- $\mathcal{M}(x) \neq y$,
- $y \succ_x \mathcal{M}(x)$,
- $x \succ_y \mathcal{M}(y)$.

2) MANY-TO-ONE MATCHING

In the many-to-one (or one-to-many) matching modes, each agent of one side can be matched with multiple agents of the other side but the reverse is not valid. Similar to one-to-one matching, $\mathcal{P}(x) = \{y_1, y_2, x, y_4, y_5, \ldots\}$ illustrates that agent x prefers y_1 to y_2 ($y_1 \succ_x y_2$), and prefers keeping the position unfilled over other people like y_4 and y_5. Unlike one-to-one matching, each agent y has a positive quota q_y to represent the maximum number of agents in the set $\mathcal{Y}$ it can be matched with. Generally, many-to-one matching can be defined as follow:

Definition 4: The outcome of many-to-one matching model is a matching $\mathcal{M}$: $\mathcal{X} \cup \mathcal{Y} \mapsto \mathcal{X} \cup \mathcal{Y}$ such that the three following constraints are satisfied:

- $|\mathcal{M}(x)| = 1$ for every agent $x \in \mathcal{X}$ and $\mathcal{M}(x) = x$ if x is unmatched,

- $|\mathcal{M}(y)| = q_y$ for every agent $y \in \mathcal{Y}$; if the number of agents in $\mathcal{M}(y)$ is k and $k < q_y$, then $\mathcal{M}(y)$ will have $q_y - k$ copies of y,
- $\mathcal{M}(y) = x$ if and only if x is an element of $\mathcal{M}(y)$.

With this definition, $\mathcal{M}(x) = y$ means that agent x is matched with agent y, and $\mathcal{M}(y) = \{x_1, x_2, y, y\}$ indicates that the agent y with the quota $q_y = 4$ has been matched with two agents x_1 and x_2 and has two unfilled matching positions. The objective of many-to-one matching is to obtain the stable matching, which is defied in the same way as one-to-one matching.

3) MANY-TO-MANY MATCHING

In the models of many-to-many matching, the number of matchings for the agents on the both sides are not restricted to one. Denote q_x and q_y as the respective quotas for agents $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Generally, the many-to-many matching is defined as follows:

Definition 5: The outcome of many-to-many matching model is a matching $\mathcal{M}$: $\mathcal{X} \cup \mathcal{Y} \mapsto \mathcal{X} \cup \mathcal{Y}$ such that the three following constraints are satisfied:

- $|\mathcal{M}(x)| = q_x$ for every agent $x \in \mathcal{X}$; if the number of agents in $\mathcal{M}(x)$ is k and $k < q_x$, then $\mathcal{M}(x)$ will have $q_x - k$ copies of x,
- $|\mathcal{M}(y)| = q_y$ for every agent $y \in \mathcal{Y}$; if the number of agents in $\mathcal{M}(y)$ is m and $m < q_y$, then $\mathcal{M}(y)$ will have $q_y - m$ copies of y,
- $\mathcal{M}(y) = x$ if and only if x is an element of $\mathcal{M}(y)$.

C. CONVENTIONAL ALGORITHMIC SOLUTION

The basic algorithm known as the deferred acceptance (DA) was introduced firstly in [42] to find the one-to-one stable matching for the marriage problem. This algorithm can reach the convergence in the polynomial time for the one-to-one matching problems, and very fast for the many-to-one matching models. Fundamentally, DA is an iterative method over the players of sets, in which one side proposes and the other side decides to reject or accept the proposal based on PLs. With this approach, DA is completely distributed since the play is just based on the local information for deriving the decisions. Algorithm 1 shows the key procedures to implement the DA algorithm to achieve the outcome for the one-to-one matching problem. Notably, we consider a complete, strict, and transitive PLs. In addition, the number of agents of both sets are equal, thus at the stability of matching outcome, all agents are matched. The variants such as tie, incomplete PLs, or unequal cardinality of sets are modified according to the application scenarios, which are discussed additionally in the next survey section.

Algorithm 1 The Classical DA Algorithm for One-to-One Matching Problem

Input: $\mathcal{P}(x)$, $\mathcal{P}(y)$
// Preference lists of x and y
Output: $\mathcal{M}$
begin
  Initialize: $\mathcal{M}(x) = x$ & $\mathcal{M}(y) = y, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$
  // All agents x and y are unmatched
  while $\exists x\ (\mathcal{M}(x) = x)$ **do**
    $\mathcal{M}(x) \leftarrow y\ (y = \mathcal{P}(y)[0])$
    // Proposing the first $y \in \mathcal{P}(y)$ to be matched with x
    if $\mathcal{M}(y) = y$ **then**
      $\mathcal{M}(x) = y$ // Match y with x
    else
      $\mathcal{M}(y) = x'$ // y is already matched with x'
      if $x \succ_y x'$ **then**
        $\mathcal{M}(x) = y$ // Match x with y
        $\mathcal{M}(x') = x'$ // x' becomes unmatched
      else
        $\mathcal{M}(x) = x$// x is still unmatched
        $\mathcal{M}(y) = x'$// (x', y) remains matched

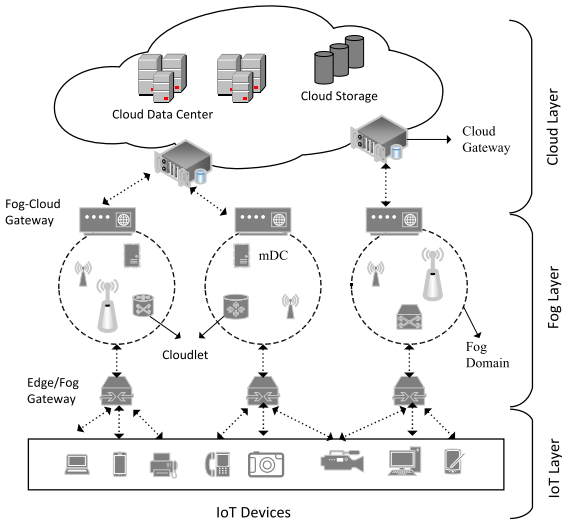

FIGURE 1. The typical architecture of IFC systems.

IV. SYSTEM AND OFFLOADING PROBLEM DESCRIPTION

A. SYSTEM MODEL

A general IFC system with three-tier architecture is illustrated as in Fig. 1. The system consists of three layer: IoT, fog, and cloud, which are connected by LAN, and WAN to provide various services for IoT-connected users such as computing, caching, storage, and networking services.

The IoT layer is recorded by a set $\mathcal{I} = \{d_1, d_2, \ldots, d_{|\mathcal{I}|}\}$ of IoT devices, which generate computation tasks recorded in a set $\mathcal{T} = \{T_1, T_2, \ldots, T_{|\mathcal{T}|}\}$. Similarly, $\mathcal{F} = \{F_1, F_2, \ldots, F_{|\mathcal{F}|}\}$ and $\mathcal{C} = \{C_1, C_2, \ldots, C_{|\mathcal{C}|}\}$ represent the sets of fog devices, and cloud servers, respectively. In practical applications, the fog devices are grouped into clusters, each provides a set of specific IoT applications for the end users. And in the large-scale IFC systems, there are multiple domains of fogs, which are federated for jointly process and offload the computation tasks. The fogs in each domain are deployed in a distributed manner.

TABLE 1. Resource state of fog nodes F_i.

Fog Node	Fog specification & Resource Status			
	M (MB)	f (GHz)	γ (cycles/bit)	CPU/GPU
F_1	128	5	500	CPU
F_2	256	10	600	GPU
F_3	128	15	750	CPU
F_4	256	10	1000	GPU

In some scenarios, there is a presence of centralized fog controllers such as fog service providers to manage the fog resources in the domains as well as the security related issues. The fog computing devices are characterized by heterogeneity in terms of computing, networking, and storage capacity depending on device types. Practically, the typical fog devices deployed in the IFC systems include networking devices such as gateways, switches, routers, and a few to name. In many scenarios, the cloudlets [43] and micro data centers (mDCs) [44], [45] are added to the fog domains to enhance the computing capability as mini-servers. Basically, the cloudlets and mDCs are resource-rich devices, which are located in a one-hop proximity of end users or mobile devices (MDs) for improving the QoS of mobile applications. Table 1 shows an example of resource states of fog computing devices with respect to memory capacity of queue buffer (M), CPU frequency (f), and CPU processing density (γ).

In the cloud tiers, virtual machines (VC) are deployed in the data centers and servers to flexibly provide the services requested by lower layers (i.e., IoT and fog layer).

For the sake of clarity, Table 2 provides notions and abbreviations used mostly in the paper.

TABLE 2. Important notions and abbreviations used in this paper.

Notion	Definition
IFC	IoT-Fog-Cloud System
F-RAN	Fog Radio Access Network
C-RAN	Cloud Radio Access Network
VFN	Vehicular Fog Network
LAN	Local Access Network
WAN	Wide Access Network
AP	Access Point
BS	Base Station
QoS	Quality of Service
QoE	Quality of Experience
FN	Fog Node
HN	Helper Node (FN that has available resources for offloading)
SFN	Surplus fog node (FN that has surplus resource for computing)
DFN	Deficit fog node (FN that lacks resource for data computing)
Helpers	A set of HNs and cloud servers
TN	Task Node (IoT node or FN that has computation tasks)
EU	End Users
UE	User Equipment
DSO	Data Service Operator
DSS	Data Service Subscriber
MD	Mobile Device
FAP	Fog Access Point
SP	Service Provider
FSP/CSP	Fog/Cloud Service Provider
VM	Virtual Machine
VRU	Virtual Resource Unit
CRB	Computing Resource Block
Outage	Fraction of computation tasks missing the deadlines
PL	Preference List
DA	Deferred Acceptance
MSDA	Multi-Stage Deferred Acceptance
SPA	Student Project Allocation
CRITIC	Criteria Importance through inter criteria correlation
TOPSIS	Technique for order of preference by similarity to ideal solution
AHP	Analytical Hierarchy Process

B. COMPUTATIONAL TASKS

Each computing task T_k can be described with a tuple $T_k = \langle A_k, O_k, B_k, D_k \rangle$, where A_k and O_k represent the input and output data size (bits) of task, and B_k is the computational resource demands (CPU/GPU cycles) to execute the task. In many application scenarios, there are latency-sensitive tasks, which require to be completely executed within the prescribed deadlines D_k.

Basically, A_k can include following features: total size (in bits or bytes), splittable or non-splitable, number of data types. The sizes of input data of tasks can be ranged from kilo-bytes to tera-bytes depending the specific applications [46]. Based on this feature, the tasks can be classified into light, medium, and heavy tasks as studied in many of existing works [15], [47] for further analyzing the impact of task sizes on the performance of computation offloading approaches.

The divisibility of tasks, particularly heavy tasks with large input data sizes is also investigated in the offloading cases. Accordingly, the whole input data of a task is definitely processed by a single computing device (e.g., FN, cloud, or even powerful IoT node) as it is unable to be splitted into data subsets. Whereas, in several scenarios, a single task can be divided into multiple subtasks with smaller data sizes. Such the task division is employed to get benefit from parallel computing since the substasks can be processed by different devices simultaneously.

A task can also be partitioned into subtasks based on the types of input data. For example, a typical AI and ML task may include multiple types of data such as text, image, video, and audio as studied in [19] and [48]. This partition is suitable for designing the efficient resource allocations in heterogeneous computing environments, in which there are limited number of devices able to handle the all data types. In other words, some devices only process text-type data, some are capable to process video-type data, and so on. Regarding the resources needed for computation offloading operations, there are many attributes included in B_k to process the task. Some of existing works just only consider B_k as the number of central processing units (CPU cycles) [25]. In another scenarios, GPU and memory requirements are considered during resource allocation for executing heavy and complex tasks such as the AI, and ML ones [49].

C. COMPUTATIONAL OFFLOADING MODELS

There are many models introduced in the literature to perform the computational offloading operations in the IFC systems. Depending on the application scenarios, the models are established appropriately to support the systems to achieve a single objective or multiple objectives simultaneously such as minimization of total energy consumption, minimization of offloading delay, and maximization of resource utilization, and fairness and balance of workload. Fundamentally,

an offloading model takes into account multiple factors including the system architecture, the task properties to derive efficient algorithms, that determine offloading locations, times to offload, and how a task is offloaded (how data of task is handled). In the following paragraphs, we summarize and discuss these relevant aspects to highlight the key features of popular offloading models in the literature.

Regarding the offloading locations, there are two major classes of offloading models including intra-layer and inter-layer offloading. The former refers to models that the offloading operations take place in the same layer, whereas the later involve multiple layers (e.g., between IoT and fog layer, between fog and cloud). Concretely, the computational offloading processes can take place only within a stratum of IFC systems where the computing devices in the same tier (e.g., the IoT, fog, and cloud tier) can share their available resources to handle the tasks cooperatively. Recently, the advance of technologies can equip with modern IoT devices more features regarding powerful resource, computing capability to process tasks locally. In combination with the emergence of device-to-device (D2D) communication technologies, the computational offloading between IoT devices is pervasive in the future IFC systems. In the same sense, the tasks can be offloaded within the fog layer and cloud layer, mainly to balance the workload as well as improve the resource utilization [50]. However, the heterogeneity of FN types exposes a challenge of communication between them. It requires unified middlewares and protocols to enable fog-to-fog communication and collaboration such as FRAMES developed in [51] to jointly offloading the tasks. Otherwise, FNs can communicate via a centralized agent such as FSP or brokers in their fog domains.

In most of application scenarios, the offloading processes involve multiple layers. For example, as per [15], a task generated by an IoT device can be processed by itself locally or offload to a FN or the cloud finally. The associated analysis reveals that the offloading locations for tasks should be flexible with respect to the task type to get the benefit of offloading operations. Concretely, the heavy tasks should be offloaded to the cloud tier, while the medium tasks are processed by FNs. In addition, the light tasks can be computed locally by IoT devices if they have sufficient resource or offloaded to FNs, otherwise. As the tasks can be splittable, one part of task can be processed by IoT node and the other by the fog or cloud. Finally, there exist several application scenarios, in which the upper layers require the lower layers to execute the task. Theses uncommon offloading models include cloud offloading to fog/IoT and end user devices, fog offloading to the IoT and end user devices for specific purposes of applications [18].

The determination of times to offload tasks is an important aspect in the offloading models. Generally, offloading is needed when TNs are unable to process the tasks locally, or processing them may not satisfy the QoS requirements. Although the modern IoT devices and end user equipment can process some types of tasks locally, the majority of tasks (e.g., complex and heavy tasks, and sporadic tasks emergency cases) generated in the IoT layer are offloaded to the upper layers. However, the task offloading incurs additional cost such as communication delay and energy consumption. Therefore, the offloading model requires an inclusion of mechanism to monitor the system performance, traffic flow rates, network conditions that can support to make the offloading decisions appropriately. For example, the FOGPLAN framework in [22] can provide the dynamic offloading strategies to adapt to the dynamic change of QoS requirements. By observing and analyzing the task processing queue of FNs constantly, tasks currently resided in the processing queues of these FNs must be offloaded to HNs if the predicted processing delays are no longer to meet the deadlines of tasks. The network reliability is also concerned in the fog networks since it directly impacts on the communication delay of offloading processes [21].

The offloading models also specify how the input data of tasks is offloaded and processed. Generally, a full offloading method is applied for a task when its whole data is indivisible and processed by a single HN. Conversely, as a divisibility of task is enabled, a partial offloading scheme can be used to offload a fractional part of task to HNs while the other part of task is processed locally by TN. In the most of studies, a task is assumed to be decomposed into two subtasks, thus there needs only one HN to offload the subtask. As the subtasks are totally independent, the task division is an effective technique employed in the offloading models to cope with the heterogeneity of computing device resources, and simultaneously improve the performance of computing operations. For example, according to the FEMTO model in the work [52], each task is divided into two subtasks with different data sizes, which are then processed by the IoT node and offloaded to the fog entity respectively. This method contributes to minimizing the energy consumption of task offloading while achieving the workload fair among the fog nodes and satisfying the deadline constraints of tasks. Similarly, the partial offloading is utilized in the task offloading models for the heterogeneous fog networks to reduce the task execution delay [26]. Dividing a task into multiple (more than two) subtasks is also considered in [48] to exploit the parallel computation of subtasks at different FNs. As analyzed in [26], compared to the full offloading model, the partial offloading offers more advantages in terms of delay reduction, energy saving, resource utilization, and workload balancing. The independence of subtasks enabling the parallel processing of subtasks is obviously a key to achieve these advantage. However, in practice, some or all subtasks of a tasks can exist a data dependency relation. For example, the output of a subtask can be an input data for another subtask. Thus, completing the task requires a subtask scheduling plan to with respect to the subtak processing order. This in turn can impact the performance of partial offloading models. For instance, as evaluated and analyzed in [48], a number of subtasks for a task can be optimized depending on the system context. In addition, not all tasks

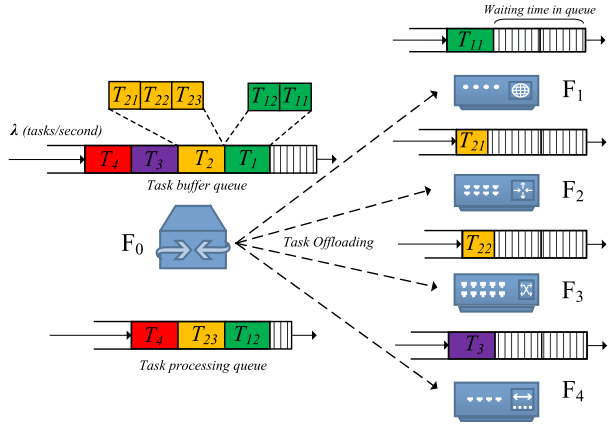

FIGURE 2. A dynamic computational offloading model is proposed in [48] that integrates partial and full offloading to balance the workload in the fog layer. The full offloading plan is used for task T_3, while the subtasks T_{11} of T_1, subtasks T_{21} and T_{22} of T_2 are offloaded partially by F_1, F_2 and F_3 respectively. T_4 is processed locally by F_0.

should be divided because more subtasks can probably lead to a coupling resource problem. An offloading framework in FRATO is then introduced based on many factors such as the FN resource status (e.g., queue status, computing capability), task request rates, and task properties (e.g., divisibility) to offer a dynamic offloading policy. As illustrated in Fig. 2, FRATO dynamically applies the partial offloading and full offloading modes for the tasks based on the queue status of FNs. In this way, FRATO is able to significantly reduce the offloading delay as well as improving the resource utilization, especially in cases of high rate of task requests. A similar investigation is presented in [51] that considers three models of task processing, in which the subtasks can be executed in sequential, parallel, and mixed processing order.

D. OPTIMIZATION PROBLEMS OF COMPUTATIONAL OFFLOADING

Denote $\mathbb{C} = \{\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k, \ldots\}$ as the set of objective functions, established by individual computing nodes (i.e., IoT nodes, FNs, or clouds) and by the system for the computational offloading performance at a given time. Some of typical objective functions concerned in the literature include total consumption energy, average task execution delay, total payment cost of resource usage, fairness and workload balancing index, and outage probability. Moreover, there also present objective functions of individual resources to indicate the inherent selfishness and rational of HNs. These kinds of objective functions are referred to as utility ones, which correspond to the benefits and revenues of available resource provision. Summarily, the generic optimization problem in the IFC systems can be represented in the following form:

$$P: \ \min(\mathbb{C}_i) \ \&/ \ \max(\mathbb{C}_j) \ \&/ \ \max(\mathbb{C}_k) \ \&/ \ \ldots \quad \text{s. t. Constraints.} \tag{1}$$

Depending on the application scenarios, the problem **P** can be in form of single or multi-objective model. Regardless the ultimate objectives of problems, the constraints involve the resource competition, resource limitations, and task scheduling. Concretely, a HN can receive multiple requests for task offloading. However, a certain number of requests are accepted to be processed owing to the limitation of resource such as limited buffer capacity, low residual energy. Furthermore, scheduling the tasks in HNs is considered to respect to the QoS requirements. From the global point of view, the problem becomes a combinatorial problem, which is proven to be NP-hard due to the natural presence of coupling resource problems [53]. Therefore, achieving the globally optimized solution is infeasible, especially in the large-scale systems. In addition, there is an extensive cost of overhead to collect the global information. These issues urge the need to design the distributed algorithms to support the computational offloading processes efficiently.

V. MATCHING-BASED MODELS FOR COMPUTATION OFFLOADING PROBLEMS

Many models and associated algorithms have been proposed to support distributed computation offloading processes in the fog-based environment. In the following, we review them according to the different types of matching models (i.e., one-to-one, one-to-many, and many-to-many) described in the previous section.

A. ONE-TO-ONE MATCHING

A task assignment problem is formulated in [54] to describe the computation offloading in vehicular fog networks (VFNs). In these networks, vehicles with available computing resources can act as vehicle FNs to offload tasks of user equipment (UEs); hence contributing to a reduction of overload on the base station (BS) during the peak time as well as improve QoS or QoE (e.g., delay). Given that sharing resources is conditioned naturally in the context of VFNs, a contract-based incentive mechanism is proposed to promote FNs to perform task offloading. Due to the heterogeneity of resource states, FNs are classified into a set of types, each has different contract item (i.e., reward) formulated by the BS. Considering the task offloading in a certain time slot, each UE has a task that must be offloaded by a certain FN; hence the task assignment problem in this context is treated as a one-to-one matching game. In addition, UEs rank the vehicles by using a preference function G that encapsulates the delay performance and resource pricing. Accordingly, as a task generated by UE U_n is offloaded and processed by a vehicle V_m, $G_{n,m}$ is defined by:

$$G_{n,m} = \frac{1}{D_{n,m}} - P_m, \tag{2}$$

where $D_{n,m}$ is the total delay of offloading, and P_m is the price for using the resource of V_m. P_m is zero initially, and will increase according to the price rising rule proposed in the price-based stable matching algorithm. The simulation based evaluation and analysis show that the proposed resource allocation and task assignment scheme

can achieve sub-optimal performance in terms of social welfare and offloading delay compared with the optimal solutions. Importantly, the matching-based solution offers lower complexity of computation because of the nature distribution of DA algorithm.

A one-sided one-to-one matching is applied in [55] to develop a distributed algorithm for task offloading in vehicular fog computing (VFC) environment. In the considered offloading scenario, tasks sent from user vehicles (UVs) can be offloaded either by nearby vehicles with residual resources acted as vehicle fog servers (VFSs) or by the remote edge server through road side units (RSUs). The inherent presence of information uncertainty in the vehicular network typically featured by time-varying state of vehicle-to-vehicle (V2V) channels for offloading, available resources and volatility of VFSs leads to a lack of global information in the server side (i.e., the edge server) to derive the globally optimal offloading solution. The matching theory is used in this situation to provide a stable and efficient alternative. Recall that, in each time slot, a task generated by UV can be offloaded to only one VFS, and each VFS is able to process at most one UV's task; hence the task offloading problem is transformed into one-to-one matching game. In addition, an online learning technique is augmented to cope with the information uncertainty by introducing upper confidence bound (UCB) concept inspired from reinforcement learning techniques. The fundamental of UCB is to estimate the future state based on the historical observations while considering the uncertainty of these data known as confidence bound (CB). Furthermore, to capture the volatility of VFSs, the concepts of occurrence awareness and matching conflict awareness are embedded in CB. Consequently, the authors propose a preference function used by a UV i to rank a certain VFS j at a time slot t, which is defined as

$$U_{i,j,t} = \frac{1}{D_{i,j,t-1}} + CB - H_j, \tag{3}$$

where $D_{i,j,t-1}$ is the historical offloading delay at time slot $t-1$, and H_j is price for using the resource of VFS j. The simulation analysis demonstrates that the proposed approach can efficiently alleviate the severe impacts of volatility and resource conflict. More importantly, it enables the system to obtain close-to-optimal delay performance compared to the case of global information availability.

An integration of Stackelberg game and matching game is formulated in [56] to study the task allocation in three-tier fog networks targeting in the patient health monitoring applications. Periodic tasks (i.e., patient health data analysis) and sporadic tasks (e.g., emergency case) are sent and requested from home agents (HA) to the cloud node (CN). In turn, CN assigns FNs to execute these tasks such that the task deadlines are met. Transfer is considered in this game for the interaction of HAs, CN, and FNs. With this configuration, the objective of system is to maximize the resource utilization while minimizing the outages. From the game perspective, the objective is to maximize the utilities of three players. To achieve these objectives simultaneously, the author divides the problem into three sub-problems. Accordingly, a pricing model is proposed to optimize the price of per resource unit, thus maximizing the utility of CN. Hence, HAs are based on the prices and the deadline constraint of task to derive the required resources such that it maximize the utilities of HAs. Finally, CN allocates FNs to HAs efficiently to maximize the utility of FN as well as maximizing the resource utilization and minimizing the outages. While the first two sub-problems are solved by Stackelberg game, the last is addressed using the one-to-one matching-based algorithm. The evaluation analysis shows that the proposed matching-based algorithms can handle the sporadic tasks with satisfied deadline compared to the greedy offloading solutions because the tasks are allocated in the more appropriated resources for computing. In addition, the proposed solution also offers an improved resource utilization at the fog computing nodes.

B. MANY-TO-ONE MATCHING

Most of matching-based algorithms are many-to-one types to apply for two distinct sets including the task set $\mathcal{T}$ and the computing device set $\mathcal{R} = \mathcal{I} \cup \mathcal{F} \cup \mathcal{C}$. With this model, a certain computing node in $\mathcal{R}$ can process multiple tasks, thus resulting in a many-to-one matching problem.

Motivated by the emergence of device-to-device communication (D2D) paradigm, modern IoT devices in the IoT networks can share and allocate available resources among themselves to enable interoperability of processes such as sensing and actuation tasks. A roommate matching model is developed in [57] for pairing these IoT nodes, which are deployed in the same fog domain. Based on the state of resources, each device can determine its quota, indicating number of IoT devices it can pair to share the resources. In addition, the utility function accounting for energy consumption and resource pricing cost is established and measured to use in the preference list construction of nodes. The Irvings matching algorithm [58] is applied and refined to endure a stable pairing between IoT devices. In particularly, compared to the pairing model between IoT node with access points (APs), the stable pairings of IoT nodes in the same domain gains more benefits in terms of energy consumption reduction, and resource utilization improvement compared to the random and greedy pairing approaches. Such the benefits are resulted in from the utilization of IoT nodes with available resources, which can take charge as computing nodes to serve tasks appropriately. The matching-based mechanism ensures to construct a stable matching between pairs of IoT nodes as well as pairs of IoT node and AP without resource conflicts.

The work [59] considers the task offloading carried out by a set $\mathcal{F}$ of FNs such as APs, routers, switches. The set of IoT nodes in the IoT layer generates a corresponding set $\mathcal{T}$ of computation tasks, which belong to different types of IoT applications. Recall that there is a limited number q of tasks

offloaded by a certain FN due to the limitation of resource (i.e. computing, buffer storage). Therefore, the task offloading problem can be viewed as a many-to-one matching game between these two sets. According to [59], the agents of sets construct the preference lists based on utility functions, which account for the communication cost, waiting time in queues, and the execution delay of task. Accordingly, the utility of a task $t \in \mathcal{T}$ is calculated as follow:

$$U_t^f = \frac{1}{D_{tf}^c + W_{tf}}, \tag{4}$$

where D_{tf}^c is the communication delay cost required to transmit the task t from the IoT device to the fog node f, and W_{tf} is the expected waiting time in the queue of f before the task is being processed. In the other side, the agent f of computing resource can obtain the utility according the following equation.

$$U_f^t = \frac{1}{D_{tf}^c + W_{tf} + D_{tf}^{ex}}, \tag{5}$$

where D_{tf}^{ex} is the execution delay to complete the task t by f.

The association of waiting time into the utility function leads to the presence of externalities of matching problem, in which the preference lists of agents can change after a pair is matched. Therefore, the DA-based algorithm requires a cost of overhead resulted in from the exchange of control packets among FNs to adjust the decision makings (i.e., acceptance or rejection of proposals) over iterations. With this approach, the outcome of matching game is to achieve a two-sided exchange-stable (2ES) matching, which handles the externality efficiently. The simulation results show that the proposed algorithm outperforms greedy and random offloading solutions in terms of worst total completion time, mean waiting time per task, mean total time per tasks, and fairness.

The work [60] studies a dynamic task offloading combining the partial and full offloading in the fog-cloud networks to minimize the total energy consumption. In this model, a computation task can be processed locally by TN or offloaded by HN or by the cloud server. In addition, the tasks can be divided into multiple independent subtasks, which can be processed in parallel by HNs and the cloud. Virtually, at a certain time of offloading decision making, the system is modeled by two sets of agents including the set of TNs $\mathcal{T}$ and the set $\mathcal{H}$ of helpers including HNs and cloud. Each helper $H_i \in \mathcal{H}$ constructs its PL based on a service efficiency (SE) indicating the channel quality (i.e., transmitting data rate) from TNs to it, whereas EE (Energy Efficiency) is used by TNs to rank the agents of helper set. In mathematical form, $EE(k, i) = R_{k,i}/P_{k,i}$, where $R_{k,i}$ is the CPU computation capability and $P_{k,i}$ is the computation power when a task T_i is offloaded by a helper H_i. The work then proposes SMETO algorithm based on the DA procedure and the constructed PLs to achieve the one-to-many stable matching between $\mathcal{T}$ and $\mathcal{H}$. Evaluated by the simulation analysis, the outcome of matching shows its benefit in reducing significantly the offloading consumption energy compared to the random approach.

A task offloading framework known as METO is presented in [61] aiming to reduce the total energy consumption and overall full offloading delay in the IoT-fog network. In this network model, each IoT device generates a single task, and the resource of each fog node (FN) is represented by a number q of virtual resource units (VRU). In addition, there is no local computing enabled the IoT nodes, thus the tasks are offloaded by the fog nodes. This offloading model leads to a form of one-to-many matching problem between the IoT device set $\mathcal{I}$ or the corresponding task set $\mathcal{T}$ and the fog node set $\mathcal{F}$, in which q_i is the quota of agent $F_i \in \mathcal{F}$. As considering jointly multiple criteria (i.e., energy consumption minimization and delay minimization) for the offloading decision-making, METO employed a hybrid CRITIC and TOPSIS-based technique to produce the preferences of both sets. CRITIC (criteria importance through inter criteria correlation) is used to evaluate the criteria and determine the weights of resource allocation strategies, whereas TOPSIS (technique for order of preference by similarity to ideal solution) uses these weights for ranking the agents of opposite sets. With this approach, the produced preference lists are strict, complete and transitive, therefore ensuring to obtain the stable matching. Using the simulation-based comparative analysis, METO shows its advantage in reducing the total consumption energy as well as the overall delay compared to the baseline algorithms including ME [59], SMETO [60], and a random resource allocation policy.

Similar to METO, a one-to-many matching model between the task set $\mathcal{T}$ and the FN set $\mathcal{F}$ is used in [62] to seek for an efficient task offloading algorithm in the IoT-fog systems. However, the system considers the presence of fog service providers (SPs), each of which manages the resources of fog nodes in its domain. Consequently, the task offloading problems is transformed into a student project allocation (SPA) game [63], in which IoT devices (or tasks), FNs, and SPs correspond to students, projects, and lectures respectively. To obtain the multi-objectives of system is challenging due to the selfishness and rational of individual agents. Alternatively, while the objective of IoT device is to minimize the offloading delay as well as the consumption energy, SPs aim to maximize the hosting cost and minimize the outages (i.e., number of tasks exceeding the their deadlines). The work further presents a DA-based distributed task offloading algorithm called SPATO to tackle the challenge. In particular, the preference lists of agents are constructed using the analytical hierarchy process (AHP) [64] that accounts for multiple criteria of system wise objectives to obtain the rankings. The simulation results indicate that the proposed algorithms enable the network to achieve a reduced offloading delay and energy consumption as well as minimum outage compared to the random offloading plan and SMETO [60].

In the same consideration of task offloading problem as studied in [62], an efficient offloading algorithm called LETO is proposed in [65], aiming at balancing the workload of FNs. A one-to-many matching model between a task set $\mathcal{T}$ and a FN set $\mathcal{F}$ with minimum and maximum quotas is formulated to access the impact of resource capability of FNs on the workload distribution strategy. In addition, with respect to the deadlines of tasks, the PLs of TNs and FNs are constructed based on the expected offloading delay and the deadlines, respectively. Basically, $f_i \succ_{t_k} f_j$ if $D_{k,i} < D_{k,j}$, where $\{f_i, f_j\} \in \mathcal{F}$, $t_k \in \mathcal{T}$, and $D_{k,i}$ is the total offloading delay if t_k is processed by f_i. In the other side, $t_i \succ_{f_k} t_j$ if $d_i < d_j$, where d_i is the deadline of t_i. The work then introduces a multi-stage deferred acceptance algorithm (MSDA) to achieve the fair and pareto-optimal matching. Based on the simulation analysis, LETO is able to balance the workload of FNs efficiently while minimizing the outages.

The work [47] introduces an algorithm abbreviated by DATS for offloading dispersive tasks in the fog and cloud networks. Given the presence of TNs and helpers (i.e., coalition of FNs and cloud) in the network, the tasks can be processed by either partial or full offloading mode dynamically. In particularly, a task can be splitted into multiple subtasks, which are then processed by different helpers in parallel to reduce the overall task execution delay. DATS incorporates two algorithms to achieve the objective of task offloading minimization, which are progressive computing resource competition (PCRM) and synchronized task scheduling (STS). Concretely, PCRM is a one-to-many matching-based algorithm to yield an efficient resource allocation strategy between task set $\mathcal{T}$ and resource set $\mathcal{H}$ of helpers. A new index called processing efficiency (PE) is defined to support the production of preference profiles for the helpers. PE encapsulates communication and computation delay to examine the delay-oriented performance of resource allocation strategy. Recall that PE is calculated as follows for fog a FN m and the cloud k when they execute a task T_n:

$$PE(n, m) = \frac{1}{r_{n,m}} + \frac{\eta_n}{f_m} + \frac{\mu_n}{r_{m,n}}, \tag{6}$$

where $r_{n,m}$ is data rate from TN n to FN m, η_n is processing density of T_n, f_m is CPU frequency of FN m, and μ_n is output-input ratio of T_n.

$$PE(n, k) = \frac{1}{r_n^t} + \frac{\eta_n}{f_k} + \frac{\mu_n}{r_n^r}, \tag{7}$$

where r_n^t, r_n^r are transmitting and receving data rate from TN n to the cloud. Whereas, TNs rank the agents of helpers based on the QoS that helpers can provide. Alternatively, a TN prefers to match with a helper which minimizes the offloading delay. Second, STS algorithm is proposed to optimize the subtask assignment and scheduling for each task given the matching obtained by PCRM. The extensive simulation analysis is conducted to evaluate the performance of DATS under the impact of many factors including task size, quota of helpers, and network bandwidth. Summarily, DATS can significantly reduce the task offloading delay compared to random and greedy offloading policies.

Another one-to-many matching game is modeled in [66] for assigning the fog resources to serve the requests sent from the end users (EUs) in the IoT networks. Considering the minimum and maximum quotas of FNs (i.e., the minimum and maximum number of EUs that a fog can serve), a multi-stage differed acceptance (MSDA) algorithm is developed to adjust the resource allocation strategies to reach the stable matching. EUs are based on QoS metrics (i.e., response latency) provided by FNs to derive PLs, whereas FNs take into account the fog load distribution to rank EUs. The outcome of matching allows an efficient assignment, which minimize the delay experienced by users while balancing the load of FNs as compared to the random and greedy resource allocation approaches.

A problem of allocation of FSPs to IoT devices is studied in [67]. Taking into account the heterogeneity of system, the IoT devices are assumed to have different services requested periodically. Likewise, FSPs vary in terms of services that they can provide. This configuration is equivalent to a many-to-one matching model, where some FSPs can serve multiple IoT devices. In particular, incomplete PLs with ties are produced by the agents of both sides. That is because some FSPs are absent in the PL of a certain IoT node if they have no services being requested by the IoT node. Meanwhile, some IoT nodes may have the same raking positions in a PL of a certain FN if they requests the same services. Furthermore, the service access time duration (i.e., long or short) is restricted by FSPs. A truthful and Pareto optimal mechanism is employed to achieve the stability of matching. Through the achieved matching, the SPs can allow short or long access to IoT devices efficiently to use the non-money services respecting to deadlines of tasks. In addition, compared to the random allocation strategy, the proposed approach enables to achieve the maximized best allocation, in which the maximum number of IoT devices are served by the best FSPs in their PLs.

Considering the provision of content and services in the fog-based system, the works [68], [69] employ the one-to-many matching model to formulate the resource allocation problem. The requests sent from EUs are handled by FNs or cloud server depending on type of requests (i.e., content retrieval or computing). The caching technology is employed in the fog layer to accelerate the data and service access for EUs. In the form of two-sided matching game, EUs and FNs construct their PLs based on different preference functions. Concretely, an EU ranks a FN the best if it can provide the content and service with the minimum latency. Meanwhile, a FN prefers to serve the request of EU with minimum energy consumption incurred. Based on simulation analysis, the proposed algorithm demonstrates its benefit when improving the cache hit ratio, reducing the energy consumption, and service response delay.

Similar to [66], the work [70] introduced a FoGMatch framework to perform the task scheduling in the IoT-Fog network. In the absence of cloud servers, FNs are responsible to receive and process the tasks requested from the IoT devices. In addition, the limitation of fog resources in terms of CPU and RAM allows a certain number of tasks processed by a single fog at a scheduling interval. With this system configuration, the one-to-many matching game between the IoT set and FN set is applied model and study the resource allocation problem in the fog stratum. Link quality (i.e., data rate) and required resource for computing are two measurements used by the IoT nodes and FNs respectively to rank the agents in the opposite sets. In other words, an IoT node i ranks a FN j the best if the data rate from TN i to FN j is the highest and FN has the maximum available resource. In other side, a FN prefers to serve a task if executing it consumes the largest amount of available resource. Compared to Min-Min and Max-Min optimal scheduling approaches, FoGMatch shows its advantage in improving the makespan of IoT service execution, and resource utilization of FNs.

The work [71] focuses on minimizing the total energy consumption during computation offloading processes for cache-enabled F-RANs. In the considered system, all task requests are sent from UEs to the centralized cloud through FAPs. Then, the cloud is responsible for deciding simultaneously EU-FAP association and task offloading strategies (i.e., which tasks are processed by FAPs, cloud) to achieve the systematic objective. Recall that the objective is constrained by the resource limitation of FAPs (i.e, computing, storage) and deadlines of tasks. By modeling the UE-FAP association problem as a one-to-many matching game, a greedy algorithm based on the DA procedure is designed. PLs of agents of both sides are con constructed based on energy consumption measurement. Hence, a swap matching condition is introduced as a constraint to evaluate the stability of matching. The work further proved that the proposed algorithm can achieve the stable matching when there is no presence of blocking pair or swap matching in the outcome of matching game. Through evaluating the algorithm by simulation approaches, the results show that consumed energy of network can be reduce significantly through efficient and stable EU-FAP association, thus enabling the green F-RANs.

A one-to-many matching game is established to model the association problem of fog network, in which each IoT node (user) can be associated with only a cloudlet while a cloudlet can have multiple IoT nodes matched with it [72]. However, there a limited number of IoT nodes connecting to a cloudlet to respect its maximal workload. In addition, the presence of wireless interference between IoT nodes located in proximity regions when connecting wirelessly to the cloudlets indicate external effect in the matching game. This externality makes PLs of agents change whenever a pair of IoT node and cloudlet is matched. The work introduces a concept called swap matching to handle externalities and then achieve the stable matching outcome. The extensive simulation are provided to show the benefits of proposed algorithms, that include the latency minimization and throughput enhancement compared to the random association approach.

The work [73] concerns the joint optimization problem of radio and computational resource allocation in the IoT-Fog computing systems to optimize the system performance in terms of service delay, user satisfaction, and system cost. Such the problem involves three entities including the set of IoT EUs, FNs, and CSPs (which manage the resources of FNs). From the matching perspective, the mutual interaction of these sets can be modeled in a SPA problem since they corresponds to students, projects, and lectures respectively. To handle the external effect, a procedure called user-oriented cooperation (OUC) is developed. Fundamentally, OUC is a strategy to remove possible blocking pairs in the matching given the presence of externality by swap operations, which evaluate the change of utility values of agents. As a swap is applied for any two current pairs, and the corresponding utility values are changeable, the two pairs is considered as blocking ones. With this way, the proposed algorithm can achieve the stable matching with an addition cost of computation complexity resulted in from the swap operations. Regarding the performance, the simulation results show that the proposed framework enables the system to achieve low service latency as well as minimized outages.

A two-sided matching model is proposed [74] for data stream processing. Applying the micro-services to server the DAG-based stream processing applications, the matching is configured to allocate the micro-services to fog and cloud computing resource. Regarding the preference relation construction, the micro-service ranks the resources based on their processing time. In addition, the resources rank the miro-service according to their residual bandwidth. The stable matching achieved by the DA algorithm offers the mutual benefits for two sides (i.e., micro-service side and resource side). The simulation results demonstrate that the proposed matching mechanism can help the system to reduce significant processing time of stream while lowing the total stream traffic traversed through the fog-fog and fog-cloud paths.

C. MANY-TO-MANY MATCHING

The work [75] integrates the Stackelberg game and matching game to study the computing resource allocation problem in three-tier IoT fog networks. The considered network consists of multiple clusters, each includes a set of FNs and is managed by a centralized data service operator (DSO). These FNs are responsible to provide resources to serve the services requested by data service subscribers (DSSs) such as mobile phones, and IoT devices such that QoS in terms of service delay is satisfied. The work first models the interaction of DSOs and DSSs as a Stackelberg game, in which DSOs are leaders and DSSs are followers. Based on the resource price announced by the leaders, the followers can optimize the resource amount measured by number of CRBs required

to achieve the desired QoS. When the optimal resources demanded by DSSs are determined, the framework is to come to resource allocation problems. Given the determined CRBs and the available resources of FNs, the many-to-many matching game is applied to model the interaction of DSOs and FNs. Finally, for each cluster of FNs owned by a DSO, the resource allocation is investigate to assign appropriate CRBs of FNs to serve the requests of DSSs. This problem is modeled as a many-to-many matching game, which aims at maximizing the utility values of FNs and DSSs. The performance of framework is evaluated by simulation scenarios, which further show that the proposed approach is able to maximize the utility of all entities (i.e., DSOs, DSSs, and FNs) while satisfying the QoS demanded by DSSs.

The work [76] studies the problem of placement of virtual functions (VFs) on FNs such that they can serve IoT applications with maximal QoS (i.e., minimized worst application time and outage probability). In this considered scenario, VFs are referred to as software, middle ware that can perform the computation, storage, and networking tasks. In addition, each application is composed by a set of atomic services (i.e., tasks), which must be processed in sequential order (i.e., chain) such as following a sense-process-actual workflow. A many-to-many matching game is applied to model the placement problem of a set of VF types ($\mathcal{V}$) on a set of FNs ($\mathcal{F}$). Concretely, a FN can contain multiple types of VFs depending on the available resource of FN represented by computing resource blocks (CRBs), and each VF type can appear in different FNs. The work then introduces two utility functions for the agents of both sets to support the creation of PLs. In the side of VF set, the utility function $U_z(f)$ of a VF $z \in \mathcal{V}$ is formulated as follow when placed on a FN f.

$$U_z(f) = r_f - r_z, \tag{8}$$

where r_f is the available CRBs on FN f and r_z is the number of CRBs required to load the VF z on the FN f. Based on this function, the order for any two FN f and f' in the PL of VF z is as follow: $f \succ_z f' \Leftrightarrow U_z(f) < U_z(f')$. In the other side, the utility function $U_f(z)$ of FN f takes into account the occurrence probability of VF z that appears in the FN set. Accordingly, $U_f(z)$ is calculated by:

$$U_f(z) = h_z(1 - \sum_{f \in \mathcal{F}} \frac{\tau_{f,z}}{n}), \tag{9}$$

where h_z is the occurrence frequency of VF z in the IoT application set $\mathcal{A}$, $\tau_{f,z} = 1$ if v_z is placed on FN f; otherwise $\tau_{f,z} = 0$, and n is the total number of types of VFs in the network. Based on this utility function, the preference relation of two certain VF z and z' ranked by FN f as follow: $z \succ_f z' \Leftrightarrow U_f(z) < U_f(z')$. In other words, FNs prefer to allocate VFs such that they have higher values of occurrence frequency in the set $\mathcal{A}$ [76]. With this PL construction, the work applies the DA procedure to sketch out the distributed VF placement algorithm named blind matching game (BMG). Additional analysis is provided to show that the proposed algorithm can lead to a stability convergence, at which the outcome of matching game is in form of strictly-two-sided exchange-stability (S2ES). Alternatively, there is no existence of blocking pair or swap matching which can break the current matching pair to change positively the utility values of agents. Consequently, at the stability, BMG can achieve a sub-optimal performance in terms of minimization of the work application completion time and the outages. In addition, it also outperforms other baseline algorithms including the random and greedy offloading approaches.

A recent study as per [77] investigates the data offloading problem in the fog network, in which FNs are responsible for receiving and then processing the data periodically transmitted by the subscribed IoT devices. Given the heterogeneity of FN resource, FNs can be classified into two sets: $\mathcal{S}$ of surplus FNs (SFNs) and $\mathcal{D}$ of deficit FNs (DFNs). As these definitions, SFNs have available resources for data processing, while DFNs are characterized by the lack of resource for handling the requests of their subscribers. In addition, each FN is managed and owned by an unique FSP, thus leading to the nature of selfish and rational, indicating that it tends to maximize the its own profit without considering the system wise performance maximization. The objective focused in the paper is to design an efficient offloading policy such that it can maximize the monetization of FNs subject to the required QoS. To achieve the objective, a matching-based algorithm is proposed that model the interaction of SFNs and DFNs as a many-to-many matching game without quotas. In this game, the agents of both sets produces their PLs based on their own utility functions. For each pair of FNs ($f_s \in \mathcal{S}, f_d \in \mathcal{D}$), their utility functions are formulated as follow if they match.

$$U_{f_s}(f_d) = K_d . x_d, \tag{10}$$

where K_d is the maximum data packets that can be offloaded to f_s from f_d, and x_d is normalized payoff received by f_s when processing a data packet offloaded from f_d. Therefore, $U_{f_s}(f_d)$ refers to the total profit received by f_s when offloading the data packets from f_d. Meanwhile, f_d is interested in the maximum number of packets that can offload to f_s. In other words, the utility function of f_d is $U_{f_d}(f_s) = K_d$. For any two DFNs f_d and $f_{d'}$, their preference relation determined based on the utility function values is represented as follow: $f_d \succ_{f_s} f'_d \Leftrightarrow U_{f_s}(f_d) > U_{f_s}(f_{d'})$. Similarly, for two SFNs f_s and $f_{s'}$, $f_s \succ_{f_d} f_{s'} \Leftrightarrow U_{f_d}(f_s) > U_{f_d}(f_{s'})$. Based on these PLs, the proposed DA based algorithm is designed to achieve for a stable matching. The analysis is provided to prove its stability convergence, thus ensuring the feasibility and efficiency of algorithm. The primary simulation results show that the proposed algorithm is able to maximize the monetization while satisfying the demanded QoS of the subscriber users (i.e., total latency of data packer processing).

VI. CHALLENGES AND OPEN RESEARCH ISSUES

The review as discussed in the previous section exposes the potential of matching theory in solving the computational offloading problems in the distributed manner. Many models

TABLE 3. Summarization of the matching theory-based solutions for distributed computation offloading in the IFC systems.

Matching Model	Reference	Sets of Players	Technical Solution & Features	Objective
One-to-One	[54]	UEs & FNs	- Matching with transfer - Integration of pricing mechanism to update PLs - Contract and matching integration	- Delay Minimization - Utility Maximization
	[55]	UVs & VFSs	- Matching with transfer - Integration of pricing mechanism to motivate resource sharing - Deal with information uncertainty using online learning & UCB	- Offloading delay minimization - Outage reduction
	[56]	HAs & FNs & CN	- Integration of Stackelberg & Matching game - Incorporation of mechanism for resource pricing - Set priority to sporadic tasks during offloading	- Resource utilization maximization - Minimization of outages
Many-to-One	[57]	IoT nodes	- Roommate matching problem with quota - Application of Irvings matching algorithm to endure a stable pairing	- Reduction of D2D energy consumption - Increase of resource utilization
	[59]	IoT nodes & FNs	- Matching with externalities - Two-sided exchange stable (2ES) matching	- Reduction of worst completion time - Reduction of mean waiting time per task - Reduction of mean time per task - Improvement of fairness index
	[60]	TNs & Helpers	- Dynamic plan with partial & full offloading - PLs are achieved based on EE & SE	- Minimization of energy consumption
	[61]	IoT nodes & FNs	- CRITIC technique is used to determine weights of multiple criteria - TOPSIS mechanism is used to create PLs	- Minimization of energy consumption - Minimization of outages
	[62]	IoT nodes & FNs & FSPs	- SPA matching problem - Using AHP to rank the opposite agents in PLs	- Minimization of energy consumption - Reduction of offloading delay - Minimum outages
	[65]	IoT nodes & FNs	- One-to-many matching with minimum & maximum quota - Multi-state deferred acceptance algorithm (MSDA) - Pair & Pareto-optimal task assignment	- Load balancing - Minimization of outages - Reduction of Offloading delay
	[47]	IoT nodes & Helpers	- PCRM: Stable matching-based for resource allocation - STS: Optimal scheduling for subtasks of each task	- Reduction of offloading delay - Resource utilization improvement
	[66]	EUs & FNs	- One-to-many matching with minimum and maximum quotas - MSDA algorithm	- Balancing the fog resources - Low response delay
	[67]	IoT devices & FSPs	- Matching without transfer - Incomplete PLs with ties - Develop a truthful and Pareto optimal mechanism to reach the matching stability	- Improvement of resource utilization - Efficient resource access with respect to long and short duration
	[68], [69]	EUs & FNs	- Employment of caching technology - Two-sided matching with incomplete PLs	- Improving cache hit ratio - Reducing energy consumption - Reducing service response delay
	[70]	IoT devices & FNs	- PLs are constructed based on consolidation polity - PLs are strict, transitive, and complete without ties	- Reduction of makespan - Improvement of resource utilization
	[71]	UEs & FAPs & Cloud	- Joint EU-FAP association & optimal task offloading - Integration of swap matching condition to ensure the stability	- Minimization of energy consumption - Green F-RANs
	[72]	IoT nodes & Cloudlets	- Matching with externalities - Integration of swap matching condition to achieve the stable outcome	- Minimization of latency - Enhanment of throughput
	[73]	IoT nodes & FNs & CSPs	- SPA problem - Matching with externality - Introduction of user-oriented cooperation (OUC) procedure to handle the externalities	- Low service latency - Minimized outages
	[74]	Microservices & Fog and Cloud	- DAG computation tasks - Two-sided Matching - DA algorithms for achieving the stable matching	- Low service latency - Minimized stream traffic
Many-to-Many	[75]	DSOs & FNs & DSSs	- Stackelberg game between DSOs & DSSs - Many-to-many model between DSOs & FNs - Many-to-many model between DSSs & FNs - Matching with transfer	- Maximization of utility - Improvement of resource utilization
	[76]	VFs & FNs	- Matching with externality - Strictly-two sided exchange-stability (S2ES)	- Minimizing the worst application time - Minimizing the outage probability
	[77]	SFNs & DFNs	- Matching with transfer and without externalities - Matching without quotas	- Maximize monetization of FNs - Improvement of resource utilization - Minimization of outages

and algorithms have been introduced to apply for different computing scenarios. However, there still exist several challenges appeared in new context of IFC systems. These section explores and investigate such issues. The associated research directions also are included to discuss the full potential of matching theory to address the issues.

A. MATCHING WITH DYNAMICS

In many application scenarios, the matching model should consider the dynamic of environment such as the mobility of fog devices, time-varying tasks. In these contexts, the preferences of agents might change accordingly at each time scheduling interval. Consequently, the time dimension must be accounted for in designing the matching solution.

B. MATCHING WITH GROUPS

In many scenarios, a group of players of a set prefers to match with a single agent of the other set. For example, a groups of tasks with the same type should be processed by fog devices supporting to process this type specially. That will lead to the task placement problem.

A similar issue might also appear in the federated fog based systems, where many domains (clusters) of fog networks are connected and a groups of IoT nodes in a certain domain prefers to be processed by the fog networks of other domain.

C. MATCHING WITH EXTERNALALITY

The nature of resource competition in the computing environment potentially leads to externalities in the matching problem, which are not investigated widely in the existing literature. The interference is only a factor making the continuous change of PLs of agent [78] in the fog and edge computing environment. For example, in the many-to-one matching model, the scheduling of tasks at a single fog can be served as an external that impact on the consistence of PLs. To the best of our knowledge, there has been no research works in the literature considering this kind of externalities in modeling the matching problems and designing the matching-based algorithms.

The presence of sporadic tasks is added as an external source since it can make the task scheduling plan change. In some scenarios, it can be addressed by offloading these tasks to the cloud. However, as the the clouded-based solution is inappropriate, FNs are considered to be alternatives to process the offloaded tasks. This situation may result in a change of PLs of some agents since the scheduled tasks must be postponed.

There are additional sources acting as externalities in may contexts of computing systems. Common ones include the system fault, network unreliability, which directly impact on the task offloading operations. Equivalently, PLs are immediately changed in these contexts because, for example, some HNs in the PLs are inaccessible. Thereby, there requires matching models that take into account these situation to enable the system reconfigure responsively.

D. SECURITY AND PRIVACY OF DATA AND END USERS

The heterogeneity and distributed nature of fog computing environment poses potential risks regarding security and privacy of data and EUs. Therefore, the choice of offloading locations is not only to achieve the improved performance but also guarantee reliability, security, and privacy criteria. This aspect has been not considered during constructing PLs in the reviewed studies, that, in other hand, open future directions.

E. NEW OFFLOADING APPLICATION SCENARIOS

All the reviewed works consider that the computation tasks can be totally processed in either parallel or serial manner. In many practical applications, the computation tasks is more complicated such as DAG tasks (Directed Arched Graph), which require a complex framework for scheduling since there exist parallel and serial computation processes [79]. Typical DAG tasks are related to the modern AI and ML applications such as real-time video processing [80], and automation in the industrial internet [81]. The presence of scheduling complexity can be considered as an external effect impacting directly on the consistence of PLs of agents.

F. APPLICATION OF AI AND ML-BASED TECHNIQUES

AI and ML tools provide efficient techniques to analyze and predict the statues of system accurately. Reinforcement learning is a such kind of techniques [82], [83], which can help to build PLs efficiently through online learning mechanism (i.e., exploitation and exploration). Thus, using these in the context of computational offloading enable the system to make dynamic and efficient offloading decisions. In addition, deep learning (DL) can be used to approximate and examine the matching outcomes [84].

VII. CONCLUSION

The matching theory has been widely applied to offer distributed algorithms in scenarios, where the optimal solutions are infeasible or feasible with incurring expensive expenditure and high computation complexity by the centralized global optimization approaches. The intrinsic feature of architecture of the IFC systems characterized by a geographic distribution of computing devices over large-scale exposes the suitability of matching-based distributed algorithms for perform the computation offloading and resource allocation-related problem. This paper surveys the literature regarding the matching theory-based solutions for distributed computing offloading in the IFC systems. Based on a brief description of matching theory, related concepts and matching models are identified and differences among them are presented. These different models are used to critically review the application scenarios and algorithms proposed in the existing literature in the area of computational offloading. The remaining challenges and corresponding open issues are discussed thoroughly to motivate research directions.

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] Y. Saleem, N. Crespi, M. H. Rehmani, and R. Copeland, "Internet of Things-aided smart grid: Technologies, architectures, applications, prototypes, and future research directions," *IEEE Access*, vol. 7, pp. 62962–63003, 2019.

[3] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Industrial IoT data scheduling based on hierarchical fog computing: A key for enabling smart factory," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4590–4602, Oct. 2018.

[4] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "The Internet of Things for logistics: Perspectives, application review, and challenges," *IETE Technical Review*, pp. 1–29, 2022.

[5] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "Toward the Internet of Things for physical internet: Perspectives and challenges," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4711–4736, Jun. 2020.

[6] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.

[7] H. Tran-Dang and D.-S. Kim, "An information framework for Internet of Things services in physical internet," *IEEE Access*, vol. 6, pp. 43967–43977, 2018.

[8] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. 5th Int. Joint Conf. INC (IMS IDC)*, 2009, pp. 44–51.

[9] A. Botta, W. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st, Ed., MCC Workshop Mobile Cloud Comput. (MCC)*, 2012, pp. 13–16.
[11] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Mar. 2018.
[12] Y.-J. Ku, D.-Y. Lin, C.-F. Lee, P.-J. Hsieh, H.-Y. Wei, C.-T. Chou, and A.-C. Pang, "5G radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 46–52, Apr. 2017.
[13] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2022.
[14] T. Q. S. Quek, M. Peng, W. Yu, and O. Simeone, *Cloud Radio Access Networks: Principles, Technologies, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[15] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, Apr. 2018.
[16] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, 2016.
[17] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.
[18] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, Oct. 2018.
[19] H. Tran-Dang and D.-S. Kim, "FRATO: Fog resource based adaptive task offloading for delay-minimizing IoT service provisioning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 10, pp. 2491–2508, Oct. 2021.
[20] K. H. Abdulkareem, M. A. Mohammed, S. S. Gunasekaran, M. N. Al-Mhiqani, A. A. Mutlag, S. A. Mostafa, N. S. Ali, and D. A. Ibrahim, "A review of fog computing and machine learning: Concepts, applications, challenges, and open issues," *IEEE Access*, vol. 7, pp. 153123–153140, 2019.
[21] J. Yao and N. Ansari, "Fog resource provisioning in reliability-aware IoT networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8262–8269, Oct. 2019.
[22] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, and J. P. Jue, "FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5080–5096, Jun. 2019.
[23] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.
[24] G. Lee, W. Saad, and M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2244–2258, Apr. 2019.
[25] K. Guo, M. Sheng, T. Q. S. Quek, and Z. Qiu, "Task offloading and scheduling in fog RAN: A parallel communication and computation perspective," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 215–218, Feb. 2020.
[26] Z. Liu, Y. Yang, K. Wang, Z. Shao, and J. Zhang, "POST: Parallel offloading of splittable tasks in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3170–3183, Apr. 2020.
[27] Y. Yang, Z. Liu, X. Yang, K. Wang, X. Hong, and X. Ge, "POMT: Paired offloading of multiple tasks in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8658–8669, Oct. 2019.
[28] S. Durand and B. Gaujal, "Complexity and optimality of the best response algorithm in random potential games," in *Proc. Symp. Algorithmic Game Theory (SAGT)*, 2016, pp. 40–51. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01404643
[29] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.
[30] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1826–1857, 3rd Quart., 2018.
[31] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: A stochastic-based perspective," *J. Grid Comput.*, vol. 18, no. 4, pp. 639–671, Dec. 2020.
[32] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107496.
[33] Z. Han, Y. Gu, and W. Saad, *Matching Theory for Wireless Networks*. USA: Springer, 2015.
[34] E. A. Jorswieck, "Stable matchings for resource allocation in wireless networks," in *Proc. 17th Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2011, pp. 1–8.
[35] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 82–95, Jan. 2011.
[36] F. Pantisano, M. Bennis, W. Saad, S. Valentin, and M. Debbah, "Matching with externalities for context-aware user-cell association in small cell networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 4483–4488.
[37] D. Gusfield, "The structure of the stable roommate problem: Efficient representation and enumeration of all stable assignments," *SIAM J. Comput.*, vol. 17, no. 4, pp. 742–769, 1988.
[38] C.-K. Lam and C. G. Plaxton, "On the existence of three-dimensional stable matchings with cyclic preferences," *Theory Comput. Syst.*, vol. 66, no. 3, pp. 679–695, Jun. 2019.
[39] K. Eriksson, J. Sjöstrand, and P. Strimling, "Three-dimensional stable matching with cyclic preferences," *Math. Social Sci.*, vol. 52, no. 1, pp. 77–87, Jul. 2018.
[40] K. Iwama and S. Miyazaki, "A survey of the stable marriage problem and its variants," in *Proc. Int. Conf. Informat. Educ. Res. Knowl.-Circulating Soc. (ICKS)*, 2008, pp. 131–136.
[41] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.
[42] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, Jul. 1962.
[43] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
[44] V. Bahl, "Emergence of micro datacenter (cloudlets/edges) for mobile computing," Microsoft Devices & Netw. Summit, USA, Tech. Rep. 56, 2015, vol. 5.
[45] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Comput. Netw.*, vol. 130, pp. 94–120, Jan. 2018.
[46] A. Ahmed, H. Arkian, D. Battulga, A. J. Fahs, M. Farhadi, D. Giouroukis, A. Gougeon, F. O. Gutierrez, G. Pierre, P. R. Souza, M. A. Tamiru, and L. Wu, "Fog computing applications: Taxonomy and requirements," 2019, *arXiv:1907.11621*.
[47] Z. Liu, X. Yang, Y. Yang, K. Wang, and G. Mao, "DATS: Dispersive stable task scheduling in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3423–3436, Apr. 2019.
[48] H. Tran-Dang and D.-S. Kim, "Impact of task splitting on the delay performance of task offloading in the IoT-enabled fog systems," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 661–663.
[49] S. Bian, X. Huang, Z. Shao, and Y. Yang, "Neural task scheduling with reinforcement learning for fog computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
[50] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2017.
[51] M. Al-Khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via fog-2-fog collaboration," *Future Gener. Comput. Syst.*, vol. 100, pp. 266–280, Nov. 2019.
[52] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M.-T. Zhou, "FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4388–4400, Jun. 2019.
[53] T. Wang, Y. Liang, W. Jia, M. Arif, A. Liu, and M. Xie, "Coupling resource management based on fog computing in smart city systems," *J. Netw. Comput. Appl.*, vol. 135, pp. 11–19, Jun. 2019.
[54] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[55] H. Liao, Z. Zhou, X. Zhao, B. Ai, and S. Mumtaz, "Task offloading for vehicular fog computing under information uncertainty: A matching-learning approach," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 2001–2006.

[56] N. Joshi and S. Srivastava, "Task allocation in three tier fog IoT architecture for patient monitoring system using Stackelberg game and matching algorithm," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2019, pp. 1–6.

[57] S. F. Abedin, M. G. R. Alam, N. H. Tran, and C. S. Hong, "A fog based system model for cooperative IoT node pairing using matching theory," in *Proc. 17th Asia–Pacific Netw. Operations Manage. Symp. (APNOMS)*, Aug. 2015, pp. 309–314.

[58] R. W. Irving, "An efficient algorithm for the 'stable roommate' problem," *J. Algorithms*, vol. 6, no. 4, pp. 577–595, Dec. 1985.

[59] F. Chiti, R. Fantacci, and B. Picano, "A matching theory framework for tasks offloading in fog computing for IoT systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5089–5096, Dec. 2018.

[60] Y. Zu, F. Shen, F. Yan, L. Shen, F. Qin, and R. Yang, "SMETO: Stable matching for energy-minimized task offloading in cloud-fog networks," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–5.

[61] C. Swain, M. N. Sahoo, A. Satpathy, K. Muhammad, S. Bakshi, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "METO: Matching-theory-based efficient task offloading in IoT-fog interconnection networks," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12705–12715, Aug. 2020.

[62] C. Swain, M. N. Sahoo, and A. Satpathy, "SPATO: A student project allocation based task offloading in IoT-fog systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[63] D. J. Abraham, R. W. Irving, and D. F. Manlove, "Two algorithms for the student-project allocation problem," *J. Discrete Algorithms*, vol. 5, no. 1, pp. 73–90, 2007.

[64] A. Satpathy, M. N. Sahoo, L. Behera, C. Swain, and A. Mishra, "VMatch: A matching theory based VDC reconfiguration strategy," in *Proc. IEEE 13th Int. Conf. Cloud Comput. (CLOUD)*, Oct. 2020, pp. 133–140.

[65] C. Swain, M. N. Sahoo, and A. Satpathy, "LETO: An efficient load balanced strategy for task offloading in IoT-fog systems," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Sep. 2021, pp. 459–464.

[66] A. Abouaomar, A. Kobbane, and S. Cherkaoui, "Matching-game for user-fog assignment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[67] A. Bandyopadhyay, V. K. Singh, S. Mukhopadhyay, U. Rai, F. Xhafa, and P. Krause, "Matching IoT devices to the fog service providers: A mechanism design perspective," *Sensors*, vol. 20, no. 23, p. 6761, Nov. 2020.

[68] B. Assila, A. Kobbane, and M. El Koutbi, "A many-to-one matching game approach to achieve low-latency exploiting fogs and caching," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–2.

[69] B. Assila, A. Kobbane, A. Walid, and M. El Koutbi, "Achieving low-energy consumption in fog computing environment: A matching game approach," in *Proc. 19th IEEE Medit. Electrotechnical Conf. (MELECON)*, May 2018, pp. 213–218.

[70] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and N. Kara, "FoG-Match: An intelligent multi-criteria IoT-fog scheduling approach using game theory," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1779–1789, Aug. 2020.

[71] C. Wang, Y. Sun, and Y. Ren, "Distributed user association for computation offloading in green fog radio access networks," in *Proc. Inf. Commun. Technol. Conf. (ICTC)*, May 2020, pp. 75–80.

[72] M. Ali, N. Riaz, M. I. Ashraf, S. Qaisar, and M. Naeem, "Joint cloudlet selection and latency minimization in fog networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4055–4063, Sep. 2018.

[73] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

[74] N. Mehran, D. Kimovski, and R. Prodan, "A two-sided matching model for data stream processing in the cloud–fog continuum," in *Proc. IEEE/ACM 21st Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2021, pp. 514–524.

[75] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[76] F. Chiti, R. Fantacci, F. Paganelli, and B. Picano, "Virtual functions placement with time constraints in fog computing: A matching theory perspective," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 980–989, Sep. 2019.

[77] K. E. S. Desikan, V. J. Kotagi, and C. S. R. Murthy, "A novel matching theory-based data offloading framework for a fog network with selfish and rational nodes," *IEEE Netw. Lett.*, vol. 3, no. 4, pp. 172–176, Dec. 2021.

[78] B. Gu, Z. Zhou, S. Mumtaz, V. Frascolla, and A. K. Bashir, "Context-aware task offloading for multi-access edge computing: Matching with externalities," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[79] X. Fu, B. Tang, F. Guo, and L. Kang, "Priority and dependency-based DAG tasks offloading in fog/edge collaborative environment," in *Proc. IEEE 24th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2021, pp. 440–445.

[80] X. Zhang, A. Pal, and S. Debroy, "EFFECT: Energy-efficient fog computing framework for real-time video processing," in *Proc. IEEE/ACM 21st Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2021, pp. 493–503.

[81] L. Yang, C. Zhong, Q. Yang, W. Zou, and A. Fathalla, "Task offloading for directed acyclic graph applications based on edge computing in industrial internet," *Inf. Sci.*, vol. 540, pp. 51–68, Nov. 2020.

[82] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," *J. Commun. Netw.*, vol. 24, no. 1, pp. 1–16, Feb. 2022.

[83] S. Misra, S. P. Rachuri, P. K. Deb, and A. Mukherjee, "Multiarmed-bandit-based decentralized computation offloading in fog-enabled IoT," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10010–10017, Jun. 2020.

[84] S. S. Ravindranath, Z. Feng, S. Li, J. Ma, S. D. Kominers, and D. C. Parkes, "Deep learning for two-sided matching," 2021, *arXiv:2107.03427*.

HOA TRAN-DANG (Member, IEEE) received the B.E. degree in electrical and electronics engineering from the Hanoi University of Science and Technology (HUST), Vietnam, in 2010, the M.S. degree in electronics engineering from the Kumoh National Institute of Technology (KIT), South of Korea, in 2012, and the Ph.D. degree from the University of Lorraine, France, in 2017. He currently works with the Department of IT Convergence Engineering, KIT, as a Research Professor. His research interests include wireless sensor networks, the Internet of Things (IoT), the physical internet, and radio resource management in wireless industrial networks.

DONG-SEONG KIM (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003. From 1994 to 2003, he worked as a full-time Researcher with ERC-ACI, Seoul National University. From March 2003 to February 2005, he worked as a Post-doctoral Researcher at the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently the Director of the Kit Convergence Research Institute and the ICT Convergence Research Center (ITRC Program) supported by Korean Government at the Kumoh National Institute of Technology. He is an ACM Senior Member. His current research interests include real-time IoT, industrial wireless control networks, networked embedded systems, and Fieldbus.