UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Telecomunicazioni

Cross-Layer Resource Allocation Frameworks in Modern Networks

Candidate
Benedetta Picano

Supervisors
Prof. Romano Fantacci

Dr. Francesco Chiti

Dr. Giulio Bartoli

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXII, 2016-2019

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

我工作用力，玩更用力

I work hard, I play harder

Acknowledgements

Words fail to express how grateful I am to my advisor, Prof. Romano Fantacci. His esteemed guidance, constant encouragement, and deep academic background have helped me in countless ways in finding a good direction during my PhD studies and improving my research skills. I wonder if he knows he is my first role model both professionally and personally.

My deep gratitude goes to Prof. Zhu Han, who never tires of pushing all of us, his students, to become the best version of ourselves.

Furthermore, I would like to thank Dr. Francesco Chiti for his guidance without any imposition, allowing me to develop my personal research vision and critical thinking.

I also wish to express my sincere gratitude to Dr. Giulio Bartoli for his keen insights, daily support and precious teachings.

I am grateful to my father, because he taught me to dream big, and to my mother, because she taught me how to rise when dreams end. Thank to my dear sister, because my accomplishments are her accomplishments.

My appreciation also goes to Ruoguang Li, Junhua Wang, Zhongyu Miao, in China, and Xuelin Cao, Dawei Chen, and Xunsheng Du, in USA, for the unforgettable moments spent together all around the world, and for their strong and constant support during my PhD life, despite distance.

Last, but not least, I would like to thank Alessio Bonadio for being such a good friend, and Francesca Nizzi, for giving me such happy three years.

To all of them I just say thank you, but from the depths of my heart.

Abstract

Recently, the massive diffusion of high-performance mobile devices and the increased request of novel services has given rise to an unprecedented growth in low-latency, multimedia and computational intensive applications demand. To fully accomplish this, in modern networks we have to face with several issues as spectrum scarcity, congestion, or coexistence of heterogeneous networks, to name the few. The new challenges here have transformed the layered concept of network into a whole complex functionally integrated system. In addition, the term resource has deeply changed its traditional meaning, until now typically referred to physical resource, i.e., communication channel, storage blocks, and so on, going towards a more flexible definition which defines the resource as a set of heterogeneous physical and logical entities correlated to each other. As a consequence, the resource allocation problem in modern networking is interpreted in a broader sense in order to meet the requirements of flexible and advanced solutions. This thesis aims at identifying suitable solutions in cross-layer point of view for specific classes of problems typical of modern networks. In particular, the thesis starts by taking into account the major classes of problem that afflict the next generation wireless networks and applying, for each problem category, techniques able to properly meet the crucial aspects of the considered optimization problem. Therefore, the ultimate aim of this thesis is the formulation of cross-layer frameworks to take into account several network issues. Matching theory, chaos theory, machine learning and queuing theory have been applied to investigate their potential in wireless network systems, starting from the study of interference management, unlicensed spectrum allocation, computational offloading, and virtual functions placement, even to the analysis of the service provider perspective through the service demand prediction and price control analysis.

The main contributions of this dissertation can be summarized as follows.

- An overview of the fundamental concepts, definitions and insights of the theoretical frameworks involved in the resource allocation solutions presented.
- Design and implementation of novel matching algorithms able to coexist with the external influences due to the system conditions or to interfering devices. Different algorithms have been developed for different application scenarios, such as the interference management, carrier aggregation, virtualization and computational offloading in various network environments. In addition, system stability has been extensively studied.
- Application of chaos theory to the nonlinear time series analysis forecasting, in order to predict the customers service demand in smart city contexts, aiming at improving accuracy without the typical machine learning drawbacks.
- Development of cross-layer frameworks to analyze the problems typical of virtualization taking into account both the service provider perspective and the users' point of view, adopting a cross-layer approach.
- Performance analysis in strict delay constrained computations scenarios in presence of users mobility by applying the queuing theory.
- Critical discussion of performance comparison and proposed approaches for designing resource allocation strategies in next generation wireless networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Novel Network Paradigms

Live as if you were to die tomorrow.
Learn as if you were to live forever.

Mahatma Gandhi

1.1 Introduction

Recently, new communication paradigms have emerged, radically changing the way in which wireless networks are designed, organized and studied. Furthermore, the ever increasing traffic and service demand for intensive and real-time applications is resulted in a consistent and continuous search for new flexible and self organized network structures. It is in this perspective that major efforts have been made to propose architectural solutions able to guarantee mobility support, low transmission delay, high data rate and user experience. Several approaches have been developed to pursuit these objectives at various levels. In particular, one of the most promising approach implies to move the cloud close to the users. That insight has been practically translated into the introduction of platforms for computation and storage between users and cloud, aiming at alleviating the significant network delay due to the transmission from and towards the far data centers. Alongside the emergence of ever more complex network structures, as represented in Figure 1.1, the efficient management of system resources has played a crucial

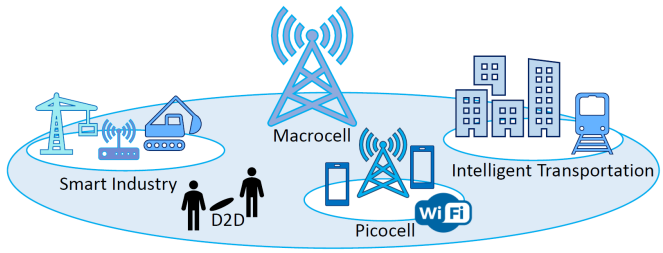

Figure 1.1: A heterogeneous wireless network.

role. For example, the proper exploitation and reuse of the wireless channels or computation nodes, necessarily impact on the network performance. For all these reasons, efficient and fast allocation solutions have become imperative within the new network environments. During years, countless approaches have been proposed by research community to correctly manage and share system resources, and many mathematical frameworks have gained attention in resource allocation field over time. In addition to the increase in complexity, the novel network paradigms have given rise to more and more challenging problems, tracing the way towards distributed, self-organizing, and context-aware optimization techniques. In parallel, during years, even the definition and meaning of the *resource* term is deeply muted, assuming a more flexible, wide, and interdisciplinary connotation. In fact, the intrinsic multifaceted nature of modern network problems implies that the meaning of the resource heavily depends on the aim of the problem afforded, and its identification may vary from the traditional physical meanings, until more abstract concepts such as service prices or accuracy on prediction users demand. Another novel aspect introduced by the advent of the emergent network, is the focus on the users' perspective and subjective experience in receiving a service. Therefore, the importance of accurate psycho-socio-mathematical models to effectively describe the knowledge evolution and opinion formation in social networks, the socio-economic dynamics in crime modeling or energy consumption, and the more general decisions making processes of rational and irrational players, has been emphasized. The combination of all these factors has played an essential role in the process by which the network is conceived as an organism composed of many correlated parts, able to influence each other and the whole system itself. Due to

the high number of involved factors and the massive amount of connections among the diverse parts of a network, the main challenge is to provide a fusion between objective criteria and human behavior to properly manage and allocate the network resources, by ensuring an adaptive and cross-layer dynamic allocation and decision processes framework. A crucial issue hard to solve in dynamic resource allocation is that, in order to face the tomorrow's wireless services, it is essential to design resource management schemes to combine time varying network conditions, different users service requirements, heterogeneous environments, and moderate prices.

Essentially, the thesis addresses some typical problems of the next generation wireless networks, and applies different advanced frameworks to provide suitable and flexible solutions to them. For each problem taken into account, the framework used has been presented in all its part, and its application to the given context motivated with a critical discussion about its strengths and weaknesses. The solutions proposed in this thesis can be categorized in four basic mathematical tools: matching theory, chaos theory, machine learning and the queuing theory. Finally, the solution achieved is validated and analyzed by performance comparison with alternative strategies.

1.2 Emerging Wireless Communications and Network Architectures

With the objective of accomplishing the new challenges posed by the diffusion of heterogeneous service requirements, real time applications, high levels of mobility and so on, numerous novel communication technologies and network infrastructure models have been theorized to boost the network performance. In the following subsections these technologies will be briefly introduced and their potentials presented.

1.2.1 Internet of Things Network

Nowadays, Internet of Things (IoT) represents a breakthrough paradigm to allow ubiquitous connectivity among humans and physical objects at any time and any place. Moreover, often IoT devices are supposed to be smart, so they can communicate, coordinate, and share information with each other in an autonomous mode, in order to take decisions and perform their tasks. In this context, the 5G wireless communication networks are considered the

glue technology, that will enable the connection of billions of heterogeneous devices with a close to zero end-to-end latency, in a high reliable and efficient mode [171]. Furthermore, 5G networks will enable a functional integration of computing and communication capabilities thus allowing intelligent IoT applications able to deeply modify several aspects of our lives (e.g. healthcare, energy, waste management, smart cities, industry, mobility). Despite the 5G potential benefits, several challenges have to be addressed before IoT paradigm effectively becomes a wide spread reality [219]. Among these, efficient resource management is crucial due to the expected massive number of IoT devices. In fact, inefficient resource management can lead to a significant reduction of the IoT system performance. In particular, in the IoT context, different kind of resources have to be be managed: communication (i.e., spectrum, power, backhaul/fronthaul capacity), storage and computing. Toward this goal, the opportunistic use/reuse of the spectrum is a key element. It allows the sharing of the spectrum among different communications, but requires strategies for avoiding or limiting the mutual interference under suitable limits.

1.2.2 Edge-Fog Computing Network

For many years cloud computing (CC) has efficiently met the users demand, by providing software and hardware resources for storage and computation, available on demand and remotely by users. With the advent of new communication realities such as IoT, the communications have become definition of ubiquitous and pervasive connections among smart objects, able to receive and send data from and towards the surrounding environment. The interconnections and data exchanges among intelligent devices resulted into networks with highly complex topologies, strict delay requirements, massive volume of data to be processed and transmitted. Roughly speaking, novel network paradigms have to faced with heterogeneous smart devices generating massive volume of data traffic and applications requests. Furthermore, the mobility of the devices represents a crucial aspect of the next generation applications, posing many concerns about the seamless service continuity, or the real-time execution constraints accomplishment for some classes of applications, i.e., augmented reality, health care applications or recognition assistance [72]. All these new challenges have triggered the tendency to migrate towards novel solutions, typically based on the deployment of computational nodes, lower in processing and storage supply, to the edges of the

network, in order to reduce the network response latency and the delay [168]. In this sense, novel network paradigms based on this approach are named edge computing (EC) [166] and fog computing (FC), and are able to provide low latency response and service continuity to mobile users. However, in comparison to the existing public cloud based solutions, i.e., Microsoft Azur and Amazon AWS, the processing capacity of EC solutions is strongly limited [69]. In order to address both the cloud and EC/FC issues, the functional integration of these two approaches in a same computing infrastructure has recently gained momentum [72]. This solution enables massive improvements in system performance and users quality of service (QoS), introducing higher levels of flexibility for rapid computations and in presence of high mobility patterns.

1.2.3 Network Function Virtualization

Network function virtualization has recently gained popularity as paradigm to improve the flexibility of the networks, to provide effective service provisioning and to maximize network infrastructure utilization. In general terms, virtualization refers to decoupling of the software implementation of the network functions from the underlying physical hardware, supporting reconfigurable hardware, general purpose servers, storage, processing and so on [96, 103]. Virtualization offers unprecedented perspectives of abstraction, slicing and sharing of resources among different parties, facing with heterogeneous service capabilities, environments massively interconnected and ultra dense. to support the development of distributed applications and specialized services. There exist numerous levels of virtualization and different practical variations depending on the specific application context, some examples are constituted by virtual machines, virtual memory, etc. However, typically, taking into account the expected exponential growth in number of users, and consequent traffic and rich communication contents [96], the major challenges are the large scale management of multiple and heterogeneous resources, i.e., processing units, memories, and bandwidth, and the optimal regulation of the network traffic by designing efficient routing and forwarding strategies [123].

1.2.4 Device to Device Communication

The explosion in data traffic demand registered during last years has triggered the necessity to find new communication solutions. The device-to-device (D2D) communication technology has been proposed to tackle this issue, establishing direct links between pairs of devices in geographical proximity without the direct support of the base station [76, 85, 250]. The D2D communication has gained a great deal of attention thanks to its promising potential in providing high levels of throughput, effective offloading strategies and increasing network coverage also at the edge of the networks.

At the expense of these promising improvements in reference to the traditional cellular networks, the D2D technology poses various challenges which need to be addressed to make the D2D communications a reality. In fact, one of the most critical point in such technology is the severe interference experienced by users due to the channel reuse, as well as the procedure to discovery the neighbors devices. During years, many efforts have been made to limit this issue and solution such as power control, coding strategies, and more general interference avoidance techniques [96, 120, 190].

1.2.5 LTE-Unlicensed

Recently, the ever increasing growth of the advanced wireless services and applications demand has led to the available spectrum scarcity [139] problem. Within this context, the LTE-Unlicensed (LTE-U) standard has emerged based on the intuitive concept that the unlicensed spectrum can be exploited to offload the traffic of the traditional cellular network [7]. In practical terms, LTE-U is based on the Carrier Aggregation (CA) technology, expanding bandwidth up to a maximum of 100 MHz through the usage of contiguous or non-contiguous Component Carriers (CC) within the same frequency band, or non-contiguous CCs belonging to different bands [6]. Currently, the carrier aggregation assignment mode is still an open issue [139]. The benefits of this novel approach is expressible in terms of greater bandwidth, higher bit rate and lowering the spectrum segmentation. Furthermore, LTE-U is designed to promise wider coverage and seamless services. Typically, LTE-U proposes the the exploitation of the 5GHz unlicensed spectrum [81], due to its less congested condition in comparison to the 2.4GHz spectrum. Despite the LTE-U technology implies several advantages, its application is limited by the severe infrastructure costs deriving from its concrete application, and

by the significant disadvantages experienced by the unlicensed users [10, 96]. In fact, LTE-U generally results in performance degradation of Wi-Fi users, due to the high levels of interference caused by the licensed users. Such interferences can be minimized by some corrective strategies, such as the limitation of the transmission power or the selection of the clearest channel [81].

1.3 Thesis Organization

During this dissertation, several theoretical frameworks are proposed and discussed in relation to their performance and applicability to some of the major classes of resource allocation problems in the wireless networks. This dissertation is organized as follows. In Chapter 2 a universal characterization of the resource allocation problem in the wireless networks is presented, providing the basic mathematical background for the formulation of an optimization problem in the wireless networks research field. A great deal of emphasis has been given to the unavoidable evolution of the meaning behind the resource allocation problem in wireless networks, and how the change of its nature has led to new techniques and approaches for solving the modern networks issues. They are matching theory, chaos theory, machine learning and the queuing theory.

In Chapter 3 and in Chapter 4 the matching theory is applied to two main problem categories: the interference management, and the computational offloading problem, respectively. As regards the management interference, the matching algorithms have been developed to meet the problem dynamics, in order to promote the proper system resource reuse. Similarly, the application of the matching theory framework to the computational offloading problem has been widely addressed in different context and scenarios. Matching theory has made much progresses compared to game theory by providing more flexible practical applications, without requiring closed form solutions for the users objective functions.

In Chapter 5 the offloading problem is still investigated, by focusing on the application of the queueing theory to the performance analysis of an edge computing system designed to support real time applications demands and users mobility. Despite many theoretical assumption, numerical simulations exhibit the validity of the proposed approach.

Chapter 6 finds application in the smart city context, in which a service

provider needs to predict the users demand to properly administrate the system resources. The chapter is mainly focused on the forecasting of nonlinear time series, in which the chaos theory principles are applied to predict the users service demand. Chaos theory offers good performance in terms of accuracy on the predicted values in presence of chaotic behavior of the time series. Furthermore, it avoids the major drawbacks of the more sophisticated machine learning techniques, often characterized by long training processes.

In Chapter 7 machine learning is applied to analyze the network service provider revenue. Cross-layer frameworks consisting of the virtual network functions placement, the users demand prediction through the federated learning paradigm, and the users allocation are realized in order to maximize the service provider profit. Furthermore, a neural networks approach [293] previously presented in literature has been applied to approximate the solution of an optimal price control problem with chaotic constraints.

Finally, conclusion are drawn in Chapter 8, which highlights that the works presented in the previous chapters address the resource allocation problem by applying and re-designing many different frameworks up to the development of cross-layer frameworks, in order to consider not only a narrow view of the problem, i.e., one optimization goal under certain constraints, but a more global vision of the problem addressed.

Chapter 2

Theoretical Frameworks for Resource Allocation Problems

Don't listen to what they say. Go see.

Chinese Proverb

2.1 General Resource Allocation Formulation

Over the past decades, the field of wireless communications has been at the forefront of an unprecedented development that has considerably influenced numerous aspects of the design and control of next generation wireless networks, throwing new challenges and deeply changing the approach in solving technical problems and solution strategies concept. The ever increasing overcrowding of wireless network has laid the foundation for more demanding services, in which a wise and efficient resource exploitation is essential to guarantee a satisfactory user experience. Throughout the evolution of wireless communications, even the meaning of resource is radically changed. Typically, the basic idea behind the traditional resource allocation is the efficient management and allocation of channels, to reduce the interference, and to properly share the spectrum, increasing network data rate. With the new era of networks, more stringent delay constraints have been required for delay-sensitive applications. The ever increasing demand for heavy computation-intensive services, the necessity for re-configurable networks and computing systems have opened the doors towards wider meanings

of the resource term. For example, in virtualization or offloading problems, storage and processor units represent the system resources; furthermore, in smart city perspective and intelligent transportation systems, resource assumes a totally different connotation, in which the term might be declined as transportation services, urban water or electricity infrastructures, and so on.

Whatever the considered scenario and the meaning of resource is, its identification strictly depends on the aim of the problem, i.e., the objective function and, in order to optimize the latter, the available resources, i.e., the optimization variables, have to be known.

Generally speaking, let $\aleph$ be the objective function of our problem. The form of $\aleph$ is strictly related to the type of the problem we want to solve, and can be expressed in terms of sum, product, or other functions of the resources, i.e. Ψ, in order to fit the real network behavior as well as possible [106]. Typical examples in network performance optimization are represented by overall throughput, energy efficiency, or QoS [106]. Limitations over quantities, for example the maximum transmitted power, the minimal distortion, or the maximal delay, are defined as problem constraint $\mathbf{\Phi}$. Both $\aleph$ and $\mathbf{\Phi}$ can be linear, nonlinear, convex or non convex and their form unavoidably impacts on the strategy resolution of the problem.

Hence, in next generation wireless networks class, one fundamental form for the resource allocation problem is given by [106]

$$\max_{\mathbf{\Psi}} \aleph \tag{2.1}$$

$$\text{s.t.} \quad \mathbf{\Phi}.$$

Whether $\aleph$ and $\mathbf{\Phi}$ are linear or convex or have some nice forms, the problem (2.1) can be solved with the Lagrange multiplier method. In formal terms, the Lagrange function associated to (2.1) can be written as

$$L = \aleph + \lambda \mathbf{\Phi}, \tag{2.2}$$

where λ identifies the Lagrange multiplier. Then, L is differentiated over $\mathbf{\Psi}$ and the solution for λ is found. By using constraint $\mathbf{\Phi}$, the optimal solution for problem (2.1) can be found [38, 106]. However, λ might not be solvable, or, again, the optimal solution may not be always easy to obtain from $\mathbf{\Phi}$ [38, 106].

When both $\aleph$ and $\mathbf{\Phi}$ are convex functions, convex optimization algorithm

such as the barrier method, the steepest descent method, or the gradient method can be applied. Even though this type of mathematical approaches bring to clean and elegant solutions fast to reach, the hard point here is that the objective function and its constrains have to be traced back to a pretty handy form or it needs approximation. In practice, its description needs accuracy under certain conditions, and the complexity of the reality behavior does not always allow valuable approximations [38, 106]. In fact, generally speaking, the success of these classes of methods is directly related to the quality of the choices made about the approximation functions. Another major drawback of these kind of methods, is that only simple problems, for example small number of users and simple channel model, can be addressed with these mathematical techniques[38, 106].
With the advent of more complex, more dense, more heterogeneous and high-demanding networks, the necessity for more flexible and scalable approaches is born and, within this context, matching theory, chaos theory, and machine learning have gained momentum.

2.2 Matching Theory Framework

Matching theory represents a flexible mathematical framework, arising from the economics field, with the aim of attempting the formalization of the process creation of mutually beneficial relations among elements, called matching agents, belonging to two distinct sets over time [96]. The Matching theory framework deals with the *stable marriage problem*, consisting of two sets of elements, hereafter one set of man and one of women, that have to be paired on the basis of their personal preference with the elements of the opposite set, and avoiding the presence of *blocking pairs* (BPs). By starting considering a matching as (man, woman) pairs, a BP is defined as (man,woman) pair, in which both the elements have incentive to leave their current situation looking for a better partner [96, 97, 104, 163, 211]. The state-of-art algorithm reaching a stable matching, applicable to a wide class of matching models, is represented by the Nobel-prize winner Gale and Shapley (GS) algorithm [96, 97, 104, 211]. The GS algorithm is based on the usage of the *preference list* associated to each element of both the two sets, indeed, each matching agent makes proposal to its most preferred element belonging to the opposite set. Then, each element, which receives one or more proposals, accepts or rejects according to its personal preferences and w.r.t. its quota,

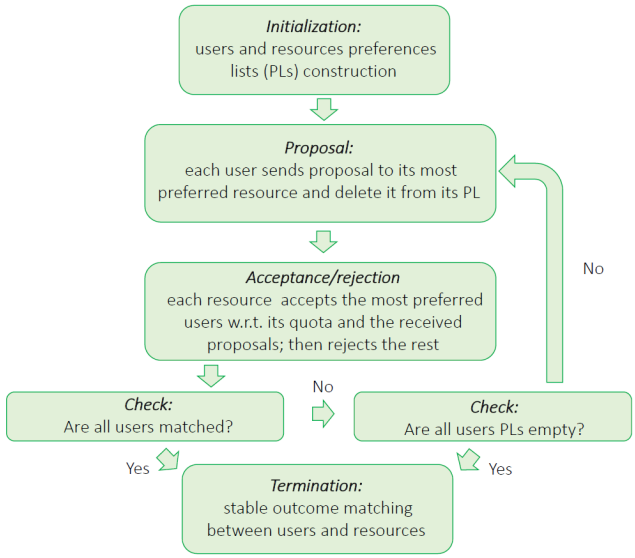

Figure 2.1: Gale-Shapley algorithm flow chart.

that is the number of elements with which can be matched. The procedure is iterative, summarized in flow chart in Figure 2.1, and many works have formulated its distributed version [96]. During years, the intrinsic nature of matching theory in being suitable for distributed contexts, has determined its success and contributed to increase its wide application in many practical fields. Furthermore, another asset of matching theory, even compared to its predecessor game theory, is its flexibility and handling in terms of its *preference list metrics*, corresponding to the utility functions of game theory, which does not require strict requirements such as closed forms. There exist many classifications about matching game. The most well known is probably those based on the value of the quota, for which we have [96, 97, 104]

- One-to-one matching. In this type of matching, the elements of each set can be matched with at most one agent of the opposite set.

- Many-to-one matching. It means that each agent of one set can be matched with a number of elements of the opposite set greater than one, according to its quota, but each element of the opposite set can accept at most one proposal.

- Many-to-many matching. This type of matching defines matching in which the elements of both the two sets considered, can be matched with more than one element of the opposite set, up to their quota.

Due to the structure of the matching theory framework, during last decade it emerged as promising tool to pursuit a proper resources allocation in wireless scenarios. Typically, wireless oriented matching models view the resources and users as two sets of players and belong to one of the following categories [96]

- Canonical matching: this class represents the fundamental matching class, in which each user (resource) involved in the game has preferences on the resources (users) solely on the basis of the information available on this resource (user) and at that user (resource) itself.

- Matching with externalities: this class is made up of matchings in which there exist interdependences and correlations between the preferences of the matching agents. Typical examples are represented by the interference derived from the shared use of the same frequency by more than one user close enough to each other, that severely changes the preferences of users towards channels. This thesis is focused on the application of this type of matching games, particularly common in context-aware allocations, load balancing disciplines and interference management.

- Matching with dynamics: this class of matching games includes matchings that have to be dynamically changed, over time, due to environmental changes such as users mobility, fast fading, and so on. However, for a given time period, the matching problem is traced into one of the two matching classes previously introduced.

From the mathematical perspective, both the second and the third matching classes are formulated as the canonical matchings, previously detailed at the beginning of this section, but as regards the presence of the externalities, the achievement of a stable matching is not trivial, since the stability depends not only on the agents set, but also on the whole matching. For dynamic matchings, time-dependent states can be introduced to study dynamic stability.

2.3 Chaos Theory Framework

Chaos theory has emerged in the later 19th century as a branch of physics specialized in the study of nonlinear dynamic systems, whom long-term behavior is typically unpredictable. A nonlinear dynamics describes the evolution of a system during time, throughout nonlinear equations of motion characterizing its behavior. In general terms, nonlinear dynamics systems perform as chaotic when they exhibit sensitive dependence on initial conditions. Such characteristic is considered the most strong signature of chaos and, intuitively, it describes unpredictability of the system in the long term. One direct consequence of the unpredictability, hence, of the presence of chaos, is that it masks the determinism hidden behind the systems, and which implies that tiny differences on the initial state can lead significant variations on the final system states. Furthermore, each initial condition unambiguously determines the time evolution of the system [47, 144, 165]. In order to observe the hidden deterministic behavior of a chaotic systems, it is necessarily to study underlying dynamics of its orbits, through the strange attractor, resulting from the phase space reconstruction process [43, 125]. The phase space represents all possible states of a system, for which each possible state corresponds to one unique point in the phase space. From a theoretical perspective, system dynamics can be described, for example by an explicit system of first-order differential equations but, due to the exceptionally complexity of many nonlinear phenomena, such approach is impracticable in the most of real cases. This unfeasibility in performing classical differential analysis has led to the necessity of a more general system dynamics description, then resulted in the approximation of a multidimensional map. Under these assumptions, recently chaos theory has found wide application in the nonlinear time series analysis. Indeed, the reconstruction of the phase space associated to a time series with chaotic characteristics can be applied, in order to catch the hidden behavior and geometrical dynamics of the system and properly predict its trend during time. Due to the Takens' delay embedding theorem, which affirms the existence of a map between the real phase space and its reconstructed version [32], the PSR is provided by associating, for each element a_i of the chaotic time series $\mathcal{A}$, the vector in the form

$$\mathbf{b}_i = [a_i, a_{i+\tau}, \ldots, a_{i+(m-1)\tau}], \tag{2.3}$$

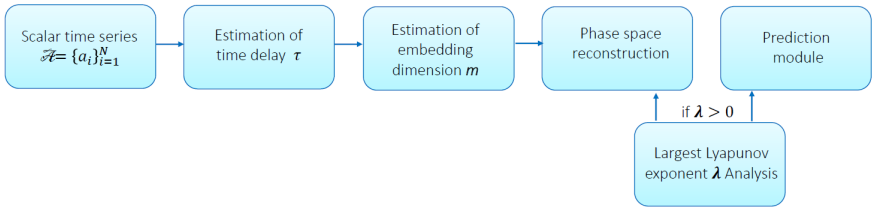

Figure 2.2: Chaos Theory Framework

where τ is the time delay and rules the temporal distance between two successive points in the phase space vector, while m is the embedding dimension and is responsible for the observation of the determinism underlying the system associated to $\mathcal{A}$. The choice of proper values for both τ and m is crucial in the analysis of time series by using the chaos theory framework. In fact, a wrong choice on the m value introduce redundancy that implies degradation on the predictive performance of the framework [124]. Likewise,a good estimation of time delay τ is very important. Indeed, whether τ is very large, successive elements are too unrelated to each other, while a small value of τ implies very low level of additional information among successive points in the phase space. Details about the computation of the optimal values for τ and m have been provided on Chapter 6. Within this context, this kind of chaotic analysis has gained momentum into the forecasting problems field. Roughly speaking, forecasting problems involve the prediction about the future behavior, δ steps ahead in the future, of time series $\mathcal{A}$. More in depth, a is suitable for τ if it maximizes the knowledge about $a_{i+\tau}$ from a_i, minimizing the redundancy between $a_{i+\tau}$ and a_i, . The proper value of delay τ can be estimated by evaluating the amount of mutual information between pairs of points $a_{i+\tau}$ and a_i, and τ that minimizes the mutual information between observations [82]. Therefore, the histogram of the probability distribution of the dataset is created. In addition, the mutual information between a_i and $a_{i+\tau}$, by increasing time the delay τ, can be given by [159]

$$I(\tau) = \sum_{f,j} p_{f,j}(\tau) \log p_{f,j} - 2 \sum_{f} \log p_f, \tag{2.4}$$

in which p_f represents the probability that a_i is in the f-th bin of the histogram, and $p_{f,j}$ expresses the probability that a_i and $a_{i+\tau}$ are in the f-th and j-th bin of the histogram, respectively.

In order to select a proper value for m, the false nearest neighbors method [126] is largely applied. The insight behind this strategy is the exploitation of the presence of chaos, which induces an exponential divergence on the trajectories of close points. Consequently, there exist some points in the dataset that are neighbors in the embedding space but for which their behavior exponentially diverges during time. The method acts, for each point in $\mathcal{A}$, in finding its closest neighbor in m dimension. After that, the ratio between their distance in $m+1$ dimensions and m dimensions is computed. Finally, the attention is posed on the ratio value. If it is is greater than a fixed threshold r, the neighbor is considered false. In formal terms, the false nearest neighbors function can be defined as in (2.5), taking fixed the threshold T

$$F_{fnn}(T) = \frac{\sum_{i=1}^{N-m-1} \Theta\left(\frac{|\mathbf{b}_i^{(m+1)} - \mathbf{b}_j^{(m+1)}|}{|\mathbf{b}_i^{(m)} - \mathbf{b}_j^{(m)}|} - T\right)\Theta(\frac{\sigma}{T} - |\mathbf{b}_i^{(m)} - \mathbf{b}_j^{(m)}|)}{\sum_{i=1}^{N-m-1} \Theta(\frac{\sigma}{T} - |\mathbf{b}_i^{(m)} - \mathbf{b}_j^{(m)}|)}, \quad (2.5)$$

where Θ is the Heaviside step function. Once the phase space reconstruction process is completed, to the chaotic behavior of $\mathcal{A}$ has to be verified, through the analysis of the largest Lyapunov exponent. The insight of such technique is the study of the distance of two vectors, close in the phase space during the time transition [184]. Let $\mathbf{g}$ and $\mathbf{h}$ be two trajectories, which are neighbors in the state space, then their mutual distance evolves according with

$$\mathbf{g}_{n+1} - \mathbf{h}_{n+1} = \mathbf{J}_n(\mathbf{g}_n - \mathbf{h}_n) + O(||\mathbf{g}_n - \mathbf{h}_n||^2), \quad (2.6)$$

in which $\mathbf{J}_n$ represents the $m \times m$ Jacobian matrix associated to $\mathbf{F}$. Furthermore, considering Λ_i the eigenvalue of $\mathbf{J}$, the Lyapunov exponents are expressed by [124]

$$\lambda_i = \lim_{N\to\infty} \frac{1}{2N} ln|\Lambda_i^{(N)}|. \quad (2.7)$$

When forecasting is provided, a measure of the gap between the predicted and the real value of the time series has to be analyzed, in order to test the accuracy of the method. Depending on how far in the future the forecasting

horizon is, we refer to short-, medium-, and long-term forecasting. Despite there exist many different metrics to evaluate the forecasting error [c11], the insight behind the formulation of forecasting problems is the minimization of the gap between the real value of the time series $\mathcal{A}$, i.e., a_i, and its forecast value $\hat{a}_i$, hence

$$\frac{1}{x}\sum_{i=1}^{x}|\hat{a}_{i+\delta} - a_{i+\delta}|, \tag{2.8}$$

where x represents the number of the samples in test data.

2.4 Machine Learning Framework

Machine learning is a relatively young technology, whose principal aim is to learn parameter models on the basis of some training data. Similarly to the way humans learn from experience, algorithms learn from data. The crucial point here is how to re-create the brain learning process. There is no definitive standard to solve this kind of problem but, during years, many recurrent patterns were noted among the proposed solutions. The whole learning approach generally follows a common structure, composed of an initial data manipulation and features extraction. The features are useful information exploited by the learning algorithms to chooses a learning model and subsequently search for its parameters. There exist three main categories of machine learning which are given by [223]

- *Supervised learning*: the main goal here is to assign a label to data, following a model trained on a labeled dataset provided by the supervisor, acting as a teacher. The labeled dataset is represented by a set of input and output parameters.

- *Unsupervised learning*: this approach is based on the detection of patterns and structures in existing data, without the involvement of labels or responses.Typical applications of this class of learning paradigm are clustering and dimensionality reduction.

- *Reinforcement learning*: this approach is halfway between the previous two branches. The teacher role is played by the environment and provides hints to the learnin g system which receives feedback on the basis of its answers.

Therefore, machine learning aims at defining parameter models on the basis of some training data. Typically, despite traditional machine learning models reach remarkable effectiveness, the learning procedures are very burdensome and the models need centralizing of the training data on a single site, for example a datacenter. Recently, to overcome these significant drawbacks of machine learning, federated learning is emerged. The emerging federated learning moves the machine learning process to the devices level. Accordingly to this approach, mobile phones collaborate to learn a shared model using data trained on the device. Federated learning improves users privacy decoupling the learning process and the data storage. Furthermore, machine learning models are computed on mobile devices, instead of on centralized machines. This novel computation paradigm is practicable since today's high performance mobile phones are powerful enough to run machine learning models.

Roughly speaking, a machine learning model can be identified by a loss function depending on the data sample z and on a parameter vector $\mathbf{w}$, i.e., $f_z(\mathbf{w})$, which captures the error introduced by the model on the basis of the training data [256]. Let m be the number of end-devices (EDs), where each ED i has its local data Ω_i, $i = 1, \ldots, m$. The collective loss function can be defined as [169, 256]

$$F_i(\mathbf{w}) = \frac{1}{|\Omega_i|} \sum_{z \in \Omega_i} f_z(\mathbf{w}), \tag{2.9}$$

where $|\Omega_i|$ is the number of elements belonging to Ω_i. Then, from (2.9) follows that the global loss function over all the Ω_i, $i = 1, \ldots, m$, is given by [169, 256]

$$F(\mathbf{w}) = \frac{\sum\limits_{i \in \{1,\ldots,m\}} |\Omega_i| F_i(\mathbf{w})}{\sum\limits_{i \in \{1,\ldots,m\}} |\Omega_i|}. \tag{2.10}$$

As well explained in [256], the direct consequence from (2.9) and (2.10), is searching $\mathbf{w}^\star$ such that

$$\mathbf{w}^\star = \arg\min F(\mathbf{w}). \tag{2.11}$$

Therefore, the federated learning approach consists of successive interactions between the clients and server sides, and, during each algorithm iteration round u, just a subset the EDs are involved in the training procedure. More in depth, the general framework can be summarized as

- In parallel, each ED i involved in the training procedure updates its local parameter vector $\mathbf{w}_i(u)$, built accordingly to [257], on the basis of Δ_i;

$$\mathbf{w}_i(u) = \hat{\mathbf{w}}_i(u-1) - \alpha \nabla F_i(\hat{\mathbf{w}}_i(u-1)), \tag{2.12}$$

  where α is the learning rate and $\hat{\mathbf{w}}_i(u-1)$ is the term $\mathbf{w}_i(u-1)$ after global aggregation.

- The server side computes the weighted average as proposed in [169] and defined by

$$\mathbf{w}(u) = \frac{\sum\limits_{i \in \{1,\ldots,m\}} |\Omega_i| \mathbf{w}_i}{\sum\limits_{i \in \{1,\ldots,m\}} |\Omega_i|}. \tag{2.13}$$

Distributed data training following the federated learning dictates results in several advantages in terms of clients privacy. In fact, the training procedure on the clients site, allows users to keep protected their sensitive information and to upload the parameter vector $\mathbf{w}_i$, which does not expose the client to any privacy issue, since, given $\mathbf{w}_i$, to retrieve Ω_i is not straightforward.

2.5 Queueing Theory Framework

The queueing theory deals with the analysis of queuing, i.e., the mathematical study of the waiting in lines. In practical terms, queues contain customers which require service, wait for service if it is not instantaneous, and they leave the system after service accomplishment. The customers can identify people, objects, information, or computer tasks [91]. Queueing theory provides tools to analyze the systems behavior and its response in reaction to randomly arising demands. The process describing arrivals is stochastic, and the knowledge about the probability distribution describing the interarrival times is necessary. Furthermore, many situations can be modeled with queuing theory. Customers arrivals may or may not occur in the same time, and customers may decide to wait service without any limitation on queuing time. Differently, if the queue grows excessively, the customers may avoid the entrance in the system, i.e., they have balked [91,246]. When a customer enters in the system and waits for service, after a certain amount of time, it may lose patience and to have reneged, i.e., leave the system without service completion. The same discussion can be easily extended to service processes.

Also in this case, a crucial role is played by the probability distribution describing the sequence of customer service times [91, 246]. Again, the service can be provided by a single server or by a batch, which represents a very common situation in parallel processing scenario. Furthermore, the number of customers in the queue can be used to rule the the service process which may rule its computation efficiency on the number of customers waiting for service. Another essential ingredient of the queuing analysis is the queue discipline adopted to select the customers for service when they are waiting for it. In this regards, one of the most frequently used discipline is the first-in, first-out (FIFO), applied also in Chapter 5. However, literature is rich of numerous queue service schemes. In some queueing processes there is a physical limitation to the amount of Some queuing processes, named finite queuing systems, imposes a physical limitation to the amount of requests in the system, i.e., a maximum system size, and, once that limit is reached, any customer is allowed to enter in the system until space returns available, i.e., a service completion occurred.

In Chapter 5 the queuing theory is exploited to conduct performance analysis in real-time and delay constrained computational offloading scenarios, by dealing with systems with and without finite capacity and reneging.

Chapter 3

Interference Management with Matching Theory

Two benefits from one action.

Chinese Proverb

3.1 An Efficient Resource Allocation Scheme for Applications in LR-WPANs based on a Stable Matching with Externalities Approach

3.1.1 Motivation

The large scale diffusion of a massive number of smart devices implementing the IoT reality have progressively led to the need for efficiently support the LR-WPANs, avoiding severe performance degradation due to the presence of interference, resulted by the concurrent use of a same band. Currently, the LR-WPANs are considered a promising solution to a vast class of modern wireless applications [16, 128, 130, 236]. The main advantages of the LR-WPANs are the energy efficiency, crucial in the IoT field, and the additional usage of the TV white space (TVWS) bands. This approach moderates the spectrum scarcity problem for the next IoT networks, even if the LR-

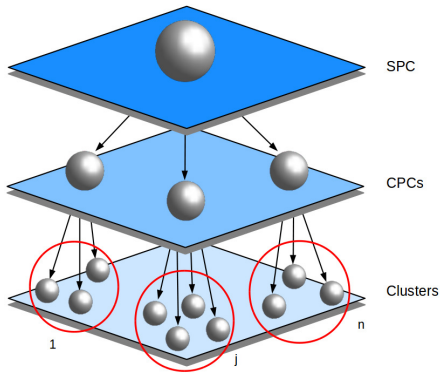

Figure 3.1: Reference scenario.

WPANs is strictly related to sensors applications such as personal medical health care, or industrial and home automation [16, 130].

Within this context, the design of proper resource allocation schemes is essential to guarantee the LR-WPAN performance maximization.

3.1.2 Contributions

This work presents the main results obtained in paper [26], focused on the IEEE 802.15.4m LR-WPAN, in which a resource allocation matching strategy is proposed, aiming at minimizing the mean system dropping probability. Since the designed matching algorithm belongs to the matching game with esxternalities class, the matching stability is deeply discussed by providing a theoretical proof about the achievement of a final stable matching configuration. Therefor, the main contributions of this work are given by

- Formulation of the average system dropping probability minimization problem
- Modeling of the proposed problem as a matching game with externalities;
- Design of a post-matching procedure in order to reach matching stability, on the basis of the stability definition proposed in [35];
- Performance evaluation through the comparisons with the random allocation strategy, a modified version of the Brelaz algorithm [136, 222,

294], and the potential game [282].

Finally, in comparison to the previous literature, this work proposed a join use of the matching theory principles with the graph theory, in order to perform the resource allocation strategy. In addition, despite the achievement of the stable matching outcome in the matching game with externalities is not a straightforward, the proposed approach terminates in a stable matching.

3.1.3 Related Works

During years, game theory (GT) has been extensively applied to propose solutions to a wide class of resource allocation problems [93], [94]. For example, non cooperative GT is applied in [297], in which a one-to-one stable match is reached to guarantee an energy-efficient resource allocation. Despite the large diffusion of GT in the literature, such mathematical tool suffers of some important limitations on its applicability, i.e., a kind of knowledge on players' actions is required, and so on [105]. Under these conditions, MT gained momentum to overcome the restrictions of GT, [94], [71, 77, 87, 95, 191, 195, 213, 277, 296].

Paper [94], reaches a stable matching to maximize both LTE and WiFi users' throughput, in LTE-U systems. In [195] a context-aware stable matching with externalities is proposed, pairing small cell base stations and user UEs, on the basis of interference and network congestion. Differently, the paper [95] adapts the matching theory to maximize network throughput, considering the service requirements associated to D2D users and UEs. In [191], authors use matching theory to solve resource allocation problem in delay tolerant networks, while in in [77] matching theory is applied to cognitive radios. Graph theory is applied in [213], where a three steps algorithm is designed, in which the load of each small cell , the average channel gain and data rates are taken into account. Then, throughout a graph coloring procedure, the interference is minimized and, after that, an advantageous resource allocation is performed on the basis of the instantaneous channel gain.

3.1.4 Reference Scenario

AS reference scenario we consider the three-tier network architecture depicted in Figure 3.1, where the SuperPAN coordinator (SPC) is arranged

to provide connections to a set of ChildPAN Coordinators (CPCs), each of which handles a cluster [235] (i.e., subnet) composed of a given number of end-devices [3]. The end-devices belonging to a given cluster, in their turn, connect to the appropriate CPC by accessing a suitable TVWS band exploiting the CSMA/CA technique [3], implemented according to the non-acknowledged mode as in [26, 60, 110].Then, we assumed that each cluster has a limited extension and that the CPCs are in fixed positions. Furthermore, we supposed the same transmission power in each cluster and uniform interference level within a cluster.

Let $\mathcal{B} = \{b_1, \ldots, b_\rho\}$ be the set of TVWS bands and the set of CPCs $\mathcal{C} = \{c_1, \ldots, c_n\}$. The number of underlying end-devices is indicated as $\mathcal{S} = \{s_1, \ldots, s_n\}$, while s_j represents the number of end-devices belonging to the cluster j, with $j = 1, \ldots, n$ [26]. It is important to note that each end-device in a given set s_j is only supported by its appropriate CPC j on the assigned band $b_k \in \mathcal{B}$. Therefor, we can define the $n \times p$ allocation matrix $\mathbf{\Delta}$, in which the generic element $\delta_{j,y}$ is 1 if the TVWS band y is assigned to the cluster j or zero otherwise [26].

By focusing on a single cluster, the performance in terms of data packet dropping probability, strictly depends on the whole number of the end-devices in the interfering clusters, forming the so called *collision domain* [60, 110] defined below in (3.2).

From a theoretical perspective, clusters h and j can be defined as mutual interfering if, using a same TVWS band y, the following inequality is verified

$$P\beta d_{h,j}^{-\alpha} \geq T_h, \tag{3.1}$$

in which P represents the CPC transmission power, β is the path loss coefficient, α is the path loss exponent, and $d_{h,j}$ is the geographical distance between c_h and c_j). Furthermore, T_h defines the power threshold, threshold below which the mutually interference is so weak that it does not affect a data packet transmission (i.e., it does not give rise to access collisions [4]). Therefore, each other cluster $h \in \mathcal{C}$, $h \neq j$, allocated on the same TVWS band $y \in \mathcal{B}$, belongs to the set of the potential interference clusters for the cluster j, i.e., $\mathcal{I}_j^*$, if $P\beta d_{h,j}^{-\alpha} \geq T_h$.

AS direct consequence, if the interference between two clusters is lower than T_h, we have independent collision domains, even if they reside on the same TVWS band. It is straightforward to note that the overall number of end-devices involved in the collision domain, considering a given cluster c_j,

with $c_j \in \mathcal{C}$ allocated to a TVWS band y, with $y \in \mathcal{B}$, $\Omega_j \in \mathbb{N}$, is expressed by

$$\Omega_j = s_j + \sum_{y \in \mathcal{B}} \delta_{j,y} \sum_{k \in \mathcal{I}_j^*}^{n} s_k \delta_{k,y}. \tag{3.2}$$

Consequently, from (3.2), it follows that Ω_{j^*} depends on the entries values $\{\delta_{k,y}\}$ of the allocation matrix Δ.

Finally, if we have $\delta_{k,y} = 0 \; \forall k \in I_j$ and $\delta_{j,y} = 1$, from (3.2) $\Omega_j = s_j$, i.e., Ω_j includes the minimum number of end-devices.

3.1.5 Problem Formulation

This work focuses on the minimization of the average $\bar{p}_{drop}$, which is given by

$$\bar{p}_{drop} = \frac{1}{n} \sum_{j=1}^{n} p_{drop}(\Omega_j), \tag{3.3}$$

where $p_{drop}(\Omega_j)$ is the average data packet dropping probability experienced by cluster c_j, derived in accordance with [59, 60, 110].

Therefore, the formulation of the resource allocation problem is

$$\min_{\boldsymbol{\Delta}} \; \bar{p}_{drop} \tag{3.4}$$

$$s.t. : \sum_{y \in \mathcal{B}} \delta_{j,y} = 1, \qquad j = \{1, \ldots, n\}, \tag{3.5}$$

where constraint (3.5) imposes that only one band is assigned to each cluster $\in \mathcal{C}$. From above, the problem results to be a mixed integer non-linear problem that is typically NP-hard to solve [31]. The intrinsic complexity of problem (3.4), makes infeasible the whole class of approaches based on classical optimization. Similarly, the game theory based strategies lack in providing scalable solutions and significantly slow convergence time. Therefore, in order to reach a suboptimal solution with acceptable time complexity, the problem of finding the configuration of Δ which minimizes $\bar{p}_{drop}$ w.r.t. the constraint (3.5) has been modeled in terms of matching game with externalities. In addition, we have exploited the graph theory to build the mutual interferences graph, i.e., $\mathcal{G}_{\mathcal{I}} = (\mathcal{V}, \mathcal{E})$, in which the vertices $\mathcal{V}$ are identified

by the clusters $\mathcal{C}$, and $\mathcal{E}$ is the set of edges such that, given two clusters, i.e., c_j and c_h, they are connected through an edge if they are mutual interfering CPCs. In formal terms, there exists an edge between c_j and c_h *if* $c_j \in \mathcal{I}_h^*$, and $c_j \in \mathcal{I}_h^* \implies c_h \in \mathcal{I}_j^*$.

3.1.6 Matching Method

The proposed algorithm is supposed to be performed at the SPC site, since it is equipped with more computational capability and less tight energy consumption constraints than all other network devices in addition to a full awareness of the interference scenario. Hence, according to [138, 299]. In other words, the SPC represents a central unit with full awareness of the whole network. For this reason, the SPC is able to evaluate the CPCs and TVWS bands preferences lists. In fact, all the devices are assumed with fixed position, while the mutual interference between two adjacent clusters evaluated at the CPCs level. The proposed centralized approach finds motivation in the following advantages

1. Low latency in updating the preferences lists at each step of the algorithm;

2. Avoid connections at each step of the algorithm between the SPC and the CPCs, hence lowering the network congestion and the energy consumption;

3. Speed up the convergence of the allocation process.

For the sake of simplicity, the notation of $b_j \succeq_{c_i} b_k$ is introduced to express that the cluster c_i prefers band b_j to b_k.

Generally speaking, $\text{rank}(c_i, b_j)$ identifies the position of bandwidth b_j in the preference list rank of c_i. Hence, b_k is the preferred band for a given c_i, i.e., $\text{rank}(c_i, b_k) = 1$ *iff*

$$b_k = \arg\min_{y \in \mathcal{B}} \{\Omega_i^y\}. \tag{3.6}$$

Similarly, the preference list of bands over clusters is built on $\bar{p}_{drop}$ values, resulting from the assignment to a cluster, i.e., c_j, of each available band in $\mathcal{B}$. Then, the cluster c_j is included in the preferences list of the band with the smallest $\bar{p}_{drop}$ value [26]. Also the preferences list of a given TVWS band is ranked in a descending order considering the $\bar{p}_{drop}$ values. The proposed

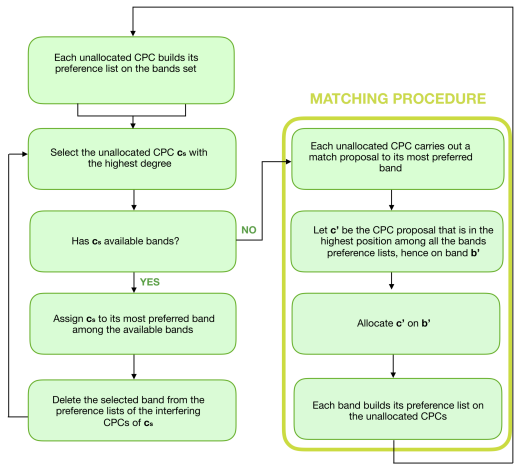

Figure 3.2: Flowchart of the proposed MT algorithm.

strategy realizes a many-to-one matching between the $\mathcal{C}$ and $\mathcal{B}$ sets and it recall, for each possible assignment (c_j, b_k), a procedure to delete the chosen band b_k from the preference lists of the potential interfering nodes of c_j. As consequence, different colors, expressing different bands, are assigned to potential mutual interfering clusters.

The proposed method can be then summarized as

1. selects vertex $c_j \in \mathcal{G}_{\mathcal{I}}$ with the highest degree, among the unallocated CPC set. If c_j has not an available band, goes to step 4), otherwise
2. c_j is allocated on its preferred TVWS band b_h derived according to the procedure proposed;
3. b_h is immediately deleted from the preference lists of each node in $\mathcal{I}_j^*$, then return to step 1);
4. each unallocated CPC proposes its most preferred TVWS band to the SPC;
5. the TVWS bands build their preferences lists;
6. let c' be the CPC proposal that is in the highest position among the TVWS bands preference lists, and let b' be the TVWS band that has

c' in the highest position on its preference list in comparison with the positions of the other received proposals on all the TVWS bands preference lists.

7. c' is allocated on b';

8. preference lists of all clusters are updated according to the new context conditions, repeat from step 1).

3.1.7 Algorithm Complexity

The complexity of the proposed matching algorithm starts with the preference lists construction. As initial step of the method, all the n CPCs build their preferences lists on the ρ channels. Therefore, the temporal complexity of this first step results

$$O(n\rho \log \rho) \tag{3.7}$$

In order to consider the worst case, all the CPCs are supposed mutually interfering. Hence, the first selected ρ CPCs need of different bands. In this case we have to perform a greedy allocation and to drop out the ρ bands from the preferences list of all the remaining $n-\rho$ unallocated CPCs. Hence, for the computational complexity of this step we have

$$O\Big((n-1)\rho^2\Big). \tag{3.8}$$

Subsequently, the number of the CPCs having to update their preferences lists at each step (from $\rho+1$ to n), is equal to the number of the remaining unallocated CPCs at each step. In the same way, all the TVWS bands, at each step (from $\rho+1$ to n), have to compute their preferences list on the basis of the remaining unallocated CPCs. Hence, the resulting complexity of the allocation strategy for this phase results to be

$$\begin{aligned} &O\Big(\sum_{w=0}^{n-\rho-1}(n-\rho-w)\rho\log\rho\Big)+ \\ &+O\Big(\sum_{w=0}^{n-\rho-1}\rho(n-\rho-w)\log(n-\rho-w)\Big) \end{aligned} \tag{3.9}$$

From (3.7)-(3.9), it follows that the overall implementation complexity of the proposed algorithm is

$$O(n\rho\log\rho)+O\Big((n-1)\rho^2\Big)+ \\ +O\Big(\sum_{w=0}^{n-\rho-1}(n-\rho-w)\rho\log\rho\Big)+ \\ +O\Big(\sum_{w=0}^{n-\rho-1}\rho(n-\rho-w)\log(n-\rho-w)\Big) \tag{3.10}$$

which can be approximated as [26]

$$O\Big((n-\rho)\log(n-\rho)\Big) \tag{3.11}$$

3.1.8 Matching Stability Analysis

The resource allocation problem presented before belongs to the class of matching problems with *externalities*, meaning that there exist correlations and interdependencies between the preferences of the involved parties [104]. In practical terms, when a CPC is allocated on a resource, preferences of other players are affected by this action, and the the preferences of each player results dependent by the entire matching [104]. Therefore, after the generic assignment (c_j, b_k), the preference lists of CPCs and SPC are updated, allowing consistency between the algorithm results and the highly dynamic network context.
Due to the strong relation between the preference lists, it is not possible to guarantee matching stability through the proposed strategy. As consequence, a post-matching procedure is proposed to obtain a final stable matching. In order to do that, on the basis of the novel stability definition introduced in [35], the following concepts are recalled

Definition 1. *Let $\mathcal{M}$ be a matching and let $(e,\mathcal{M}(e))$ and $(f,\mathcal{M}(f))$ be two pairs, a swap matching is given by $\mathcal{M}_{e,f}=\{\mathcal{M}\}\setminus\{(e,\mathcal{M}(e)),(f,\mathcal{M}(f))\}\cup\{(e,\mathcal{M}(f)),(f,\mathcal{M}(e))\}$.*

Definition 2. *Let $\mathcal{M}$ be a matching. $\mathcal{M}$ is stable iff $\nexists$ $(e,\mathcal{M}(e))$ and $(f,\mathcal{M}(f))$ s.t. both the following condition are verified*

- $\forall z\in\{e,f\}$, $U_z(\mathcal{M}_{e,f})\geq U_z(\mathcal{M})$;

- $U_{sys}(\mathcal{M}_{e,f}) > U_{sys}(\mathcal{M})$.

where U_z is the local utility function relative to the z-th cluster, defined as $U_z(\mathcal{M}) = \Omega_z^{\mathcal{M}(c_z)}$, $\forall z \in \mathcal{C}$, where $\mathcal{M}(c_z)$ is the partner $\mathcal{B}$ of c_z, associated by matching $\mathcal{M}$. U_{sys} is the global system utility function, expressed as $U_{sys} = \bar{p}_{drop}$ [26]. Furthermore, $\mathcal{M}$ is the current matching and $\mathcal{M}_{e,f}$ is the matching obtained by swapping e and f. Furthermore, the stability introduced in Definition 2 approves a swap if it produces improves mean system dropping probability. More specifically, we define a blocking pair as follows

Definition 3. *A pair $c_i, c_j \in \mathcal{C}$ s.t. $(c_i, \mathcal{M}(c_i)), (c_j, \mathcal{M}(c_j)) \in \mathcal{M}$, with $c_i \neq c_j$, is a blocking pair if $U_{sys}(\mathcal{M}_{c_i,c_j}) > U_{sys}(\mathcal{M})$ and no player c_i or c_j is worse off after swap.*

Let $\mathcal{BP}$ be the set of BPs accordingly to *Definition 3*. For each $c_i, c_j \in \mathcal{BP}$ s.t. $\mathcal{M}(c_i) = b_h$, let $\mathcal{BP}_i^h$, $\mathcal{BP}_i^h \subseteq \mathcal{BP}$ s.t.

$$\mathcal{BP}_i^h = \left\{ c_k \in \mathcal{BP} \;\middle|\; \begin{array}{c} \mathcal{M}(c_k) = b_n \wedge \text{rank}(c_i, b_n) = 1 \\ \wedge \quad \text{rank}(c_k, b_h) = 1 \end{array} \right\}. \tag{3.12}$$

Hence, we can introduce $\Delta U = U_{sys} - U'_{sys}$, where U'_{sys} is mean system dropping probability after exchange.

Therefore, the method reaches a stable matching after the following steps

- detect the set of BPs $\mathcal{BP}$;
- for each c_i in $\mathcal{BP}$, select $\mathcal{BP}_i^h$;
- if exists, find $c_j^* \in \mathcal{BP}_i^h$ which maximizes ΔU;
- divide the previously matched pairs (c_i, b_h) and $(c_j^*, \mathcal{M}(c_j^*))$;
- build new pairs $(c_i, \mathcal{M}(c_j^*))$ and (c_j^*, b_h).

THe termination of the presented procedure is guaranteed since the search space of the blocking pairs has finite dimension, having the set $\mathcal{C}$ a finite number of elements. Furthermore, the swap is provided just if $\Delta U > 0$, ensuring that two pairs can exchange each others a finite number of times and there are no infinite loops.

Lemma 1. *The post matching procedure algorithm reaches a stable matching.*

Proof. Let $\mathcal{M}$ be the assignment configuration resulted by the proposed matching algorithm. Similarly, let $\mathcal{M}'$ be the outcome matching of the post matching procedure. Then, $\mathcal{M}'$ is a stable matching in reference to Definition 2. Therefore, in the following, we show that $\forall c_i \in \mathcal{M}'$ there not exists a $c_j \in \mathcal{M}'$ s.t. the three conditions $\Delta U_i(\mathcal{M}') \geq \Delta U_i(\mathcal{M}^{'}_{i,j})$, $\Delta U_j(\mathcal{M}') \geq \Delta U_j(\mathcal{M}^{'}_{i,j})$ and $\Delta U_{sys}(\mathcal{M}') \geq \Delta U_{sys}(\mathcal{M}^{'}_{i,j})$ [26]. For the sake of simplicity, we name the previously three conditions as t_i, q_j, r_{sys}, respectively. Hence, we prove that the logical expression $(t_i \wedge q_j \wedge r_{sys}) = \mathit{false}, \forall c_i, c_j \in \mathcal{M}'$. Therefore, we analyze the following possibilities

- if $(\neg t_i \vee \neg q_i \vee \neg r_{sys}) = \mathit{true} \Rightarrow (t_i \wedge q_j \wedge r_{sys}) = \mathit{false}$;
- if $(t_i \wedge q_i) = \mathit{true}$ in outcome matching $\mathcal{M}'$, since post matching procedure acts detecting $c_i, c_j \in \mathcal{M}$ s.t. t_i and q_j are verified, necessarily results $\neg r_{sys} = \mathit{true}$, otherwise c_i and c_j would be exchanged during runtime execution and in $\mathcal{M}'$ should be verified $(t_i \wedge q_i) = \mathit{false}$.

□

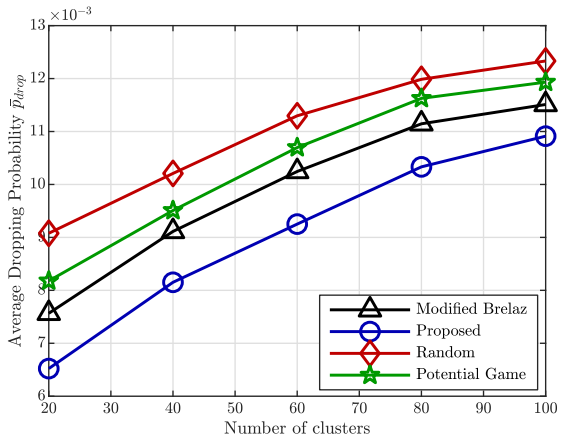

Figure 3.3: Mean total system dropping probability performance.

3.1.9 Numerical Results

In order to validate the applicability of the proposed framework to the system dropping probability minimization problem in LR-WPAN systems, per-

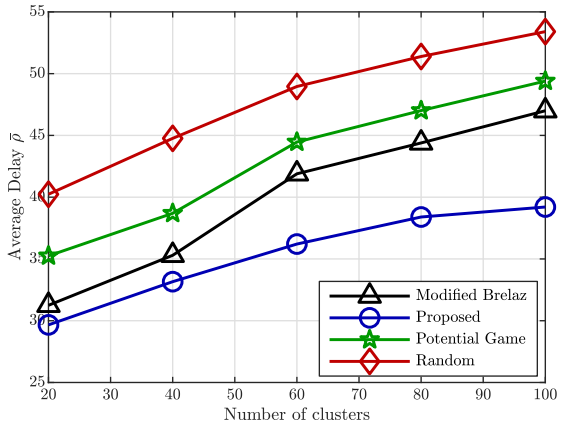

Figure 3.4: Mean data packet transmission attempts performance.

Table 3.1: Simulation parameters

Parameter	**Value**
path loss	free space model
transmission power	0 dBm
power threshold	-82 dBm
LR-WPAN radius	180 m
number of CPCs	$(0, 100]$
cluster size	$[15, 20]$

formance comparisons with the basic random allocation method (RA), the potential game (PG) [282], and with a method based on the use of the well known Brelaz algorithm [136, 222, 294] in a modified form are also presented and critically discussed. In the RA case, the SPC randomly assigns TVWS bands $\in \mathcal{B}$ following an uniform distribution, and without considering mutual interference constraints. Furthermore, the Brelaz's algorithm is a greedy algorithm that solves the graph coloring problem [136, 222, 294]. Since in the problem addressed in this work the number of available channels is fixed and that the network access has to be provided to all the CPCs demanding for it, the Brelaz's algorithm in its traditional form cannot be applied. Therefore, the modified Brelaz's algorithm applied consist consists of

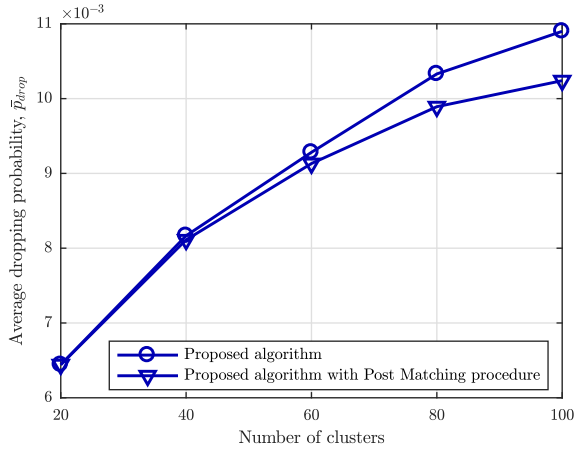

Figure 3.5: Mean total system dropping probability with and without post-matching procedure.

1. start from the most interfered CPC;
2. assign to the CPC the first available band among the bands set and delete the selected band from the available bands pool of each neighbor CPC;
3. if no band in $\mathcal{B}$ set is available, assign to the CPC the band used by the farthest interfering CPC from itself;
4. update the available bands pool of each neighbor CPC;
5. select the next CPC with the maximum number of available bands. In case of a tie, choose the most interfered CPC. If there is still a tie, choose one from them randomly;
6. go to step 1) until all CPCs are allocated.

System performance with the different allocation disciplines are expressed in terms of $\bar{p}_{drop}$, mean data packet transmission attempts, $\bar{\rho}_a$, given by the mean number of access attempts performed, in accordance with the CSMA/CA discipline, to achieve a success in packet delivery, averaged over all the active clusters and, finally, *Jain's index*, representing the system fairness. Furthermore, the robustness to estimation errors affecting input data,

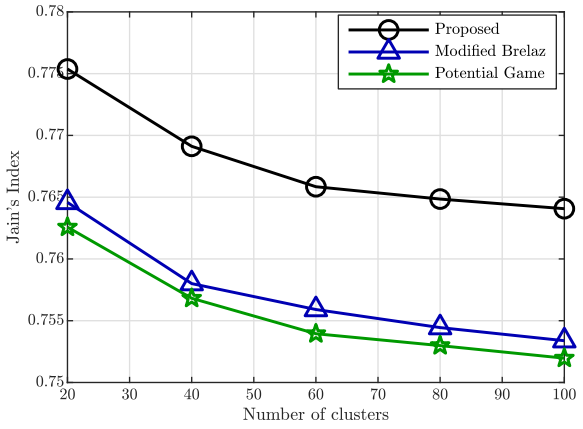

Figure 3.6: Jain's fairness index performance.

for example due to interference between CPCs pairs, has been taken into account.
Table 3.1 exhibits the simulation parameters, in which the power threshold T_h and the transmission powerhave been set in accordance with [4] and [50], respectively. The CPCs have been spatially distributed within the LR-WPAN area according to a Poisson Point Process (PPP). The network is assumed with regular circular topology having radius of $10\sqrt{10}$ m, a number of clusters $n \in (0, 100]$, a number of end-devices for each cluster independently selected according to an uniform rule in $[15, 60]$, and a number of TV white space bands equal to 8 without differences in terms of channel quality.

In performing our analysis we have focused on the case of a set of 8 available TVWS bands to be allocated to a variable number of clusters as specified in Table 3.1. In addition to this, for the considered CSMA/CA scheme, we have assumed according to [110], the minimum and maximum value of the backoff window length (slots), see the Appendix, equal to 2^3 and 2^8, respectively. As a consequence, the value of the number of backoff stages m, results to be equal to 6. Under such assumptions, Figure 3.3 shows $\bar{p}_{drop}$ for the proposed matching algorithm in comparison with the modified Brelaz algorithm, PG and RA alternative [26]. The figure highlights, as expected, a non decreasing behavior of $\bar{p}_{drop}$ with respect to the number of active clusters for all the considered alternatives. However, the better behavior of

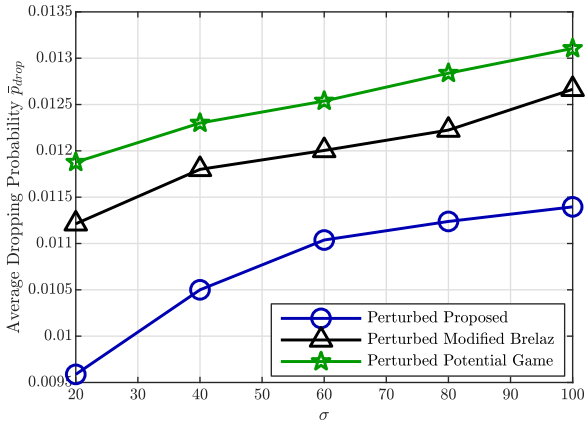

Figure 3.7: Algorithms robustness performance.

the proposed solution is clearly evident. Specifically, with low CPCs density values, the modified Brelaz method outperforms the RA scheme by 16%, the PG approach by 10%, while the proposed approach has a gain of 28%. Furthermore, in high density conditions, the modified Brelaz improves the RA scheme by 5%, while the PG by 3%, and the proposed scheme has a gain of 13% that confirms the superiority of the proposed solution where we have to face with a huge number CPCs as in an ultra dense scenarios. .

The benefits of the proposed technique in terms of achieved $\bar{\rho}_a$ are then illustrated in Figure 3.4 by increasing the clusters number.

In Figure 3.3, we can note that, for a number of CPCs equal to 60, we have a gain for the proposed approach of $12\%, 19\%, 22\%$ with respect to the modified Brelaz, PG and RA, respectively [26]. In Figure 3.3, by setting a target value for the packet dropping probability equals to 10^{-2}, the number of supported CPCs for the comparison approaches are 56,49,36, respectively, while the value of 74 CPCs is managed by the proposed strategy. Figure 3.5 shows the $\bar{p}_{drop}$ trend of the proposed algorithm with and without the post-matching procedure. From this figure it is evident that the allocation configuration resulting from the proposed matching scheme is close to a stable matching configuration, especially for small number of CPCs. Differently, as the number of the CPCs increases, the application of the proposed post-matching approach improves the performance.

In order to provide an accurate analysis, the Jain's index, which is a fairness measure widely adopted in resource allocation field [100, 218], has been considered. Therefore, the Jain's index is defined as follows [26, 118]

Definition 4. *Let $v \in \mathbb{R}_+^m$ be a given vector. We can define the Jain's fairness index as $\mathcal{J} : \mathbb{R}_+^m \to \mathbb{R}_+$ s.t.*

$$\mathcal{J}(v) = \frac{(\sum_{i=1}^{m} v_i)^2}{m \sum_{i=1}^{m} v_i^2}, 0 \leq \mathcal{J}(v) \leq 1. \tag{3.13}$$

Figure 3.6 shows the $\mathcal{J}(v)$ trend for the proposed approach and the considered alternatives, as the number of CPCs grows. Since the fairness increases the closer the $\mathcal{J}(v)$ gets to 1, Figure 3.6 shows that the proposed scheme outperforms the proposed comparison methods, for all the considered number of CPCs.
Finally, Figure 3.7 depicts the influence of the estimation errors in evaluating (3.1) and $\mathcal{I}_j^*$, on the performance in terms of $\bar{p}_{drop}$ for the alternatives taken into account. In performing our analysis we have modeled the estimation errors as *iid* Gaussian random variables with standard deviation σ. As it is evident in Figure 3.7, as σ grows, $\bar{p}_{drop}$ worsen for both the schemes, however, the proposed scheme achieves better results.

3.1.10 Conclusions

The minimization of the system dropping probability in LR-WPANs performing access with the CSMA/CA discipline has been performed through the formulation of a two-sided matching framework. The proposed scheme combines matching theory with a graph coloring method, aiming at reducing the interefring CPCs and, consequently, the data packet dropping probability averaged over all the CPCs. More in depth, the proposed algorithm consists of two sub-procedures: the graph coloring phase, to assign bands only to non interfering CPCs, the second one in which the graph coloring procedure is relaxed and it assigns clusters to bands, taking into account the SPC perspective. In addition, the framework provides a method to divide unstable pairs, exchange their respective partners, in case that their swap may produce a performance improvement. Finally, performance has been provided also in terms of the Jain's index and robustness to estimation errors of the starting conditions.

3.2 Efficient Matching for Almost Blank Subframes Allocation in Ultra Dense Networks

3.2.1 Motivation

The challenge of the fifth generation (5G) of wireless systems is to overcome fundamental limits of existing cellular networks. In fact, in the future mobile networks, increments are expected in mobile data volume and heterogeneity in technologies and paradigms, whose differences need to be integrated and managed [18,36]. Ultra-dense networks have been identified as promising way to massively improve communication capacity and spectral efficiency, since network densification reduce the transmission distance and large resource reuse.However, interference in ultra-dense networks still remains a concrete challenge, due to unpredictable complex nature, compared to traditional networks, due to the dense and random deployment of network infrastructure and different association policies [27, 167, 279].

3.2.2 Contributions

In this subsection the work presented in [27] is discussed, and the proposed cell-centric solution that allows to mitigate both cross and co-tier interference analyzed. More in depth,the enhanced inter cell interference coordination (eICIC) is adopted to limit cross-tier interference, while a new co-tier interference management strategy is proposed. In fact, eICIC is a low-complexity technology in which, periodically, a number of subframes are partially muted using almost blank subframes (ABS) at the macrocell base station (MBS) to lower the impact on the most vulnerable small cell UE (SUE), [27, 28]. However, in UDNs the close proximity of small cells may provide significant levels of interfernece, especially for the edge SUEs [27]. THerefore, a proper allocation management between ABSs and small cells plays a crucial role. For all these reasons, a centralized algorithm aiming at minimizing a novel concept of *cross-layer interference* metric is proposed. The framework designed uses the matching theory principles to provide a stable matching between the set of small cells and the set of available subframes left almost blank by the macrocell and used by the small cells to communicate with their most vulnerable users [27]. The proposed solution aims at improving the throughput of most critical users with an affordable computational complexity [27]. The effectiveness of the proposed scheme has been verified in

comparison with benchmark methods [27].

3.2.3 System model

A heterogeneous network where many small cells are densely deployed in the macrocell area, operating on the same frequency band, is considered. Furthermore, a CloudRAN approach is assumed, in which cells are handled by a central entity. Homogeneous Poisson point processes have been used to deploy Small cells and UEs within the network area. The UEs in each cell are divided in two sets depending on their received power, i.e., *critical UEs* and *non-critical UEs* [27]. The presented work focuses on the critical UEs, which experience a signal-to-noise-ratio (SNR) lower than a given percentile [27]. Let $\mathcal{S}$ be the set of S small cells in the area, while $\mathcal{U}_s$ and U_s represent the set and the number of critical UEs in the s-th cell, respectively [27]. To limit the cross-tier interference, an eICIC based discipline is applied, for which the macrocell does not transmit data in the periods during which small cells communicate with the most vulnerable UEs. In practical terms, since in dense small cells scenarios the co-tier interference occurs inevitably when two nearby small cells access the same ABS simultaneously, the assignment of different ABSs to interfering cells becomes of paramount importance. We assume to have a set $\mathcal{R}$ of R available ABSs (or ABSs' portions), with $R < S$ and each small cell can use only one ABS to communicate with its critical UEs, hereafter referred to simply UEs. Let $\mathcal{S}_r$ be the set of small cells which use the r-th ABS, the goal of the proposed approach is to detect the sets $\mathcal{S}_r$ that optimize system performance through the matrix $\mathbf{M}$, whose dimensions are $S \times R$,in which the generic element, $m_{s,r}$, is one if the r-th channel is assigned to the s-th small cell, zero otherwise. As direct consequence, the u-th UE connected to the s-th cell has an average signal to noise plus interference ratio (SINR) equals to

$$\Gamma_{u,s}(r) = \frac{P_{u,s}}{N + D\Upsilon_u + \sum\limits_{c \in \S - \{s\}} m_{c,r} P_{u,c}}, \tag{3.14}$$

in which $P_{u,c}$ is the power received by the u-th user from the c-th small cell, Υ_u represents the power received by the u-th user from the macrocell, D is the macrocell power level reduction in ABSs, and N is the AWGN noise power.

In order to take into account the quality of service (QoS) of the provided

connection, we impose that a SUE can communicate only if its SINR value is higher than a given threshold Γ_T, otherwise it is considered to be in outage. The system throughput is calculated as the sum of the capacities of the *no-outage UEs*, normalized by the bandwidth, that is

$$T = \sum_{r \in \mathcal{R}} \sum_{s \in \S} m_{s,r} \sum_{u \in \mathcal{U}_s} a(u) \log_2(1 + \Gamma_{u,s}(r)), \tag{3.15}$$

with $a(u) = 1$ if $\Gamma_{u,s}(r) \geq \Gamma_T$, $a(u) = 0$ otherwise.

3.2.4 Problem formulation

The aim of the proposed scheme is to increase the network efficiency in terms of throughput by reducing the interference among small cells in ABSs taking into account QoS requirement (i.e., Γ_T) and cells fairness. The traditional UE-centric problem formulation is the maximization of the sum-rate of all UEs in the area, however, the solution of this problem can be computationally unaffordable for a UDN, given the huge number of UEs and base stations. Indeed, the central entity should know the actual SINR value of each UE for each interference scenario. This would require high signaling overhead, high computational complexity and the algorithm should be performed for every change in UEs SINR values.

Differently, we formulate a cell-centric problem. In this case, measurements are performed and sent to the central entity only during the set-up phase: the s-th SBS measures the level of signal (interference) received from the c-th small cell in a co-channel deployment, $I_{s,c}$ with $c = 1, \cdots, S$ and $c \neq s$, while $I_{s,s} = 0$. In particular, we introduce a new cross-layer interference metric that takes into account both the physical and the MAC layer interference. The first one considers the effective amount of received interference, while the second is an interference abstraction defined as the number of neighboring users communicating on the same channel [70]. Therefore, the cross-layer interference defined here, considers the physical interference experienced by each SBS on a given ABS weighted by the total number of UEs allocated on that ABS. The minimization of this metric, allow cells with a higher number of UEs to perceive lower interference levels, and the

problem can be formulated as

$$\min_{\mathbf{M}} \sum_{r\in\mathcal{R}} N^r(M) \sum_{s\in\mathcal{S}} \sum_{c\in\mathcal{S}-\{s\}} m_{s,r} m_{c,r} I_{s,c} \tag{3.16}$$

$$\text{s.t.} : \sum_{r\in\mathcal{R}} m_{s,r} = 1, \quad s \in \mathcal{S} \tag{3.17}$$

with $N^r(M) = \sum_{s\in\mathcal{S}} m_{s,r} U_s$ the total number of UEs on the r-th ABS.

3.2.5 Proposed solution

In reference to the analyzed problem, the matching theory is applied to provide a suitable match between the set of SBSs, $\mathcal{S}$, and the set of available ABSs, $\mathcal{R}$. Under the considered assumptions, each SBS can use only a single ABS, while different SBSs can use the same resource to improve the spectrum efficiency. Hence, each ABS can be assigned to one or more SBSs, but the number of SBSs that can use the same ABS is limited by the mutual interference. Hence, the proposed centralized algorithm builds a preference list for each cell over ABSs and a preference list of ABSs over cells. Cells rank their preferences taking into account the level of received interference, while ABSs balance the selfish behavior of the SBSs taking into account the number of UEs communicating on it.

3.2.6 Preference lists

As stated before, matching is performed on the basis of preference lists that are built by SBSs and ABSs to rank potential matching. In particular, we indicate with $\mathcal{P}_s$ the preference lists of the s-th cell whose elements belong to $\mathcal{R}$ and with $\mathcal{P}_r$ the preference lists of the r-th ABS, whose elements belong to $\mathcal{S}$. The s-th SBS ranks ABSs based on local metric, according to following the preference function,

$$f_s(r) = -\sum_{c\in\mathcal{S}-s} I_{s,c} m_{c,r} \quad r \in \mathcal{R} \tag{3.18}$$

that is, each SBS ranks the ABSs based on the PHY-interference (i.e., level of received interference). Hence, $\mathcal{P}_s$ is given by the indexes (r) of $f_s(r)$ sorted in descending order. Similarly, the r-th ABS ranks SBSs based on the number of UEs allocated on it

$$f_r(s) = \sum_{c\in\mathcal{S}-s} m_{c,r} U_c + U_s \quad s \in \mathcal{S} \tag{3.19}$$

hence, the ABS takes into account the MAC-interference i.e., the number of UEs communicating on a given ABS, thus giving less utility to SBSs with few UEs. $\mathcal{P}_r$ is given by the indexes of $f_r(s)$ sorted in descending order. It is clear that SBSs preference over ABSs set depends strictly on the SBSs already allocated, and both ABSs and SBSs preference lists change during the matching game. Moreover, these are mutually influenced by previous assignments, resulting in interdependencies and correlations among the preferences. In order to deal with such conditions, the preference list of each element in both involved sets have to be updated upon each assignment. This type of matching game belongs to the matching game with externalities class [105].

3.2.7 Proposed Matching Strategy

The proposed algorithm builds iteratively the outcome matching as follows

1. Each $r \in \mathcal{R}$ and each unassigned $s \in \mathcal{S}$ build their own preference list, $\mathcal{P}_r$ and $\mathcal{P}_s$ respectively, according to (3.19) and (3.18);
2. each unassigned $s \in \mathcal{S}$ sends a match proposal to its most preferred ABS r^s, that is the element in first position in its preference list (i.e., $r^s = \mathcal{P}_s(1)$);
3. let $\mathcal{A}$ be the ABSs set that receive one or more allocation requests from the interested SBSs, $\mathcal{A} \subseteq \mathcal{R}$. Each $z \in \mathcal{A}$ selects its favorite matching proposal s^z among those submitted, that is the request that occupies the highest position i_{s^z} on its preference list;
4. let $\mathcal{F}$ be the set of the indexes of the favorite matching proposals received by the $\mathcal{A}$ elements, $\mathcal{F} = \{i_{s^z}\}_{z \in \mathcal{A}}$. The algorithm selects the minimum index in $\mathcal{F}$, that identifies the most satisfied assignment among the current possibilities. More in depth, supposing $j_{u^v} = \min \mathcal{F}$, the new allocation is the (u, v) pair.
5. the SBS u is deleted from the set of the unallocated SBSs;
6. steps $1) - 6)$ are repeated until all SBSs are allocated.

In the matching problems with externalities, the stability analysis is a very complex and challenging issue. In this respect, there exist many stability definitions, and in this paper we consider the following one, known as the *two-sided exchange-stability* (2ES) [35].

Definition 5. Let $\mathcal{M}$ be a given matching, and let $\mathcal{M}(s)$ be the ABS partner of the SBS s in the matching $\mathcal{M}$, $\mathcal{M}$ is a 2ES matching *if* there not exists a pair of SBSs (s_1, s_2) s.t.:

1. $P_{s_1}(\mathcal{M}(s_2)) \geq P_{s_1}(\mathcal{M}(s_1))$ and
2. $P_{s_2}(\mathcal{M}(s_1)) \geq P_{s_2}(\mathcal{M}(s_2))$ and
3. $P_{\mathcal{M}(s_1)}(s_2) \geq P_{\mathcal{M}(s_1)}(s_1)$ and
4. $P_{\mathcal{M}(s_2)}(s_1) \geq P_{\mathcal{M}(s_2)}(s_2)$ and
5. $\exists z \in \{s_1, s_2, \mathcal{M}(s_1), \mathcal{M}(s_2)\}$ s.t. at least one of the previous conditions is strictly verified.

The above definition expresses that both SBSs and ABSs involved in the swap have to get benefit from the swapping. In this paper we provide a post matching procedure to provide a stable outcome matching according to Definition 5. More in depth, the post matching procedure acts on the allocation matrix $\mathbf{M}$ resulting from the application of the proposed matching strategy, and its behavior can be summarized through the following steps:

1. each $s_1 \in \mathcal{S}$ attempts the swap with each $s_2 \in \mathcal{S}'$, where $\mathcal{S}'$ is defined as $\mathcal{S}' = \mathcal{S} \setminus \{s_1\}$;
2. for each pair (s_1, s_2), conditions $1) - 5)$ of Definition 5 are evaluated;
3. if at least one of the conditions $1) - 5)$ of Definition 5 results unverified, the swap is denied;
4. otherwise, if all the conditions $1) - 5)$ of Definition 5 are satisfied, the matching is redefined as
$$\mathcal{M} = \mathcal{M} \setminus \{(s_1, \mathcal{M}(s_1)), (s_2, \mathcal{M}(s_2))\} \bigcup \{(s_1, \mathcal{M}(s_2)), (s_2, \mathcal{M}(s_1))\};$$
5. repeat previous steps until there not exists in $\mathcal{M}$ a pair (s_1, s_2) for which the swap is allowed.

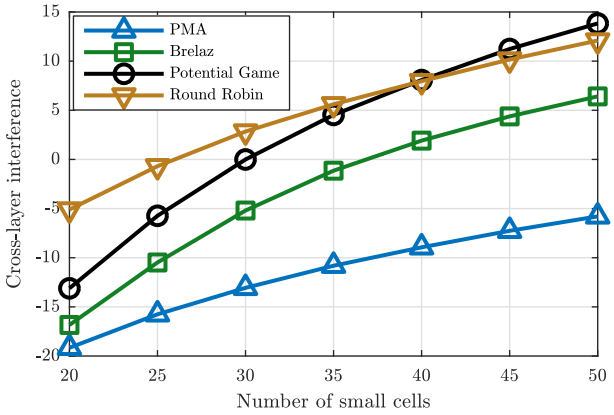

Figure 3.8: Cross-layer interference metric vs S, when $R = 4$ and $\Gamma_T = -2dB$.

3.2.8 Numerical Results

In order to properly test the behavior of the proposed framework, a dense scenario consisting of a variable number of small cells with mean value in the range $[10-50]$, deployed on squared area whose side is 100 m, is considered. Small cells transmitted power is 26 dBm and the pathloss exponent is 3.9.

According to the 4G standard, the number of ABSs per frame should be assumed in the range $[1, 3]$ [5]. Here, we suppose that also half-subframe can be assigned to a cell, hence the number of considered ABS, R, is in the range $[2, 6]$. Moreover, we focus on performance of the most *critical UEs* that are usually allocated on ABSs. In particular, we consider the UEs in each cell that receive a SNR lower than the 20-th percentile[1]. Simulation outputs of several realizations have been averaged to make the results independent on the particular distribution of UE devices and small cells. To validate the efficiency of of the proposed scheme in terms of throughput increase we consider three different benchmark methods: *(i)* the *Round Robin* that successively assigns ABSs to the cells with a circular order; *(ii)* the graph-coloring approach based on *Brelaz* algorithm [222] that tries to assign different colors (i.e., ABSs) to interfering cells; *(iii)* the solution based on *potential game*, specifically proposed in [282] for ABSs allocation in UDNs,

[1]Different percentiles have been evaluated, but the general behavior of the proposed scheme is not affected by this value.

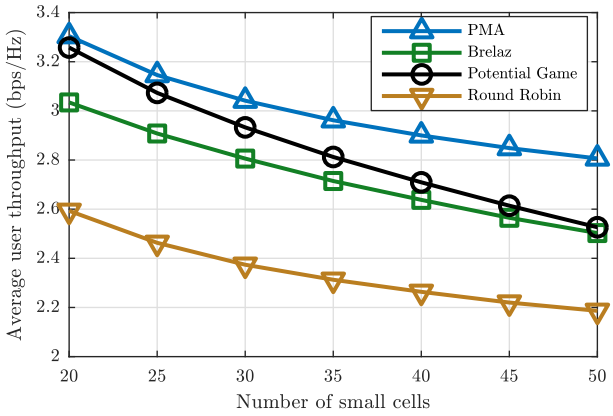

Figure 3.9: Mean throughput per UE ($\Gamma_T = -2$dB) vs S, when $R = 4$.

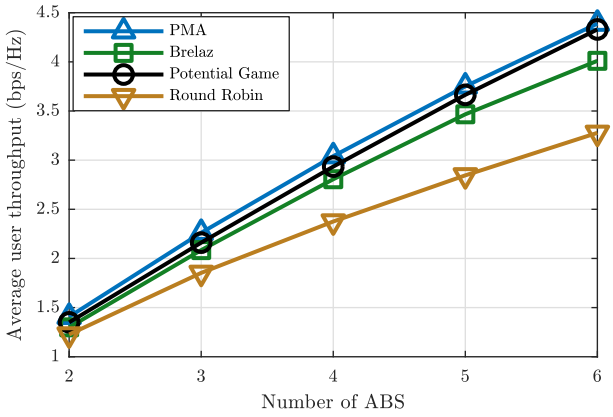

Figure 3.10: Mean throughput per UE ($\Gamma_T = -2$dB) vs R, when $S = 30$.

where for a given number of iterations, a randomly selected cell choices the ABS that senses as less interfered.

First of all, we want to show the behavior of the *cross-layer interference metric*, as defined in Sect. 3.2.4, that is the utility function we want to minimize. This is shown in Figure 3.8 as a function of the number of cells in the reference area, in comparison with benchmarks. We can see that the proposed matching algorithm (PMA) performs better than the alternatives, especially for a high number of SBSs. As a consequence, the proposed scheme always achieves higher throughput, especially for a high cell density. This can be seen from Figures 3.9 and 3.10 where the actual effect of the interference on the critical UEs is evaluated by means the *average user throughput* calculated as the system throughput defined in (3.15) normalized to the number of critical UEs in the area. This is shown as function of the number of cells when $R = 4$ in Figure 3.9 and as a function of the number of ABSs when $S = 30$ in Figure 3.10. The proposed method has a significant gain in comparison with benchmark methods, and it is possible to observe that the minimization of the selected utility function corresponds to the increase of the system throughput. This is because the considered cross-layer interference metric takes into account not only the level of interference among cells, but also how many UEs are subject to a given level of interference, thus increasing the throughput (i.e., reducing the interference on a cell that has few users is less useful than reducing the interference on a cell that has a higher number of UEs). Differently, previously proposed methods are based only on the physical level of interference received by the cells: a binary model in Brelaz and a generalized interference model (with different interference layers) in Potential Game. The proposed matching algorithm performs well also in terms of outage as shown in Figure 3.11. In particular, in this figure the outage of PMA and Potential Game strategies is provided as a function of Γ_T for different values of S and R. We can see that the PMA always outperform Potential Game, even if for a reduced number of cells and channels (i.e., few degrees of freedom) the two algorithms have almost the same performance. As expected, the outage increases as the QoS requirement becomes more stringent. As only one ABS can be assigned to each SBS for communicating with its critical UEs (i.e., each ABS has a fixed number of physical resources - PR - that can be assigned to UEs), it is important that UEs in densely populated cells experience higher SINR and hence higher throughput per Hz. In this way, even if the number of PRs

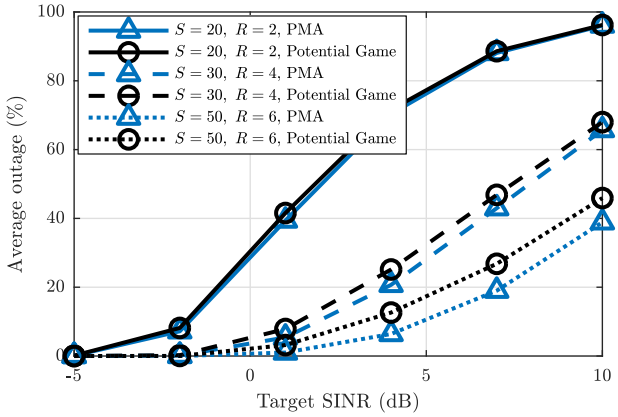

Figure 3.11: UEs in outage vs Γ_T for different values of S and R.

for UEs is lower than in the other cells, the throughput per PR is higher. This allows a *fairer distribution* of the throughput among UEs of different cells. This is shown in Figure 3.12 where the Jain index $\left(\frac{\left(\sum_i x_i\right)^2}{n\sum_i x_i^2}\right)$ of the averaged throughput of the cell normalized to the number of UEs per cell is shown. As expected the proposed method presents the highest fairness.

3.2.9 Conclusions

Network densification is one of the key technologies to achieve improvements of communication capacity in 5G wireless systems. Despite the promising communications performance, the close proximity resulting from densification makes interference a challenging issue. In order to limit this problem, a solution to reduce both cross and co-tier interference is proposed. In particular, eICIC is adopted to limit cross-tier interference, and a novel strategy based on matching theory principles is designed to assign almost blank subframes allocation to small cells, thus managing also the co-tier interference. A new cross-layer interference metric is proposed to catch both the level of received interference and the number of users in the cells. The suitability of the proposed method has been proved in terms of different metrics in comparison with benchmark methods.

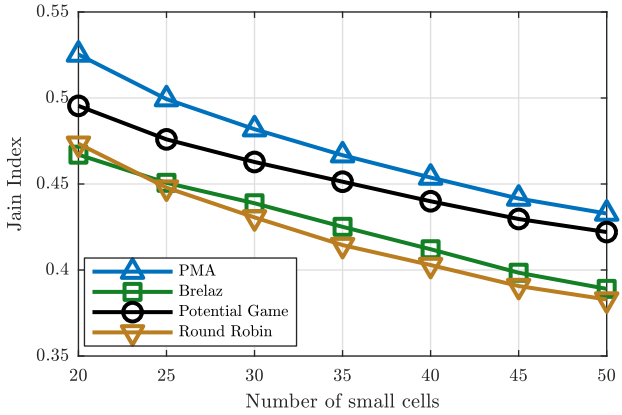

Figure 3.12: Jain index (fairness) of the average user throughput among cells.

3.3 A Low Complexity Matching Game Approach for LTE-Unlicensed

3.3.1 Motivation

One of the most crucial problem in modern networks is the scarcity of available spectrum [139], caused by the ever-growing demand of intensive applications and the diffusion of the novel network paradigms such as IoT, LTE assisted, V2V communications [61], and so on. Recently, the LTE-Unlicensed (LTE-U) standard has gained momentum as technology to offload cellular traffic to the unlicensed bands[7], by performing Carrier Aggregation (CA) methodology, typical of the LTE-Advanced (LTE-A) [7]. Despite LTE-U provides higher bit rate and bandwidth, spectral efficiency, seamless services and coverage, its uncontrolled application causes significant disadvantages to unlicensed users. As consequence, carrier aggregation assignment mode in the unlicensed spectrum is still an open issue [139].

3.3.2 Contributions

This study, resulted in paper [58], investigates the resource allocation problem with the LTE-Unlicensed technology, aiming at increasing unlicensed system performance. The aim of the work is the maximization of the un-

licensed overall network sum rate. The problem is formulated as a many-to-many matching game between UEs and component carriers (CCs), i.e., channels, according with the CA specifications. Differently from [94], this work proposes a more realistic approach since paper [94] assumes the complete knowledge about the unlicensed system conditions, which is not a truthful hypothesis. As consequence, we only suppose that each UEs has knowledge about the signal-to-interference-plus-noise-ratio (SINR) on each channel. Within this context, a matching based algorithm for the resource allocation problem in 5GHz spectrum has been proposed.

Summarizing, the contributions are given by

- Formulation of the resource allocation problem in the unlicensed spectrum, adopting the LTE-U technology;
- Design of a low complexity matching theory based algorithm to perform a many-to-many relationship between UEs and channels;
- Algorithm choices selection aiming at minimizing the inter-cell interference (ICI), through an approximated interference factor detecting the mutual interference level between the small cells.
- Extensive numerical simulations to validate the behavior of the proposed solution, in comparison with the well known Hungarian algorithm (HA).

3.3.3 System Model

The reference scenario, as depicted in Figure 3.13, is composed by a heterogeneous cellular network consisting of femtocells, picocells, a set of UEs $UE = \{ue_1, \ldots, ue_n\}$, and a set of unlicensed small cells (uSCs) $SC = \{sc_1, \ldots, sc_m\}$. The uSCs are arranged for transmission in the unlicensed spectrum, and each UE_i is served by its closest uSC sc_j. The set of unlicensed channels is given by $CH = \{ch_1, \ldots, ch_l\}$. Let p and q be the capacity of each uSC and channel, respectively, expressed as the maximum number of UEs connected to a particular uSC and the maximum number of UEs allocated on a channel, for each uSC. Furthermore, each UE can request up to 2 CCs in the unlicensed spectrum, in addition to its standard licensed band. The bandwidth of each CC has been assumed of 20MHz, and the UEs distributed with the Poisson Point Process (PPP).

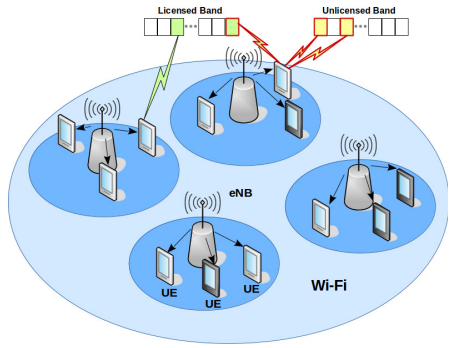

Figure 3.13: Reference network composed by UEs and SCs, where each UE is served by one and only one SC

Assuming that each UE ue_i is served only by uSC sc_j, the received signal-to-interference-plus-noise ratio (SINR) $\Gamma^y_{i,j}$ from sc_j to UE_i, given a channel ch_y can be defined as [58]

$$\Gamma^y_{i,j} = \frac{P_j g_{j,i}}{N_0 + \sum_{k \in \Phi} P_k g_{k,j}}, \tag{3.20}$$

where P_j represents the uSC transmission power, $g_{j,i}$ is the channel gain, and Φ is the set of interfering uSCs. Furthermore, N_0 expresses the noise term.

I t is important to highlight that the interference in (3.20) is estimated considering that the interference generated by the eNB of the k-th uSC is the same for all the UEs belonging to the k-th uSC. This is reasonable, because the scenario considered is highly densified with a limited coverage area. Therefore, considering an unlicensed channel ch_h and the set of uSCs that have at least one own subscriber allocated on ch_h, i. e., S_h, the following condition has to be verified [58]

$$d_h(sc_d, sc_e) \leq thr, \quad \forall d \in S_h, d \neq e, \tag{3.21}$$

in which $d_h(d,e)$ is the interference power between two uSCs, and thr is the maximum acceptable mutual interference level between two interfering uSCs.

The inequality in (3.21) rules the ICI factor. Let sc_s be the small cell such that $sc_s \in S_h$ with t UEs allocated on ch_h. We have that if $t = 0$, ch_h

is *free*, while if $t < q$, then ch_h is *undersubscribed*. Differently, if $t = q$, ch_h is *full*, otherwise, if $t > q$, then ch_h is *oversubscribed*.

3.3.4 Problem Formulation

The resource allocation problem in LTE-U systems has been formulated as the maximization of the incremental network sum rate, as

$$\max_{N_{j,y}} \sum_{y \in CH} \sum_{j \in S_h} \sum_{i \in I(j)} \frac{W_y}{N_{j,y}} log_2(1 + \Gamma^y_{i,j}), \tag{3.22}$$

s.t.:

$$N_{j,y} \leqslant q, \quad \forall j = 1, \ldots, m, \qquad \forall y = 1, \ldots, l, \tag{3.23}$$

$$\sum_{i=1}^{n} \delta_{i,j} \leq p, \qquad \delta_{i,j} \in \{0, 1\}, \qquad j = 1, \ldots, m, \tag{3.24}$$

$$d_y(sc_d, sc_e) \leq thr, \quad \forall sc_d, sc_e \in S_y, \quad d \neq e, y = 1, \ldots, l, \tag{3.25}$$

$$\Gamma^y_{i,j} \geq \Gamma_{min}, \forall i = 1, \ldots, n, \forall j = 1, \ldots, m, \forall y = 1, \ldots, l \tag{3.26}$$

$$P_{j,i} g_{j,i} \leq \Theta, \qquad \forall i = 1, \ldots, n, \forall j = 1, \ldots, m, \tag{3.27}$$

in which *I(j)* is the set of UEs supported by uSC j on channel y. Then, W_y is the channel bandwidth and $N_{j,y}$ is the set of UEs allocated on channel y, and served by uSC j. Furthermore, $\delta_{i,j}$ is a binary value where $\delta_{i,j} = 1$ if UE_i is served by sc_j and $\delta_{i,j} = 0$ otherwise. In addition, constraint (3.23) and (3.24) represent that the number of UEs allocated on a given channel, under a certain uSC, and the number of UEs that an uSC can serve are both upper bounded. Similarly, the inequality in (3.25) is referred to the upper bound on the acceptable interference power. Finally, constraints (3.26) imposes the minimum value of SINR for each UEs, while in (3.27) we protect the unlicensed system, by requiring that the UEs' interference is less or equal to the thermal noise Θ.

3.3.5 Algorithm Characterization

The student-project-allocation algorithm (SPA) algorithm [11], belonging to the whole class of matching algorithms, consists of a set of lecturers who offer a range of projects to a set of students who apply for them.

In our problem the set of lecturers are identified by the uSCs, while the projects are the unlicensed channels. Finally, students are the UEs.

The proposed algorithm, hereafter referred as asymmetric-SPA (A-SPA), achieves a many-to-many matching between the UEs and channels, according to the constraints of the optimization problem. For each channel, each UE measures the SINR as in (3.20), and builds its own preferences list by sorting the resulting values in descending order. In the initial phase of the A-SPA algorithm, all the UEs are unallocated. Then each UE proposes its allocation on its most preferred channel. Let ch_r be the channel submitted by UE_i connected to the $k-th$ uSC. The UE UE_i is temporarily allocated on the proposed channel ch_r if one of the following condition is satisfied [58]

1. ch_r is free;
2. ch_r is full or undersubscribed and on channel ch_r there is at least one UE_j connected to uSC k;
3. ch_r is full or undersubscribed, on channel ch_r there is no UE_j connected to uSC k and the latter doesn't interfere excessively with the other uSCs of each UE_j present on ch_r;
4. ch_r is full or oversubscribed but the SINR value associated to UE_i is greater than the worst SINR value associated to the other UEs currently allocated in ch_r.

Let x be the UE in $UE_{sc}^{\bar{ch}}$ with the lowest SINR value; this value is compared with the SINR value of UE_i. The UE associated to the worst SINR value is then scheduled for reallocation during the next round and it is replaced with UE_i. In the channels allocation process, we take into account the power interference threshold. The same procedure is repeated until all the UEs are not allocated.

3.3.6 Numerical Results

The system performance evaluation is provided by comparing the proposed A-SPA algorithm with the HA, the well-known combinatorial optimization

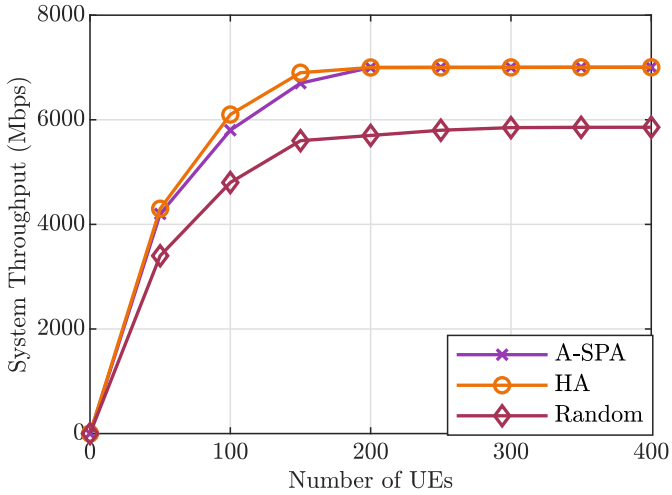

Figure 3.14: Total system throughput

algorithm to solve the assignment problems [185]. The considered reference network is a circular cellular network highly densified, consisting of $N \in [0, 400]$ UEs, $K = 5$ uSCs, $M = 16$ unlicensed channels to $20MHz$. The SINR value was assumed as a uniform random distribution within $[10, 35]$ dB.

Figure 3.14, shows the comparison between the A-SPA and the HA in terms of overall system throughput. As it is straightforward to note, the HA performs slightly better than the proposed A-SPA algorithm, and the two algorithms have approximately the same achieved throughput and their results are very close to each other. Also performance in terms of robustness to perturbed input data is almost the same for both algorithms, as shown in Figure 3.15. Finally, Figure 3.16 exhibits the trend of system throughput as a function of the number of uSCs, and it shows that a low UEs-to-SCs ratio is better than a higher ratio. The major difference between HA and A-SPA is strongly tangible when we consider their computational complexity: the HA presents an original complexity of $O(n^3)$, become $O(n^4)$ considering the CA mechanism, while the proposed A-SPA has a computational complexity equals to $O(n^2)$. Thus, despite the two algorithms have similar performances as regards system throughput, the remarkable gap between time complexity highlights the efficiency of the A-SPA. The expressions of complexity for

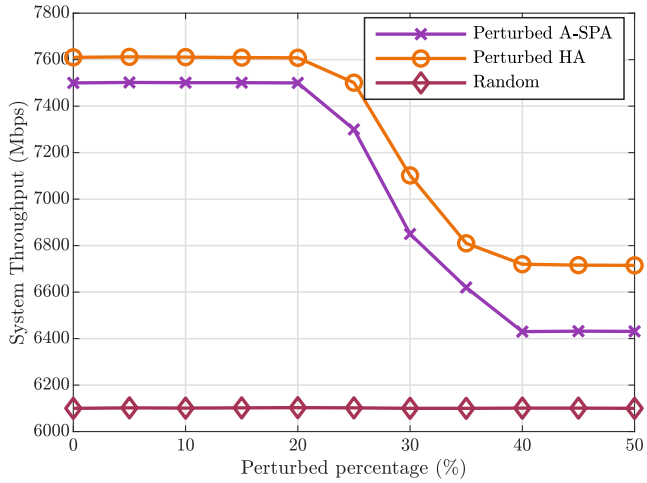

Figure 3.15: Total system throughput with perturbed input data

both A-SPA and HA are respectively

$$n_{UE}(n_{SC}+n_{UE})+\frac{n_{UE}}{2}(n_{SC}+n_{UE})+S(\frac{n_{UE}}{2})\approx \\ \approx O(n_{UE}^2) \tag{3.28}$$

and

$$n_{SC}+n_{UE}+n_{UE}+n_{UE}+n_{UE}+n_{UE}(n_{UE}^3)\approx \\ \approx O(n_{UE}^4) \tag{3.29}$$

where n_{UE} and n_{SC} are respectively the number of UEs and uSCs.

3.3.7 Conclusions

In this work a many-to-many matching algorithm has been proposed to address the resource allocation problem in LTE-U networks. We have proposed a modified version of the SPA algorithm which has the similar performances in comparison to the HA, as regards the system throughput and robustness. However, the HA exhibits a higher computational complexity, resulting in a lower adaptability to dynamic contexts. Therefore, the proposed A-SPA scheme represents a more efficient alternative to the HA for the resource allocation problem in mobile networks.

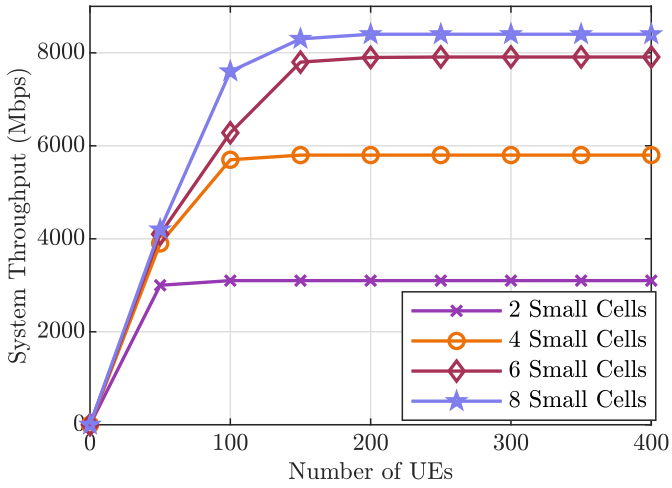

Figure 3.16: Algorithm scalability increasing the number of small cells

Chapter 4

Computational Offloading with Matching Theory

The more I practice, the luckier I get.

Jerry Barber

4.1 A Matching Theory Framework for Tasks Offloading in Fog Computing for IoT Systems

4.1.1 Motivation

Fog computing is an emerging paradigm that extends cloud computing to the network edges. In particular, fog computing refers to a distributed computing infrastructure confined on a limited geographical area within which some IoTs applications/services run directly at the network edge on smart devices having computing, storage, and network connectivity, named fog nodes, with the goal of improving efficiency and reducing the amount of data that needs to be sent to the cloud for massive data processing, analysis and storage [57]. In particular, with the aim at efficiently enabling tasks computation offloading, several approaches have been proposed in the literature.

Recently, the advances in the IoT applications and services have produced a deeply modification of this scenario. Despite the data centre approach al-

lows fast tasks computations, the latency of communications between the CC and the end devices (EDs) makes the CC solution not affordable, especially for EDs located at the network borders. Furthermore, in novel IoT scenarios as industrial applications, smart grids, smart building, intelligent transportation systems and video surveillance to name the few, we have to take into account the huge amounts of data generated from a high number of sensors and other devices [34,86,288]. The integration of FC in IoT based systems implies many advantages for all the IoTs applications where a low latency is essential in taking decisions and actions. In this context, the main problem to face is the identification of a suitable policy to efficiently handling the tasks offloading from EDs to the fog nodes (FNs) [67,153,188,232,233].

4.1.2 Contribution

The strategy presented to offload computationally intensive tasks from EDs to FNs, has been originally proposed in [57]. The main goal here is to provide an effective, low complexity and distributed algorithm to offload computation intensive tasks originated at the EDs level to FNs, by considering all tasks with the same priority. Differently from previous approaches [53,220], the tasks offloading problem is modeled as a matching game with externalities reaching a stable configuration of the outcome tasks offloading solution. In particular, a distributed matching algorithm to minimize the worst total task completion time has been designed.The existence of a direct link between any couple of EDs and FNs has been assumed and the performance comparisons with different tasks offloading schemes is also provided in order to highlight the advantage of the proposed technique and its suitability to pursuing an efficient tasks offloading in FC for IoT applications with low latency requirements. Furthermore, comparisons with the GT based approach presented in [220] is given in terms of time complexity and signaling overhead in order to further pointed out the advantages of the proposed method.

4.1.3 System Model

For the sake of simplicity, in what follows, the boldface letters denote matrix and vectors. In particular, with capital letters we denote matrix and with lower-case letters vectors. All vectors, if not specified, are column-wise vectors. The considered scenario, depicted in Figure 4.1, consists of a set $\mathcal{F}$ of FNs, with cardinality n, i.e., $|\mathcal{F}| = n$, equipped with a central processing

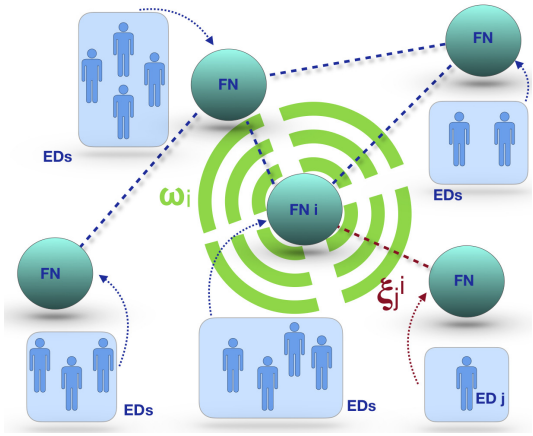

Figure 4.1: Reference Scenario.

unit (CPU) with a different computational capability and storage capacity on board. Furthermore, a pool of tasks $\mathcal{J}$, with $|\mathcal{J}| = m$ has been considered, in which each task belongs to a certain application type, with the set of applications types denoted by $\mathcal{T}$, with $|\mathcal{T}| = \nu$. For readability, i is the index on $\mathcal{F}$, and j the index on $\mathcal{J}$. Finally, u represents the index on $\mathcal{T}$. Supposing $\mathcal{J}$ originated by a set of EDs interested in offloading computational intensive tasks to FNs, a fixed position for EDs and FNs are assumed. The overall run time of tasks offloaded to FN i is defined as

$$\tau_i = \mathbf{a}_i^T \mathbf{t}_i + g(\mathbf{a}_i^T \boldsymbol{\Theta}) \mathbf{c}_i^{\text{miss}} + (\mathbf{a}_i^T \boldsymbol{\Theta} - g(\mathbf{a}_i^T \boldsymbol{\Theta})) \mathbf{c}_i^{\text{hit}}, \tag{4.1}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{m \times t}$, with entries $\theta_{j,u}$ equal to 1, if tasks j is of type u, zero otherwise. In addition, $\mathbf{t}_i \in \mathbb{R}^n$ is the vector of the execution time with entries $t_{j,i}$ denoting the time required to execute task j on FN node i. Moreover, $\mathbf{c}_i^{\text{miss}} \in \mathbb{R}^t$ is the caching cost vector, denoting the additional time required to retrieve a given application on FN i having a cache miss [112]. Likewise, $\mathbf{c}_i^{\text{hit}}$ is the time in case of a cache hit [112]. Furthermore, $\mathbf{a}_i \in \mathbb{R}^m$ is the allocation vector of FN i, for which the j-th entry is 1 if task j is allocated on i or 0, otherwise. Finally, $g(\cdot)$ in (4.1) is defined as

$$g(\rho) = \begin{cases} 0 & \rho = 0; \\ 1 & \rho \geq 1. \end{cases} \tag{4.2}$$

Therefore, in order to evaluate the overall completion time of the task j, we have also to take into account the communication cost that, for the purposes of our analysis, we consider only due to the network delay ξ_j^i, derived according to [53], related to the link between the ED requiring the offloading of task j and the selected *destination* FN i. Hence, the overall completion time for the task j results to be

$$\psi_j = \sum_{i=1}^{n} (\tau_i + 2\xi_j^i)\gamma_{j,i}, \tag{4.3}$$

where $\gamma_{j,i}$ is a binary function that is equal to 1 if task j is assigned to FN i, zero otherwise.

In performing the proposed analysis, the constraint that all the tasks have to be offloaded has been imposed. As a consequence, we must have

$$\sum_{i=1}^{n} \mathbf{a}_i = \mathbf{1}, \tag{4.4}$$

where $\mathbf{a}_i$ represents the allocation vector of FN i defined before and $\mathbf{1}$ a vector with all entries equal to 1.

4.1.4 Problem Formulation

In order to enable real time applications, the worst total completion time minimization problem has been formulated as

$$\min \max_{j \in \mathcal{J}} \psi_j \tag{4.5}$$

s.t.

$$\sum_{i=1}^{n} \mathbf{a}_i = \mathbf{1}, \tag{4.6}$$

It is straightforward to note that the tasks offloading problem represented in (4.5) and (4.6) exhibits high computational complexity that makes prohibitive the use of any centralized method. This motivated the design of a self-organized approach, in which tasks and FNs establish a suitable computation resources pooling, interacting each other without the intervention of a central coordinator. As a consequence, we propose here a matching game method by taking into account specific context information, as the expected task completion and waiting time that depend on the outcomes of the computation offloading process at each processing step. The proposed approach

aims at providing a computation offloading strategy capable of minimizing the worst task total completion time and achieving, in comparison with other alternatives, better results also in terms of mean task waiting, mean total task completion time and fairness. Furthermore, we also demonstrate that the proposed tasks offloading strategy gives rise to a stable matching according to the Definition 6 given in the next Section.

4.1.5 Proposed Solution and Algorithm

Let $\mathcal{J}$ and $\mathcal{F}$ be two sets of players. A *matching game* is the pair $(\mathcal{J}, \mathcal{F})$ on which are defined two preference relations $\succ_j$, $\succ_i$ allowing each player, i.e., a task $j \in \mathcal{J}$ to be offloaded, to express preferences over the opposite players, i.e., FNs $\in \mathcal{F}$, and vice versa.

Let $O_j(\cdot)$ and $D_i(\cdot)$ be the utility functions of task j and FN i, respectively. We can say that if $O_j(i_1) > O_j(i_2)$, task j prefers FN i_1 to FN i_2 and such situation is expressed by $i_1 \succ_j i_2$. Similarly, $j_1 \succ_i j_2$ means that FN i prefers task j_1 to j_2, hence $D_i(j_1) > D_i(j_2)$. Note that, differently from [80,119,208,260], we have to take into account that in our case, due to the specific requirements of the tasks offloading problem, the preferences list of each player are dependent on the others players' preferences, i.e., the preference of offloading a task to a given FN is influenced by the number of tasks already offloaded to that FN [105]. Such class of matching problems is identified in the literature as matching problems with *externalities*, meaning that the players' choices mutually influence and affect the other agents' decisions, by creating interdependencies and correlations among players' preferences. As a consequence, the preferences list of each involved participant has to be updated upon each assignment and, hence, both $O_j(\cdot)$ and $D_i(\cdot)$ result to be time dependent. However, for the sake of notational simplicity, the dependence on the time index is not made explicit for both $O_j(\cdot)$ and $D_i(\cdot)$ in what follows.

The utility function of a generic task j, $O_j(\cdot)$, is evaluated by considering the cost of the communication link ξ_j^i between the ED interested to offload task j to a given *destination* FN, i.e., FN i and the waiting time $\omega_i^j(\zeta)$ that the task j has to suffer before being processed if offloaded to FN i at time ζ. Hence, we have

$$O_j(i) = \frac{1}{\omega_i^j(\zeta) + \xi_j^i}, \forall i \in \mathcal{F} \tag{4.7}$$

Likewise, the utility function of a generic FN i, $D_i(\cdot)$ is given as:

$$D_i(j) = \frac{1}{\omega_i^j(\zeta) + \xi_j^i + t_{j,i}}, \forall j \in \mathcal{J} \tag{4.8}$$

where the term $t_{j,i}$ denotes the time needed to compute task j at FN i.

From what stated before, the proposed tasks offloading algorithm provides a distributed procedure to realize a many-to-one matching between the sets $\mathcal{J}$ and $\mathcal{F}$ [105]. The proposed algorithm is performed in two main matching steps in order to ensure as final result that all the tasks are offloaded to the appropriate FNs. The first step begins after the completion of the neighbors discovery phase by each ED interested in the tasks offloading process in order to acquire a context awareness concerning the FNs computation capabilities and the communication cost towards each possible *destination* FN $\in \mathcal{F}$. Beside, a modified version of the Deferred Acceptance Algorithm (DAA) [84, 211] in the many-to-one version [88], is performed to gather a preliminary matching $\mathcal{M}_1$. Briefly, the modified DAA operates as follows:

- each ED requesting the computation offloading of an intensive task, i.e., task j, selects the most preferred FN in accordance with $\succ_j$;
- each FN i accepts any new task offloading request till the maximum number χ_i of tasks computable in parallel at the FN i, is reached;
- If the FN i receives a new task offloading request more preferred than at least one of those already accepted, when the χ_i value is already reached, the FN i rejects the less preferred task offloading request among those ones previously accepted (based on $\succ_i$) and accepts the new one.
- whenever an ED suffers a task offloading rejection from a FN, differently from the DAA implementation, it does not delete that FN from its preference list.

After the completion of the first step the FNs send out to the EDs an updating for the waiting time suffered by the next (if any) submitted task offloading request.

Hence, the second step of the proposed algorithm is performed as follows:

1. each ED having a task, i.e., task j, not already offloaded, evaluates $O_j(i)$, $\forall i \in \mathcal{F}$;

2. each FN i evaluates $D_i(j)$, for $\forall$ task j not already offloaded $\in \mathcal{J}$;
3. each ED having a task, i.e., task j, not already offloaded submits the offloading request to its top-ranked FN in compliance with $\succ_j$;
4. each FN i receives the task offloading requests from the interested EDs and selects its favorite one among those submitted;
5. FNs send out an updating for the waiting time suffered by any new submitted task offloading request, hence, each interested ED repeats 1) until each task offloading request is satisfied.

4.1.6 Stability Analysis

In accordance with the standard MT [35], we have that the proposed tasks offloading algorithm is stable if it results to be a *two-sided exchange-stable* (2ES) matching, according to the following Definition 6.

Definition 6. A given outcome matching μ is a 2ES matching *iff* there not exist a pair of tasks (j_1, j_2) s.t.:

1. $O_{j_1}(\mu(j_2)) \geq O_{j_1}(\mu(j_1))$ and
2. $O_{j_2}(\mu(j_1)) \geq O_{j_2}(\mu(j_2))$ and
3. $D_{\mu(j_1)}(j_2) \geq D_{\mu(j_1)}(j_1)$ and
4. $D_{\mu(j_2)}(j_1) \geq D_{\mu(j_2)}(j_2)$ and
5. $\exists z \in \{j_1, j_2, \mu(j_1), \mu(j_2)\}$ s.t. one of the previous conditions is strictly verified.

More in general, the 2ES matching definition given in Definition 6 implies that two EDs can modify their tasks offloading decisions only if both get advantage from the swapping. In the same way, if two FNs want to swap two tasks offloading requests, the involved EDs have to get advantage on this. Briefly, a swap is allowed only if both the involved players, i.e., FNs or EDs requesting tasks offloading, strictly improves their utility. Hence, we verify below that the proposed tasks offloading procedure reaches a 2ES matching according to Definition 6.

Let $\mathcal{A}_1$ be the set of tasks offloaded at the end of the first step of the proposed strategy. It can be easily noted that the cardinality of $\mathcal{A}_1$ is an

integer with values $\in [1, \sum_{i=1}^{n} \chi_i]$. Without loss of generality, we can say that $\forall$ offloaded task $j \in \mathcal{A}_1$, the associated EDs cannot modified their tasks offloading decision (all tasks $\in \mathcal{A}_1$ are stably offloaded to the appropriate FNs) due to the fact that $\mathcal{O}_j(\cdot)$ and $\mathcal{D}_i(\cdot)$, defined as in (4.7) and (4.8), cannot change because all the terms on which they depend remain constant with respect to any possible swap.
Now we show that also for each task $\bar{j}$ that remain to be offloaded at the end of the first step of the proposed algorithm, i.e., $\forall \bar{j} \in \mathcal{A}_2 = \{\mathcal{J} \setminus \mathcal{A}_1\}$, we have a stable matching by providing a reduction ad absurdum.

We start our analysis by assuming that at the time instant ζ^* of step 2 completion, i.e., all tasks have been offloaded to FNs, there exists a tasks pair (j_1, j_2) s.t. conditions $1) - 5)$ of Definition 6 are verified. Therefore, by focusing on the condition 1), we must have

$$O_{j_1}(i_2) \geq O_{j_1}(i_1) \tag{4.9}$$

By supposing that j_1 and j_2, at the allocation time ζ, with $\zeta < \zeta^*$, are offloaded to FNs i_1 and i_2 respectively, we must have $O_{j_1}(i_1) \geq O_{j_1}(i_2)$, hence,

$$\frac{1}{\omega_{i_1}^{j_1}(\zeta) + \xi_{j_1}^{i_2}} \geq \frac{1}{\omega_{i_2}^{j_1}(\zeta) + \xi_{j_1}^{i_2}} \tag{4.10}$$

As a consequence, we must have $\omega_{i_1}^{j_1}(\zeta) = \omega_{i_2}^{j_1}(\zeta^*)$, otherwise (4.9) would be impossible, because at its second step the proposed algorithm does not remove any previous accepted task offloading request and, hence, the expected waiting time on a given FN could not decrease. Likewise, we can perform the previous analysis by focusing on the conditions $2) - 4)$ of Definition 6 again having at the end that it must be $\omega_{i_1}^{j_1}(\zeta) = \omega_{i_2}^{j_1}(\zeta^*)$. As a consequence, we have that the condition 5) of the Definition 6 cannot be verified. This contradicts our assumption and, hence, we can conclude, according to Definition 6, that the proposed tasks allocation algorithm is a 2ES matching.

4.1.7 Numerical Results

The good behavior of the proposed computation offloading strategy is highlighted here by providing numerical results derived by resorting to extensive computer simulations. In particular, for each performance result point we have considered 10^3 independent simulation rounds with randomly located

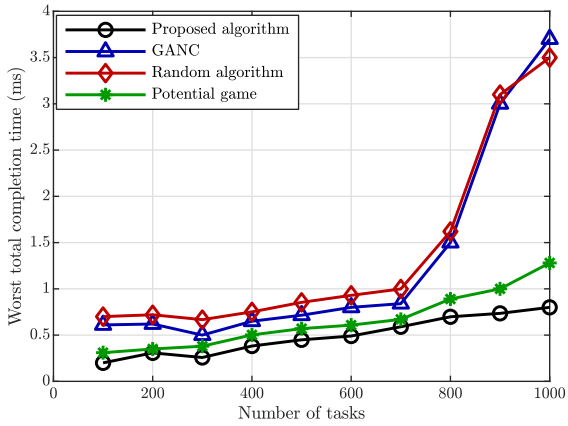

Figure 4.2: Worst task total completion time comparisons.

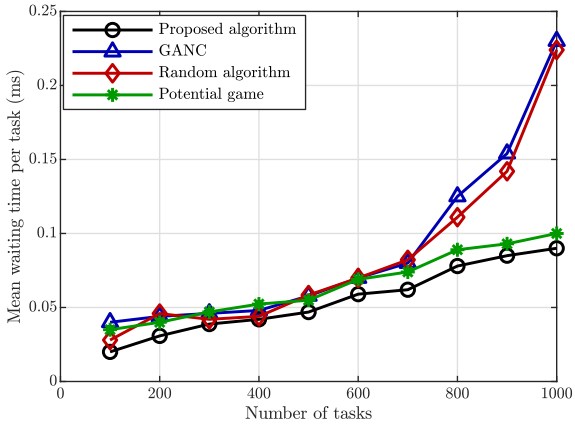

Figure 4.3: Mean task waiting time comparisons.

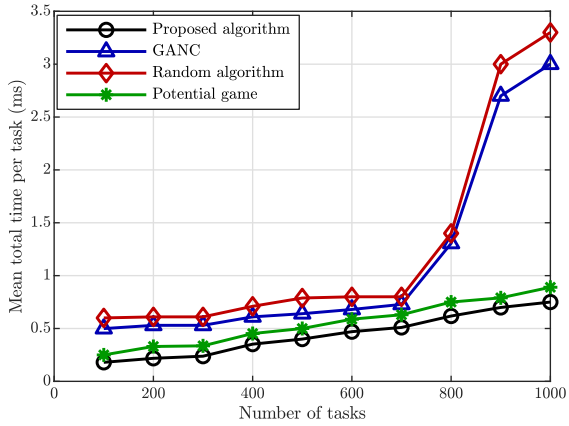

Figure 4.4: Mean task total time comparisons.

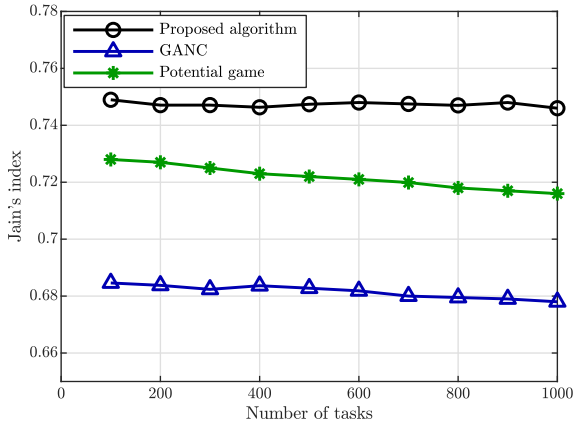

Figure 4.5: Jain's index comparisons.

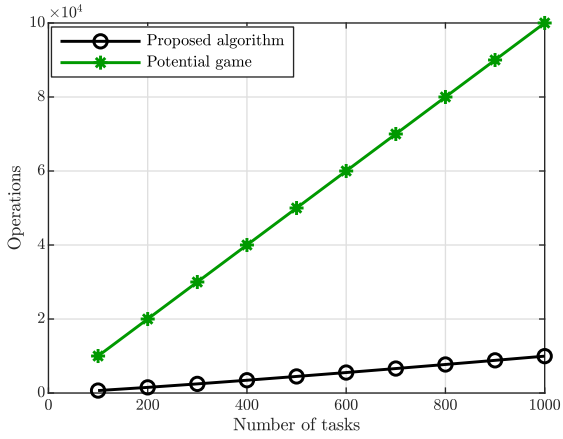

Figure 4.6: Time complexity comparisons.

EDs and FNs to get the mean value $[112, 135, 210]$. Specifically, the system conditions have been stressed considering as simulation parameters a number of FNs $n = 5$, $t = 15$ application types, a total number of tasks $m \in \{100, 1000\}$ each consisting of a uniformly distributed number of instructions within the interval $[10000, 50000]$. The spacial distribution of FNs and EDs within a circular area of radius 100 m has been modeled according to a Poisson Point Process (PPP). In order to test the validity of the proposed tasks offloading scheme in an actual environment, we have considered a x86 *Intel Sandy Bridge* as FNs architecture. More in depth, we have evaluated the performance of the proposed tasks offloading scheme by assuming that FNs may be equipped by a CPU selected with an equal probability among five possible Intel processor cores alternatives: the *Core i*7, *Core i*5, *Core i*3, *Pentium* and *Celeron*, with a CPU clock rate of 3.6GHz, 2.7GHz, 2.4GHz, 1.9GHz, 2.8GHz, respectively. Our performance analysis has been carried out by provided numerical results related to the *worst total task completion time* ($\bar{T}_M$). However, for the performance analysis completeness, we have considered also the evaluation of the *mean task total completion time* ($\bar{T}_T$), the *mean waiting time* ($\bar{T}_W$) and the *Jain's index* (I_J), defined as:

Definition 7. *Let $v \in \mathbb{R}_+^k$ be a given vector. The Jain's fairness index is*

defined as $I_J : \mathbb{R}_+^k \rightarrow \mathbb{R}_+$ *s.t.*

$$I_J(v) = \frac{(\sum_{l=1}^{k} v_l)^2}{k \sum_{l=1}^{k} v_l^2}, 0 \leq I_J \leq 1. \tag{4.11}$$

Note that the fairness increases the closer the I_J gets to 1. In order to highlight the better behavior of the proposed tasks offloading scheme, we have considered, for comparison purposes, a Greedy algorithm based on the nearest FN criterion (GANC), according to which each ED offloads its computation on the nearest FN[1], and a random approach (RAND) according to which each ED randomly offloads its intensive computation task to one FN $\in \mathcal{F}$. Furthermore, we consider also for comparison purposes the approach outlined in [220], named hereafter as potential game. In this case, differently from [220], we have considered that EDs can offload their tasks computation only to FNs with the goal of minimizing $\bar{T}_T$.

The results given in Figure 4.2 highlight the better performance of the proposed algorithm in terms of $\bar{T}_M$ in comparison with the GANC and the potential game methods. From this follows that our matching strategy allows to match more strict real time task constraints than the considered alternatives. Likewise, the better behavior of the proposed tasks offloading scheme is again evident in Figure 4.3 and Figure 4.4 in terms of $\bar{T}_W$ and $\bar{T}_T$, respectively. Moreover, from these figures, it is possible to note that the GANC scheme gives rise to a significant increase of $\bar{T}_W$ and $\bar{T}_T$ for medium to high values of the parameter m, while the proposed approach ensures better performance for all the considered m values. It is also evident in these figures that the proposed solution overcomes all the other offloading alternative methods.

Figure 4.5 shows the performance of the proposed tasks offloading scheme in comparison with the alternative approaches in terms of the Jain's index. This figure depicts the good behavior of the proposed scheme by guaranteeing a Jain's index value closer to 1 w.r.t. GANC and potential game alternatives, for all the considered tasks offloaded values. We stress that this result highlight that our algorithm provides a quite well-balanced total completion time for all the offloaded tasks, avoiding performance degradation due to workload peaks (i.e., congestion) on some FNs. However, it is also evident from our results that the potential game has performance slightly

[1]The spacial distribution of EDs and FNs modeled as a PPP favors the GANC criterion, avoiding significant imbalances as regards the offloading requests.

worst than the proposed approach. In this respect, in order to complete our analysis, we verify below that the proposed method clearly outperforms the potential game alternative (even in its low complexity implementation, named suboptimal potential game in [220]) in terms of time complexity and signaling overhead. We start our analysis by evaluating the computational complexity and signaling overhead of the proposed algorithm. Towards this end, we focus on the worst case occurring when each FN has a capacity equal to 1 and the tasks to be offloaded have the same preferences order. Indeed, under these assumptions, at the end of the first step, the number of allocated tasks result 1 out of m. Hence, under the assumed worst case hypothesis, the tasks allocation process is completed in exactly m steps, giving rise to an overall signaling overhead equal to

$$\sum_{s=0}^{m-1} m - s \;=\; \frac{m(m+1)}{2}, \tag{4.12}$$

Likewise, from [220], we have that for the suboptimal potential game the $\epsilon - NE$ equilibrium is reached with a signaling complexity given by:

$$\frac{m^2}{\epsilon}(m-1). \tag{4.13}$$

From above, it is straightforward to note that, in the worst case, our algorithm requires m steps to converge. The total time complexity is $\mathcal{O}(mlogm)$ while for the suboptimal potential game case it results equal to $\mathcal{O}(m/\epsilon)$ [220].

Figure 4.6 compares the time complexity of the proposed method with that one of the sub-optimum potential game, with $\epsilon = 0.01$, according to [220]. This figure highlights a significant gain for the proposed method, hence confirming the convenience in its use to perform tasks offloading in FC to support low latency IoT applications where it is also important to lower the computation complexity and signaling overhead.

4.1.8 Conclusion

This work has provided a distributed DAA based strategy to perform an effective offloading of computational intensive tasks in FC for IoT systems enabling low latency applications and services. In the proposed framework, a many-to-one matching game with externalities has been modeled ensuring a stable matching outcome, and complexity and signaling overhead analysis has been discussed. The performance of the proposed approach has been

evaluated in terms of worst total task completion time, mean task total completion time, main waiting time and Jain's index. Performance comparisons with a greedy (GANC), random (RAND) and a more recent proposed method, i.e., the potential game approach, based on the GT, have been presented in order to validate the good behavior of the proposed solution. Furthermore, we have also verified that the proposed MT based approach positively compares with the potential game alternative in terms of computational complexity and signaling overhead.

4.2 Virtual Functions Placement with Time Constraints in Fog Computing: a Matching Theory Perspective

4.2.1 Motivation

The large adoption of the virtualization and *infrastructure as a service* models has opened the doors towards heterogeneous service capabilities and resources pervasively distributed and interconnected close both to end users and physical objects [56]. AS consequence,FC can be viewed as a highly virtualized platform that provides processing, storage, and networking capabilities made available by *edge* nodes (e.g., mobile devices, routers, and micro data centers) to support the development of distributed applications. Devices endowed with sensing and actuation capabilities could be considered nodes that offer specialized services (e.g., acquisition of measurements from the environment in which they are deployed and, whenever possible or needed, perform control actions) [18, 56].

FC is expected to support services and applications that demand widely distributed deployments and real-time interactions. Especially in the area of the IoT, the availability of processing, network and storage resources close to sensors, actuators and end users can help in coping with low-latency application requirements, while also partially relieving the central Cloud from processing big amounts of data [56].

An application deployed on a Fog infrastructure can be conceived as a set of independent services which cooperate each other to realize the application goal, and typically interact in a *sequential* order, especially in IoT scenarios demanding for *sense-process-actuate* workflows [42, 56, 227]. These

services, ultimately composed of atomic services, are expected to be provisioned by software capabilities, i.e., programs, that can be executed on top of a virtualized resource infrastructure. We refer here to such software capabilities as Virtual Functions (VFs). It is worth clarifying that a VF may serve more than one service and, consequently, more than one application. However, the usage of a VF may differ from service to service, according to specific applications requirements, and, consequently, it requires different computation times and resource usage [56]. In this context, *orchestration* mechanisms should be put in place to achieve a cost-effective utilization of infrastructural resources for Quality of Service (QoS) aware deployment and management of application components (i.e., services) [56, 264]. A major problem is, thus, how to effectively distribute services to support composite applications provisioning on top of a Fog resource infrastructure.

4.2.2 Contributions

The contributions of paper [56] are here presented and discussed and, in first instance, can be identified with the two proposed VFs placement approaches in a fog domain. The considered solutions formulate a matching game with externalities, aiming at minimizing both the worst application completion time and the number of applications in outage, i.e., the number of applications with an overall completion time greater than a given deadline. The first proposed matching game is established between the VFs set and the FNs set by taking into account the ordered sequence of services (i.e., chain) requested by each application. Conversely, the second proposed method overlooks the applications service chain structure in formulating the VF placement problem, with the aim at lowering the computation complexity without loosing the performance. Moreover, the strategies aim at minimizing both the worst application overall completion time and the number of applications that suffer an overall completion time greater than their deadlines (*outage*). Furthermore, in order to complete our analysis, the stability of the reached matchings has been theoretically proved for both the proposed solutions. Finally, performance comparisons of the proposed MT approaches with different alternatives are provided to highlight the superior performance of the proposed methods. In order to confirm the good behavior of the proposed solutions, in comparison with different alternatives, i.e., greedy and random algorithms, and a modified version of the potential game presented in [220], performance results derived by extensive computer simulations are

presented. Finally, a theoretical proof of the stability of the proposed placement policies has been provided.

4.2.3 Related work

Resource management and allocation problems have been extensively investigated in the fields of Cloud Computing (CC) [164] and, more recently, of NFV [113].

Several approaches have been proposed for addressing the problem of optimizing the use of physical resources in Data Centers to achieve an effective placement of VMs onto physical hosts to account for conflicting requirements on performance and operational costs [164]. Wang et al. [255] address the problem of service composition in a data center network by formulating a multi-objective constraint optimization problem that aims at maximizing an overall QoS utility value, and minimizing network resource utilization. To this purpose, Anchor [275] is a MT based resource management architecture that performs VM allocation requests to servers, using both offline and online algorithms, allowing clients and operators to express a variety of distinct resource management policies.

Herrera and Botero [113] provide a state of the art of resource allocation strategy for the deployment of network services in VNF enabled network infrastructures. Several works proposed approaches for VNF placement and service chaining with the goal of minimizing costs, while fulfilling Service Level Agreements. Liberati et al. [154] propose a stochastic algorithm based on reinforcement learning, that maximizes an expected mapping reward in the long term. The reward function can be formulated to steer the controlled system towards the desired performance (e.g., costs minimization, load balancing, maximization of the acceptance rate). Pham et al. [199] address the problem of VNF placement for deploying service chains on virtual resource infrastructures, while minimizing both traffic and operational costs. They propose an approach combining a sampling-based Markov approximation technique with MT.

Resource management in a FC environment differs from such previous works on several aspects [42, 228]: *i)* FNs have limited resources *ii)* clusters of FNs are typically distributed on a suitable area, and *iii)* computation tasks are typically delay sensitive. Several works have focused on the efficient utilization of resources of the network infrastructure, typically including FC, as well as Cloud resources. Brogi and Forti [42] propose a model

for representing FC infrastructures and applications and propose a two-step algorithmic approach for finding eligible deployments for an application into a FC infrastructure. The proposed approach consists in a pre-processing phase aiming at reducing the search space and a backward search procedure that provides an eligible deployment. On the other hand, Foglets [217] is a programming model and software infrastructure for the deployment of applications on FC environments which embodies algorithms and protocol for the discovery and allocation of Fog resources for application deployment. Taneja and Davy [241] propose a network-aware module, mapping algorithm aiming at guaranteeing efficient resource utilization in distributed application deployment in a Cloud-FC environment. Both resource nodes and application modules are sorted according to the available capacity and requirements. Moreover, associations are performed whenever QoS constraints are satisfied. Zhang et al. [288] focus on a resource allocation problem in a FC environment consisting of a large number of FNs deployed by services providers at different locations to provide data services to subscribers. They propose a joint optimization approach that combines Stackelberg game and many-to-many matching to achieve an optimal and stable performance in the resource selection and allocation between the FNs, services providers and subscribers.

To the best of our knowledge, applications delay requirements have been only recently considered in the literature [34, 227, 228]. Specifically, Skarlat et al. propose in [227] a resource model for a FC architecture and also formulate a Fog service placement problem that maximizes utilization of FC resources, while taking into account application QoS constraints, especially deadlines on the execution time. The problem is formulated as an Integer Linear Program. These results have been successively extended in [228] by proposing a genetic-based algorithm heuristics. Finally, Bittencourt et al. compared in [34] the performance of three different resource allocation policies (i.e., concurrent, First Come-First Served, and the delay-priority strategies) in terms of efficient resource utilization and support to application QoS constraints. Differently from [228], [34] and [227], where an hierarchical three-layers Fog network[2] with the Cloud at the highest level of the network is considered, we focus here on a two-layers network, without considering the presence of the Cloud.

Furthermore, differently from [34], in which service processing is allowed

[2]Due to the different network architecture adopted in [228] and [227], their genetic and evolutionary approaches cannot be considered as proper benchmark methods for our approach.

just one hop away from the origin of the request, we consider the possibility to offload the service computation on FNs that are distant more than one hop from the origin of the request. Finally, as in [34, 227, 228], we associate at each application a time deadline for the application completion. However, differently from [227, 228], we do not introduce any prioritization mechanism for the applications, and we do not consider the presence of the Cloud in the network.

4.2.4 System Model

The problem consists in VFs placement in a network of FNs to accommodate the requirements of a batch of applications, so that the worst application time and the number of applications in outage are minimized.

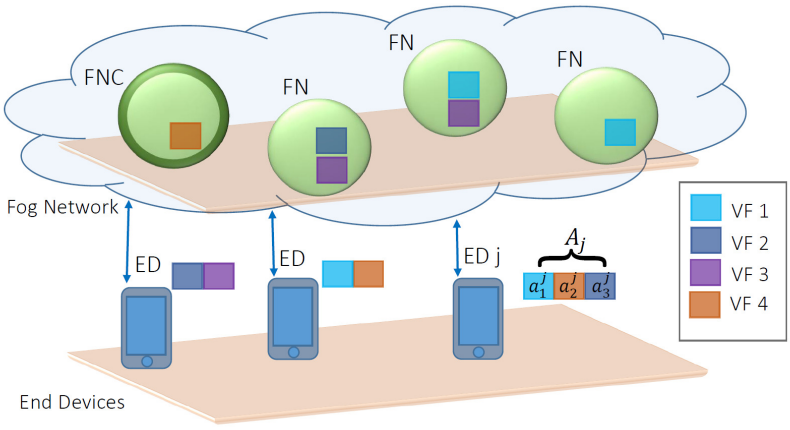

Figure 4.7: Reference Scenario. The EDs request to FNs the computation of their applications that are constituted by one or more services, each of which requires a VF.

The reference scenario, as depicted in Figure 4.7, is composed of a set $\mathcal{F}$ of n FNs. Furthermore, the presence of only one FN, among the n in $\mathcal{F}$, referred as FNs controller (FNC), able to acquire a full knowledge about the FNs capabilities, applications requirements and able to control all the FNs of the network, has been assumed. Likewise, supposing a number of applications equals to m, $\mathcal{A} = \{A_1, \ldots, A_m\}$ denotes the set of applications,

where each application A_j is associated to a suitable services chain, i.e., $A_j = \{a_1^j, \ldots, a_g^j\}$, for $j = 1, 2, ..., m$, where g is the number of services that compose A_j, and each service a_k^j, $k = 1, \ldots, g$ exploits a specific VF belonging to the set $\mathcal{V} = \{v_1, \ldots, v_p\}$, where p represents the number of VF types. Moreover, h_z, for $z = 1, 2, ..., p$, represents the occurrence frequency of the VF v_z in the set $\mathcal{A}$. Finally, a time constraint for each application, i.e., A_j, has been introduced. Such constraint means that A_j has to be completely executed within a given time deadline d_j.
Each $v_z \in \mathcal{V}$ can be loaded on a FN $f \in \mathcal{F}$ if it has sufficient computational resource blocks (CRBs) to satisfy the service demand. To this purpose, a VF v_z is associated with the amount r_z of requested CRBs. In addition, each FN $f \in \mathcal{F}$ is equipped with a central processing unit (CPU), whose capability is expressed in terms of number of CPU cycles per second c_f and r_f available CRBs, with $0 \leq r_f \leq r_{MAX}$, where r_{MAX} is the maximum number of available CRBs for each FN. As a consequence, the execution time of the k-th service of A_j requiring v_z, on the FN f results to be

$$t_{k,f,v_z}^j = \frac{s_{k,v_z}^j}{c_f}, \tag{4.14}$$

where $s_{k,z}^j$ is the number of cycles required to execute the service a_j using v_z. In order to evaluate the total time demanded by a_k^j to be accomplished on FN f, the possibility of experiencing the waiting time ω_f^j, i.e., the time needed to complete the execution of all the services, requesting a v_z, previously assigned to FN f, has to be taken into account. Furthermore,the communication cost of ED j in reaching the desired computation site, i.e., FN f, has been supposed given by the sum of the following two contributions

1. the communication cost due to link the ED j and its closest FN f^*, r_{j,f^*}, and given by the ratio between the size in bits of a_k^j and the available transmission rate of the link connecting ED j to FN f^*;
2. the communication cost associated to the intra-cluster connection (i.e., among FNs), $q_{f^*,f}$, given by the sum of the communication cost of the links that service a_k^j has to get across in order to reach FN f from FN f^*.

Hence, the total time $T_{k,f,z}^j$ spent by a_k^j at the FN f before being completed is

$$T_{k,f,v_z}^j = \left(t_{k,f,v_z}^j + \omega_f^j + r_{j,f^*} + q_{f^*,f} \right), \tag{4.15}$$

From (4.15), it follows that the overall completion time of application A_j results in

$$C_j = \sum_{k \in \{1,\dots,g\}} \sum_{f \in \mathcal{F}} T^j_{k,f,v_z} \phi_{k,f}; \tag{4.16}$$

where $\phi_{k,f}$ is a binary variable equals to 1 if service k is executed on FN f, or to 0 otherwise. In addition to this, we refer in our analysis to the forthcoming 5G network with a transmission rate of 1 Gbit/s to estimate communication costs [13].

4.2.5 Problem Formulation

The main objective of this study is the VFs placement aiming at minimizing both the worst overall application time and the number of applications in outage. In formal terms, the optimization problem can be defined as follows

$$\min_{\mathbf{\Phi}} \max_{j \in \mathcal{A}} C_j \quad and \quad \min |\mathcal{O}| \tag{4.17}$$

s.t.

$$\sum_{j \in \mathcal{A}} \delta_j = m, \tag{4.18}$$

$$0 \leq r_f \leq r_{MAX}, \quad \forall f \in \mathcal{F} \tag{4.19}$$

$$0 \leq \sum_{z \in \mathcal{V}} \tau_{f,z} \leq p, \quad \forall f \in \mathcal{F} \tag{4.20}$$

$$\text{if} \quad \phi_{k,f} = 1 \Rightarrow \tau_{f,z} = 1, \forall f \in \mathcal{F}, k = 1, \dots, g, z = 1, \dots, p. \tag{4.21}$$

where $\mathbf{\Phi} = \{\phi_{k,f}, \forall k \in A_j, \forall f \in \mathcal{F}\}$, $\mathcal{O}$ is the set of the unsatisfied applications defined as

$$\mathcal{O} = \{A_j \in \mathcal{A} | C_j > d_j, j = 1, \dots, m\}, \tag{4.22}$$

and $|\mathcal{O}|$ denotes the number of elements in $\mathcal{O}$. In (4.22), δ_j is equal to 1 if A_j has all its services accomplished, or to 0 otherwise. Hence, (4.18) expresses that all the applications belonging to $\mathcal{A}$ have to be completely executed, hence all their services chain has to be computed. Therefore, it implies that any service of each application has to be run on *at least one* FN. Furthermore, constraint (4.19) imposes a maximum number of available CRBs for each FN in $\mathcal{F}$. Finally, $\tau_{f,z}$ is a binary variable equals to 1 if the

FN f already contains v_z, and (4.20) means that each FN can contain at most one instance of a same v_z. Finally, (4.21) means that the execution of service k on FN f implies the presence of the VF required by k, i.e., z, on FN f. In order to handle the case of a number of different VFs greater than the available CRBs at each FN, while fulfilling constraint 4.18, a discipline of VFs replacement has been adopted. In other words, this consists in allowing loading VFs at runtime, i.e., during the application execution. More in detail, a temporal *penalty* due to the additional loading time required by the VFs runtime replacement has been introduced. The temporal penalties in which each adopted approach occurred during its execution have been taken into account in the performance evaluation.

4.2.6 Matching algorithm

In order to solve the problem (4.17)-(4.21) by resorting to matching theory, a matching procedure, named Chain Based Matching Algorithm (CBMA) based on the well known Gale-Shapley algorithm (GSA) [30, 84, 163, 211], has been proposed. In this case, the matching game takes into account the services order, i.e., the position (level) of the services within the chain associated to each application. At each run, the matching procedure considers the VFs requested by the services that occupy the same level in the respective services chains. Thus, the proposed procedure acts at each level of the services chains and the VFs belonging to a given level cannot be allocated until all the ones of the previous level have been matched. Let v_z be the VF requested by the service occupying the k-th level in the A_j application services chain. The most preferred FN by A_j, i.e., f_j^*, to allocate v_z has to satisfy the following condition:

$$f_j^* = \arg\min_f T_{k,f,v_z}^j. \tag{4.23}$$

Likewise, the FNs preferences lists are formed by favoring the allocations of VFs associated to applications with a closer deadline. It is straightforward to note that the preferences lists change during the matching game. Indeed, the allocation of a VF on a FN impacts on many factors, as the waiting time of that FN, its memory availability and so on, i.e., the presence of a matching influences the future assignments. This type of matching games is referred as matching games with externalities. Since the assumption that the preferences of each element of the matching do not depend on the choices of

other elements is not satisfied, the GSA in this class of matching games does not reach a stable matching and, more in depth, there not exist an approach that solves these games and achieves a stable matching. For such reason the study of stability is not trivial and it is extremely relevant.

Summarizing, the CBMA for each level of the applications services chains acts as

1. Each application A_j builds its own preferences list on the FNs set;
2. each application j, requiring a VF z_j, proposes to allocate z_j on its most preferred FN, according to (4.23);
3. each FN that receives more than one proposal decides which one to accept in accordance with its preference criterion;
4. each FN f that at the previous step has accepted a proposal, decrements its CRBs according to the resource request by the selected VF placement proposal;
5. repeat $1)-4)$ until all the services of a same level have been processed.
6. repeat $1)-5)$ until all the services in all the applications services chains have been processed.

In order to define a faster and less complex services placement solution, this paper proposes a novel matching theory based heuristic between the VFs in $\mathcal{V}$ and the FNs in $\mathcal{F}$, aiming at providing a sub optimal solution to the VFs placement problem (4.17).

In order to reduce the computation complexity of the service planning approach, without loosing performance, we propose below an alternative matching game, named Blind Matching Game (BMG), that removes the constraint to allocate the VFs to FNs according to the order they appear in the applications services chains. In this case, it is only taken into account the occurrences of any VF within the set $\mathcal{A}$. Moreover, the VFs preferences lists are formed with the aim at minimizing the allocation space waste at the FNs.

As previously introduced, a given VF, i.e., v_z, to be loaded on the FN f, needs a CRBs amount equal to r_z. Hence, we have in this case that the utility function of v_z in being loaded on FN f results in

$$U_z(f) = r_f - r_z, \forall f \in \mathcal{F}, z \in \mathcal{V}, \tag{4.24}$$

where r_f, representing the available CRBs on FN f, given by

$$r_f = r_{MAX} - \sum_{z \in \mathcal{V}} r_z \xi_{z,f}, \tag{4.25}$$

with $\xi_{z,f}$ a binary function equals to 1 if VF z is present on the FN f, or to 0, otherwise and, as previously introduced, r_{MAX} denoting the maximum number of available CRBs for each FN. In order to build the VFs preference lists, we sort the utility functions of each VF, according to (4.24), in an ascending order, hence favoring VFs allocation on FNs for which the remaining available CRBs are minimum.

The utility function definition of a FN f with respect to a given v_z, $U_f(z)$, has been based on the criterion of pursuing the allocation of v_z, to a suitable number of FNs in order to have the value of $U_f(z)$, defined as

$$U_f(z) = h_z \left(1 - \sum_{f \in \mathcal{F}} \frac{\tau_{f,z}}{n} \right), \tag{4.26}$$

as much as possible close to zero, i.e., all FNs have allocated the v_z. In this way, the aim is to favor the individual allocation of the VFs having the higher values of the occurrence frequency in the set $\mathcal{A}$, i.e., having the higher number of requests of performing individual services.

Hence, the FN f preferences list is sorted, according to (4.26), in an ascending order with respect to the values of $U_f(z)$ for all the VFs in $\mathcal{V}$, hence favoring allocation of the VF for which the resulting updated value of the utility function is minimized.

The proposed BMG algorithm is also based on a modified version of the GSA and consists of the following steps

1. Each VF in $\mathcal{V}$ builds its preference list according to (4.24);
2. each FN in $\mathcal{F}$ builds its preference list according to (4.26);
3. each VF in $\mathcal{V}$ proposes the match to its most preferred FN;
4. each FN in $\mathcal{F}$ that receives one or more proposals accepts the preferred one among the alternatives and rejects the others;
5. each FN f that at the previous step has accepted a proposal, decrements its CRBs according to the resource request by the selected VF placement proposal;

6. repeat $1) - 5)$ until at least one FN has sufficient CRBs.

It is important to note that the matching strategy solves the *a priori* placement problem but, due to the physical network limitations, after application of a placement algorithm, all the VFs may not result loaded on at least one FN. For such reason, we have introduced the VFs replacement at run-time, allowing all the algorithm proposed and analyzed in the performance evaluation to load a VF during its execution if it is not present on any FN. As previously introduced, the VF replacement introduces a time penalty that we have taken into account in the performance analysis.

More in depth, due to the presence of the externalities, the two players sets involved in the matching game, i.e., FNs and VFs, have to necessarily update their preference lists (PLs) in order to make decisions consistent with the current system conditions. Furthermore, aiming at maximizing the consistence between the decisions and the actual conditions, each FN simultaneously accept at most one VF, that is its preferred one among the proposals received. Hence, the final many-to-many matching is iteratively built during the algorithm.

4.2.7 Practical Consideration

As previously introduced, it is straightforward to note that the proposed heuristic BMG limits the need of context information. In particular, BMG does not require the knowledge of the number of cycles required by the application services, the FNs computation capabilities, the time deadline associated to each application, the waiting and execution time at each FN. Moreover, we verify below that the BMG algorithm requires a lower computational complexity, compared to the CBMA and other considered alternatives.

In performing our computation complexity analysis, we focus on a worst case scenario in which all the applications require all the possible VF types. More in depth, each of the m applications has to sort the n FNs in accordance with its preferences. This procedure exhibits a complexity equals to

$$O(m \cdot n \log n). \tag{4.27}$$

Then, such procedure is repeated a number of times equals to the maximum chain length among the applications in $\mathcal{A}$. Supposing all the A_j with the same length and requiring all the VFs types, we can conclude that the CBMA

exhibits a computational complexity given by

$$O(p \cdot m \cdot n \log n). \tag{4.28}$$

Likewise, in the BMG approach, each FN has to sort the VFs, hence the complexity results to be

$$O(n \cdot p \cdot \log p). \tag{4.29}$$

Considering that $p \approx n$ and $m >> p, n$, the complexity of the BMG approach is lower than the CBMA one.

4.2.8 Stability analysis

It is important to note that although in the classical matching game the stability condition is reached by applying the GSA, in the matching game with externalities to obtain a stable matching outcome is a very complex and challenging issue [30, 105, 211].
In order to discuss the stability convergence of the proposed algorithms, we refer in what follows to the *strictly-two-sided exchange-stability* (S2ES) stability definition derived from the one proposed in [35].

Definition 8. Let $\mathcal{M}$ be a matching function, and let $\mathcal{M}(v_z)$ be the FN matched with the VF v_z in the matching $\mathcal{M}$, $\mathcal{M}$ is a S2ES matching *if* there not exists a pair of VFs (v_x, v_y) s.t.:

1. $U_{v_x}(\mathcal{M}(v_y)) \leq U_{v_x}(\mathcal{M}(v_x))$ and
2. $U_{v_y}(\mathcal{M}(v_x)) \leq U_{v_y}(\mathcal{M}(v_y))$ and
3. $U_{\mathcal{M}(v_x)}(v_y) \leq U_{\mathcal{M}(v_x)}(v_x)$ and
4. $U_{\mathcal{M}(v_y)}(v_x) \leq U_{\mathcal{M}(v_y)}(v_y)$ and
5. $\exists \psi \in \{v_x, v_y, \}$ s.t. at least one of the conditions $1) - 2)$ is strictly verified and
6. $\exists \phi \in \{\mathcal{M}(v_x), \mathcal{M}(v_y)\}$ s.t. at least one of the conditions $3) - 4)$ is strictly verified.

The above definition means that at least one of among the VFs and at least one among the FNs involved in the swap have to get benefit from the swapping improving its condition.

In what follows, we verify that both the CBMA and the BMG approaches converge to a $S2ES$ matching outcome.
We start the stability analysis by focusing on the CBMA. In particular, we suppose that there exists a pair of VFs (v_x, v_y), belonging to applications x and y respectively, s.t. the conditions $1) - 2)$ of Definition 8 are verified. Supposing VFs v_x and v_y s.t. $\mathcal{M}(v_x) = f$ and $\mathcal{M}(v_y) = g$ respectively, from conditions $1) - 4)$, we have:

$$T^x_{k_x,g,v_x} \leq T^x_{k_x,f,v_x}, \tag{4.30}$$

$$T^y_{k_y,f,v_y} \leq T^y_{k_y,g,v_y}. \tag{4.31}$$

As detailed in Section 4.2.7, the completion time may or may not vary during time, so we cannot make any assumption about (4.30) and (4.31) and about the satisfaction of condition 5) of Definition 8. Instead, the FNs preferences lists are built taking into account the time deadlines of each application that proposes a given VF on such node. Let b_x and b_y be the time deadlines associated to applications x and y respectively. According to our assumptions that whether f prefers y to x, then $b_y \leq b_x$. In the same way we can show that whether g prefers x to y, we have $b_x \leq b_y$. Consequently, we necessarily have $b_x = b_y$ and none of f and g gains in switching. In conclusion the CBMA reaches a $S2ES$ matching outcome.
Likewise, we can conduct the stability analysis of the BMG approach. In order to discuss the stability, in this case we start considering the situation in which there exists a pair of VFs (v_x, v_y), with v_x allocated on FN f and v_y assigned to FN g, s.t. v_x prefers g and v_y prefers f to their actual partners. Hence, we have:

$$(r_{\mathcal{M}(v_y)} - r_{v_x}) \leq (r_{\mathcal{M}(v_x)} - r_{v_x}), \tag{4.32}$$

$$(r_{\mathcal{M}(v_x)} - r_{v_y}) \leq (r_{\mathcal{M}(v_y)} - r_{v_y}). \tag{4.33}$$

It is straightforward that (4.32) and (4.33) can be verified only if

$$r_{\mathcal{M}(v_x)} = r_{\mathcal{M}(v_y)}. \tag{4.34}$$

From (4.34) follows that v_x or v_y cannot improve their condition and that there not exists any ψ s.t. condition 5) of Definition 8 results true. Hence we can conclude that the reached matching is S2ES.

4.2.9 Simulation results

In carrying out the performance evaluation for the proposed approaches, a cluster formed by 8 heterogeneous FNs, distributed with a Poisson Point Process in a circular area with a radius equals to 40m, has been considered. Furthermore, each FN has a CPU frequency uniformly selected in the set $\{2.4, 3.6, 4.0\}$ GHz. Moreover, the following assumptions have been made:

- the FN computational capability uniformly selected, in terms of CRBs, in the set of values $\{100, 200, 300\}$;
- the number of different VFs, for each A_j service chain, uniformly distributed in the integer interval $[0, p]$, with $p = 10$, and the associated CRBs values uniformly selected in the range values $[35, 80]$;
- heavy tailed distribution of the occurrences of the VF types;
- 64 bits instructions for each requested service uniformly distributed in the range of integer values $[300, 500]$ with a mean cycles per instructions (CPI) equals to 8.
- the temporal penalty, associated to each FN, uniformly selected in $[0.5, 1.5]$ ms. Furthermore, each missed VF type in the FN network results in the addition, in the overall completion time, of the temporal penalty associated to the FN on which the VF is dynamically loaded.
- the temporal deadline, associated to each A_j, uniformly selected in $[0.2, 2.8]$ ms.

For the sake of simplicity, all the simulation parameters and their corresponding values have been reported in Table I, while Table II reports the system parameters details.

According to these assumptions, the system performance have been obtained by resorting to computer simulations in terms of *wasted CRBs*, calculated over all the algorithm iterations, *mean overall completion time per application* (T_C), calculated as the ratio between the sum of the individual application completion times and the total number of applications, *worst overall completion time per application* (T_W) and *outage probability* P_O. In order to highlight the advantages of the proposed solutions, the following methods have been considered as benchmark:

- The *random placement method* (RPM) that, for each VF type, selects the FN to allocate it with uniform probability;
- The *greedy placement method* (GPM), based on the selection of the FN, for each VF, with the highest amount of CRBs.
- The *weighted greedy placement method* (GGPM) a heuristic that, iteratively, places the highly requested VFs on the FN with the highest CPU frequency.
- The near optimal potential game (POT) proposed in [220], adapted to our scenario, for which the utility function of each VF is equal to the execution time (4.14).

In particular, with reference to the POT scheme, due the prohibitive temporal complexity (exponentially growing with n), we have considered its sub-optimal version [220], according to which the players involved in the game can deviate from the Nash equilibrium up to ϵ, with $\epsilon = 0.01$. This modification results in a strong reduction of the time complexity, that results equal to $\mathcal{O}(\frac{n}{\epsilon})$, as detailed in [220]. All the proposed RPM, GPM and GGPM start by allocating one instance of each VF. Then, if there are still available CRBs, iteratively repeat the allocation until all the FNs are full. Figure 4.8 provides comparisons in terms of *wasted CRBs* as a function of the number of different VFs to be allocated. This Figure points out the better behavior of the proposed methods with, in particular, the BMG scheme clearly outperforming all the other considered alternatives, hence achieving the best management of the computational resources. As direct consequence of the performance presented in Figure 4.8, it is evident to see that the proposed method guarantees a greater number of VFs types loaded on the FNs, in comparison with the other alternatives. Then, in general terms, it can be affirmed that the greater the number of VFs types on the FNs, the higher is the odds of finding the required VF in the network, avoiding to occur in the temporal penalties. Such insight has been confirmed by Figures 4.9-4.10 that show performance comparisons in terms of the mean overall completion time and worst overall completion time per application, respectively, as a function of the number of applications to be performed. In particular, in terms of mean overall completion time, in Figure 4.9, the CBMA and BMG achieve better results considering the other alternatives, with the CBMA overcoming the BMG. The same behavior is evident in Figure 4.10, where

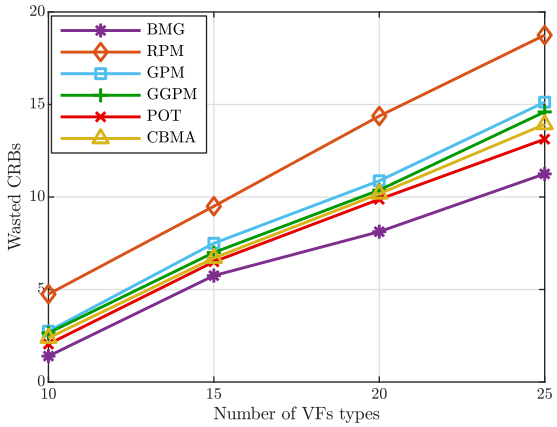

Figure 4.8: Wasted CRBs, supposing $n = 8$ FNs, and $m = 100$ applications requests.

the worst T_W is given as a function of the number of applications to be considered. This is due to the intrinsic trend of both the MT based approaches to deploy on the FNs the most requested VFs. In fact, in both approaches the presence of more than one instances of the highly requested VFs in the network increments the chances to properly select the FN that brings more advantages in terms of overall completion time. As evident in Figure 4.11, increasing the VF types, the system performance get worse. However, also in this case, the proposed approaches reduce the worst overall completion time, by ensuring higher responsiveness despite in presence of heavy applications demand. By guaranteeing better performance in Figure 4.9 and Figure 4.10, the proposed approaches provide a better compliance with the applications deadlines, that implies lower values of outage probability, hence, of the number of applications which do not match their temporal deadlines as depicted in Figure 4.12. From Figures 4.9-4.12, it is evident that all the considered methods are overcome by the CBMA. However, on the basis of the resulting temporal complexity presented in Sec. 4.2.6, the BMG alternative represents the best tradeoff between high performance and low temporal complexity. Furthermore, in terms of convergence time, the BMG and CBMA complete their computation in 0.023926 and 0.051621 s respectively, as reported in Table II.

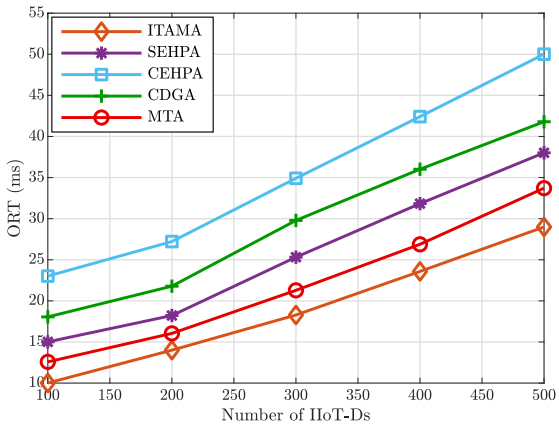

Figure 4.9: Average application completion time, for $n = 8$ FNs, and $p = 10$ VFs types.

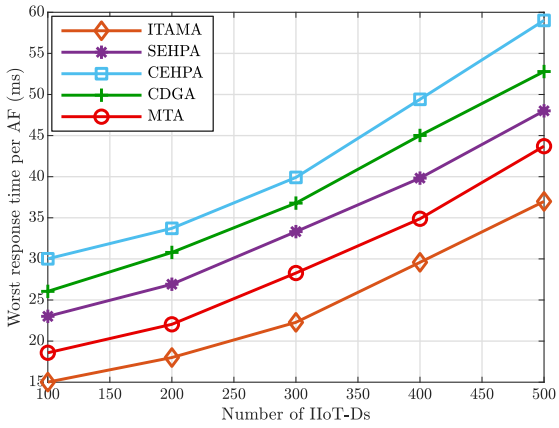

Figure 4.10: Worst application completion time, for $n = 8$ FNs, and $p = 10$ VFs types.

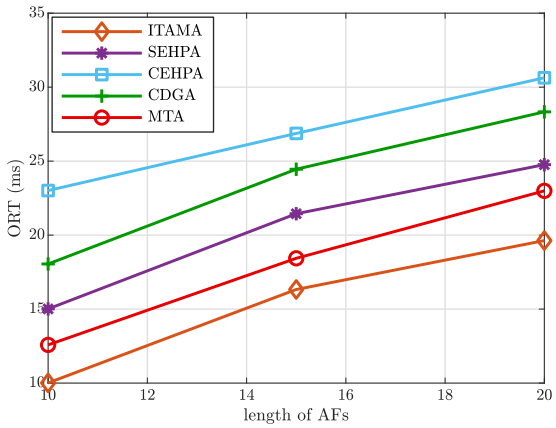

Figure 4.11: Worst application completion time, for $n = 8$ FNs, by varying the maximum size of the service chain, i.e., the number of VFs types.

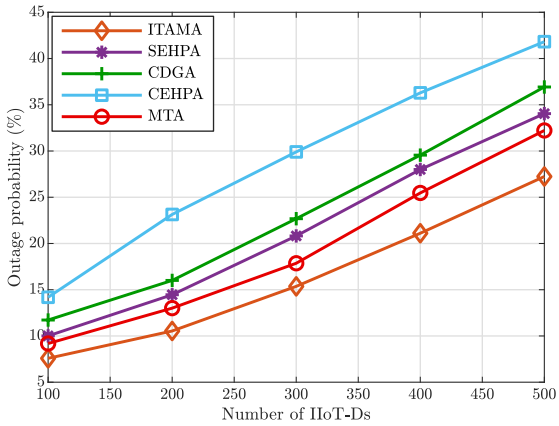

Figure 4.12: Outage probability comparison for $n = 8$ FNs, and $p = 10$ VFs types.

Table 4.1: Simulation Parameters

Parameter	value
Network radius	40 m
CPU frequency	{2.4, 3.6, 4.0} GHz
FN CRBs	{100, 200, 300}
Number of VFs types	10
VF CRBs	[35,80]
Bits per instruction	64
Number of instructions	[300, 500]
CPI	8
Time penalty	[0.5,1.5] ms
Time deadline	[0.2, 2.8] ms

4.2.10 Conclusions

The proposed VFs placement schemes, namely CBMA and BMG, are based on the application of the matching theory principles, with the aim of minimizing both the worst overall application time and the applications outage probability. Performance comparisons with different services placement methods, i.e., a random scheme, two greedy disciplines and a game theory based approach have been presented to highlight the advantages of the proposed schemes. In particular, it can be concluded that the CBMA approach achieves the best performance with a higher computation complexity and a more heavy signaling overhead, in comparison with the BMG solution.

Finally, for both the CBMA and BMG matching approaches the matching stability according to the strictly-two-sided exchange-stability definition has been theoretically proved.

Table 4.2: System Parameter

Operating System	Ubuntu 16.04
Programming Platform	Matlab
CPU	Intel i7-3770 3.40Ghz
RAM	16 GB
Convergence BMG	0.023926 s
Convergence CBMA	0.051621 s

4.3 A Matching Game with Discard Policy for Virtual Machines Placement in Hybrid Cloud-Edge Architecture for Industrial IoT Systems

4.3.1 Motivation

Nowadays, the ever increasing diffusion of novel IoT applications has triggered efforts, from both academia and industry, in searching for new solutions and strategies to enable novel applications in different scenarios, such as smart cities, intelligent transportation, pollution monitoring etc. Within this context, also due to the progresses in the Industrial Wireless Networks (IWNs) [152], a novel IoT subset, named Industrial IoT (IIoT), has risen in reference to the machine-to-machine domain and industrial communication technologies with automation applications [127, 226, 268, 278]. The large attention that the IIoT research branch has recently gained is due to the improvement on sustainability and safety deriving from the usage of wireless technologies in industrial environments, in which self-organizing systems and context-aware devices interact to provide flexible, reliable and adaptive industrial environments, with high quality services [252]. Nevertheless, there exist several major challenges that have to be addressed to make IIoT a reality. Among all the difficulties that have to be dealt with in order to properly realize IIoT systems, a crucial point is that the IIoT devices are typically computationally constrained, i.e., equipped with scarce processors, and possess low battery capacity which implies finite lifetime

and energy [44, 252, 287]. Furthermore, the IIoT devices generally support mission and safety-critical applications with strict timing or end-to-end constraints on task executions [78, 226]. Generally speaking, it is important to note that network model architecture severely impacts the accomplishments of IIoT-devices temporal deadlines [12, 49, 51, 143, 150, 226, 237, 242]. In order to meet the constraints and specifications imposed by the novel IIoT applications, many papers in literature suggest to adopt networks structures that move computation close to the IIoT-devices, and capable of adapting to different flows requirements, such as different time delay or quality of service constraints. In particular, it is stated in [49] that a completely flat approach is not suitable for IIoT environments, while a hierarchical network structure with decentralized computation, i.e., a fog computing structure, leads to significant improvements on both the delay and computation aspects [49, 51, 143].

Recently, the Edge Computing (EC) has emerged as a promising solution to face the strict latency constraints of IIoT applications [162, 240], moving computation and storage to the edges of the network, nearby the IIoT-devices [162], bypassing the drawbacks of the traditional cloud computing architecture [162, 240], mainly in terms of severs network congestion and high response delay. Furthermore, in order to facilitate the fulfillment of the delay-sensitive applications, the deployment of virtual machine replica copies (VRCs) of applications on the network edges gained momentum [292].

4.3.2 Contributions

This work focuses on a hybrid cloud-EC IIoT-devices architecture, and proposes a matching theory based VRCs allocation on the EC servers (ECSs), aiming at jointly minimizing both the mean IIoT-devices system response time, referred in the following as overall response time (ORT), and the number of IIoT-devices that suffer a response time greater than their time deadline.

In practical terms, the main contributions of this study can be summarized as follows:

- The paper addresses the problem of the VRCs placement in a hybrid cloud-edge computing architecture, in which each EC server can host more than one VRC. The presence of more than one VRC is a crucial point in order to increase the system responsiveness and provide low

service latency. In this respect, a joint optimization problem formulation has been proposed, aiming at minimizing both the ORT and the number of IIoT-devices that miss their deadline, which is a very important metrics in intensive and time critical applications;

- A matching game with externalities between the set of the ECSs and the applications requested by IIoT-devices has been formulated to find a sub-optimal solution to our problem. Different from the matching based approaches available from the existing literature, the proposed matching placement strategy resorts to a *discard policy* consisting in canceling the requests which missed their deadline, in favor of requests with the associated deadline not yet expired. The proposed algorithm is able to manage IIoT-devices requests formed by an arbitrary number of applications, differently from previous papers presented in literature such as [292];

- Despite in the matching games with externalities stability is not a trivial issue, stability of the outcome matching has been discussed and proved;

- Performance comparisons with the SEHPA and another heuristic proposed in [292], the matching strategy presented in [17], as well as a greedy algorithm are provided to validate the e good performance of the proposed framework.

4.3.3 Releted Literature

As regards the EC and FC research topic, and specifically the one about offloading, there exist many papers which deal with this problem. In particular, authors in [17] proposes a matching game with externalities among IoT devices and the fog nodes (FNs) to jointly solve the problem of the fog selection and the network latency minimization. The matching theory then has been exploited also in papers [291], [108], and [92]. Paper [291] addresses the problem of the resource allocation for the device-to-device communications in the cellular networks, by modeling the problem as a many-to-many matching game with externalities, aiming at maximizing the sum rate subjected to the user equipment quality of service (QoS) constraints. Differently, paper [108] applies the matching game with externalities principles to the average delay packets minimization in pairing mobile stations

with access points, within an offloading scenario, by also taking into account the monetary aspect in performing offloading and the energy consumption perspective. The offloading has been addressed also in paper [92], in which a mobile EC (MEC) scenario has been considered, and a task assignment procedure based on MEC servers computation capabilities, channel conditions, and time delay constraints have been taken into account. In fact, in paper [92], aims at minimizing the overall system energy consumption, by considering the heterogeneous QoS deadlines associated to each device requiring task computation.

The virtual machines (VMs) placement and the workload assignment problems are studied in paper [259], considering a mobile EC scenario, in order to minimize the hardware resources consumption for deploying VMs, taking into account heterogeneous applications latency requirements. The problem addressed in [208] concerns the minimization of the service delay, by considering both the processing delay and the transmission delay. The paper proposes a method that provides the processing time reduction through the virtual machine migration, while it applies a power control mechanism to improve the transmission delay. Furthermore, paper [292] addresses the problem of the optimal VM replica placement in a hybrid Cloud-EC scenario, aiming at minimizing the average system response time and service provision cost, given different applications types and ECSs with limited capacity.

Differently, the introduction of novel network architecture paradigms, such as FC or EC, on the IIoT scenario is still at the beginning [143].

A fog architecture in the IIoT-devices network is considered in paper [51] that aims at minimizing the network delay over time w.r.t. energy consumption constraints. In order to address this problem, authors in [51] propose a distributed approach in which they apply both the Lyapunov optimization and the Gibbs sampling. Paper [49] shows the higher performance of a hierarchical fog approach compared to the classical flat design and proposes a queue model system analysis, introducing two priority classes to schedule the IIoT data. Finally, a network architecture combining the centralized software defined network approach and the EC paradigm is proposed in paper [150]. The authors in [150] consider IIoT flows belonging to two different delay constraints, and an adaptive optimal transmission routing is proposed, by exploiting the path difference degree.

This paper considers a combined edge-cloud architecture and it aims at jointly minimizing both the ORT and the number of IIoT-devices which do

not terminate computation before their time deadline. Different from authors in [292], this paper considers as performance metric the number of computations that do not terminate before their associated deadline. It is important to note that the latter represents a significant performance index, especially when time critical application are handled. Furthermore, papers [292] and [143] consider applications requesting only one function, while this paper models the IIoT-devices workflows as an ordered sequence of functions that, to meet the assigned deadline, have to be completed before the expiration of the corresponding time constraint. More in detail, the proposed VRCs deployment strategy realizes a matching game with externalities between the ECSs and the IIoT-devices application requests, represented by chains of functions. In particular, the paper provides a discussion and proof about the stability on the outcome matching, that is not a trivial issue in this class of matching games. It is also important to highlight that, although authors in [17] design a matching algorithm that reaches a stable matching outcome by pairing and decoupling together IoT devices and FNs, until the stability is reached, our algorithm has been developed by pursuing the complementary approach, in which the pairs, once formed, cannot be broken. In addition, another main improvement introduced by our paper in comparison with paper [17, 92, 108, 291], is the *discard policy* characterizing our matching algorithm, for which the applications with the corresponding time deadline not yet expired are promoted to avoid their timeout missing, by sacrificing the tide of the applications which have already suffered the expiration of their deadline. Finally, performance comparisons with two VRCs deployment algorithms proposed in [292], and a greedy approach has been provided, in order to highlight the better behavior of the proposed solution.

4.3.4 System Model

As an IIoT practical scenario, this paper focuses on the automation/control applications domain, considering the IIoT-devices workflows formed by both delay sensitive applications (i.e., closed-loop or interlocking processes) and delay insensitive applications as those related to monitoring/supervision activities [226], characterized by a less stringent time deadline on their execution.

In order to formalize the considered Cloud-Edge architecture scenario, we consider a set of IIoT-devices $E = \{j, j = 1, ..., |E|\}$, with fixed position according to [226], each of which requires an application flow (AF) w_j, $j =$

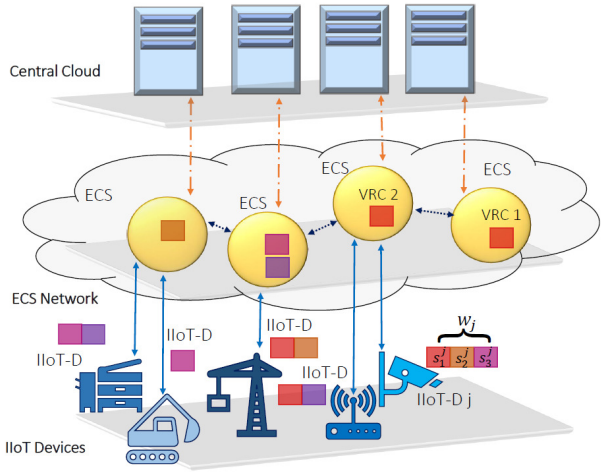

Figure 4.13: Reference Scenario: the IIoT-devices request to their nearby ECS the computation of their AFs. The ECS compute on itself, or on a nearby ECS, or on the remote Cloud.

$1, ..., |w_j|$. Each AF w_j, for its whole execution, requires an ordered sequence of services requiring functions, without repetition, generally located on the remote Cloud. Hence, each AF w_j is given by $w_j = \{s_k^j, k = 1, \ldots, |w_j|\}$, where each service s_k^j requests a function belonging to the set of the whole functions S. Equivalently, we can say that the maximum length of w_j is the number of function in S. Let F be the set of Edge Computing Servers (ECSs) $F = \{i, i = 1, ..., |F|\}$, heterogeneous in processing capability and storage, on which the Cloud, situated in the remote area of the network, can create VRCs of the applications in S. Supposing the referred network ideally divided into different non-overlapped areas, each ECS represents the local applications server for the surrounding network area, referred hereafter as its service area. In other words the service area can be viewed as the geographical area within which any ED can be reliably linked to the appropriate ECS. As previously stated, each ECS i in F has a limitation, in terms of storage resource blocks (SRBs) and referred hereafter as d_i, $0 < d_i \leq d_{MAX}$, where d_{MAX} is the maximum number of available SRBs for each ECS, that limits the number of VRCs that it can contain. In this regards, each VRC requires a given amount of SRBs, depending on the type of corresponding function

in S and, furthermore, each ECS can hold more than one VRC of of more than one function of S. Denoting with n_i the number of IIoT-devices that belong to the i-th network area, we have $\sum_{i \in F} n_i = |E|$. As introduced at the beginning of this section, the E set is formed by computational intensive applications, with different requirements on their timeout execution. In this regards, a time deadline a_j, $j = \{1, ..., |E|\}$, associated to the execution of each AF has been introduced, where A represents the set containing the AF deadlines associated to each IIoT-device in E.

Furthermore, the ECSs network has been modeled as a graph $\mathcal{F} = (F, H)$, where the set of vertices F is the set of the ECSs, and H represents the set of links among the ECSs. In this respect, we suppose that there exists always a path between two ECSs in the network. Finally, we assume that each ECS contains a routing table in which, for each pair of ECSs, the shortest path between those ECSs is stored.

Computation Time Model

In general, being the ECSs heterogeneous in terms of computation capabilities, the execution time of the service s_k^j mainly depends on which ECS hosts the computation, as well as the size of s_k^j. In particular, considering the computation of s_k^j on ECS i, as discussed later, its execution time depends on a range of factors among which the central processing unit (CPU) frequency of the host ECS i, the number of parallel computation ϕ_i that ECS supports, and the length of s_k^j itself. Furthermore, given the IIoT device j demanding for the computation of its AF $w_j = \{s_k^j, k = 1, ..., |w_j|\}$, denoting with c_k^j the number of cycles required to execute the k-th service, and with v_i the CPU cycles per time unit of ECS i, we have that the execution time of service k on the ECS i is equal to

$$\eta_{k,i}^j = \frac{c_k^j}{v_i}. \tag{4.35}$$

An additional term that has to be taken into account is the waiting time q_i on the ECS i, due to the time needed to complete the execution of all the services previously assigned to ECS i. Therefore, the total time spent by a function request k on the ECS i results to be:

$$\tau_{k,i}^j = \eta_{k,i}^j + q_i^j, \tag{4.36}$$

defining q_i^j as the queuing time[3] experienced by IIoT-device j by computing s_k^j on ECS i. Likewise, we obtain the execution time of s_k^j on the Cloud, i.e., $\eta_{k,C}^j$, by considering in (4.35) the CPU cycles per time unit of the Cloud given by v_C. Furthermore, let q_C^k be the waiting time on the Cloud, then it follows that the time spent by s_k^j on the Cloud is

$$\tau_{k,C}^j = \eta_{k,C}^j + q_C^k. \tag{4.37}$$

Transmission Delay Model

In order to fully characterize the response time for a generic function request s_k^j,$j = 1, \ldots, |E|$, $k = 1, \ldots, |w_j|$, we need to consider the network delay. In this respect, we suppose that each ECS i, in first instance, receives all the computation requests from the IIoT-devices within its service area and then, each s_k^j, will be eventually rerouted (if needed) on another ECS of the EC network, or towards the Cloud.

To simplify our analysis we have assumed :

- network delay is represented by the time required to information transmission, considering negligible the access and the set up delay;
- a fixed data rate for the wireless transmission delay between IIoT-devices and their local ECS;
- a negligible high velocity data rate for the links connecting two ECSs.
- a negligible high velocity data rate for the links connecting the ECSs to the Cloud[4].

As direct consequence, we have the total time required to execute all the AF w_j, originated within the service area of ECS i, is given by

$$\psi_{w_j} = \sum_{k=1}^{|w_j|} \sum_{\delta \in F} (\tau_{k,i}^j + \chi_k) x_{k,\delta} + (1 - x_{k,\delta})(\tau_{k,C}^j + \chi_k), \tag{4.38}$$

where χ_k is the transmission time of s_k^j, given by the ratio between the size in bits of s_k^j and the fixed data rate of the link between the IIoT-device

[3]The queuing time on the ECS is also strictly related to the ECS CPU frequency and to the number of parallel computation ϕ_i that ECS is able to perform.

[4]It is important to note here that the IIoT-devices, in order to reach the Cloud, have to perform the connection towards their local ECS.

requiring the computation of w_j and its local ECS, Finally, $x_{k,\delta}$ is a binary variable equals to 1 when the k-th service is executed on ECS δ, 0 otherwise. Hence, the mean overall response time (DORT) for each ECS domain i is given by

$$T_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \psi_{w_j}; \tag{4.39}$$

Finally, considering only the k-th level of w_j, we have that its completion time is equal to

$$\psi_{w_j}^k = \sum_{\delta \in F} (\tau_{k,i}^j + \chi_k) x_{k,i}^j + (1 - x_{k,i}^j)(\tau_{k,C}^j + \chi_k). \tag{4.40}$$

4.3.5 Problem Formulation

Differently from [292] where the main goal was the minimization of the ORT, this paper considers a joint optimization of both the ORT and the number of AFs that suffer a response time larger than their own time deadline. This is due to the fact that we consider intensive and time-critical applications and, in this respect, the mean time value considered in [292] as performance metrics, cannot provide a good estimation of the behavior of an algorithm in a context in which there are stringent deadlines. For this purpose, we introduce the set $\mathcal{U}$ representing the unsatisfied AFs,defined as

$$\mathcal{U} = \{w_j \in \mathcal{W} | \psi_{w_j} > a_j, j = 1, \ldots, |E|\} \tag{4.41}$$

In formal terms the optimization problem can be formulated as follows

$$\min \frac{1}{|F|} \sum_{i \in F} T_i \quad and \quad \min |\mathcal{U}| \tag{4.42}$$

s.t.

$$0 \leq d_i \leq d_{MAX}, \forall i \in F, \tag{4.43}$$

where constraint (4.43) expresses that each ECS i has limitation about the SRBs. Problem (4.42)-(4.43), as that proposed in [292], is generally NP-hard, making it necessary to resort to sub-optimal approaches. In this respect, we apply here the MT in order to find a sub-optimal fast solution.

4.3.6 Matching Game Formulation

The formulation of a matching game between the F and the S sets has been proposed to define relations reciprocally advantageous for all the agents in F and S [30], based on their own individual preferences. Below, first of all, we introduce the formal definition of matching [212], contextualized to our problem:

Definition 9. Let F and S be two sets of players. A *matching game* is the pair (F, S) on which are defined two preference relations $\succ_i, \succ_z$ allowing each player, i.e., a ECS $i \in F$, to express preferences over the opposite players, i.e., applications $z \in S$, and vice versa.

Let $P_z(\cdot)$ and $Q_i(\cdot)$ be the utility functions of application z and ECS i, respectively. We can say that $z_1 \succ_i z_2$ means that ECS i prefers to host application z_1 to z_2, hence $Q_i(z_1) > Q_i(z_2)$.

It is important to note that, in the proposed problem, the preferences list of each player is dependent on the others players' preferences, i.e., an existing matching influences the preferences of the unmatched elements, and a partner suitable at the start of the game cannot be the best choice during the evolution of the same game [30]. As a consequence, the preferences list of each involved participant, has to be updated after each algorithm assignment.

Applications preference list

Given an AF w_j, let k be the level of the AF, the application z required by s_k^j prefers the ECS that minimizes its response time, hence the ECS i such that

$$P_z(i) = \frac{1}{\psi_{w_j}^k} \tag{4.44}$$

is maximum. Hence, each application z sorts the ECSs in descending order, according to (4.44).

ECSs preference list

Each ECS preference list is built considering the time deadline associated to each w_j, whose s_k^j, requiring z, belongs. The preference list consists of the

applications sorted in descending order in accordance with

$$Q_i(z) = \frac{1}{a_j} \tag{4.45}$$

aiming at maximizing $Q_i(z)$.
The proposed Individual Time Aware Matching Algorithm (ITAMA) is a modified version of the Gale-Shapley one (GSA) [30, 84, 211]. The ITAMA is iteratively run on each level of the AFs chain. More in detail, during the k-th run, the matching procedure considers the functions requested by the k-th service request of each AF. All the ECSs are subjected to storage limitation, since to host an application implies storage resources utilization. Hence, when a ECS cannot host anything else, the ECS is not considered to be a possible host for the computations. Furthermore, a *discard policy* (DP) has been introduced in order to limit the number of unsatisfied AFs, i.e., the number of AFs belonging to $\mathcal{U}$. The DP consists of dismissing the services of the AFs that missed their deadline from the ECSs where they are allocated. The idea behind the DP is to offload these services towards the Cloud, avoiding to cause additional unsatisfied AFs. Furthermore, services requiring computation are actually rerouted toward the Cloud if their response time does not worsen more than 200 ms computing on Cloud, in order to avoid worst response times too high.
Summarizing, the ITAMA for each level k of the AFs chains acts as follows:

1. each function z, belonging to S and required by s_k^j, builds its own preferences list on the ECSs set;

2. each function z, proposes to allocate an own VRC on its most preferred ECS, according to (4.44);

3. each ECS that receives more than one proposal decides which one to accept in accordance with its preference criterion 4.45;

4. each ECS checks the deadlines of the associated to the AF to which s_k^j belongs to, i.e., w_j. If there are some AFs that miss their deadlines, the DP starts.

5. repeat $1) - 5)$ until all the services in all the applications services chains have been processed.

4.3.7 Complexity Analysis

In order to perform the computational complexity analysis of the proposed strategy, we focus on a worst case scenario, in which the following assumptions have been taken into account:

1. each AF requests all the functions in $|S|$;
2. at each level of the AFs, all the functions have the same favorite ECS, hence, all the applications send their proposal to the same ECS.

As it is evident, the assumptions introduced above imply a considerable worst case scenario.

The first steps of the proposed placement strategy consists of building the preference lists of both the parts involved in the matching game, i.e., the ECSs and the functions, respectively. In order to construct its own preference list, each function needs to sort the elements belonging to the F set, according to (4.45). The computational complexity of this procedure, for each function, is clearly

$$O(|F| \log |F|), \tag{4.46}$$

that extended to all the S set is

$$O(|S| \cdot |F| \log |F|). \tag{4.47}$$

Similarly, the computation complexity required by the creation of the ECSs preference lists can be obtained in the same way. Hence, the associated computational complexity results given by

$$O(|F| \cdot |S| \log |S|). \tag{4.48}$$

From (4.47) and (4.48) follows that the preference lists building processes are the heaviest parts, in terms of temporal complexity, of the proposed placement algorithm. Hence, the overall computational complexity requested for the creation of the preference lists is

$$O(|S| \cdot |F| \log |F|) + O(|F| \cdot |S| \log |S|). \tag{4.49}$$

Generally speaking, the preference lists building procedure is repeated a number of times equals to the length of the longest AF among the elements of W. As a consequence of assumption 2), previously introduced, all the AFs request all the functions, hence, the length of each AF is the same for all the AFs, and it is equal to $|S|$. Indeed, the total computational complexity is

$$O(|S| \cdot |F| \cdot |S| \log |S|). \tag{4.50}$$

4.3.8 Stability Analysis

Although in the classical matching problem the GSA reaches a stable matching, for matching game with externalities, as that one proposed here, there not exists any matching algorithm that definitely achieves outcome matching which is definitely stable. Since in the matching games with externalities the preferences change during the evolution of the game, to obtain a final stable matching is a very complex and challenging issue [30, 211]
In order to discuss the stability of the ITAMA, we introduce the *strictly-two-sided exchange-stability* (S2ES) stability definition, as a modification of the one proposed in [35].

Definition 10. Let $\mathcal{M}$ be a matching function, and let $\mathcal{M}(z)$ be the ECS matched with the function z in the matching $\mathcal{M}$, $\mathcal{M}$ is a S2ES matching *if* there not exists a pair of functions (z_1, z_2) s.t.:

1. $P_{z_1}(\mathcal{M}(z_2)) \geq P_{z_1}(\mathcal{M}(z_1))$ and
2. $P_{z_2}(\mathcal{M}(z_1)) \geq P_{z_2}(\mathcal{M}(z_2))$ and
3. $Q_{\mathcal{M}(z_1)}(z_2) \geq Q_{\mathcal{M}(z_1)}(z_1)$ and
4. $Q_{\mathcal{M}(z_2)}(z_1) \geq Q_{\mathcal{M}(z_2)}(z_2)$ and
5. $\exists \psi \in \{z_1, z_2\}$ s.t. at least one of the conditions $1)-2)$ is strictly verified and
6. $\exists \phi \in \{\mathcal{M}(z_1), \mathcal{M}(z_2, \}$ s.t. at least one of the conditions $3) - 4)$ is strictly verified.

In other words, Definition 10 means that a swap is allowed if it implies an improvement to almost one between the players involved, and the remaining elements do not get worse. In order to prove the ITAMA stability, we suppose that there exists a pair of functions (z_1, z_2), belonging to AFs w_{j_1} and w_{j_2} respectively, s.t. the conditions $1) - 2)$ of Definition 10 are verified. Let z_1 and z_2 be s.t. $\mathcal{M}(z_1) = i_1$ and $\mathcal{M}(z_2) = i_2$ respectively, from conditions $1) - 4)$, we have:

$$P_{z_1}(i_1) \leq P_{z_1}(i_2), \tag{4.51}$$

$$P_{z_2}(i_2) \leq P_{z_2}(i_2). \tag{4.52}$$

Since the ECSs are heterogeneous in terms of CPU frequency, number of parallel computations, and the algorithm allows to discard services, the completion time may or may not vary during time, and we cannot make any assumption about (4.51) and (4.52), hence about the satisfaction of condition 5) of Definition 10. Instead, the ECSs preferences lists are built considering the time deadlines of each corresponding AF. Supposing a_1 and a_2 the time deadlines associated to w_{j_1} and w_{j_2} , if z_1 prefers i_2 to i_1, then $a_2 \leq a_1$. Likewise, if z_2 prefers i_1 to i_2, we have $a_1 \leq a_1$. Then, we necessarily have $a_1 = a_2$ and neither i_1 nor i_2 gains in switching. Therefore, the ITAMA is S2ES stable.

4.3.9 Numerical Results

In order to highlight the good behavior of the proposed matching scheme, we have resorted to extensive computer simulations. For each plot point, we considered 3×10^4 independent simulation rounds with all the parameters selected as detailed below. Both the IIoT-devices and ECSs have been assumed distributed according to a Poisson Point Process in a circular area with a radius equals to 800 m. Each ECS has a CPU on board with a frequency uniformly selected in the set $\{2.4, 3.6, 4.0\}$ GHz. Furthermore, we have assumed that each ECS has a storage capability, expressed in terms of SRBs, uniformly selected in $[500, 1000]$, and that each VRC of a given function requires an available storage space on the ECS selected with a uniform distribution in $[100, 200]$ SRBs. The number of different functions has been assumed equals to 20. Consequently, the AFs length is an integer uniformly distributed between 1 and 10, and each AF cannot require more than once the same function. In addition to this, the AFs deadlines have been set considering a uniform distribution in the system of both the sensitive and insensitive applications. As regards the former, the associated deadlines have been uniformly selected in $[10, 100]$ms, as specified in [226], while to the latter we have uniformly assigned deadlines in $[150, 500]$ms. Furthermore, a heavy tailed distribution has been used to model the occurrences of the function types in the AF chains. The connection link between the IIoT-devices and the local ECS has been set equal to 100 Mbit/s, while the data rate between two ECSs and between ECS and Cloud equals to 1 Gbit/s. Finally, each service consists of a number of 64 bits instructions uniformly distributed in the range of integer values $[100, 3500]$ with a mean cycles per instructions (CPI) equals to 8, and the system performance has been mea-

sured considering a number of AFs variable within the range $[100, 500]$, with a number of ECS equals to 8.

Hereinafter, the system performance is provided in terms of

- ORT, expressed as the sum, for all the ECS, of its DORT as defined in (4.39), and divided for the number of ECSs in the system;
- *worst response time per AF* (WRT), that is the response time of the AF in the system that suffers the larger response time;
- *outage probability* (OP), calculated as the percentage of AFs in the system that are afflicted by a response time greater than their own deadline.

The system performance analysis has been provided by resorting to comparison to the following methods already proposed in [292]

- The *Clustered Enhanced Heuristic Placement Algorithm* (CEHPA) that is a cluster based solution which identifies a subset of suitable ECSs, selecting them considering the distance from the nearest ECS, to support a given application. Then, the best ECS is found, considering the minimization of the ORT;
- The *Substitution Enhanced Heuristic Placement Algorithm*(SEHPA), that is an almost-optimal solution, as highlighted in [292], to the VRCs placement problem aiming at minimizing the ORT, by considering all the possible ECSs for each application that has to be replicated.

Furthermore, performance comparisons have been provided also with the matching approach proposed in [17], here referred as *Matching Theory Algorithm* (MTA), of which we have adapted the behavior to our scenario, by using the same metrics considered in the ITAMA, but without the DP and by following the algorithm procedure explained in [17]. Finally, comparison with the *closer deadline greedy algorithm* (CDGA) has been proposed, and that, for each level k of the AFs, the method acts as follows

- let z be the application requested by the function request s_k^j, whose associated deadline a_j is the closest one among the deadlines of the other unallocated ARs belonging to the k-th level of the AFs. If does not exist any VRCs on the ECSs, and there is not any available ECS on which a VRC of z can be loaded, then s_k^j is offloaded to the Cloud;

Table 4.3: Computational Complexity

ITAMA	$O(\|S\| \cdot \|F\| \cdot \|S\| \log \|S\|)$
SEHPA	$O(\|S\|^2 \cdot \|F\|^3)$
CEHPA	$O(\|F\| \cdot \|S\|^2 \cdot (\|F\| + \|S\|))$

- if any application requested at level k, and already allocated, has lead to the placement of a VRC of z on an ECS, a VRC of z is allocated on the nearest available ECS, if at least one ECS is available;

- if there exists one function request, at level k, that has lead to the allocation of z on an ECS, s_k^j is offloaded on the nearest ECS containing a VRC of z;

- each queue in the system, i.e., the queue on the communication links and on the ECSs, is managed by serving first the ARs with high priority.

We start our analysis by comparing the system performance of the proposed hybrid EC-Cloud architecture in comparison with a Cloud based solution in Figure 4.14 in terms of achieved ORT values as a function of the number of IIoT-devices. As it is evident in the Figure, the hybrid EC-Cloud approach using the ITAMA strategy results in significant performance gain in comparison with the Cloud based alternative. The good behavior of the proposed allocation framework is also highlighted in Figure 4.15 where the ORT behavior is shown as a function of the maximum length of the AFs in the system in comparisons with the different allocation approaches. Likewise, from the results provided in Figure 4.16 it is straightforward to note the good improvement on the AF mean response time when the ITAMA is applied. Furthermore, thanks to the ECSs that prefers AFs with close deadlines, the good behavior of the proposed approach is also confirmed in Figure 4.17 in terms of WRT.The same considerations can be applied to the results given in Figure 4.18 in terms of achieved OP as a function of the number of IIoT-devices in the case of 18 ECSs. Finally, the computational complexity of the proposed strategy and of the alternatives presented in [292] has been reported in Table 4.3.9. The notable advantage of the the proposed approach is clearly evident in this table in comparison with the other considered alternatives.

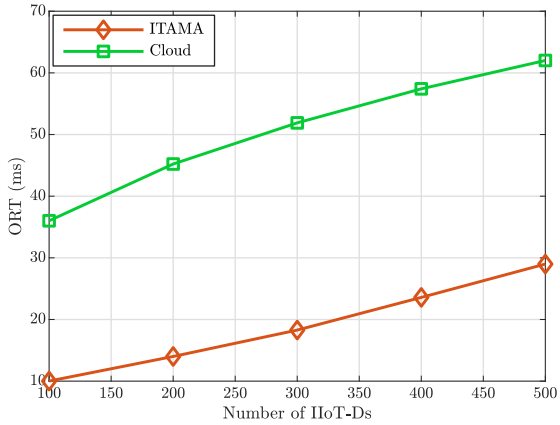

Figure 4.14: ORT Cloud comparisons as a function of the number of IIoT-devices

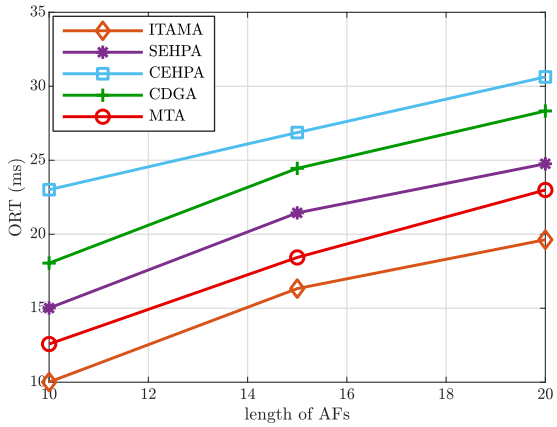

Figure 4.15: ORT as a function of the AFs length

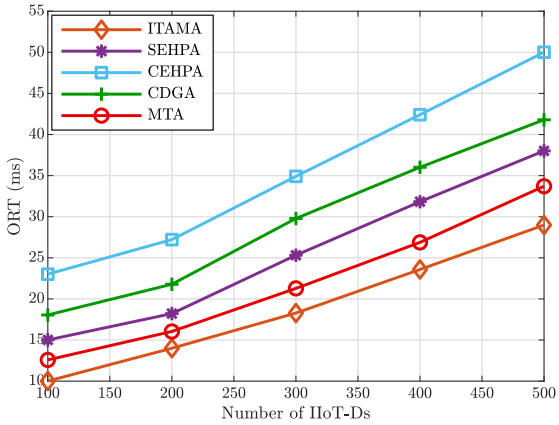

Figure 4.16: ORT as a function of the number of IIoT-devices.

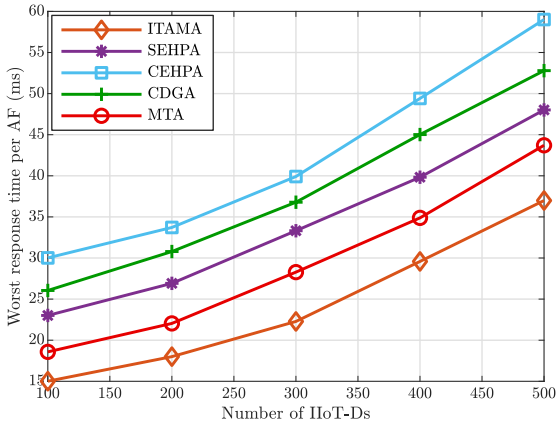

Figure 4.17: Worst response time per AF for 8 ECSs, as a function of the number of IIoT-devices.

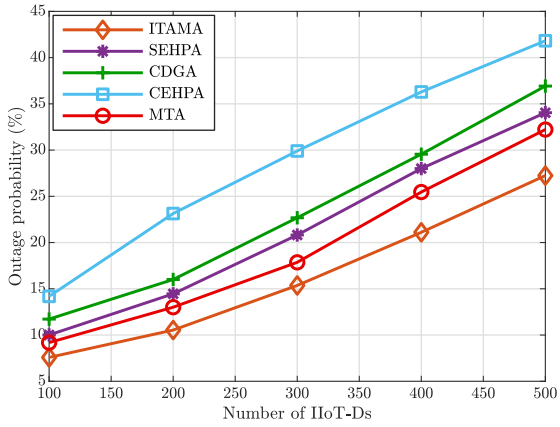

Figure 4.18: Outage probability for 8 ECSs, as a function of the number of IIoT-devices.

4.4 Conclusion

The study addresses the VRCs placement problem in a hybrid EC-Cloud network structure for a IIoT scenario. The proposed framework resorts to the formulation of a matching game with externalities between the applications requested by the IIoT-devices and the ECSs. The proposed placement strategy, i.e., ITAMA, realizes the matching game in order to minimize both the ORT and the number IIoT-devices that suffer a response time greater than their own time deadline. A theoretical discussion and proof about the stability of the reached matching have been also provided, according to the strictly-two-sided exchange-stability definition. Finally, the validation of the better behavior ITAMA solution in comparisons with different alternatives, namely SEHPA and CEHPA, recently presented in literature has been provided by presenting extensive computer simulation results.

4.5 A Matching Game for Tasks Offloading in Integrated Edge-Fog Computing Systems

4.5.1 Motivation

During last decade, the ever increasing expansion of IoT to numerous different application areas, such as smart buildings, smart city, augmented reality, e-health, and video analytics, and so on, has given rise to an increasing demand of both the intensive computation capabilities and high network bandwidth to support the massive volume of heterogeneous data traffic with strict real time constraints [62, 98, 253, 288, 290]. Edge and Fog computing paradigms have been widely studied to overcome latency and network congestion drawbacks of the traditional cloud architecture, typically characterized by data centers located in remote network areas, far away from end users, i.e., significant transmission latency, network congestion and scarce adaptability to real-time context [62,288]. In practical words, to move computation towards network edges, close to the end users may extend the cloud architecture benefits to network edges, by deploying computation nodes, named fog nodes (FNs), arranged to perform intensive computation tasks with low latency. Similarly, edge computing aims at process directly at the devices level at the edge of the network, promoting the collaborative sharing of storage and processing capabilities [62,269]. The integration of these two paradigms into a whole system considers that the edge devices (EDs) may take advantage in computing their tasks on a close ED higher in available processing capability or on a FN. Furthermore, since EDs are usually limited in battery supply and computation resources, sometimes they may be unable to execute a given task, or due to existing process acting in background on itself, an ED may prefer to compute on a close ED with wider available processing capability. Tasks offloading allows to overcome these drawbacks by allocating the computation of tasks to nearby devices. In particular, we focus here on an integrated EC-FC system where an ED can offload its task to a FN in alternative to nearby EDs, depending on advantages on a reduced processing time or overall energy consumption. Among the possible D2D alternatives, we have considered the network-assisted D2D communication mode, where the cellular base station (BS) reserves a suitable number of resources to support D2D communications between pairs of EDs.

4.5.2 Contributions

This work presents the analysis and the results discussed in [62] main contributions of this paper are the following

- The computational offloading problem formulated in terms of a matching game with externalities and incomplete preference lists. In fact, each ED requiring computation selects the computation site based on the minimization of its completion time, and is able to compute on itself, on a neighbors ED or on a specific FN belonging to a subset of the FNs forming the Integrated Edge-Fog Computing System under consideration;

- Proposal of a matching theory based allocation strategy to reach a suboptimal solution to minimize both the system energy consumption and the longest task overall completion time. Furthermore, since stability is not a trivial issue in the matching game with externalities, a post matching procedure has been provided to guarantee the two-sided-exchange stability [35];

- Numerical results derived by extensive computer simulations to validate the good behavior of the proposed method by providing performance comparisons with the potential game approach, other matching based strategies recently proposed in literature, and, finally, with an alternative solution based on the exclusive use of the far cloud.

  The contribution of this paper consists of proposing a method to achieve a suboptimal solution for the joint optimization of the normalized energy consumption and the worst task completion time, i.e., the highest delay experienced by an ED to complete its task computation. In this regard, a novel post matching procedure to ensure the system stability is proposed. It is important to note that in this work a matching game with externalities has been proposed, which differs from those presented in [175] and in [200], in which the interactions and relations among the preferences of the elements involved in the matching are neglected. Finally, the good behavior of the proposed approach is validated by presenting performance comparisons with the algorithms proposed in [53,200], the GS algorithm, the well known potential game approach [90], and the offloading considering exclusively the far cloud as computation site.

4.5.3 Related Works

The tasks offloading problem plays a crucial role in the FC/EC research area and it has been widely considered in the recent literature with reference to different application scenarios [52, 53, 101, 175, 204, 243, 289]. In particular, in [204], the Lyapunov optimization method is used to minimize the average energy consumption for the execution of all tasks. In [243] the problem of the computation offloading in a hierarchical fog-cloud architecture with unmanned aerial vehicles has been addressed. In this case the authors propose a two-phase algorithm in which in the first phase the association between users and cloud is pursuit, while the second phase consists of the optimization of the remaining variables. Authors in [101] propose a fiber based cloud-mobile edge computing architecture to relieve the high congestion and high latency due to the computation on the cloud. The problem of the computational offloading of the intensive tasks to the cloud is studied in [289] with the aim at minimizing the energy cost clock frequency of the mobile devices, by resorting to the alternative optimization and the difference convex function programming. In [52] the authors consider a ultra-dense network and address the delay minimization of the task offloading problem in a mobile edge computing context.

Recently, matching theory has been widely used in the computational offloading problem. One example is represented by [175], in which the Gale-Shapley (GS) algorithm is applied to reach a stable matching between EDs, in order to offload a task originated on a device on a more powerful neighbor. Paper [53] gives a different point of view of the problem, by proposing a collaborative task execution in an EC network, aiming at minimizing the system energy consumption. The energy-efficient task assignment has been achieved by using a graph-matching-based approach. Finally, paper [200] proposes a joint optimization of the system overhead in multi-server mobile EC. In particular, the offloading decisions are provided by applying matching theory. Furthermore, authors in [273] propose the allocation of blockchains on edge nodes, considering a cloud/fog architecture, by modeling the offloading price control as a two stage Stackelberg game. In practical terms, in [273] the cloud/fog providers set the resource price and the followers are constituted by the edge devices, which purchase resource and perform the mining tasks. The blockchain networks are analyzed also in [122], in which the social welfare is maximized, as regards the entry into the network. Furthermore, paper [151] addresses the heterogeneous resource allocation problem in a fog-

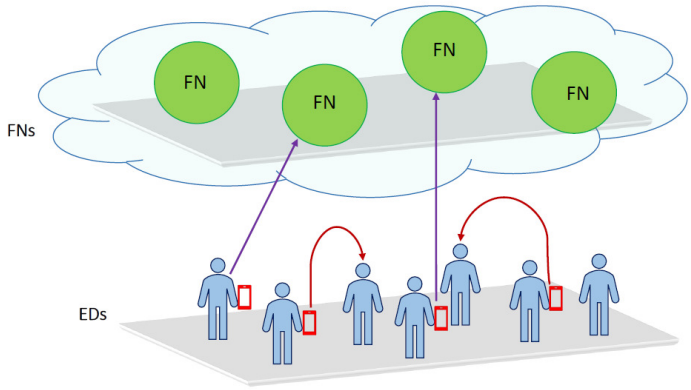

Figure 4.19: Reference scenario.

IoT network, by involving an improved version of a genetic algorithm to find a suboptimal solution for the minimization of the system energy consumption problem.

4.5.4 Reference Scenario

In performing our analysis we refer to the application scenario depicted in Figure 4.5.4. In particular, we suppose the presence of a cellular system in which the base station (BS), reserving a set $\mathcal{K}$ of channels, each of bandwidth B, to support D2D links for a set $\mathcal{E}$ of n EDs needing task computation. Furthermore, we suppose the presence of a task manager unit (TMU), responsible for controlling our framework by implementing the proposed algorithm. The EDs have been assumed with limited mobility according to a pedestrian scenario, so that we can assume with a good approximation that their position does not change during the execution of the proposed planning algorithm and corresponding EDs tasks execution.

The EDs set is assumed heterogeneous in terms of computation capability and power consumption. Hence, each ED j belonging to $\mathcal{E}$ is characterized by a given computational capability expressed in terms of number of central processing unit (CPU) cycles per second q_j, and power consumption.

Moreover, we consider FNs equipped with a more powerful computation capability and able to communicate with a given ED $\in \mathcal{E}$ by means of

a shared (according to the Time Division Multiplexing (TDM) approach) channel of bandwidth B reserved to this purpose by the cellular BS. It is important to note that, in this case, the used channel has the same bandwidth of those reserved to D2D communications, but they are not the same. We assume also that each task computation request is originated at a given ED and that no more than one task can be executed on a same ED. In particular, each ED j, as in [53], makes available a part η_j of its computation capacity q_j, to accept an offloaded task, since any ED may run tasks in background. Consequently, the available processing capacity for a task on ED j results in

$$c_j = q_j \eta_j. \tag{4.53}$$

Likewise, for each FN $y \in \mathcal{F}$, we have

$$c_y = q_y \eta_y. \tag{4.54}$$

In particular, in this case we have assumed that the FN is able to run till χ_y tasks in parallel. Moreover, in order to preserve its lifetime due to the limited battery supply, each ED $j \in \mathcal{E}$ devotes a limited amount of energy, referred hereafter as $\pi_{MAX,j}$, to the computation of an offloaded task. Differently, the FNs have not any constraint on the amount of energy to be devoted to the computation of the offloaded tasks.

Task Completion Time

According to the energy constraint individually introduced by each ED interested in a task computation in order to preserve its life time we may have:

- the task can be computed locally or on an ED close to the ED j, if the hosting ED has alloocated an enough energy amount to perform the task j computation;
- if the computation of task j requires an energy consumption greater than the energy allocated by the available EDs computation capabilities, the task is offloaded on one FN among those in line of sight (LoS) of the ED j, i.e., $\mathcal{F}_j$.

If a task runs on the ED j where it is originated, the overall task completion time is exactly its makespan, that is

$$\tau_j = \frac{s_j}{c_j}, \tag{4.55}$$

where s_j is the size of the task expressed in terms of the number of required CPU cycles.

Alternatively, if the ED j offloads its task to the FN by means of the shared cellular link, in addition to the task makespan, we have to consider the task *access delay* a_j, i.e., the time needed to access the *direct link* between the ED j and a FN according to the TDM scheduling, the task *response delay* d_j, i.e., the time occurring to access the *reverse link* from a FN to the ED j to notify the task computation response, the *waiting time* ω_j^y experienced by ED j at the FN y site waiting for task computation [5], and the task *request /response transmission time*. Hence, we have

$$\tau_{j,y} = \frac{s_j}{c_y} + \omega_j^y + i_{j,y} + o_{j,y} + a_j + d_j, \tag{4.56}$$

where $i_{j,y}$ and $o_{j,y}$ are the transmission time needed to send out task j to the FN y and the task response latency from FN to ED j, respectively. More in depth, assuming the same data rate for transmissions on the direct and reverse link, R_y, we have $i_{j,y} = \iota_j / R_y$ and $o_{j,y} = \upsilon_j / R_y$,

where ι_j, υ_j denote the the task request and response size (both in bits), respectively. Finally, ω_j^y is the waiting time experienced by the task j before the start of its computation on the FN y, due to the computation time of all the tasks previously offloaded on the FN according to a FIFO policy. Conversely, in the case of offloading to a nearby ED, by taking into account that only one task can be accepted at a time, we have to consider the mutual interference due to a possible concurrent use of the same D2D link by other interfering EDs pairs. For such reason, the overall task completion time $\tau_{j,z}^k$ for the ED j, when its task is offloaded to the ED z, with $z \in \mathcal{D}_j$, through the channel k, can be defined as

$$\tau_{j,z}^k = \frac{s_j}{c_z} + \frac{(\iota_j + \upsilon_j)}{R_k^j}, \tag{4.57}$$

where R_k^j is the achievable data rate for the ED j, related to the use of the channel k and defined as follows

$$R_k^j = B \log_2 \left(1 + \frac{P_d \beta \phi_{j,z}^{-\alpha}}{\sigma + \sum\limits_{a \in \mathcal{E} \setminus \{j\}} P_d \beta \phi_{z,a}^{-\alpha} \delta_{z,a}} \right), \tag{4.58}$$

[5] The waiting time at the FN y site for a given task depends on χ_y and on the number of tasks previously offloaded to the FN according to the procedure outlined in Section 4.5.6.

where the terms denote: $\phi_{z,a}$ the distance between the ED z and the ED a; β the path loss coefficient; α the path loss exponent; σ the Additive White Gaussian Noise; $\delta_{z,a}$ a binary value equal to 1 if the ED a offloads its task to a nearby ED by using the same channel for the D2D communication, 0 otherwise; and, finally, P_d the ED transmission/receiving power used at any involved ED.

The resulting overall completion time for task originated at the ED j, results in

$$T_j = \tau_j \lambda_j + (1 - \lambda_j) \Bigg[\sum_{k \in \mathcal{K}} \sum_{z \in \mathcal{D}_j} \sum_{y \in \mathcal{F}_j} (\tau^k_{j,z} \delta_{j,z} \zeta_{j,k} + (1 - \delta_{j,z}) \tau_{j,y}) \Bigg], \quad (4.59)$$

where λ_j is a binary value equal to 1 if ED j has a task to compute and it does not offload it, 0 otherwise. Furthermore, $\delta_{j,z}$ is a binary value equal to 1 if ED j offloads its task to ED z, with $z \in \mathcal{D}_j$, or 0 otherwise, while $\zeta_{j,k}$ is equal to 1 if task j selects channel k to offload its task on a nearby ED.

Energy Consumption

In order to provide a complete analysis, the overall energy consumption needed to complete a task computation process at a given site is evaluated. Considering a tagged ED $\in \mathcal{E}$, having a task to be computed, i.e., the ED j. If the ED j locally executes it, the corresponding energy consumption results in

$$\pi_j = \tau_j \rho_j \leq \pi_{MAX,j}, \quad (4.60)$$

where ρ_j is the power consumption of the ED j CPU, that can be derived as in [134]. Likewise, if the ED j offloads its task to the FN y, the energy consumption of its computation is

$$\pi_{j,y} = \rho_y (\frac{s_j}{c_y}) + P_f (i_{j,y} + o_{j,y}), \quad (4.61)$$

where P_f is the transmission/receiving power assumed the same at the ED and FN y site and ρ_y is the FN CPU power consumption. Finally, if ED j offloads its task to a nearby ED z, with $z \in \mathcal{D}_j$, whose CPU power consumption is ρ_z, the resulting energy consumption for the task computation on ED z can be defied as

$$\pi_{j,z} = \rho_z (\frac{s_j}{c_z}) + P_d (i_{j,z} + o_{j,z}) \leq \pi_{MAX,z}, \quad (4.62)$$

where P_d represents the ED transmission/receiving power when the offloading between EDs is achieved by resorting to a D2D link. Hence, the normalized system energy consumption, i.e., the overall energy needed to complete the computation of all the tasks normalized to the number n of EDs having a task to be computed, can be defined as

$$\mathcal{E}_{sys} = \frac{1}{n} \sum_{j \in \mathcal{E}} \pi_j \lambda_j + (1 - \lambda_j)[\sum_{z \in \mathcal{D}_j} \sum_{y \in \mathcal{F}_j} (\pi_{j,z} \delta_{j,z} + \pi_{j,y} \gamma_{j,y})], \tag{4.63}$$

with $\gamma_{j,y}$ equals to 1 if ED j offloads its task to the FN y, or 0 otherwise.

4.5.5 Problem Formulation

The aim of this study is the minimization of both the normalized system energy consumption and the worst overall completion time. As consequence, in formal terms, the tasks offloading problem can be expressed by

$$\min \mathcal{E}_{sys}(\lambda, \delta, \gamma) \ \textit{and} \ \min_{\lambda, \delta, \zeta} \max_{j \in \mathcal{E}} T_j(\lambda, \delta, \zeta) \tag{4.64}$$

s.t.

$$\lambda_j + \sum_{z \in \mathcal{D}_j} \delta_{j,z} + \sum_{y \in \mathcal{F}_j} \gamma_{j,y} = 1, \quad \forall j \in \mathcal{E}; \tag{4.65}$$

$$\forall j \in \mathcal{E} \sum_{z \in \mathcal{D}_j} \delta_{j,z} > 0 \Rightarrow \sum_{k \in \mathcal{K}} \zeta_{j,k} = 1; \tag{4.66}$$

$$\forall j \in \mathcal{E} \sum_{z \in \mathcal{E}} \delta_{j,z} \leq 1, \tag{4.67}$$

$$\pi_j \leq \pi_{MAX,j}, \forall j \in \mathcal{E}, \tag{4.68}$$

where λ, δ, γ and ζ are vectors whose elements are $\lambda_{i,j}, \delta_{j,z}, \gamma_{j,y}, \zeta_{j,k} \forall j, z \in \mathcal{E}, \forall y \in \mathcal{F}$, and $k \in \mathcal{K}$, respectively. Constraint (4.65) imposes that each ED has to necessary choose only *one* computation site among itself, a nearby device or on one FN, constraint (4.66) expresses that, in case of offloading on a nearby device, only one channel has to be selected. Constraint (4.67) means that each ED can compute no more than one task. Finally, constraint (4.68) means that the energy consumption of each ED cannot exceed the allocated energy amount to perform computation of the offloaded task. The problem formulated in (4.64)-(4.67) is a mixed integer linear programming (MILP), generally NP-hard to solve [2], hence, this motivates us to search for a suboptimal matching based solution to solve it.

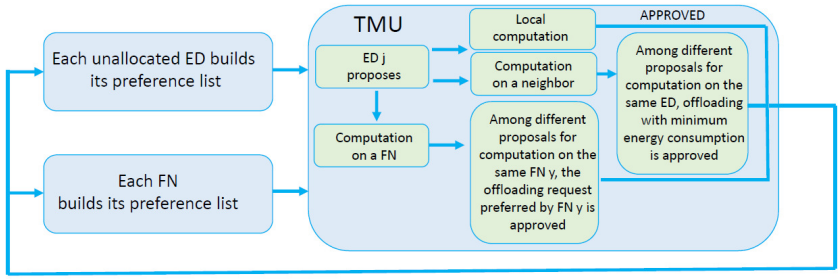

Figure 4.20: Proposed matching theory strategy

4.5.6 Proposed Solution and Algorithm

The tasks offloading problem can be easily formulated in terms of matching game, in order to establish mutually beneficial relations between the the EDs requiring task computation, and the set of all the possible computation sites. In particular, for a task originated at a given ED, i.e., ED j, due to its position, the length of its task and the energy capabilities of its EDs neighbors, the set of the whole possible computation set changes for different EDs. By assuming $\mathcal{A}_j$ the set of all the possible computation sites for ED j, $\mathcal{A}_j$ may include the ED j, in dependence of its energy capability in relation to the length of the task needing computation. Similarly, $\mathcal{A}_j$ may contain one or more neighbors EDs, while the FNs in the LoS of the ED j represent possible computation sites for task j Generally speaking, since each ED j has not the access to each other ED element and to each FN of $\mathcal{F}$, the EDs preferences lists are not complete, i.e., the number of elements constituting the preferences lists change on the basis of the considered ED. Furthermore, it is straightforward to note that $\mathcal{A}_j$ changes during the matching game, since, whenever a given ED $\in \mathcal{A}_j$ is selected for the offloading of a task, it is inhibited to accept computation of any other task. In the proposed matching approach the EDs requiring a task computation and the FNs individually express their task computation preferences defined later.

4.5.7 EDs and FNs preference list

Each ED j needing task computation is a selfish and rational player with the aim of minimizing its task overall completion time, according to (4.55), (4.56) and (4.57). In particular, each ED j forms its preference list over $\mathcal{A}_j$ by sorting the computation sites in order to find

$$\min_{\lambda,\delta,\zeta} T_j(\lambda, \delta, \zeta). \tag{4.69}$$

The FNs preferences lists are built in order to pursue the minimization of the total system energy consumption. Therefore, each FN sorts the EDs which request task offloading on itself, by finding

$$\min_{\lambda,\delta,\gamma} \mathcal{E}_{sys}(\lambda, \delta, \gamma). \tag{4.70}$$

It is important to note that the allocation of a task on a computation site can impact the preferences of other tasks, i.e., may reduce the number of EDs able to accept a task offloading or may increase the access delay due to the use of the shared D2D channels.

4.5.8 Allocation Strategy

The proposed algorithm provides different matching types according to the computation site type. In particular, it is allowed a many-to-one matching between the EDs set and the FNs, and a one-to-one matching among the EDs. The algorithm runs at the cellular TMU site which has a full awareness of the application scenario in terms of EDs/FNs location, computation capabilities and communication channels conditions. The TMU collects the preferences list of each ED requiring task computation and takes decisions according to an iterative procedure that, for each task, selects the most appropriate computation site that guarantees the best trade off between the overall task completion time and the system energy consumption. At each step of the task allocation strategy, a modified version of the Gale-Shapley algorithm (GSA) [84, 88, 211, 212] is applied by following the steps

1. Each ED, i.e., ED j, having a task to compute creates its preference list on the available computation sites in $\mathcal{A}_j$, aiming at minimizing its overall task completion time.

2. Each ED having a task to compute proposes its preferred computation site to the TMU.

3. The TMU acts as follows

   (a) Each ED requiring locally computation is enabled to execute its task;

   (b) Each ED that receives more than one computation request, accepts the ED that minimizes $\mathcal{E}_{sys}$;

   (c) Each FN that receives more than one computation request, accepts the ED with the highest position in its preference list among the proposals received;

4. The allocated tasks are removed by the unallocated tasks set $\mathcal{T}$ as well as each ED with an assigned task computation is no longer considered available for a further task allocation. Finally, the TMU updates the waiting time of all the FNs that have accepted one task;

5. Each unallocated ED updates its reference list;

6. Each FN updates its preference list;

7. Repeat steps $1) - 7)$ until $\mathcal{T}$ is empty.

In Figure 4.20 the proposed matching based offloading framework is summarised. As regards the study of the complexity of the proposed algorithm, we consider the worst case scenario in order to perform the time complexity analysis. Under such assumption, we suppose that any couple of EDs can be connected together by D2D links. In this respect, it is easy to note that, at the first step, when the whole $\mathcal{E}$ set is unallocated, the number of all the possible computation sites for each ED is exactly equal to n. Hence, the complexity of the proposed solution is in the order of $\mathcal{O}(n \cdot (n \cdot log(n))$. However, it is important to note that the worst case scenario previously analyzed is strongly pejorative and very unlikely in actual application scenarios.

4.5.9 Stability Analysis

Within the matching games with externalities context, there not exists a unique definition of stability. In this respect, we consider the stability definition proposed in [35] and we adapt it to our model by introducing the following definitions:

Definition 11. A blocking pair (BP) is defined as a pair of tasks (t_1, t_2) if both t_1 and t_2 improved their overall completion time after exchanging their partners and eventually the assigned channel.

Definition 12. Let (u, v) be a BP. The exchange between the components of the BP (u, v) is allowed if all the following conditions are satisfied:

1. $T_{t_1}^{(u,v)} < T_{t_1}$ and $T_{t_2}^{(u,v)} < T_{t_2}$;
2. $P_{sys}^{(u,v)} \leq P_{sys}$.

where $T_{t_1}^{(u,v)}$, $T_{t_2}^{(u,v)}$ expresses the overall completion time of task t_1 and t_2 respectively after the exchange, and similarly $\mathcal{E}_{sys}^{(u,v)}$ is the system power consumption after the exchange. Definition 12 means that the exchange is provided when both task t_1 and t_2 improve their overall completion time and also the system power consumption does not get worse. Since a matching $\mathcal{M}$ is stable if there is no BPs for which the exchange is allowed, as defined in Definition 12, we propose a strategy to reach a final outcome matching that acts as follows

1. given an outcome matching $\mathcal{M}$, all the BPs belonging to $\mathcal{M}$ are detected;
2. for each BP $(u, v) \in \mathcal{BP}$, the requested exchange is attempted;
3. conditions 1) and 2) of Definition 12 are evaluated;
4. if conditions 1) and 2) of Definition 12 are verified, the (u, v) swap is allowed.
5. repeat steps 1)-4) until at the previous iteration a swap has occurred.

The method terminates in a stable matching and has a temporal complexity of $\mathcal{O}(n \cdot \phi)$, where ϕ is the total number of switched occurred in the post matching procedure. Since the conditions to allow the swap are very strict, we can say that $\phi << n$.

4.5.10 Numerical Results

Furthermore, performance comparisons with alternative methods are given in order to highlight the better behavior of the proposed solution. The presented numerical results have been derived by considering a reference scenario with a different number of heterogeneous EDs. Moreover, for each ED

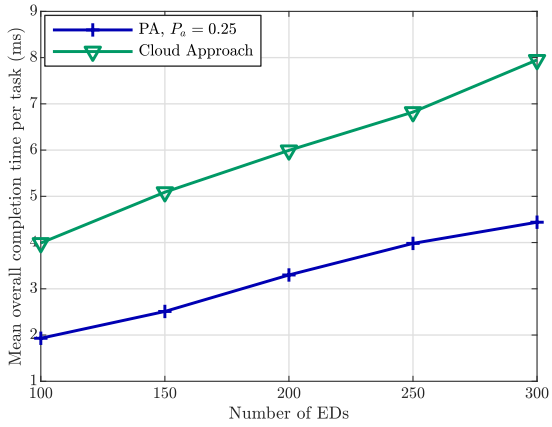

Figure 4.21: System performance by assuming a cloud architecture and fog architecture.

we have assumed a CPU working frequency uniformly distributed within the set $\{0.2, 0.4, 0.6\}$ GHz with the associated computation energy consumption evaluated as in [134]. Likewise, we have assumed a set of 4 FNs with CPU working frequency uniformly distributed within the set $\{0.8, 1.2, 1.4\}$ GHz, 4 channels with a bandwidth of 20 MHz reserved by the BS for support D2D communications and two channels at a fixed data rate equal to $20Mbit/s$ reserved by the BS to allow communications on the *direct* and *reverse* link from and towards FNs and cloud, respectively. The parameter $\pi_{MAX,j}$ for any $ED_j \in \mathcal{E}$ has been assumed as uniformly distributed within the interval $[0.18, 0.27]$ kJ. Furthermore, cloud[6] is supposed equipped with a CPU with working frequency uniformly selected within the set $\{3.2, 4.8, 5.6\}$ GHz. Each ED has to compute a task formed by number of instructions N selected with an equal probability in $[1500, 2300]$, with each instruction requiring a number of mean clocks per instruction (CPI) uniformly selected within $\{12, 8, 4, 2\}$. Furthermore, we have assumed a 64 bits instructions format, and the length (bits) of the associated task computation output uniformly selected in $[N/2, N]$. The specific time deadline T_h constraints for real time

[6]Once linked to the BS by means of the shared wireless channel, the EDs are connected to the cloud through an high speed (i.e., optical) link for which we have assumed negligible the transmission delay.

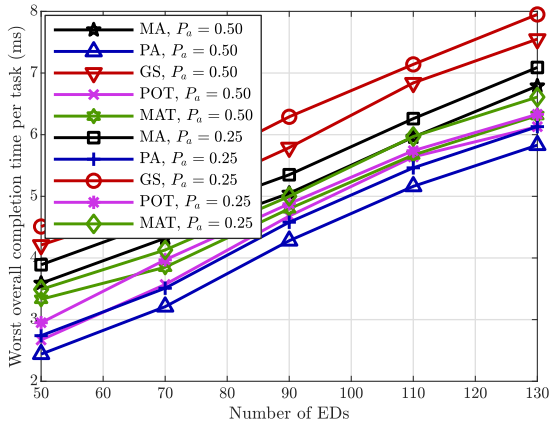

Figure 4.22: Worst overall task completion time comparisons

applications have been considered for each task uniformly distributed in the interval $[0.8, 2.4]$ ms. Finally, the EDs have been spatially distributed within a circular area of radius 100 m according to a Poisson Point Process (PPP). Transmission/receiving power for D2D communications is 0.1 W, while for communications between EDs and the FNs, it is 0.8 W. Differently, as regards the cloud specifications, we supposed, according to [1], a CPU cycles frequency equals to 5.6 GHz and able to run 32 tasks in parallel, against the 8 of each FN.

The performance of the proposed task offloading scheme are compared with the following alternative approaches

- The bipartite matching graph algorithm (MA) proposed in [53];
- The classical Gale-Shapley (GS) matching algorithm (i.e., without considering the preference list updating);
- A matching game derived by that proposed in [200] (MAT) without considering the power control procedure being not of interest in our case;
- The well known potential game [90] (POT) in which each ED and FN aims at minimizing its computation time;

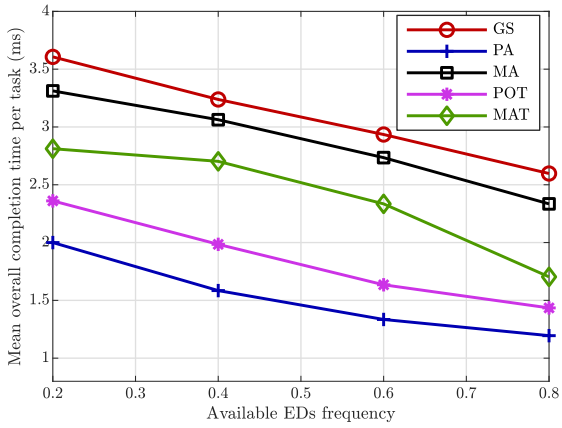

Figure 4.23: Mean overall completion time per task as a function of Pa for 100 EDs.

- An offloading approach that considers only a far cloud as possible alternative.

In particular, the results analysis has been conducted considering as performance metrics:

- The *mean overall task completion time* ($\bar{T}_C$), defined as the sum of all the task completion times T_j given by (4.59) divided by the number of the EDs that require computation;
- The *worst overall task completion time* (T_W);
- The *normalized system energy consumption* (S_{sys});
- The *outage probability* (P_{out}) defined as the probability that a given task does not complete its computation within the associate time deadline.

As a consequence, Figure 4.21 shows the mean task overall completion time as a function of the EDs number. The Figure depicts performance comparison when the computation offloading is provided on the hybrid edge-fog computing scenario analyzed in the paper, or supposing a cloud architecture, in which all the task computation requests are all offloaded to a far cloud. As

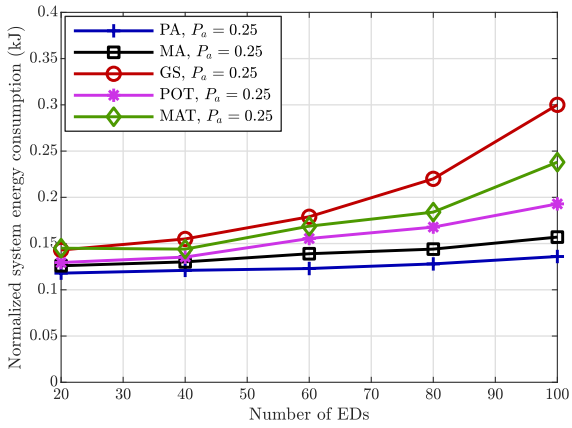

Figure 4.24: Normalized system energy consumption comparisons.

it is easily to note, the integrated edge-fog computing system based on the proposed offloading algorithm (PA) reaches significant improvement on the mean overall task completion time. The validity of the PA is confirmed in Figure 4.22, where T_W is again given as a function of the number of EDs requiring a task computation. Furthermore, we show in Figure 4.23 the mean overall task completion time considering a fixed number of EDs equals to 100, for different values of the probability that one ED has not to compute a task. The superiority of our approach is again evident in this figure, also in comparison with different approaches such as GS, MA, MAT and POT. In order to complete our analysis, we compare in Figure 4.24 the $\mathcal{E}_{sys}$ parameter related to the proposed method with the considered alternatives, without taking into account any constraint for the T_h value, in order to be compliant with the original definitions of the comparison methods. This figure shows that the proposed method clearly outperforms all the considered alternatives. Finally, Figure 4.25 shows the achieved P_{out} for the application scenario under consideration as a function of the EDs number. The better behavior of the PA is again evident in the figure in comparison with the other methods by allowing a significant reduction of the energy wasted due to unfinished tasks computations.

It is straightforward to note that in Figure 4.23, Figure 4.22 and Figure 4.25, the PA and POT reach better results than other disciplines. It is

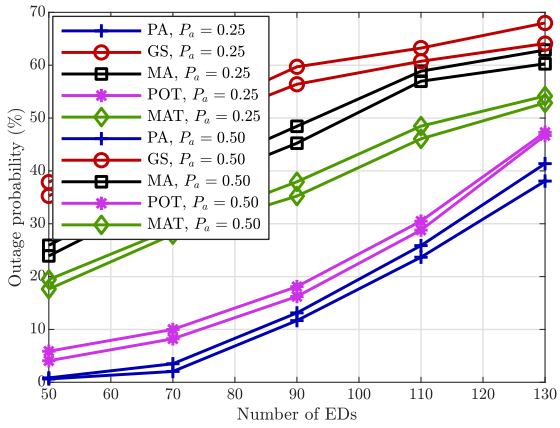

Figure 4.25: Outage probability comparisons.

due to the fact that they, after each allocation, update the system conditions on which they perform offloading allocation. Furthermore, both the approaches take into account the minimization of the longest task completion time. Differently, the other approaches do not update, or update less frequently than PA and POT, the system conditions and perform choices without considering the most recent changes into the system state. Finally, in Figure 4.24, MA approach exhibits a better behavior compared to its trends in other figures, since its main objective is the energy consumption minimization.

4.5.11 Conclusion

This study has presented an efficient tasks offloading framework, contextualized to an integrated Edge-Fog computing system, in order to provide solution to the task offloading problem. Moreover, a matching game with externalities and incomplete preferences lists between the tasks set and the computation sites set has been formulated, aiming at minimizing both the system energy consumption and the worst overall task completion time. The proposed algorithm achieves a different type of matching according to the nature of the computation sites. In particular, the task offloading strategy realizes a many-to-one matching between the tasks and the FNs, and a

one-to-one matching between the tasks and the EDs sets. Furthermore, a post matching procedure has been proposed, in order to provide matching adjustments, allowing the computation sites exchanges whether system performance improves and no one is worse off. The performance of the proposed tasks offloading strategy have been evaluated in terms of mean/worst overall task completion time, mean task communication time, total system energy consumption and outage probability, i.e., the probability that a given task does not complete its computation within the associate time deadline. Finally, performance comparisons with different approaches have been provided in order to validate the effectiveness of the integrated Edge-FoG computing solution based on the proposed tasks offloading technique.

Chapter 5

Queuing Theory for Delay Sensitive Computational Offloading

Things may come to those who wait, but only the things left by those who hustle.

Abraham Lincoln

5.1 Performance Analysis of an Edge Computing System for Real Time Computations and Mobile Users

5.1.1 Motivation

The exponential diffusion of mobile devices and applications demanding for real-time computation and processing services, resulted during last decades, has unambiguously implied a massive emergence of searching strategies able to handle data-intensive and delay sensitive applications. The rise of this class of services demand, typically tactile internet, augmented reality, online games, mission critical applications, autonomous driving processes, or intelligent transportation system (ITS) services, has brought new challenges in terms of ensuring continuity of service, low latency, energy efficiency and

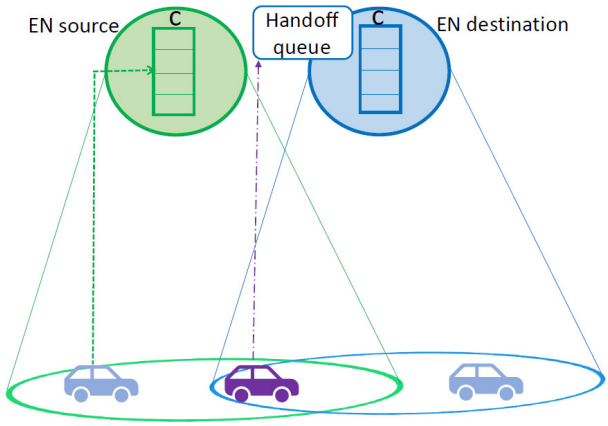

Figure 5.1: Handoff mechanism

high access speeds. Among all these aspects, ensuring continuity of service to mobile users is a very critical issue to be addressed, in order to guarantee high level of users experience [203].

Moreover, in addition to these real time applications, the well-known ITS currently constitutes a wide scenario in which users mobility and real time/delay sensitive applications strictly coexist [40, 55, 74, 194, 214, 298].

Recently, the new paradigm of the edge computing (EC) [166] has emerged as a promising methodology to support such real time applications for mobile users, in which the edge node are arranged provide computation capabilities to all the mobile users within a specific area. In such a context, the EC solution provides high speed computation services, low transmission delay and communication congestion, hence favoring applications requiring real time low latency computations and communications.

5.1.2 Contributions

In reference to the results and the analysis presented in [75], this study presents an analytical method based on the queuing theory analysis to derive the performance of an EC system devoted to provide computation facilities to mobile users demanding computation of real time applications. It is important to note that the proposed analysis is quite general and it is pretty applicable to the whole class of real time data traffic implying mobility. Ac-

cording to our approach, each EN has been modeled as a $M/M/C$ system with reneging, i.e, with premature departure of requests from the system before they achieve service completion or access to the service facilities. In particular, the main objective here is the minimization of the number of processors C to be allocated at each EN to ensure specific quality of service target values as regards the probability that a new arrived computation request is dropped and the probability that a computation request handoff attempt towards an adjacent EN fails. Finally, comparisons between analytical predictions and numerical results obtained by resorting to computer simulations, assuming actual application scenarios, are provided in order to validate the proposed analytical method.

5.1.3 System model

An EC system composed by a set $\mathcal{F}$ of edge nodes (ENs) to provide real time computation capabilities to mobile users (MUs), has been considered. Each EN is assumed equipped with k homogeneous central processor units (CPUs). Any MU within the service area of a given EN, i.e., the geographical area within which any MU can be reliably linked with the EN, can offload the real time applications computation to the EN throughout wireless links with a negligible latency. A computation request is dropped by an EN if at its arrival all the C CPUs are busy. It is important to note that all the applications of interest here are supposed composed by atomic processes, for which the task relocation on the destination EN is a safety procedure as regards the state of the involved programs [182]. Whenever a MU with a computation request in service at a given EN, (source EN), approaches the border of the overlapping area with an adjacent EN, i.e., enters in the service area of the adjacent EN (destination EN), a handoff procedure is started, i.e., a request for allocation of a CPU at the destination EN is submitted, according to the received signal strength (RSS) principle [68]. We assume here that the service areas of adjacent ENs overlap on a limited geographical region and that any MU entering an overlapping area (see Figure 5.1) starts to constantly monitoring the RSSs from the adjacent EN. As soon as the RSS from the adjacent EN overcomes a given threshold value, the MU submits to that EN a handoff request, i.e., the allocation request for a new CPU at the destination EN to continue the service of its application. The handoff request has to be accomplished before the involved MU definitively leaves the service area of its source EN (i.e., until the wireless communications

between the MU and the source EN are no longer possible). During the handoff request waiting time, i.e., the time elapsed from the start instant of the handoff procedure to when communications between the MU and the source EN are cleared, the computation request of the MU involved in the handoff procedure remains in service at the allocated CPU at the source EN. Therefore, we may have:

- the computation request is successfully completed before the waiting time expiration;
- the computation request is successfully passed for completing service (successful handoff) to the adjacent EN;
- the computation request is forced to termination (handoff failure) due to the waiting time expiration.

In order to derive the performance of the EC system under consideration in terms of the probability that a new computation request is dropped because at its arrival, at a given EN, all the available CPUs are busy, and the probability that a handoff attempt fails, we resort here to the use of Markovian queuing system model, i.e., an M/M/C system with reneging, with the system state assumed as the sum of the busy CPUs and the number of handoff requests waiting in the handoff queue. The goodness of the proposed approach based on the Markovian assumption will be validated later by providing comparisons between the obtained analytical predictions and the numerical results derived by performing extensive computer simulation in relation to actual application cases. A more detailed description of the proposed M/M/C queuing system is provided in what follows.

Traffic Model

In accordance with the adopted queuing system model, we assumed the rise of new real time computation requests flow within each EN serving area modeled as independent Poisson processes with the same mean generation rate per minute equal to λ_{rt}. Likewise, we assumed that the handoff requests flow towards an EN follows an independent Poisson distribution with mean attempts rate per minute λ_h. Furthermore, we consider the service time T_e requested by any real time application to a given EN exponentially

distributed with mean value given by

$$\frac{1}{\mu_e}. \tag{5.1}$$

Likewise, we denote as *connection* time for a MU the time interval during which the MU may be linked to a particular EN. Under the assumption that any MU within the service area of a given EN can be linked to the EN itself, the *connection* time of any MU corresponds to its sojourn time T_f of a MU in the service area of the EN of interest. Hence, according to [73, 99], the connection time T_f of a MU can be considered here as a negative exponentially distributed random variable with mean value given by

$$\mu_f = 0.7182\frac{V}{R}, \tag{5.2}$$

where V and R are the devices speed and the EN service area radius, respectively. Moreover, we assumed that a real time computation may stay in service at a given EN until the associated MU is within its service area. Hence, let T_H be the CPU holding time, i.e., the time during which a real time computation request stays in service on a particular CPU of a given EN, we have that

$$T_H = \min\{T_e, T_f\}. \tag{5.3}$$

Hence,

$$P\{T_H \leq t\} = 1 - P\{(T_e > t) \text{ and } (T_f > t)\}, \tag{5.4}$$

where $P(x)$ denotes the probability of event x. In our case, under the assumption of mutually independent events, we have

$$P\{T_H \leq t\} = 1 - e^{-(\mu_e+\mu_f)t}. \tag{5.5}$$

Therefore, the pdf of T_H results to be

$$f_{T_H}(t) = (\mu_e + \mu_f)e^{-(\mu_e+\mu_f)t}, \quad t \geq 0, \tag{5.6}$$

with mean value equals to

$$\frac{1}{(\mu_e + \mu_f)} = \frac{1}{\mu_H}. \tag{5.7}$$

We recall here that any new real time request that does not find an idle server (i.e., a CPU) at its arrival at a given EN is blocked and, hence, cleared from

the EN. Hence, by denoting with P_B the probability of this event and with P_{fh} the probability of a handoff failure, i.e., the probability that a handoff request is not successfully accomplished within its deadline, we have that the mean handoff attempts rate due to MUs in service at a given EN, λ_{ho}, results to be

$$\lambda_{ho} = [\lambda_{rt}(1 - P_B) + \lambda_{ha}(1 - P_{fh})]P_h, \tag{5.8}$$

where λ_{ha} is the mean handoff attempts rate towards the given EN and P_h, according to [114], is the probability that a MU having a real time computation in service gives rise to a handoff request towards an adjacent EN, defined as:

$$P_h = \frac{\mu_f}{\mu_f + \mu_e}. \tag{5.9}$$

Finally, under the assumption of an equilibrium condition, i.e., when the mean rate of MUs entering into the service area of a given EN with a computation in progres,s from all the adjacent ENs, is equal to the mean rate of MUs with a computation in progress leaving the service area of that EN towards the adjacent ENs, we have $\lambda_h = \lambda_{ha} = \lambda_{ho}$. Hence, from (5.8)

$$\lambda_h = \frac{(1 - P_B)P_h}{1 - P_h(1 - P_{fh})}\lambda_{rt}. \tag{5.10}$$

Handoff Queuing Management

According to the RSS handoff approach outlined in Section 5.1.3, any handoff request arrived at a given EN when all the C CPUs are busy is queued according to the FIFO policy, waiting for acquiring the service. Moreover, any queued handoff request is dropped from the queue without acquiring service if the connection time of the associated MU with the source EN expires.

By recalling that the state of the EN has been assumed given by the sum of the number of busy CPUs and the number of handoff requests waiting into the FIFO queue, through application of results concerning the M/M/C systems with reneging in standard queuing theory, we obtain that the probability of having the EN in the state x , P_x, when $x \leq C$ (in this case a new handoff arrival immediately accesses service) is given by

$$P_x = P_0 \frac{(\lambda_{rt} + \lambda_h)^x}{x!(\mu_H)^x}. \tag{5.11}$$

$$P_0 = \frac{1}{\sum_{b=0}^{C} \frac{(\lambda_{rt}+\lambda_h)^b}{b!(\mu_H)^b} + \frac{(\lambda_{rt}+\lambda_h)^C}{C!(\mu_H)^C} \sum_{b=C+1}^{\infty} \frac{\lambda_h^{b-C}}{\prod_{g=1}^{b-C}[(C+g)\mu_H + g\mu_q]}}. \quad (5.13)$$

$$P_B = \sum_{b=C}^{\infty} P_b = \frac{P_0(\lambda_{rt}+\lambda_h)^C}{C!(\mu_H)^C} \left[1 + \sum_{b=C+1}^{\infty} \frac{\lambda_h^{b-C}}{\prod_{g=1}^{b-C}[(C+g)\mu_H + g\mu_q]}\right]. \quad (5.14)$$

Different, for $x > C$, we have

$$P_x = P_0 \frac{(\lambda_{rt}+\lambda_h)^C}{C!(\mu_H)^C} \frac{\lambda_h^{x-C}}{\prod_{g=1}^{x-C}[(C+g)\mu_H + g\mu_q]}, \quad (5.12)$$

with P_0 given by (5.13).

From (5.11)-(5.13), it follows that the probability P_B that a new real time computation request is blocked results as in (5.14). Generally speaking, we assumed that a queued handoff request fails if it does not reach the head of the queue and it does not obtain access to the service at the destination EN before the expiration of its connection time with the source EN.

Let $P_{fh}(j)$ be the probability that a handoff attempt fails entering in the queue in position j. Then, the probability that a MU suffers a handoff attempt failure is

$$P_{fh} = \sum_{l=0}^{\infty} P_{l+C} P_{fh}(l). \quad (5.15)$$

Considering the handoff request queued in the $(j+1)$-th position, for which we suppose that the service of the associate real time application at the source EN cannot be completed before the expiration of its connection time. We will denote as P_{hd} in what follows the probability of this event which results defined as

$$P_{hd} = \frac{\mu_e}{\mu_e + \mu_f} = \frac{\mu_e}{\mu_H}, \quad (5.16)$$

Hence, by indicating with $P(j|j+1)$ the probability that a handoff computation request in position $j+1$ moves in position j before the connection time expiration, the probability that the considered handoff request reaches the head of the queue, obtaining the allocation of one of the C CPUs before the expiration of the associated connection time is given by

$$P_s = \left[\prod_{g=1}^{j} P(g|g+1)\right] P_{pr}, \tag{5.17}$$

where P_{pr}, defined later, is the probability that a CPU is assigned to the handoff request at the head of the queue before expiration of the related connection time.
Let $T_{e,i}, T_{H,i}, T_{s,i}$ be random variables exponentially distributed, denoting respectively

- $T_{e,i}$: the residual processing time of the handoff attempt request located at the i-th queue position when a change for the queue size occurs, i.e., a handoff request leaves or joins the queue;
- $T_{H,i}$: the residual time that a computation request remains allocated at a given CPU at the destination EN when the queue size changes and all the C CPUs are busy;
- $T_{s,i}$: the residual time that a MU with its handoff request at the $i-th$ queue position remains in the source EN service area.

Hence, supposing for the sake of simplicity $\alpha = (T_{H,1} \vee \cdots T_{H,C})$, $\gamma = (T_{e,1} \vee \cdots T_{e,i})$, and $\delta = (T_{s,1} \vee \cdots T_{s,i})$, it is straightforward to obtain the following relation

$$\begin{aligned} P(i|i+1) &= \text{Prob}\{(\alpha \vee \beta \vee \gamma \vee \delta) < T_{s,i+1}\} \\ &= \int_0^\infty \text{Prob}\{(\alpha \vee \gamma \vee \delta) < T_{s,i+1} = t\} f_{T_{s,i+1}(t)dt} \\ &= \int_0^\infty \{1 - e^{-[(C+i)\mu_H + i\mu_s]t}\} \mu_s e^{-\mu_s t} dt \\ &= \frac{(C+i)\mu_H + i\mu_s}{(C+i)\mu_H + (i+1)\mu_s}. \end{aligned} \tag{5.18}$$

Since a handoff attempt request, located at the head of the queue, obtains the allocation of a CPU if the connection time of the MU, associated to

that request, exceeds the CPU holding time for at least one of the busy C processors, we have

$$P_{pr} = \frac{C\mu_H}{C\mu_H + \mu_s}. \tag{5.19}$$

From (5.17), (5.18) and (5.19) follows

$$P_s(j) = \left[\prod_{g=0}^{j} \frac{(C+g)\mu_H + g\mu_s}{(C+g)\mu_H + (g+1)\mu_s}\right]. \tag{5.20}$$

Then, we have (5.21) and (5.22).

$$P_{fh}(j) = P_{hd}[1 - P_s(j)] = \frac{\mu_q}{\mu_H + \mu_q}\left(1 - \prod_{g=0}^{j} \frac{(C+g)\mu_H + g\mu_q}{(C+g)\mu_H + (g+1)\mu_q}\right). \tag{5.21}$$

5.1.4 Problem Formulation

As stated before, the scope here is to provide an analytical approach based on the use of the $M/M/C$ system model with reneging in order to derive the performance of an EC system devoted to perform real time computation to MUs. The focus here is on deriving the minimum number of CPUs to be allocated to any EN in order to meet specific service constraints in terms of probability that a new computation request is blocked and probability that a computation handoff attempt fails. Hence, our problem can be formulated as

$$\min C, \tag{5.23}$$

s.t.

$$P_{fh} \leq P_{fh,target}, \tag{5.24}$$

$$P_{fh} = P_0 \frac{(\lambda_{rt} + \lambda_h)^C}{C!(\mu_H)^C} \frac{\mu_q}{\mu_H + \mu_q}\left\{[1 - P_s(0)] + \sum_{j=1}^{\infty} \frac{\lambda_h^j}{\prod_{g=0}^{j}[(C+1)\mu_H + g\mu_q]}\right\}. \tag{5.22}$$

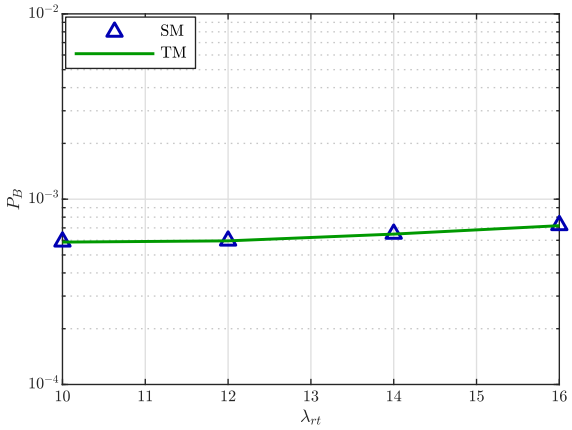

Figure 5.2: P_B as function of λ_{rt}.

$$P_B \leq P_{B,target}, \tag{5.25}$$

where $P_{fh,target}$ and $P_{B,target}$ are the maximum acceptable values referred to the P_{fh} and the P_B, respectively, in relation to specific QoS constraints. It is straightforward to note that the dominant constraint between (5.24) and (5.25) is represented by the minimum value between $P_{fh,target}$ and $P_{B,target}$. Generally speaking, since from a MU perspective having cleared a computation in progress is usually more annoying than having a new computation request blocked, we consider here a more stringent constraint for $P_{fh,target}$ than $P_{B,target}$.

5.1.5 Numerical Results

In order to validate the the proposed theoretical model (TM), based on the following assumptions

- Handoff arrivals at an EN assumed as a Poisson process with a given mean rate and independent from the Poisson arrival process of the new computation requests at the same EN;
- Independent Exponential distributions with appropriate mean values for all the other random variables of interest for our analysis,

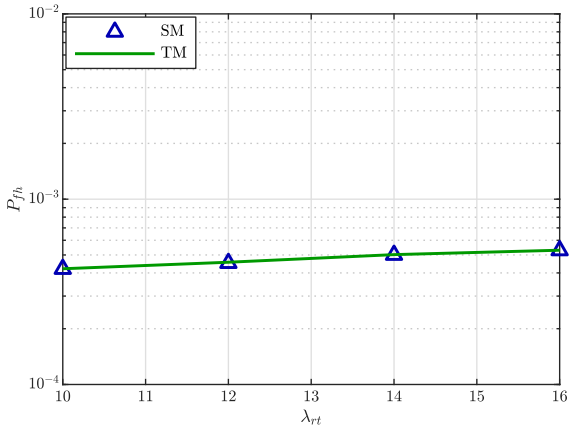

Figure 5.3: P_{fh} as function of λ_{rt}.

we present here comparisons between the obtained analytical predictions and numerical results derived by performing extensive computer simulations (SM) in the case of actual arrival and service time distributions under the assumption of an equal computation requests arrival rate and mean service time [246]. The system performance has been measured in terms of P_B and P_{fh}. We supposed ENs equipped on board of a maximum number of processors C equals to 14, and each computational request with a mean duration of 3 minutes. Furthermore, considering as reference mobility model that proposed in [99], it results $V/R = 1$ min^{-1}, and $T_{e,i}, T_{H,i}, T_{s,i}$ modeled as in [137]. With the aim at comparing the analytical predictions derived on the basis of the proposed M/M/C queuing system model with reneging with those related to an actual EC system, according to [179], the computation requests arrivals have been modeled according to a Normal distribution, with service time following a hyperexponential distribution as in [246]. Finally, it has been assumed that all the real time applications having an equal mean computation time of 0.3 minutes and target values for P_{fh} and P_B equal to 4×10^{-4} and 5×10^{-4}, respectively. All the numerical results presented here are referred to the mean values obtained from 1000 independent runs of simulations. It is important to stress that a very good agreement between the analytical predictions and simulation results is evident in all the Figures 5.2, 5.3, and 5.4. Finally, Figure 5.5 shows the minimum (i.e., op-

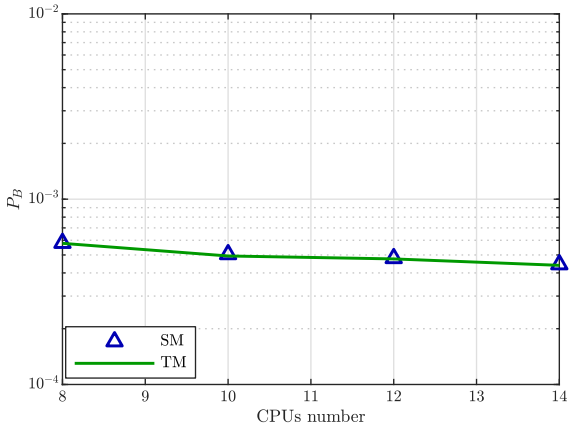

Figure 5.4: P_B as function of the number of CPUs

timum) number of CPUs as a function of the λ_{rt} required to meet different $P_{fh,target}$ and $P_{B,target}$ service constraints derived by solving the optimization problem formulated in (5.23)-(5.25). In the same figure the optimum C values derived by resorting to computer simulations by assuming the actual conditions previously introduced are also reported for comparison purposes. The good agreement is evident in Figure 5.5 and validates the effectiveness of the proposed analytical approach based on the definition of an equivalent M/M/C queuing system with reneging.

5.1.6 Conclusion

The performance analysis of an EC system in the case of real time computation requests and user mobility has been provided. A suitable handoff management scheme has been proposed and analyzed by resorting to the definition of a M/M/C queuing system model with reneging. Being the proposed analytical approach based on some simplified (i.e., Markovian) assumptions, its accuracy has been validated by providing comparisons with the simulation results derived by performing extensive runs in the case of actual service conditions. In particular, it has been demonstrated here that the proposed method represents an efficient tool to perform the design of an EC system for what concerns the minimum number of CPUs to be allocated

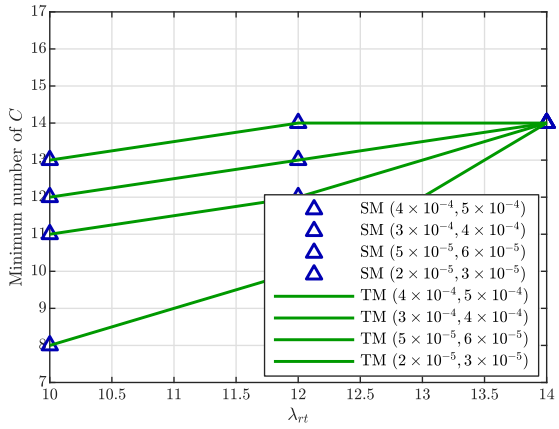

Figure 5.5: Minimum number of CPUs as a function of λ_{rt} for different ($P_{fh,target}$, $P_{B,target}$) service constraints

at each EN in order to meet specific service constraints.

5.2 Performance Analysis of a Delay Constrained Data Offloading Scheme in an Integrated Cloud-Fog-Edge Computing System

5.2.1 Motivation

During last decades, the cloud computing [24] architecture has held an undisputed dominant role in the network computing paradigms scenario, providing massive processing and storage capacity to the users. However, over the years, the large scale diffusion of devices always getting smarter, able to exchange data information to each other and with the surrounding environment, has led to a deep need to redesign the networks architectures. More in depth, the ever increasing presence of smart devices in our daily life has opened the doors to novel ubiquitous communication paradigms, such as the Internet of Things (IoT). The IoT expresses a wide reality consisting of heterogeneous smart devices generating big volume of data traffic and applications demand. Other typical aspects of the IoT may be represented by the

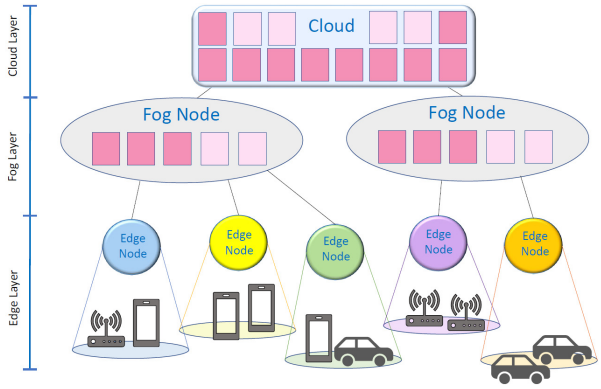

Figure 5.6: Edge-fog-cloud Architecture

mobility of the devices, which poses new challenges as concerns the seamless service continuity, or real-time execution constraints required for some classes of applications, such as augmented reality, health care applications or recognition assistance [72]. All these new challenges have triggered the tendency to migrate towards novel solutions, typically based on the deployment of computational nodes, lower in processing and storage supply, to the edges of the network, in order to reduce the network response latency [168]. The novel network paradigms based on this approach are named edge computing (EC) [166] and fog computing (FC), and are able to provide low latency response and service continuity to mobile users.

However, in comparison to the existing public cloud based solutions, i.e., Microsoft Azur and Amazon AWS, the processing capacity of EC solutions is strongly limited [69]. In order to address both the cloud and EC/FC issues, the functional integration of these two approaches in a same computing infrastructure has recently gained momentum [72]. This solution enables massive improvements in system performance and users quality of service (QoS), introducing higher levels of flexibility for rapid computations and high mobility patterns. In such a context, however, an additional issue has to be carefully considered, mainly in the service providers point of view, that is the pursuit of a proper system resources exploitation.

5.2.2 Contributions

Towards this direction, this study analyzes the performance of an integrated cloud-fog-edge computing infrastructure depicted in Figure 5.6. At the Edge layer we have sets of EC nodes (ECNs) able to offer computation to the underlying devices. Each set of ECNs is linked to a fog computing node (FCN) belonging to the Fog layer. Finally, all the FCNs are linked to the cloud [91, 246]. ECNs and FCNs are modeled as Markov queueing system with finite capacity with the entering of a computation request into the ECN or FCN ruled by the number of requests waiting for computation completion at its arrival instant, in relation to specific deadline constraints. Conversely, the cloud does not reject any computation request. As a consequence, we have resorted here to a Markov queueing system model with reneging, to take into account the premature departure of a request due to the expiration of the associated deadline.

Summarizing, the main contributions here are

- Queueing theory analysis of the considered integrated cloud-fog-edge computing infrastructure;
- Performance optimization on the basis of the social welfare metric [83, 261, 272], constrained on a target referred to the dropping probability, i.e., the probability of the occurrence of the deadline expiration;
- Validation of the proposed analytical model and obtained analytical predictions throughout comparisons with numerical results derived by performing extensive computer simulation runs under realistic world conditions.

5.2.3 Related Works

During years, queueing theory analysis has received many attentions within the field of the computational networks. As a consequence, several papers dealing with different aspects of these systems are available from the literature. In particular, in [115] the authors focus on the optimization of the number of processors in a real time computing system, where arrivals are bursty and divided into two priority job classes. The two priority service requests are managed by setting a fixed number of processors to serve the high priority queue, while the remaining servers are allocated on the basis of the waiting time on both the queues. The presence of two classes

of customers has been analyzed also in [15], in which a system M/M/1/K with weighted fair queueing discipline is studied. The conditions and performance of a computing system with finite input source is analyzed in [173], while the optimal number of servers in a computing system is a problem addressed in [173]. Furthermore, in [173] a M/M/C/m/m queueing system model is consider to derive the optimal number of processors through the fuzzy logic approach, aiming at minimizing the system maintenance cost. A novel method to assess the suitable number of servers in a queueing system with finite capacity is proposed in [23], on the basis of the level of customers satisfaction. In particular, the server optimization is performed here by considering three different metrics consisting of the system cost, its acceptability and the servers utilization rates.

Differently, the authors in [66] optimize the number of processors on the basis of a M/G/∞ system, focusing on the maximum exploitation of the production line, by considering a provider perspective. The study of the economic aspects of different versions of the computational systems have recently emerged in many papers. An example is represented by [272] where a dynamic control problem in an open Jackson network with limited capacity is formulated, in which the aim is to determine the suitable admission price to maximize the long therm social welfare system metric. Paper [193] formalizes the cloud provider maximization profit, in which both service charges and business costs are considered in the system optimization.

With the emergence of the new network paradigms, queueing theory has been extensively applied to provide stochastic traffic analysis of next generation networks. Paper [270] aims at ruling offloading considering an heterogeneous networks scenario. The paper models both the partial and the full offloading policies, via Wi-Fi and cellular networks, considering reneging and service interruptions. The main aim of the paper is the optimal tradeoff between energy efficiency and system performance, and the heterogeneous offloading interfaces are represented through on/off Markov chain models. The offloading problem is investigated also in paper [14] authors use anon/off alternating renewals process which is analyzed to derive transmission delay and offloading efficiency. Then, a model with balking is proposed taking into account the WLAN status, the number of packets waiting for transmission and the associated deadline. An alternating renewal process is also used in paper [170], in order to model the availability of the WiFi network for the offloading strategy. The paper proposes a complete theoretical queueing

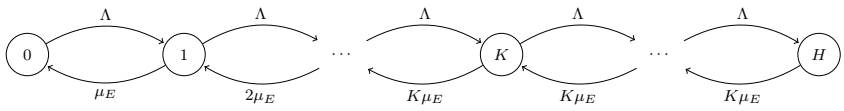

Figure 5.7: ECN Markov chain

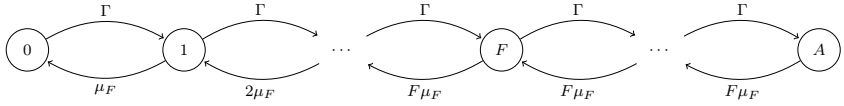

Figure 5.8: FCN Markov chain

analysis, and authors especially focus on a novel user patience metric. The customers impatience is analyzed also in paper [271], where a multi-server retrial queueing system is modeled, where customers may leave the system for balking or impatience. Furthermore, a more realistic customer behavior is formulated by using three parameters to weight probabilities. The offloading scheme in a hybrid cloud-fog computing system, for time critical application, is proposed in [230]. Paper [230] aims at minimizing the mobile devices power consumption, considering strict restrictions on the system response time. The paper [72] still considers a hybrid fog-cloud network as computational support in an IoT scenario. The paper addresses the healthcare services deployment, modeling each edge node as a M/M/C system, the public cloud with a M/M/∞ queue, and the private cloud with a M/M/c/K system. In [266] the mobile device resource sharing framework in a cloudlets network is proposed, to model offloading. The proposed framework consists of a M/M/c/K queue system, and the optimal average service time for cloudlet is provided.

5.2.4 System Model

We refer here to the three layers computing infrastructure depicted in Figure 5.6. The EC layer consists of a suitable number of ECNs, each of them located at a given Base Station (BS) of a high speed, high reliable, low latency fifth generation (5G) wireless network. Each ECN provides computation services to all mobile users within its service area, i.e., within the coverage area of the related BS of the 5G network. FCNs belonging to the

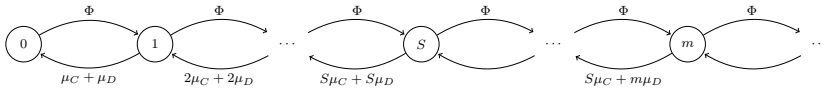

Figure 5.9: Cloud Markov chain

Fog layer, can be connected to a given number of ECNs by means of switched high speed links. FCNs provide tasks computation service to mobile users within the service area of the connected ECNs on the basis of the suitable procedure described later. Finally, we have the cloud layer where a cloud infrastructure connected by switched high speed links with all the FCNs of the Fog layer can provide computing service to all mobile users within all the ECN service areas on the basis of a suitable offloading procedure.

More in detail, we focus on a reference scenario where we have a set of tasks from mobile users requiring computation, and an integrated computing infrastructure composed of a set $\mathcal{E}$ of ECNs, several FCNs $\mathcal{F}$ belonging to the set $\mathcal{F}_\mathcal{C}$, and a cloud. Each FCN offers computation support to a subset of ECNs $\mathcal{E}_\mathcal{F}$ and, similarly, the remote cloud is devoted to provide support to $\mathcal{F}_\mathcal{C}$. Hereafter we refer to the service area of an ECN as the geographical area within which a task originated in that area can be offloaded on the corresponding ECN throughout wireless links with a negligible latency. As stated before, we assume ECNs located at the BS sites of the 5G cellular network which supports, in conjunction with a high speed low latency core network, mobile users communication connections with the integrated computing infrastructure under consideration. As mobile users move across different cells during the time needed to complete a task computation, a handover procedure can be performed between neighboring cells to guarantee a seamless connection with the integrated computing infrastructure.

In this work, due to the complexity and time consume of a task computation migration from one computation site to a neighboring one, we have considered here the communication plane (i.e., handover) separated by the computing plane (i.e., tasks computation). This means that, whenever a mobile user is no longer in the coverage of a given BS, it starts a handover procedure but its offloaded task continues to receive computation service from the original ECN, even if the mobile user is not directly linked to the original ECN anymore. This is so, because a mobile user can be linked with the remote ECN by means of any BS to which the mobile user is connected

that acts as a relay node.

The arrival processes of the task computation requests at the ECNs are assumed as independent and identically distributed Poisson processes with same mean arrival rate Λ.

At a first instance, a mobile user within a given EC service area submits a task computation request to the related ECN. This new arrived request is accepted if, on the basis of the task computation requests already in the ECN, it results, with suitable probability, that it can complete computation before expiration of its time deadline. Otherwise, the BS associated to the ECN redirects the task computation request to the linked FCN. Here again, the task computation request is accepted if, on the basis of the number of task computation requests already in the FCN, it is possible to guarantee with a suitable statistical uncertainty that its computation is completed within its time deadline. Differently, it is redirect through the core network to the cloud. Each new arrived task computation request at the cloud is accepted, hence, in this case, we can have task computations reneging due to the expiration of the tasks deadlines.

In performing our analysis we assume the task computation time at the ECNs, FCNs and cloud of the integrated computing infrastructure exponentially distributed with appropriate mean value (related to the different computation capabilities of each site). We will validate the goodness of this assumption by comparing the obtained analytical predictions with simulation results derived by assuming realistic world task computation time distributions. In addition to this, we consider that each task computation request has associated a deadline which expires after a time exponentially distributed [**?**, 140, 174], with mean value dependent on the specific computation site (EN, FN or cloud), in order to take into account the impact of the resulting different communication delays.

As a consequence, on the basis of our assumption, we have resorted to a M/M/K/H queueing system to model each ECN behavior, where K denotes the maximum number of task computations accepted by each ECN and H the number of available CPUs at each ECN. Similarly, the FCNs have been modeled as independent M/M/F/A queueing system. Finally, the cloud, being in this case the rejection of task computation request not allowed, has been modeled as a M/M/S queueing system. It is important to note that parameters K, H, F, A, S have to be derived on the basis of a suitable optimization approach in relation to specific QoS requirements in terms of

task reneging probability at the cloud site less than a target value. It is important to note that at both the ECNs and FCNs, the task computation completion is assumed guaranteed, since any new arrived task computation request is admitted if this is assured with a suitable probability, as detailed later.

Edge Subsystem Analysis

On the basis of our previous assumptions we have that each ECN belonging to $\mathcal{E}$ can be modeled as an independent M/M/K/H queue system where task computation requests are served in their order of arrival (FCFS scheduling) and with each of them requesting an exponential service time with mean value T_E. Moreover, in this case we have that the at most H task computation requests can be in the ECN at a time. This term is dependent on K in relation to a specific QoS requirement, i.e., H is the highest value for which the resulting probability $P_{EOUT}(K, H)$ that the task computation request entering the system while there are $H-1$ requests waiting for service completion does not complete its service before expiration of its deadline is less than a target value, $P_{EOUT,tg}$.

Hence, given K and P_{EOUT}, the maximum number H of task computation requests accepted by each ECN is derived under the assumption (worst case) that each task computation request accepted by ECN completes service within its deadline. It follows that the time needed to a task computation request arrived while in the ECN there are $H-1$ requests to complete its service, i.e., T, can be defined as the sum of $k = \left\lfloor \frac{H}{K} \right\rfloor + 1$ independent exponentially distributed random variables with mean values $\alpha = \frac{T_E}{K}$, and an independent exponentially distributed random variable with mean values $\beta = T_E$.

Therefore, the *pdf* of T can be obtained throughout the convolution of an Erlang distribution generating the random variable corresponding to the sum of the k independent exponential random variables previously introduced, and the exponential distribution with mean values β. After some algebraic manipulations the corresponding *pdf* results to be

$$f_T(t) = \frac{\alpha^k \beta e^{-\alpha t}}{(k-1)!} \int_0^t (t-\tau)^{(k-1)} e^{(\alpha-\beta)\tau} d\tau. \tag{5.26}$$

Hence, let E be the random variable exponentially distributed with mean value $\frac{1}{\mu_D}$ referred to the time deadline of the task computation request en-

tering the ECN system in the H-th position, we have that $P_{EOUT}(K,H)$ is

$$\begin{aligned} P_{EOUT} &= P\{T > E\} = 1 - P\{T \leq E\} \\ &= 1 - \int_0^\infty \left(\int_\tau^\infty \mu_D e^{-\mu_D \sigma} d\sigma \right) f_T(\tau) d\tau \\ &= 1 - \int_0^\infty e^{-\mu_D \tau} f_T(\tau) d\tau. \end{aligned} \tag{5.27}$$

Consequently, parameters K, H have to be defined in order to have

$$P_{EOUT}(K,H) \leq P_{EOUT,tg} \tag{5.28}$$

Once K, H have been defined to satisfy (5.28), any new task computation request arrived while the ECN is in state H is redirect to the linked FN for a successive consideration. By referring to the state diagram of the M/M/K/H queue system under consideration shown in Figure 5.7 for which we have

$$\lambda_n = \begin{cases} \Lambda, & 0 \leq n < H \\ 0, & otherwise \end{cases} \tag{5.29}$$

and

$$\mu_n = \begin{cases} n\mu_s, & 1 \leq n < K \\ K\mu_s, & K \leq n \leq H. \end{cases} \tag{5.30}$$

where $\mu_s = 1/T_s$ and the system state represented by the number n of task computation requests in the system.

Therefore, p_n,i.e., the probability of having n task computation requests in the ECN, for $1 \leq n < K$, results to be

$$p_n = p_{0,E} \frac{\Lambda^n}{n! \mu_s^n}, \tag{5.31}$$

while for $K < n \leq H$ is

$$p_n = p_{0,E} \frac{\Lambda^n}{K^{n-K} K! (\mu_s)^n}. \tag{5.32}$$

The term $p_{0,E}$, i.e., the probability of having no task computation requests in the ECN system, in (5.31), (5.32) can be derived by imposing the validating

of the state probability normalization condition $p_{0,E} + \sum_{n=1}^{\infty} p_n = 1$ as [246]:

$$p_{0,E} = \left[\sum_{n=0}^{K-1} \frac{\Lambda^n}{n! \mu_s^n} + \sum_{n=K}^{H} \frac{\Lambda^n}{K^{n-K} K! \mu_s^n} \right]^{-1}, \tag{5.33}$$

Moreover, its is easy to note that the task computation request blocking probability P_B equals to the probability to have the ECN system in state H, i.e., $P_B = p_H$.

Then, the mean time spent in the system by any task computation request results, through application of the Little's formula [247], results to be

$$T_{p,E} = \frac{\sum_{x=0}^{H} x p_x}{\Lambda(1 - p_H)}. \tag{5.34}$$

where $\Lambda(1 - p_H)$ is the mean task computation requests arrival rate at each ECN system.

Fog Subsystem Analysis

According to our assumptions, we have here again that the FCNs can be modeled as independent M/M/F/A queue systems where task computation requests are served in their order of arrival (FCFS scheduling) and each task computation request requires an exponential service time with mean value T_F. As for the ECNs, we have that at most A task computation requests can be in each FCN at a time with this term dependent on F in relation to the specific QoS requirement previously introduced, i.e., $P_{FOUT}(F, A) \leq P_{FOUT,tg}$. Also in this case, for given F and P_{FOUT} values, the number A of task computation requests accepted by each FCN, according to the considered worst case analysis, is the maximum value for which the following constraint is satisfied

$$P_{FOUT}(F, A) \leq P_{FOUT,tg} \tag{5.35}$$

where now $P_{FOUT}(F, A)$ is

$$\begin{aligned} P_{FOUT} &= P\{U > E\} = 1 - P\{U \leq E\} \\ &= 1 - \int_0^{\infty} e^{-\mu_D \tau} f_U(\tau) d\tau, \end{aligned} \tag{5.36}$$

where

$$f_U(t) = \frac{\pi^\theta \xi e^{-\pi t}}{(\theta-1)!} \int_0^t (t-\tau)^{(\theta-1)} e^{(\pi-\xi)\tau} d\tau, \tag{5.37}$$

in which, as in the edge subsystem case, we have $\pi = \frac{T_F}{F}$, $\xi = T_F$, and $\theta = \lfloor \frac{A}{F} \rfloor + 1$. In this case we have that any new task computation request arrived while the FCN is in state A is routed to the cloud. It is important to highlight that being each FCN linked with a number $\mathcal{Y}$ of ECNs, the task computation requests arrival process is Poisson with mean rate Γ, equals to $\mathcal{Y}\Lambda P_B$, resulting from the superposition of $\mathcal{Y}$ independent Poisson processes with equal mean rate ΛP_B [246]. The state diagram of the M/M/F/A queue system under consideration is shown in Figure 5.9. The system state is again represented by the number n of task computation requests in the FN and d queue system parameters given by

$$\lambda_f = \begin{cases} \Gamma, & \text{if } \exists \varepsilon \in \mathcal{E}_\mathcal{F} \text{ s.t. } n = H \\ 0, & otherwise \end{cases} \tag{5.38}$$

$$\mu_f = \begin{cases} f(\mu_F + \mu_D), & 1 \leq f < F \\ F\mu_F + f\mu_D, & l \geq A. \end{cases} \tag{5.39}$$

Hence, the state probability p_f of having f requests in the system, $1 \leq f < F$, is

$$p_f = p_{0,FN} \frac{\Gamma^f}{f!(\mu_F + \mu_D)^f}, \tag{5.40}$$

and, for $F \leq f < A$

$$p_f = p_{0,FN} \frac{\Gamma^f}{(F-1)!(\mu_F + \mu_D)^{(F-1)} \prod_{y=F}^{f} (y\mu_D + F\mu_F)}. \tag{5.41}$$

As before, the term $p_{0,FN}$, i.e., the probability of having no task computation requests in the considered FCN, in (5.40) and (5.41) can be derived by application of the state probabilities normalization condition as in (5.42), with the task computation requests blocking probability P_B, in this case, equals to the probability to have the considered FCN in state A, i.e., p_A.

$$p_{0,FN} = \left[\sum_{f=0}^{F-1} \frac{\Gamma^f}{f!(\mu_D + \mu_F)^f} + \sum_{f=F}^{A} \frac{\Gamma^f}{(F-1)!(\mu_D + \mu_F)^{F-1} \prod_{y=F}^{f} (y\mu_D + F\mu_F)} \right]^{-1}. \tag{5.42}$$

As before, the mean time spent by each task computation request in a FCN through the Little's formula is

$$T_{p,FN} = \frac{\sum_{x=0}^{\infty} x p_x}{\Gamma(1-p_A)}. \tag{5.43}$$

where $\Gamma(1-p_A)$ is the mean task computation requests arrival rate.

Cloud Subsystem Analysis

In the cloud case, we have no limitations on the accepted tasks computation requests. Hence, it follows that the cloud can be modeled as a M/M/S queue system, with FCFS selection policy and S available CPUs. In this case, we have that the cloud may be linked to a number of $\mathcal{F}$ FNs. So that, according to our assumptions, we have that the task requests arrival process is Poisson with mean rate Φ equal to $\mathcal{F}\Lambda P_A$. The associated system state diagram is provided in Figure 5.9 where the system state is considered as the number of task computation requests m in the cloud. In particular, we have

$$\lambda_m = \begin{cases} \Phi, & \text{if } \exists \eta \in \mathcal{F}_{\mathcal{C}} \text{ s.t. } f = A \\ 0, & otherwise \end{cases} \tag{5.44}$$

$$\mu_m = \begin{cases} m(\mu_C + \mu_D), & 1 \leq m < S \\ S\mu_C + m\mu_D, & m \geq S. \end{cases} \tag{5.45}$$

Consequently, the state probability p_m, that is the probability of having m tasks in the cloud subsystem, for $1 \leq m < S$, is

$$p_m = p_{0,C} \frac{\Phi^m}{m!(\mu_C + \mu_D)^m}, \tag{5.46}$$

while for $m \geq S$, we have (5.47).

$$p_m = p_{0,C} \frac{\Phi^m}{(S-1)!(\mu_D + \mu_C)^{S-1} \prod_{g=S}^{m} [g\mu_D + S\mu_C]}. \tag{5.47}$$

In(5.46) and (5.47) the term $p_{0,C}$, i.e., the probability of having no task computation requests in the cloud system, is given as in (5.48) by imposing

again the validating of the state probability normalization condition for this case.

$$p_{0,C} = \left[\sum_{m=0}^{S-1} \frac{\Phi^m}{m!(\mu_D+\mu_C)^m} + \sum_{m=S}^{\infty} \frac{\Phi^m}{(S-1)!(\mu_D+\mu_C)^{S-1}\prod_{g=S}^{m}[g\mu_D+S\mu_C]}\right]^{-1}. \tag{5.48}$$

Similarly to the previous analysis, the mean time spent by each task computation request in the cloud system is

$$T_{p,C} = \frac{\sum_{x=0}^{\infty} xp_x}{\Phi}. \tag{5.49}$$

Finally, in this case the reneging probability, i.e., the probability that a task computation request in the cloud does not complete its service due to its deadline expiration, P_D, can be obtained according to [102], as

$$P_D = T_{p,C}\mu_D. \tag{5.50}$$

5.2.5 Problem Formulation

The main aim of the study is to pursuit an optimization procedure in order derive the proper number of active processors allocated to each ECN, FCN and cloud, as well as the maximum number of computation requests accepted by each ECN and FCN, respectively. The optimization procedure is intended here as the maximization of the social welfare [83,261,272] function, through which both the users and provider points of view can be simultaneously taken into account, reaching a good trade-off between the parts involved in the proposed network infrastructure.

Furthermore, assuming Δ_{Eff} as the whole infrastructure rate of satisfied requests, the social welfare metric is defined as

$$\begin{aligned}\mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S') \\ = & U(\Delta_{Eff,E} + \Delta_{Eff,F} + \Delta_{Eff,C}) \\ & - V(T_{p,E} + T_{p,FN} + T_{p,C}) \\ & - \sum_{z=1}^{K} rb_z - \sum_{w=1}^{F} uc_w - \sum_{j=1}^{S} sd_j,\end{aligned} \tag{5.51}$$

where $\Delta_{Eff,E} = \Lambda(1 - p_H)$, $\Delta_{Eff,FN} = \Gamma(1 - p_a)$, represents the mean values of the computation requests arrival rates accepted by each ECN and FCN for which we have assumed guaranteed the service completion. Likewise, $\Delta_{Eff,C} = \Phi(1 - P_D)$ is the mean rate of computation requests completing service without deadline expiration at the cloud site. Furthermore, K', H', F', A', S' represents the system parameters values for the integrated cloud-fog-edge computation infrastructure, that have to satisfy the following architectural constraint:

$$K' \leq K, \quad H' \leq H, \quad F' \leq F, \quad A' \leq A, \quad S' \leq S. \tag{5.52}$$

In addition to this, in (5.51) we have that :

- U is the gain associated to the client service accomplishment while V is a penalty associated to each unit of time spent by a computation request in an ECN, FCN or cloud;
- Assuming the service provider operating cost associated to a CPU dependent on its location, i.e., ECN, FCN or cloud, r represents the service provider cost for CPU available at a ECN, u is the cost related to a CPU located at a FCN, while s is the cost associated to an active CPU at the cloud site;
- $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$ are binary vectors whose elements b_i, with $1 \leq i \leq K$, c_i with $1 \leq i \leq F$, and d_i $1 \leq i \leq S$ equal to 1 if at the ECN, FCN or cloud site, respectively, the i-th CPU is active, or 0 otherwise.

Hence, being our goal that of maximizing the social welfare metric defined in (5.51), we can formulate our problem as

$$\max_{\mathbf{b},\mathbf{c},\mathbf{d},K',H',F',A',S'} \mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S'), \tag{5.53}$$

s.t.

$$P_{Deadline} \leq P_{D,\ traget}, \tag{5.54}$$

$$K' \leq K, \tag{5.55}$$

$$H' \leq H, \tag{5.56}$$

$$F' \leq F, \tag{5.57}$$

$$A' \leq A, \tag{5.58}$$

$$\Phi < S\mu_C, \tag{5.59}$$

where

- Constraint (5.54) represents a QoS target constraint referred to each ECN, FCN and cloud site on the probability that a computation request does not complete its service due to its deadline expiration;
- Constraints (5.54)–(5.58) represent the architectural restricts of the integrated cloud-fog-edge computation infrastructure;
- Constraint (5.59) refers to the stability condition for the cloud subsystem, interpreted in its most strict form.

5.2.6 Proposed Heuristic

Due to the intrinsic difficulty of problem (5.53)-(5.54), an approximated heuristic is proposed to determine the suitable number of CPUs to be allocated at each computation site, i.e., ECNs, FCNs, and cloud, in addition to the maximum number of computation requests accepted at each ECN and FCN in order to guarantee a specific QoS in relation to the probability that a computation request does not complete its service at a given site due to its deadline expiration. The proposed social welfare maximization procedure is iterative and, for each computation site, acts as follow

1. Let S, F, K, H, and $A = 1$ be the maximum architectural capacity of the whole system. Start with $S' = 1$, $F' = 1$, $K' = 1$, $H' = 1$, and $A' = 1$, $\phi = 0$, $\xi = 0$, $\theta = 0$, $\pi = 0$, $\psi = 0$;
2. Compute P_D. If $P_D \leq P_{D,target}$ then terminate, otherwise evaluate
   (a) If $K' = K$, i.e., it cannot be incremented, set $\phi = 1$ and jump to b), otherwise evaluate $A_1 = \mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S')$ with S', $K' = K' + 1$, H', F', A';

(b) If $H' = H$, i.e., it cannot be incremented, set $\xi = 1$ and jump to c), otherwise evaluate $A_2 = \mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S')$ with S', K', $H' = H' + 1$, F', A';

(c) If $F' = F$, i.e., it cannot be incremented, set $\theta = 1$ and jump to d), otherwise evaluate $A_3 = \mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S')$ with S', K', H', $F' = F' + 1$, A';

(d) If $A' = A$, i.e., it cannot be incremented, set $\pi = 1$ and jump to e), otherwise evaluate $A_4 = \mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S')$ with S', K', H', F', and $A' = A' + 1$;

(e) If $S' = S$, i.e., it cannot be incremented, set $\psi = 1$ and jump to 3), otherwise evaluate $A_5 = \mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S')$ with $S' = S' + 1$, K', H', F', and A';

3. If $(\psi \wedge \xi \wedge \theta \wedge \pi \wedge \phi)$ is equal to 1 terminate, otherwise select the resource allocation among A_1, A_2, A_3, A_4, and A_5 which minimizes $\mathcal{F}(\Delta_{Eff}, \mathbf{b}, \mathbf{c}, \mathbf{d}, K', H', F', A', S')$;

4. Perform allocation in accordance with the previous step;

5. Repeat from step 2).

5.2.7 Numerical Results

In order to validate the effectiveness of the proposed analytical approach, this section provides comparisons between the obtained analytical predictions (AR) based on the assumed Markov queueing system models with numerical results (SR), derived by resorting to extensive computer simulation runs by considering actual arrival and service time distributions under the assumption of same mean values [246]. In performing our analysis we assume the cloud system equipped with a maximum number of $S = 16$ CPUs, each FCN with a maximum number F of CPUs equal to 13 maximum, and each ECN with a maximum of $K = 10$ CPUs. Furthermore, we refer to an edge layer composed of 6 ECNs and a fog layer with 3 FCNs.

According to [179], in order to test the validity of the proposed theoretical model, we have analyzed here, under the assumption of equal mean values, the case of computation requests arrivals modeled as a Normal distribution, with service time following a hyperexponential distribution as in [246]. Furthermore, we have assumed that applications require an equal

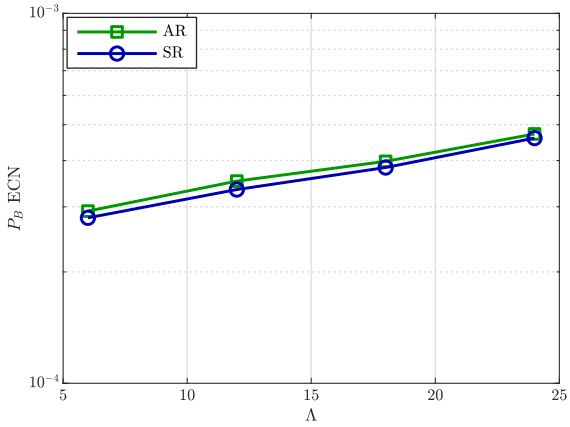

Figure 5.10: Computation requests blocking probability at each ECN as function of Λ

mean computation time at ecah computation site. in particular, in the cloud case, it results equal to 0.4 s, with a $\mu_C = 2.5\ \text{s}^{-1}$. Likewise, the mean requests computation time on each FCN is 0.2 s, with $\mu_F = 5\ \text{s}^{-1}$, while the mean requests computation time on each ECN has been assumed equals to 0.1 s, with $\mu_E = 10\ \text{s}^{-1}$. The system performance is measured in terms of computation requests completion failure probability at each ECN, FCN and cloud site, P_B limited to ECNs and FCNs, social welfare metric referred to the integrated cloud-fog-edge computing infrastructure under consideration.

All the numerical results presented here are referred to the mean values obtained from 2000 independent runs of simulations. Figure 5.10 depicts the behavior of P_B at an ECN as a function of Λ for H and K values derived in order to satisfy the constraint of a resulting $P_{EOUT}(K, H)$ less than 3.00×10^{-3}.

Likewise, Figure **??** shows the behavior of P_B at each FCN as a function of Λ with the system parameters A and F derived in order to satisfied the constraint $P_{FOUT}(F, A) \leq 3.00 \times 10^{-3}$. It is important to stress that in all previous figures a very good agreement between analytical predictions based on the considered Markov queueing system models with the simulation results derived under real world arrival and service time distributions [246].

The results related to the proposed heuristic based on the maximization

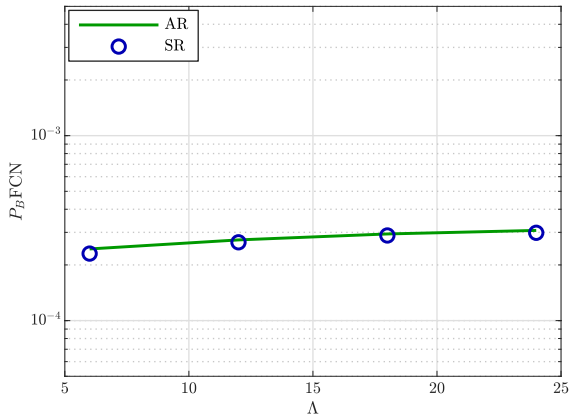

Figure 5.11: Computation requests blocking probability at each FCN as function of Λ.

of the social welfare metric, defined in (5.51) with $V = 4$, $U = 15$ per time unit, $r = 20$, $u = 16$, and $s = 12$, are provided in Figure 5.12 as a function of the parameter Λ. The values of the systems parameters, i.e., number of CPUs allocated at each ECN, FCN and cloud, maximum number of computing requests accepted by each ECN and FCN, that maximize (5.51), are shown in Figure 5.13 and Figure 5.14, respectively, as a function of Λ. Finally, in Figures 5.15-5.17 is shown the resulting computation requests completion failure probability at each ECN, FCN and cloud site, respectively, as a function of Λ. The assumed computation requests completion failure probability target value (i.e., 3.00×10^{-3}) for all the computation sites is given in Figure 5.15. In all these figures comparisons with the simulation results derived under proper assumptions for the computing requests arrival process and service time are also given in order to highlight the goodness of our Markov approach. Furthermore, it is important to note that the trends of H and A in Figure 5.13, in relation to Figure 5.14 and Figure 5.12, reveal that the size of H and A increases when the cost of the activation of new processors is higher in comparison to the increment of the size of H or A. Finally, we stress that, the ever-increasing trend of the social welfare function shown in Figure 5.12, demonstrates that the system parameters provided in Figure 5.13 and Figure 5.14 allow a good exploitation of system resources

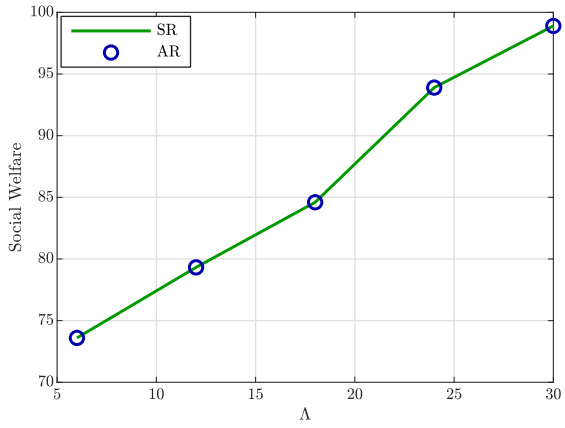

Figure 5.12: Social Welfare metric as function of Λ.

in reference to the set QoS target, on the basis of the service provision cost which characterized each subsystem.

5.2.8 Conclusions

This work has investigated the behavior of an integrated cloud-fog-edge computing infrastructure by resorting to the use of suitable Markov queueing models. A system design procedure based on an heuristic method for the maximization of a social welfare metric, conveniently defined, has been also proposed in order to derive the most appropriate values for the considered system parameters, i.e., number of CPUs to be allocated at each ECN, FCN and cloud, and maximum number of computation requests accepted by each ECN and FCN, in order to satisfy specific QoS requirements in terms of probability that a computation request does not complete its service at a given site, i.e., ECN, FCN or cloud, due to its deadline expiration less than specific target values. Finally, extensive simulations results have been provided in order to confirm the validity of the proposed theoretical approach, by exhibiting a strong closeness with the obtained analytical predictions derived under the assumption of more realistic computing requests arrivals and service time distributions.

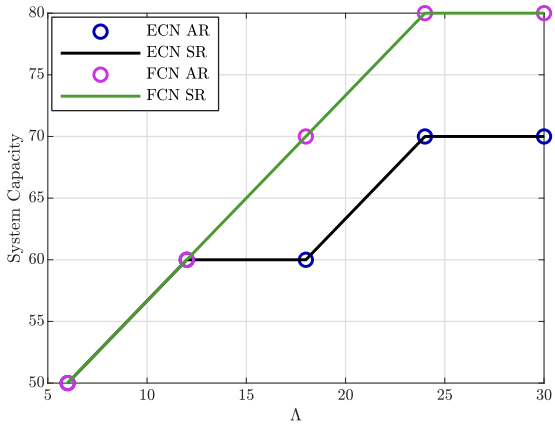

Figure 5.13: Maximum number of computation requests accepted by each ECN and FCN resulting by our heuristic as function of Λ.

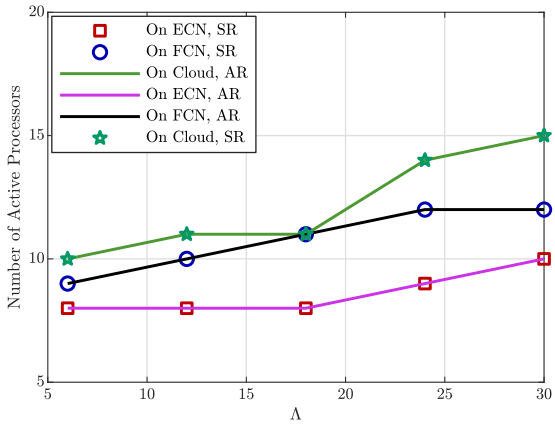

Figure 5.14: Number CPUs to be allocated at each ECN, FCN and cloud resulting by our heuristic as function of Λ.

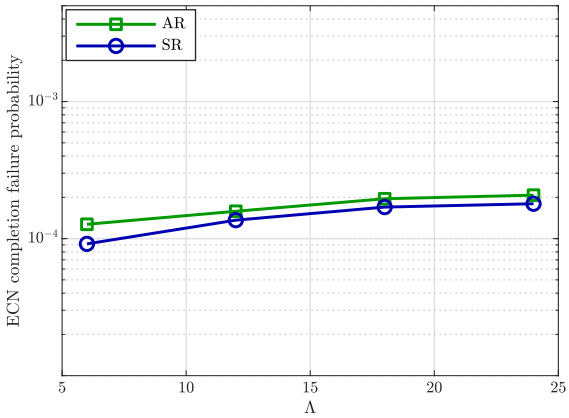

Figure 5.15: Computation requests completion failure probability at each ECN as function of Λ.

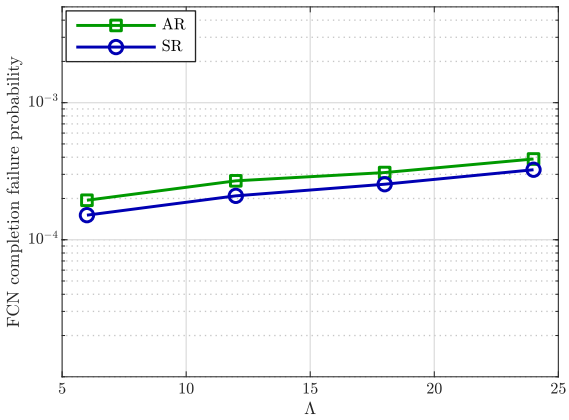

Figure 5.16: Computation requests completion failure probability at each FCN as function of Λ.

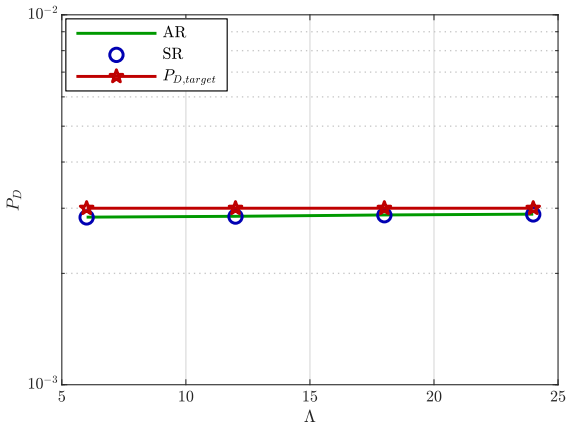

Figure 5.17: Computation requests completion failure probability at the cloud site as function of Λ.

Chapter 6

Chaos Theory as Predictive Tool

It is difficult to make predictions,
especially about the future.

Niels Bohr

6.1 Nonlinear Dynamic Chaos Theory Framework for Passenger Demand Forecasting in Smart City

6.1.1 Motivation

Recently, the ever increasing diffusion of vehicles has resulted in cities with remarkable levels of urban traffic, fatalities, injuries, and congestion [202, 283]. Within this context, the smart city concept has emerged, opening the doors towards several promising applications. In particular, an unprecedented chance to create a wide variety of new services has been provided [177, 202, 207, 283], involving the efficient data-collection and data-processing, and providing support to smart infrastructure, smart healthcare, smart governance, smart mobility, smart technology, etc. [177,202]. Furthermore, many research efforts have been made to design modern solutions to control mobility and traffic, especially in order to lower road congestion and

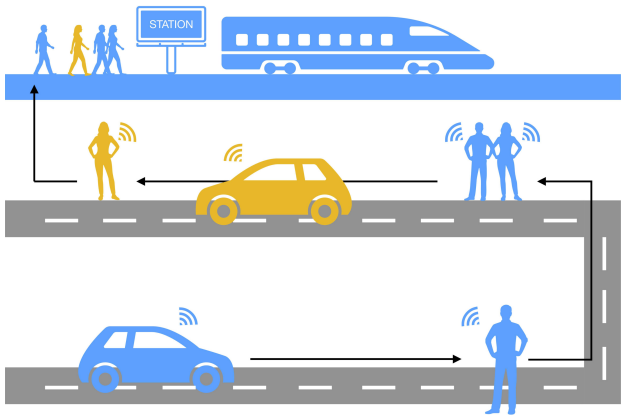

Figure 6.1: Ride-sharing service in a smart city

improve the transportation efficiency [116, 198, 202]. Therefore, the Intelligent Transportation Systems (ITS) has gained much attention, favoring the development of proper strategic solutions to reduce the drawbacks due to the growing spread of vehicles [202].

In particular, in order to limit the traffic congestion, one of the main objective is the reduction of the utilization of cars, by promoting the usage of public transportation during the rush hours [202]. Transportation network companies (TNC) such as Uber or Lyft offer peer-to-peer ride-sharing services to move people from and to homes and offices, or to the public transportation stations [202]. These types of services constitute a promising way to ensure to people a fast and comfortable solution to reduce the utilization of their own cars [202]. Within this context, by considering the TNC perspective, the accurate prediction of traffic demands plays a crucial role to properly allocate resources and, consequently, to avoid resource waste or delays on services provision [202]. For all these reasons, traffic forecasting and mobility forecasting have gained significant momentum in the area of ITS [202]. The application of forecasting procedures to the ITS [183,196,267] area focuses on the prediction of traffic conditions, a given number of hours ahead in the future [202].

In general terms, we refer to short-, medium-, and long-term forecasting. The first one involves predictions with horizons from few minutes up to few days ahead, and the second one deals with time horizons from few

days to few months ahead. Instead, the long-term forecasting is referred to predictions from quarters to years ahead [265]. Although medium and long-term forecasting generally involve the prediction about risk management and profitability planning, the short-term forecasting is frequently applied to traffic demands and mobility prediction because of its satisfactory accuracy [157, 265]. The short-term forecasting has been extensively studied in literature, and many different methods have been proposed [157, 180].

Roughly speaking, the whole family of the predictive methods can be divided into two different main branches: the approaches based on the classic time series analysis (TSA), and the techniques based on the deep learning. The former branch includes methods such as the Auto-Regressive (AR), the Auto-Regressive and Moving Average (ARMA), or the Auto-Regressive Integrated Moving Average (ARIMA) [37, 141, 181, 202], while the methods based on the latter approach are the strategies based on Artificial Neural Network (ANN), Support Vector Machine (SVM), Support Vector Regression (SVR) and so on [111, 155, 202, 205, 215]. TSA has been widely used for a very long time to solve a vast number of forecasting problems, by guaranteeing limited complexity and good performance on prediction accuracy. However, recently, the ever increasing complexity on the time series, due to the intrinsic complexity of the current problems, has lead to a performance reduction on accuracy applying the TSA. Therefore, despite TSA has provided good solution to many problems during past years, recently the high level complexity of the current problems scenarios have limited the applicability of the TSA due to its inability in performing valuable predictions on time series with complex behaviors. From the other side, the deep learning approaches reach very accurate results, but require a large amount of data to be trained, procedure that is extremely computationally expensive. Therefore, novel approaches to perform forecasting able to obtain accurate results with an acceptable complexity are essential. Within this situation, Chaos Theory (CT) has emerged as a powerful tool to perform nonlinear TSA [124, 157, 202].

6.1.2 Contributions

This application of the CT framework to the forecasting TSA has been proposed in [202] and here the corresponding results and analysis are presented. The prediction has been conducted on real data in Chengdu from Didi, a Chinese TNC, and two sets from Google dataset search, concerning the Uber

pickup requests in New York City and in Bangalore, respectively. More in depth, the main contributions of this work are:

- Validation of the chaotic behavior of the considered datasets, provided by the largest Lyapunov exponent analysis. The analysis consist of estimating the divergence rate of close trajectories associated to the scalar time series, during its evolution;

- Applications of the CT principles to design three different predictive algorithms for different datasets. In particular, the reconstruction of the phase space for each dataset has been pursuit, and proper forecasting algorithms proposed. Furthermore, the forecasting algorithms are based on local predictive mechanisms, and for the third dataset a hybrid approach is presented, combining both local and global approximations;

- Comparison of the proposed approaches with the well-known moving average (MA) model [41], that presented in [64], based on the CT as well, and the deep learning approach proposed in [46]. System performance has been provided in terms of mean squared error, mean absolute deviation and mean absolute percentage error.

6.1.3 Related Literature

The classical TSA methodologies are applied in paper [37,117,183,196,267]. Paper [183] proposes a short-term spatio-temporal forecasting approach to estimate the future taxi-passengers demand. The method aims at predicting the number of service requests that emerge at taxi ranks, by exploiting the real-time information exchanged among taxis. The paper combines both the predictive ARIMA and the time-varying Poisson models to realize the passengers demand prediction [37, 117]. Authors in [267] model the univariate vehicular traffic flow with the seasonal ARIMA, providing theoretical evidences about the suitability of this model in solving the short-term traffic conditions forecasting problems. Similarly, in [196], the study of traffic forecasting problem on large IEEE802.11 infrastructures is addressed. More in depth, authors in [196] evaluate the performance of many modified versions of the moving average and ARIMA algorithms, at different time scale, to forecast the access points load in wireless networks. Paper [196] highlights the importance of fine-grained prediction horizons and recent past data, to

obtain high levels of accuracy on the forecast values. Furthermore, many works based on machine learning approaches have been proposed. Examples are represented by paper [45, 89, 157]. Within the short-term traffic prediction, [157] combines the CT principles with the SVM, to improve the accuracy on the forecast values. In particular, authors in [157] adopt as measure of similarity the dynamic time warping to mitigate the negative effects of possible bursty points outside the neighborhood area of the processed point.

In [45] is provided a short-term passenger demand forecasting of light rail services. In particular, authors in [45] propose a novel neural networks model to fit non-stationary time series, aiming at minimizing the prediction error. Furthermore, the model formulated in [45] is based on the multi-layer perceptron one and the back-propagation algorithm is applied during the training process. Work [89] compares the forecasting performance applying both ANNs and Box Jenkins methods to airline passenger demand, calculated over the past five year daily data. An ANN strategy is also adopted in paper [9], in which a multilayer perceptron neural network is adopted to forecast the lightning occurrences. The recurrent neural networks are sequentially applied in paper [276], where the main objective is the real-time prediction of the taxi demand in the city of New York. Paper [46] proposes a forecasting framework based on the combination of the feed forward neural network and the long short term memory approach, to evaluate the electricity consumption. Differently, paper [145] predicts the urban traffic passengers flows by proposing a predictive structure based on convolutional neural networks and a graph representation of the traffic data, in order to extract the spatio-temporal information of the analyzed samples series. A spatio-temporal analysis has been provided also in [65], in which the demand for shared bicycles in three typical subway stations in the city of Beijing is predicted. The forecasting framework proposed in [65] is based on an improved version of the Xgboost method and the idea of sliding window. Within the CT approaches, in papers [64, 158], CT is applied to forecast the electricity price. Specifically, in [158] the electricity price is modeled as a multivariate time series, since it depends on many different factors. Then, authors in [158] provide the phase space reconstruction of the corresponding chaotic time series and find the forecasting function by fitting all points in the phase space, by applying the Elman model, which is one of the most common recurrent neural network. Paper [64] addresses the same problem

presented in [158] but, after phase space reconstruction, in order to improve the accuracy on the predicted results, authors use the add-weighted one-rank multi-steps prediction method [43]. Authors in [251] apply CT principles as well, to forecast the grids load data. Subsequently to the phase space reconstruction, paper [251] proposes the introduction of a weight on the largest Lyapunov exponent with the aim at mitigating the effect of significantly divergent trajectories, in order to reduce the forecasting error. This implies that during the prediction process, points close to the processed one with divergent trajectory will give a lower contribution on the prediction. Differently, paper [47] exploits CT to predict the urban daily water demand. In [47], the prediction of each point is made by considering the behavior, in the phase space, of only its nearest point. Another situation in which chaotic and nonlinear behaviors are widely present, is that of wind power generation. Within this context, the wind power exhibits fluctuations very difficult to predict.

Paper [165] deals with the system power forecasting, through the use of CT combined with the wavelet packet one. In [165] the wavelet packet theory is used to decompose the history of wind power data between high and low frequency components. Then, the frequency components are reconstructed with the single branch and the phase space is built for each single branch. Whether, during the process, the time series shows a non-chaotic behavior, prediction is performed by using back propagation neural network, otherwise through CT. The usage of both wavelet and CT has been also adopted in [144] to predict traffic in wireless sensor networks. This paper proposes the application of CT principles to both the high and low frequency parts of the original signal, and builds the predictive function considering the near points most influential than the far ones. Paper [125] addresses the forecasting of the load of power. In particular, it provided a short term prediction by combining CT with the fuzzy approach. In the method proposed by [125], closer the points in the phase space are to the value to be forecast, the greater is their impact on the predicted result. Finally, paper [29] aims at predicting a nonlinear time series for human actions and dynamic textures synthesis through a CT approach. Specifically, the phase space is reconstructed considering the corresponding multivariate time series and future predictions are made using a nonparametric data driven model, based on a kernel which is a decreasing function of the distance from the point that has to be predicted [186]. Then, the future multivariate time series values are

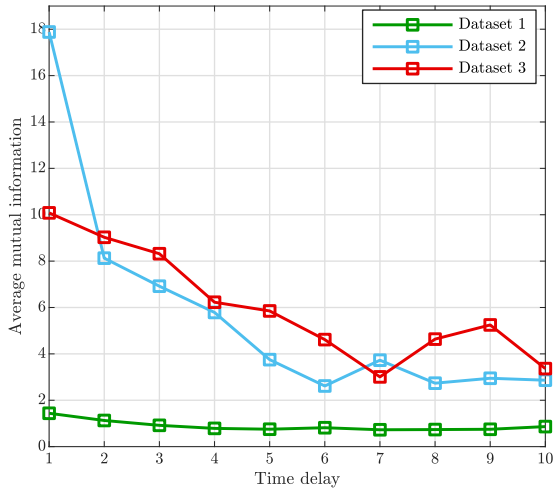

Figure 6.2: The optimal value of τ for Dataset 1, Dataset 2 and Dataset 3.

built by extracting the univariate time series from the reconstructed phase space.

6.1.4 Problem Statement

Given a scalar time series $\mathcal{X} = \{x_i\}_{i=1}^{N}$, x_i represents the value observed at time i and N is the whole number of samples of the time series $\mathcal{X}$. More in depth, $\mathcal{X}$ comes out from a previous sampling procedure in the city of interest. As regards the Didi dataset, the city of Chengdu has been modeled as a rectangle $\mathcal{R}$ of dimension $P \times Q$, where $\rho_{p,q}$, $p \in [0, P)$ and $q \in [0, Q)$, represents the region with coordinates p and q. A service request demand r_u is represented as a quintuple in the form $(r_{u,id}, r_{u,pc}, r_{u,pt}, r_{u,dc}, r_{h,dt})$, in which $r_{u,id}$ is the request ID, $r_{u,pc}$ the pick-up coordinates, $r_{u,pt}$ is the pick-up time, $r_{h,dc}$ identifies the coordinates of the destination, and $r_{u,dt}$ the time of arrival at destination. Our analysis has been conducted by grouping all the requests based on the pick-up time. Therefore, time has been partitioned into equal slots starting from 0. Hence, the k-th time slot is identified by the interval $[k \times S, (k + 1) \times S)$, where S is the time span of the interval.

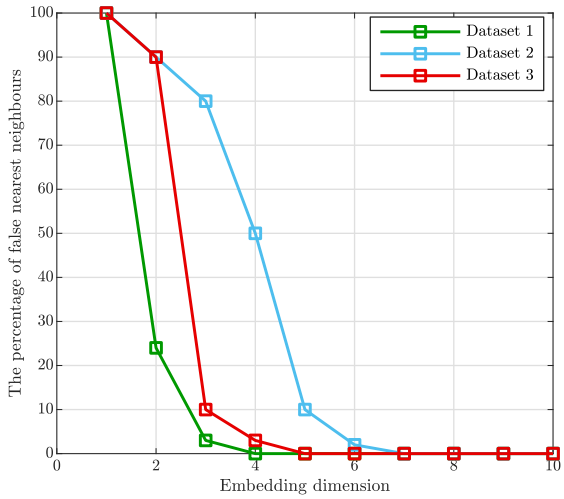

Figure 6.3: The optimal value of m for Dataset 1, Dataset 2 and Dataset 3.

Consequently, the passenger demand at the interval k, i.e., x_k, is given by

$$x_k = |\{u \in [0, \mathcal{N}) : r_{u,pc} \in \mathcal{R} \wedge r_{h,pt} \in [k \times S, (k+1) \times S)\}|, \tag{6.1}$$

where $\mathcal{N}$ is the total number of received requests and $|\cdot|$ means the number of elements belonging to the set. The other two sets have been taken on Google dataset search, and each request of the first of them is identified by the triplets (π_d, π_t, π_a), where π_d is the pick-up date, π_t the pick-up time and π_a the pick-up address. Differently, each request of the last dataset is composed of six fields, i.e, $(\mu_r, \mu_p, \mu_d, \mu_s, \mu_u, \mu_o)$, where μ_r is the request identifier number, μ_p represents the pick-up point, μ_d the driver number, μ_s the status of the ride (completed or ongoing), and μ_u and μ_o are the pick-up time and the drop off time, respectively. Given the time series $\mathcal{X}$, forecasting problem involves the prediction about the future behavior, δ steps ahead in the future. This study deals with the short-term forecasting, aiming at minimizing the forecasting error that, generally speaking, is a measure of the gap between the predicted and the real value of the time series analyzed. Despite there exist many different metrics to evaluate the forecasting error [180], here the minimization of the mean squared error

(MSE) defined as

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\hat{x}_{i+\delta} - x_{i+\delta})^2, \tag{6.2}$$

has been considered, where M represents the number of the samples in test data, and $\hat{x}_{i+\delta}$ and $x_{i+\delta}$ are the actual and the predicted values at time $i+\delta$. In addition, in order to provide an exhaustive analysis, we have also considered the mean percent error (MAPE) defined as

$$MAPE = \frac{1}{M}\sum_{i=1}^{M}\left|\frac{\hat{x}_{i+\delta} - x_{i+\delta}}{x_{i+\delta}}\right| \cdot 100, \tag{6.3}$$

and the mean absolute deviation (MAD) given by

$$MAD = \frac{1}{M}\sum_{i=1}^{M}|\hat{x}_{i+\delta} - x_{i+\delta}|. \tag{6.4}$$

It is important to note that metrics (6.2) and (6.4) highlight the variability of the forecasting error, while (6.3) expresses the error in terms of percentage on the actual data.

In the following section, in order to clarify the motivation behind the insight of the proposed forecasting algorithms, the technical background about CT principles is provided and the phase space reconstruction procedure explained.

$$F_{fnn}(T) = \frac{\sum_{i=1}^{N-m-1}\Theta\left(\frac{|\mathbf{z}_i^{(m+1)} - \mathbf{z}_j^{(m+1)}|}{|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}|} - T\right)\Theta(\frac{\sigma}{T} - |\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}|)}{\sum_{i=1}^{N-m-1}\Theta(\frac{\sigma}{T} - |\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}|)}; \tag{6.5}$$

6.1.5 Chaos Theory Approach

The class of the chaotic nonlinear dynamical systems includes the nonlinear dynamical systems whose behavior is unpredictable on the long term, and exhibit strong sensitivity to the initial conditions, that implies small changes in the initial state result in significant differences on the final states [19,192].

Generally speaking, the nonlinear dynamical systems with chaotic behavior present apparent irregular trend which hides the deterministic features of these systems. A system state is specified by a m dimensional vector $\mathbf{z}$, while the system dynamics can be expressed by [124]

$$\mathbf{z}_{n+1} = \mathcal{F}(\mathbf{z}_n), \tag{6.6}$$

in which $\mathbf{F}$ is a m dimensional map. It is important to highlight the relation between equation (6.6) and the set $\mathcal{X}$. Indeed, the elements of $\mathcal{X}$ are a sequence of scalar measurements, therefore, the conversion of the observations into state vectors has to be performed. In order to execute such conversion, the *phase space* reconstruction procedure [29,239] has been conducted. After the phase space reconstruction procedure, the important features of chaotic time series can be caught by analyzing the underlying dynamics and the geometrical structure of its corresponding *attractor*, i.e., the set of values to which the system tends to evolve.

For the sake of simplicity, hereafter we refer to the dataset derived from Didi as Dataset 1, Dataset 2 for the dataset of the Uber pickups in New York City, and finally Dataset 3 for the Uber service requests in Bangalore.

6.1.6 Phase Space Reconstruction

Given the chaotic scalar time series $\mathcal{X}$, the first step towards the comprehension of its behavior is the reconstruction of the phase space (PSR) associated with it, in order to pursuit the analysis of its strange attractor. Due to the Takens' delay embedding theorem, which affirms the existence of a map between the real phase space and its reconstructed version [29], the PSR is provided by associating, to each element x_i in $\mathcal{X}$, a vector in the form

$$\mathbf{z}_i = [x_i, x_{i+\tau}, \ldots, x_{i+(m-1)\tau}], \tag{6.7}$$

where τ is the *time delay* and m is the *embedding dimension*. As detailed in Section 6.1.6 and Section 6.1.6, a proper choice of the values of τ and m is crucial to pursuit a suitable PSR.

Time Delay Estimation

As it is evident from (6.7), m and τ strongly impact the transformation of the scalar time series to the phase space vectors represented by $\mathbf{z}_i$. According to this, time delay τ rules the temporal distance between two suc-

cessive points in the phase space vector, i.e., $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$, and its optimal value minimizes the redundancy between points $x_{i+\tau}$ and x_i, maximizing the knowledge about $x_{i+\tau}$ from x_i. This analysis is conducted by computation of the exact value of delay τ with the approach based on the evaluation of the amount of mutual information between pairs of points $x_{i+\tau}$ and x_i, with τ that minimizes the mutual information between observations [82].

In practical terms, for each measured data, the histogram of the probability distribution of the data is created. Then, by varying time the delay τ, the mutual information x_i and $x_{i+\tau}$ results expressed by [159]

$$I(\tau) = \sum_{f,j} p_{f,j}(\tau) \log p_{f,j} - 2 \sum_{f} \log p_f, \tag{6.8}$$

where p_f is the probability that x_i is in the f-th bin of the histogram, while $p_{f,j}$ is the probability that x_i and $x_{i+\tau}$ fall in the f-th and j-th bin of the histogram, respectively.

As depicted in Figure 6.2, the values of τ for Dataset 1, Dataset 2 and Dataset 3 are $\tau = 5, 6$ and 7, respectively. Indeed, Figure 6.2, shows the plot of the mutual information I when τ changes. In order to select the proper value for time delay τ, i.e., the minimum time delay, the optimal value of τ is in correspondence of the first local minimum of the I function.

Embedding Dimension Estimation

As in the case of τ, to choose a proper value of m is very important to observe the determinism underlying the system associated to time series $\mathcal{X}$, with the least computational effort. In this case, the false nearest neighbors method is adopted [126]. The idea behind this method is that, since the presence of chaos can induce an exponential divergence on the trajectories of two nearby points, there exist some points in the data set that are neighbors in the embedding space but for which their temporal evolution exponentially diverges. Hence, this method consists, for each point in $\mathcal{X}$, in finding its nearest neighbor in m dimension. Then, the ratio between their distance in $m + 1$ dimensions and m dimensions is calculated. Finally, whether the resulted ratio is greater than a fixed threshold r, the neighbor is considered false.

Hence, given a fixed threshold T, the false nearest neighbors function can be defined as in (6.5) on the top of the previous page. in which σ is the standard deviation of the data, j is index of the nearest point, and Θ is the

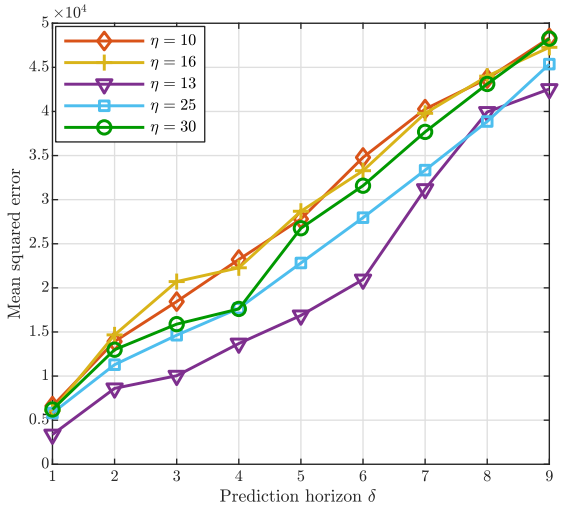

Figure 6.4: The predicted values by varying the number of considered neighbors for Dataset 1.

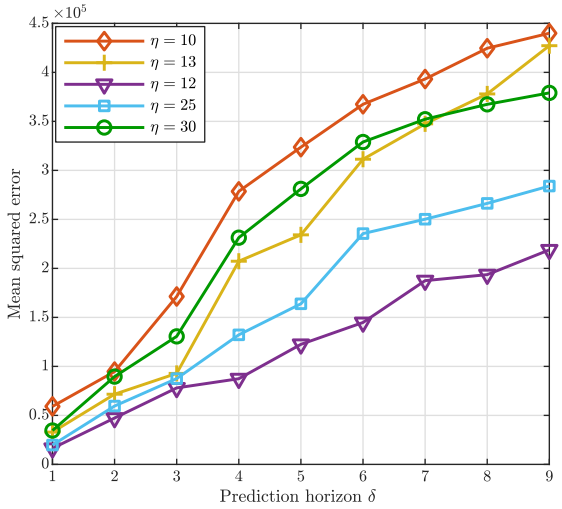

Figure 6.5: The predicted values by varying the number of considered neighbors for Dataset 2.

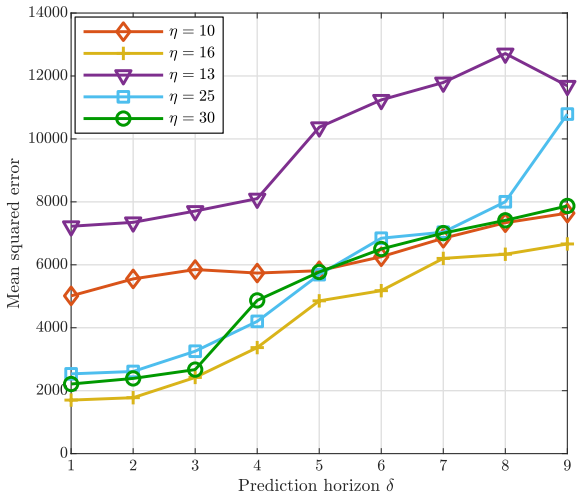

Figure 6.6: The predicted values by varying the number of considered neighbors for Dataset 3.

Heaviside step function given by

$$\Theta(\nu) = \begin{cases} 0 & \nu < 0; \\ 1 & \nu \geq 0. \end{cases} \tag{6.9}$$

Figure 6.3 shows the F_{fnn} function by varying the embedding dimension m, and the proper value of m is $m = 3, 6$ and 4, respectively. Indeed, for each set of data, the right value of m is the value for which is minimum the number of false nearest neighbors, hence, graphically, it is in correspondence of the last m value before the plot of F_{fnn} drops to zero.

Largest Lyapunov Exponent

Once the PSR has been pursued, in order to verify the chaotic behavior of $\mathcal{X}$, we analyze the largest Lyapunov exponent. There exist many approaches to check the presence of chaos in a time series [64], and the study of the largest Lyapunov exponent is one of the most used. The main idea behind such technique is the study of the distance of two close vectors in the phase space over the time transition [184]. Hence, considering two trajectories $\mathbf{y}$ and $\mathbf{x}$,

i.e., solutions of (6.6), close in the state space, the evolution of their mutual distance is given by

$$\mathbf{y}_{n+1} - \mathbf{x}_{n+1} = \mathbf{J}_n(\mathbf{y}_n - \mathbf{x}_n) + O(||\mathbf{y}_n - \mathbf{x}_n||^2), \tag{6.10}$$

where $\mathbf{J}_n$ is the $m \times m$ Jacobian matrix of $\mathbf{F}$. Then, supposing Λ_i the eigenvalue of $\mathbf{J}$, the Lyapunov exponents are given by [124]

$$\lambda_i = \lim_{N\to\infty} \frac{1}{2N} ln|\Lambda_i^{(N)}|. \tag{6.11}$$

Hence, the study of the largest Lyapunov exponent aims at evaluating the sign of the value of the largest Lyapunov exponent, that represents the rate of separation of close trajectories in the phase space. Since the positive value of such divergence rate is a strong signature of the presence of chaos, it represents a suitable criterion for establishing the chaotic nature of a time series. In this respect, here the largest Lyapunov exponent has been computed by applying the Rosenstein method [238], which is based on the estimation of the local divergence rates of trajectories over the whole data set in the phase space. Specifically, the local divergence is estimated on the neighborhood of each point of $\mathcal{X}$ in the phase space. The general idea of the method is the measurement, for each x_i in $\mathcal{X}$, of the expansion rate in a particular time span δ of the trajectories. In particular, the expansion rate for time span δ is given by

$$\mathcal{E}(\delta) = \frac{1}{t}\sum_{i=1}^{N} \ln\left(\frac{1}{|\mathcal{N}(\mathbf{z}_i)|}\sum_{\mathbf{z}_z\in\mathcal{N}_\epsilon(\mathbf{z}_i)} |\mathbf{z}_{i+\delta} - \mathbf{z}_{z+\delta}|\right); \tag{6.12}$$

where $\mathcal{N}(\mathbf{z}_i)$ is the neighborhood of point $\mathbf{z}_i$ in the phase space. More in depth, $\mathcal{N}_\epsilon(\mathbf{z}_i)$ derived from the selection of vectors $\mathbf{z}_z$ in m dimension, closer than a given value ϵ in the max norm. Hence, in order to define the neighborhood of $\mathbf{z}_i$, it is necessary to determine the indices z for which

$$||\mathbf{z}_z - \mathbf{z}_i|| \leq \epsilon. \tag{6.13}$$

The details of algorithm are reported in [238] and in our case, the resulted largest Lyapunov for all the three dataset analyzed has been reported in Table 6.1. Since the largest Lyapunov exponents are a real number greater than zero, the considered time series indeed exhibit a chaotic behavior [124, 158], which justify the validation of our proposed chaotic framework.

Table 6.1: Largest Lyapunov Exponent

Dataset	Lyapunov Exponent
Dataset 1	0.7
Dataset 2	1.6
Dataset 3	0.3

6.1.7 Time Series Forecasting

The general idea behind the prediction of the behavior of $\mathcal{X}$, is the approximation of the map $\mathcal{F}$ in (6.6).

In order to predict the future behaviors of Dataset 1, Dataset 2, Dataset 3, we apply the PSR and the suitable values for time delay τ and embedding dimension m have been calculated in accordance with Sections 6.1.5 and 6.1.6. In general term, there exist two main approaches to address the approximation problem of $\mathcal{F}$ function in (6.6): the local and the global approximation approach. One of the most used local prediction method is the neighbors based prediction, that evaluates the future behavior of the points belonging to a neighborhood around the point which has to be predicted. Then, the resulted forecast value is given by the average of the values of the neighbors points. Both the algorithms proposed for Dataset 1 and Dataset 2 constitute two general improved versions of the classical neighbors based prediction approach, in which each term is properly weighted. Despite different dataset may require different weight definitions, typically, the attribution of high weights to points close to that needing prediction, or to those that exhibit a high degree of similarity with the point to be forecast, could result in accuracy of prediction, compared with the standard version of the modified method. One strength point of local methods is that it does not involve any model for $\mathcal{F}$, while, as better explained later, the determination of the suitable number of neighbors to consider for the prediction is not a trivial issue. Furthermore, when determinism is weak on the dataset or for long term predictions, local methods result ineffective, since they tend to reproduce past trends. Differently, the algorithm proposed for Dataset 3 also considers the global approach. Roughly speaking, the global approaches constitute a more advanced technique compared to the local methods, but they often lead to more difficult problem management since modeling $\mathcal{F}$ implies the determination of many parameters.

Summarizing, as concerns the first two datasets, we propose two pre-

diction algorithms based on a local approximation of $\mathcal{F}$. Differently, the prediction procedure for Dataset 3 constitutes a hybrid approach between the local and global approximation. In order to predict the value δ steps ahead of x_i, i.e. $x_{i+\delta}$, with a local approximation of $\mathcal{F}$, we consider the neighborhood around $\mathbf{z}_i$ in the phase space, of radius ϵ, i.e., $\mathcal{N}_\epsilon(\mathbf{z}_i)$.

6.1.8 Forcasting Algorithm for Dataset 1

The algorithm that we propose to predict the future behavior of Dataset 1, i.e., the number of Didi requests collected in a given hour in the city of Chengdu, is based on the evaluation, for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, of its trajectory and its value at time $j+\delta$, hence $x_{j+\delta}$. Then, the weighted mean value over all the future behaviors of the points belonging to $\mathcal{N}_\epsilon(\mathbf{z}_i)$ is computed, and the weight β_j is defined as

$$\beta_j = \frac{1}{|x_{j+\delta} - x_i|}. \tag{6.14}$$

As it is straightforward to note, the higher is the similarity between x_j and $x_{j+\delta}$, the higher is the value of β_j. Consequently, the predicted value of $x_{i+\delta}$, i.e., $\hat{x}_{i+\delta}$, being η the number of points in $\mathcal{N}_\epsilon(\mathbf{z}_i)$, is given by

$$\hat{x}_{i+\delta} = \frac{1}{\eta} \sum_{\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)} \beta_i x_{i+\delta}. \tag{6.15}$$

Hence, the whole prediction procedure for Dataset 1 can be summarized as follows

- compute the optimal value for time delay τ according to Section 6.1.6;
- compute the optimal value for embedding dimension m according to Section 6.1.6;
- build $\mathcal{N}_\epsilon(\mathbf{z}_i)$;
- for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, measure $x_{j+\delta}$;
- for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, calculate β_j;
- determine $\hat{x}_{i+\delta}$ in accordance with (6.15).

6.1.9 Forcasting Algorithm for Dataset 2

In order to predict the trend of Dataset 2, hence the number of total Uber pickups requests for a given hour in the city of New York, we define the following weight based on the similarity between the point object of prediction x_i and the neighbor point x_j. Hence, γ_j is defined as

$$\gamma_j = \frac{1}{|x_i - x_j|}. \tag{6.16}$$

Consequently, the prediction formula is expressed by

$$\hat{x}_{i+\delta} = \frac{1}{\eta} \sum_{\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)} \gamma_i x_{i+\delta}. \tag{6.17}$$

The whole prediction procedure can be summarized in the following

- compute the optimal values for time delay τ and embedding dimension m according to Section 6.1.6 and Section 6.1.6, respectively;
- build $\mathcal{N}_\epsilon(\mathbf{z}_i)$;
- for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, measure $x_{j+\delta}$;
- for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, calculate γ_j as reported in (6.16) ;
- determine $\hat{x}_{i+\delta}$ in accordance with (6.17).

6.1.10 Forcasting Algorithm for Dataset 3

This algorithm aims at predicting the behavior of Dataset 3, hence the number of total uber service requests in Banglore in a given hour. The algorithm prediction for Dataset 3 is a hybrid approach between the local approximation and the global one. Specifically, in order to improve the accuracy about the very short term forecasting, we provide a global nonlinear approximation of $\mathcal{F}$ given by the radial basis function [124] defined as follows

$$\mathbf{z}_{i+1} = \mathbf{F}(\mathbf{z}_i) = \theta_0 + \sum_{w=1}^{g} \theta_w \phi(|\mathbf{z}_i - \zeta_w|), \tag{6.18}$$

where ζ_w are the g centers of the attractor, ζ_0 and ζ_w are coefficients, and ϕ is the Lorentzian function expressed by

$$\phi(\psi) = \frac{1}{[1 + (\frac{r}{a})^2]}, \tag{6.19}$$

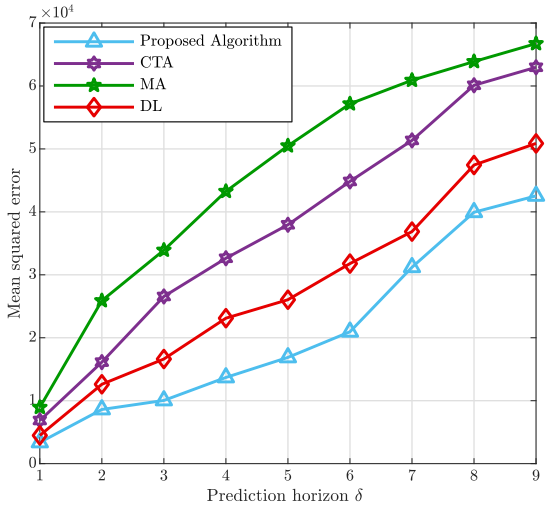

Figure 6.7: Forecasting error expressed in terms of mean squared error for Dataset 1.

with a constant. The joint utilization of both the approximation models presented in (6.17) and (6.18) has been pursued and, while the first approach ensures a lower long-term forecasting error, the second one improves the performance of the short-term forecasting. The prediction procedure acts as follows

- compute the most suitable value for time delay τ according to Section 6.1.6;
- compute the most suitable value for embedding dimension m according to Section 6.1.6;
- build the approximation of $\mathcal{F}$ according to (6.18);
- build $\mathcal{N}_\epsilon(\mathbf{z}_i)$;
- for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, measure x_{j+1};
- for each $\mathbf{z}_j \in \mathcal{N}_\epsilon(\mathbf{z}_i)$, calculate γ_j;

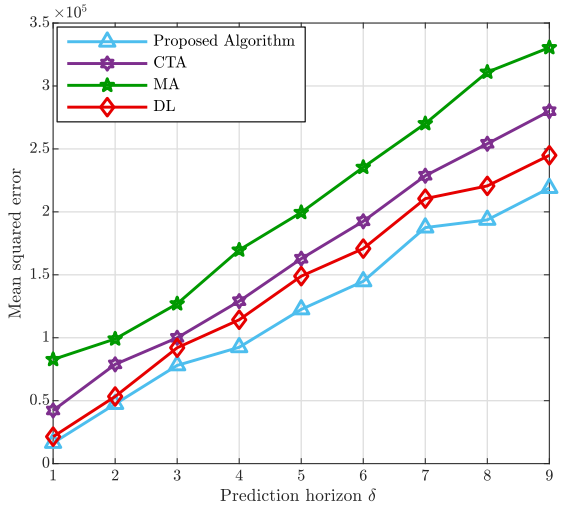

Figure 6.8: Forecasting error expressed in terms of mean squared error for Dataset 2.

- assign to $\hat{x}_{i+1}$ the most accurate value between the values obtained from the application of the local and the global approximation.

6.1.11 Practical Considerations

Since the value of ϵ directly impacts on the number of considered neighbors, the choice of such value crucially affects the prediction performance. Due to the fact that there not exists an exact method to determine the optimal value for ϵ [64], in Figure 6.4, Figure 6.5 and Figure 6.6 is reported the predictive performance of the algorithms designed for the three datasets. The better behavior is evident considering a number of neighbors $\eta = 13$ and $\eta = 12$ for Dataset 1 and Dataset 2, respectively, while for Dataset 3 the suitable value is $\eta = 16$. The time complexity is approximately the same for all the three algorithms previously presented. Let $\mathcal{X}$ be the time series composed of N scalar values, the estimation of the optimal value of τ requires a computational complexity in the order of $\mathcal{O}(N \cdot \log N)$, while by applying the procedure to find the suitable embedding dimension m the maximum amount of time taken is $\mathcal{O}(N^2 + 4 \cdot \xi \cdot t + \xi^2 \cdot N)$, where ξ is the number of

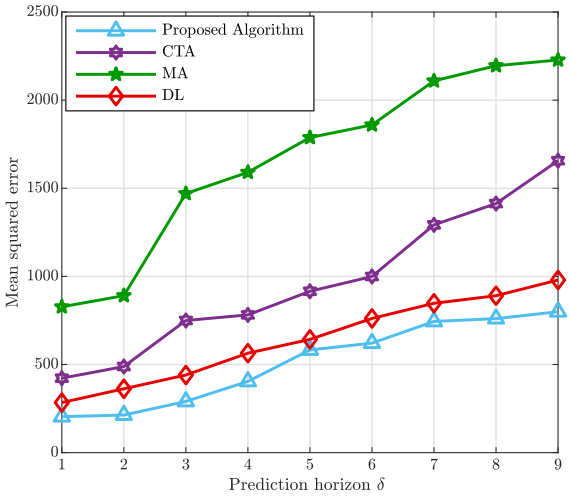

Figure 6.9: Forecasting error expressed in terms of mean squared error for Dataset 3.

Table 6.2: Order of Magnitude of Complexity

Time delay estimation	$\mathcal{O}(N \cdot \log N)$
Embedding Dimension	$\mathcal{O}(N^2 + 4 \cdot \xi \cdot t + \xi^2 \cdot N)$
Phase Space Reconstruction	$\mathcal{O}(\lambda \cdot m)$
Nearest neighbors procedure	$\mathcal{O}(N)$

considered m values. Furthermore, the phase space reconstruction exhibits a computational complexity of $\mathcal{O}(\lambda \cdot m)$, where $\lambda = N - (m-1)\tau$. Finally, the nearest neighbors procedure has a complexity in the order of $\mathcal{O}(N)$. Hence,

we can conclude that the overall time complexity is

$$\mathcal{O}(N \cdot \log N) + \mathcal{O}(N^2 + 4 \cdot \xi \cdot N + \xi^2 \cdot N) + \mathcal{O}(\lambda \cdot m) + \mathcal{O}(N). \quad (6.20)$$

The order of magnitude of the forecasting based on CT strategy steps are reported in Table 6.2.

6.1.12 Numerical Simulations

This work takes into account three sets of data related to the nonlinear time series represented by Datatset 1 derives from sampled data collected in the city of Chengdu, from Didi Chuxing, the biggest TNC in China. The dataset contains the passengers requests of one month, from 11/01/2016 to 11/30/2016, and the whole dataset contains more than 6.11 million of passengers requests. The whole area has been divided into 20×20 same-size grids. Every grid is a square with sides equal to 700 meters, and the longitude of the focus area is from 30.60E to 30.73E, the latitude is from 104.00N to 104.15N, while the considered surface is about $207.35\,\mathrm{km}^2$. Furthermore, Dataset 2 and Dataset 3 have been retrieved from *Google Dataset Search* where they are named as *Uber pickups in New York City* and *Uber request data* respectively. Dataset 2 derives from a sampling period from April to September 2014, while Dataset 3 is referred to Uber service requests data in Bangalore from 11/07/2016 to 15/07/2016.

In order to provide an exhaustive analysis, we compare our methods with the well known moving average (MA) model [41], with a complexity linear in the number of the samples considered to provide forecasting, and with the CT based approach proposed in [64] (CTA) whose complexity is in the order of magnitude of $\mathcal{O}(N^2)$. Furthermore, we also propose performance comparison with the deep neural network method (DL) designed in [46]. Furthermore, all the approaches have been applied by using 3 days of samples to forecast 5 and 9 hours ahead.

Figures 6.13, 6.14 and 6.15 show the performance comparison among the proposed algorithm, the CTA, the MA, and the DL models in terms of MSE. Despite all the four predictive approaches get worse as the prediction horizon increases, it is clearly evident as the proposed forecasting methods reach better results in all the three application dataset, by considering the same number of training days for all the methodologies applied. In fact, each algorithm guarantees a higher accuracy respect to the CTA, the MA, and the

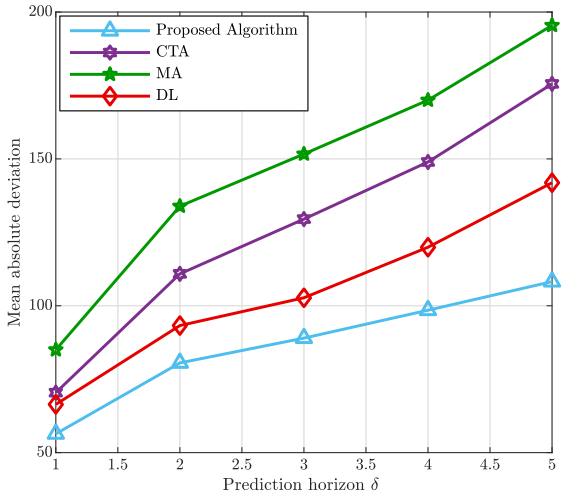

Figure 6.10: Mean absolute deviation for Dataset 1.

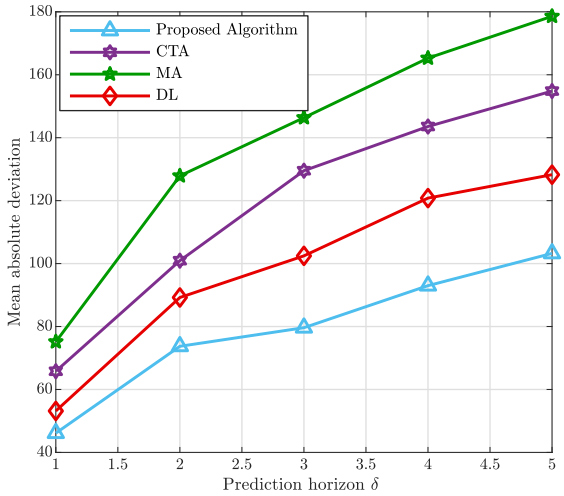

Figure 6.11: Mean absolute deviation for Dataset 2.

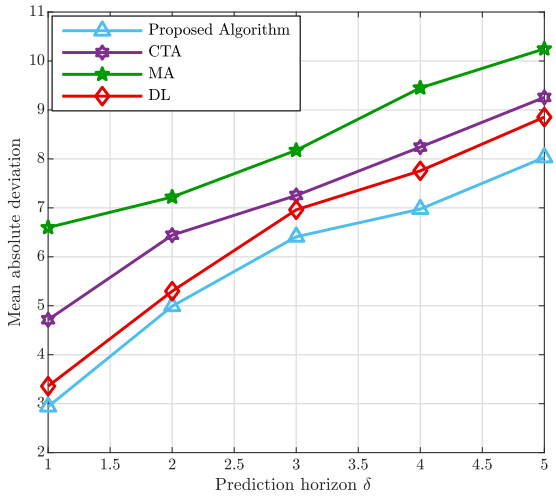

Figure 6.12: Mean absolute deviation for Dataset 3.

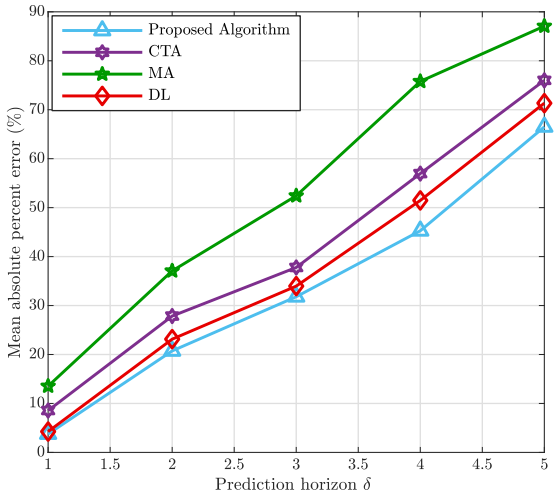

Figure 6.13: Mean absolute percentage error for Dataset 1.

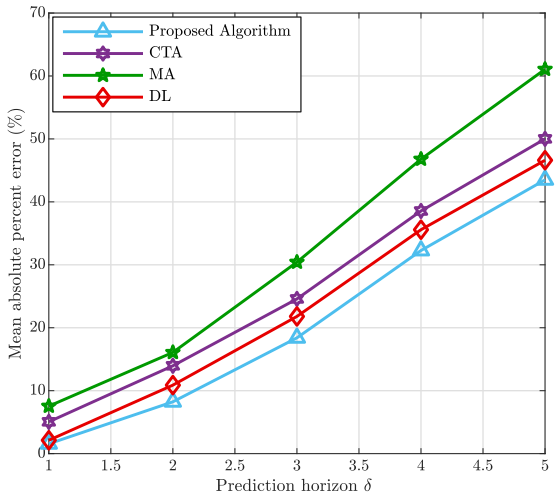

Figure 6.14: Mean absolute percentage error for Dataset 2.

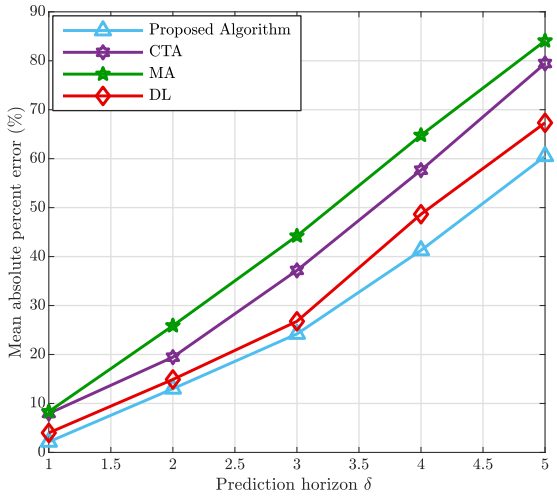

Figure 6.15: Mean absolute percentage error for Dataset 3.

deep neural network strategies. In order to analyze the strategies performance in terms of measure of the variability of the forecast errors, MSE and MAD have been represented in Figures 6.7-6.12. As is evident to note, in Figures 6.7-6.12 the forecasting accuracy is better when the proposed algorithms is adopted. Then, it is clearly evident that both the MSE and MAD increase for high values of δ. It is due to the general difficulty in predicting behavior for long interval times. In order to better quantify the meaning of the MSE and MAD, Figures 6.13 and 6.15 show the performance of the proposed algorithms in comparison with the considered alternatives, expressed in terms of MAPE metric. The results confirm the good performance of the proposed approach in comparison with the alternatives taken into account, for all the three different strategies proposed for each dataset. In conclusion, the proposed CT approaches provide a suitable solution to forecast values in complex and nonlinear dynamical systems, by investigating and capturing their underlying dynamics and geometrical structure, in the corresponding reconstructed phase space, and chasing the time series behavior, guaranteeing more accuracy than the MA, CTA, and the DL models.

6.1.13 Conclusions

This work investigates the potentialities of the CT framework contextualized to the problem of the prediction of the service requests for the TNCs. In particular, different algorithms for different real datasets have been presented. The predictive methods designed for the three analyzed dataset are based on the CT principles and the corresponding phase space has been reconstructed, the chaotic behavior studied, through the analysis of the largest Lyapunov exponent. Furthermore, a different CT based algorithm has been proposed for the different datasets studied. The validity of the proposed strategies have been confirmed by simulations and comparison with the MA, the CT based approach presented in [64], and the one discussed in [46]. Finally, system performance has been expressed in terms of mean squared, mean absolute error and mean percent forecasting error.

6.2 Passengers Demand Forecasting Based on Chaos Theory

6.2.1 Introduction

The recent increase in spread of vehicles into the metropolitan areas has resulted in massive levels of urban traffic, fatalities, injuries, and congested cities [201,283]. Within smart cities context, many research efforts have been made to significantly improve the conditions of our lives, involving the optimize use of resources of the critical infrastructures of cities, such as power, water, roads, etc [177, 201, 207, 283]. In this respect, the use of Intelligent Transportation Systems (ITS) is crucial to enable modern solutions to control mobility and traffic, especially in order to lower roads congestion and improve the transportation efficiency [198, 201]. The reduction of the number of commuters that use their own cars, in favor of public transportation, can significantly reduce the traffic flows during the rush hours. Transportation network companies (TNC) such as Uber or Lyft provide peer-to-peer ride-sharing services to move people from and to homes and offices, until the public transportation stations. These types of services constitute a promising way to ensure to people a fast and comfortable solution to use the public transportation system [201]. By the way, the accurate prediction of traffic demands has a key role to guarantee high levels of customers satisfaction and to improve companies' profits, to properly allocate resources, avoiding delays or waste of resources. For all these reasons, traffic and mobility forecasting based on big data analysis have gained momentum in the area of ITS. The general idea behind the forecasting problem in the ITS, is the prediction of traffic conditions, a certain number of steps ahead, on the basis of past observations [198, 201].

6.2.2 Contributions

Hereafter the passenger demand forecasting framework proposed in [201] is presented. The prediction has been conducted on big data collected by Didi, a Chinese TNC. More in depth, the main contributions of this work are:

- the reconstruction of the phase space, in order to study the geometrical and dynamical characteristics of the corresponding strange attractor;
- the application of a modified version of the nearest neighbors method

combined with a global predictive model to improve the accuracy on the forecast values;

- the comparison of the proposed approach with the most popular state-of-art methods, aiming at corroborating the CT based forecasting validity.

6.2.3 Problem Statement

Let $\mathcal{S} = \{s_i\}_{i=1}^{t}$ be a given scalar time series, in which s_i is the value observed at time i. $\mathcal{S}$ derives from a previous sampling procedure in the city of Chengdu in China. In particular, the such city has been modeled as a rectangle R of dimension $A \times B$. Furthermore, each service request demand r_h in the dataset consists of a quintuple in the form $(r_{h,id}, r_{h,pc}, r_{h,pt}, r_{h,dc}, r_{h,dt})$, in which $r_{h,id}$ is the request ID, $r_{h,pc}$ the pick-up coordinates, $r_{h,pt}$ is the pick-up time, $r_{h,dc}$ the coordinates of the destination, and $r_{h,dt}$ the time of arrival at destination. Since in this study the main objective is to provide a prediction about the demand, only the pick-up coordinates have been considered. In addition, requests can be clustered based on the pick-up time. In this respect, time has been partitioned into equal slots. Starting from 0, the k-th time slot is identified by the interval $[k \times F, (k+1) \times F)$, where F is the time span of the interval. Hence, since the pick-up requests outside the city are discarded, the passenger demand at the interval k, i.e. s_k, is given by

$$s_k = |\{h \in [0, N) : r_{h,pc} \in R \wedge r_{h,pt} \in [k \times F, (k+1) \times F)\}|, \tag{6.21}$$

where N is the total number of received requests.

The forecasting problem consists in predicting some next points, less or more far in the future, typically based on historical data. In short-term forecasting problems the main objective is the maximization of accuracy of predicted values, i.e. the minimization of the forecast error. The idea behind the forecast error is to measured the gap between the predicted and the actual value of the time series. There not exists only one single standard metrics to measure the forecast error, and possible metrics include the mean error, the mean absolute error, and the percent forecast error [180].

In this study the mean squared error (MSE) defined as

$$MSE = \frac{1}{n} \sum_{k=1}^{n} (\hat{s}_{k+1} - s_{k+1})^2; \tag{6.22}$$

has been minimized, where n is the number of the samples in test data, and $\hat{s}_{k+1}$ and s_{k+1} are the actual and the predicted values at time $k+1$, respectively.

6.2.4 Phase Space Reconstruction

In order to study the chaotic properties of $\mathcal{S}$, the first step is the reconstruction of the phase space (PSR) associated to $\mathcal{S}$. Based on the Takens' delay embedding theorem [29,239], The PSR has been obtained through the association of each element s_i in $\mathcal{S}$ with a vector in the form

$$\mathbf{v}_i = [s_i, s_{i+\tau}, \ldots, s_{i+(m-1)\tau}]; \tag{6.23}$$

where τ is the *time delay* and m is the *embedding dimension.*

Time Delay Reconstruction

The time delay τ rules the temporal distance between two successive points in the phase space vector and its optimal value is the one for which redundancy is minimal and the knowledge about $s_{i+\tau}$ from s_i is maximal. Let the mutual information of time delay τ be defined as

$$\begin{aligned} &I(s_i, s_{i+\tau}) = \\ &\quad = \int p(s_i, s_{i+\tau}) \log(\frac{p(s_i, s_{i+\tau})}{p(s_i)p(s_{i+\tau})}) ds_i ds_{i+\tau} \end{aligned} \tag{6.24}$$

where $p(s_i, s_{i+\tau})$ is the joint probability distribution between a s measured at a time i and at a time $i+\tau$ respectively, and the $p(s_i)$ and $p(s_{i+\tau})$ represent the marginal distributions. Generally speaking, the aim is to find the value of τ for which the average mutual information between two successive points is least [82]. Graphically, as depicted in Figure 6.16, the optimal value of τ is in correspondence of the first minimum of the I function, in our case $\tau = 4$.

Embedding Dimension

There are many approaches to estimate the suitable value of m. Hereafter, the method proposed by Kennel [126], named the false nearest neighbors, has been applied. The idea behind this method is that, since the presence

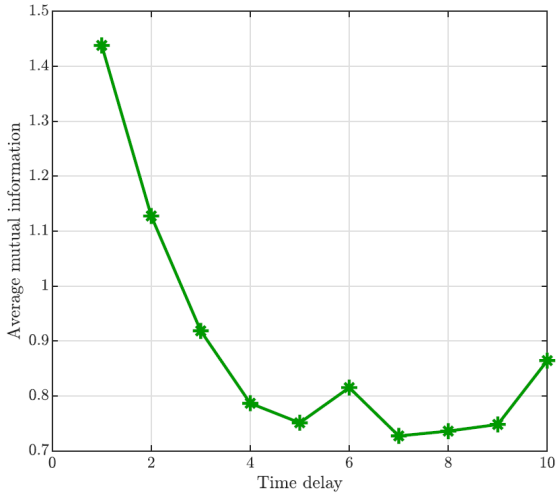

Figure 6.16: The optimal value of τ is in correspondence of the first minimum of the average mutual information plot.

of chaos can induce an exponential divergence on the trajectories of two nearby points, there exist some points in the data set that are neighbors in the embedding space but for which their temporal evolution exponentially diverges. Hence, for each point in $\mathcal{S}$, its nearest neighbor in m dimensions is considered. Then the ratio between their distance in $m+1$ dimensions and m dimensions is calculated. Finally, whether the resulted ratio is greater than a fixed threshold r, the neighbor is considered false. In formal terms, given a fixed threshold r, the false nearest neighbors function is defined in (6.26), where σ is the standard deviation of the data, j is index of the nearest point, and Θ is the Heaviside step function.

$$F_{fnn}(r) = \frac{\sum_{i=1}^{t-m-1} \Theta\left(\frac{|\mathbf{v}_i^{(m+1)} - \mathbf{v}_j^{(m+1)}|}{|\mathbf{v}_i^{(m)} - \mathbf{v}_j^{(m)}|} - r\right) \Theta(\frac{\sigma}{r} - |\mathbf{v}_i^{(m)} - \mathbf{v}_j^{(m)}|)}{\sum_{i=1}^{t-m-1} \Theta(\frac{\sigma}{r} - |\mathbf{v}_i^{(m)} - \mathbf{v}_j^{(m)}|)}; \quad (6.25)$$

Figure 6.17 shows the behavior of F_{fnn} function, and the suitable value of m is in correspondence of the last m value before the plot drops to zero,

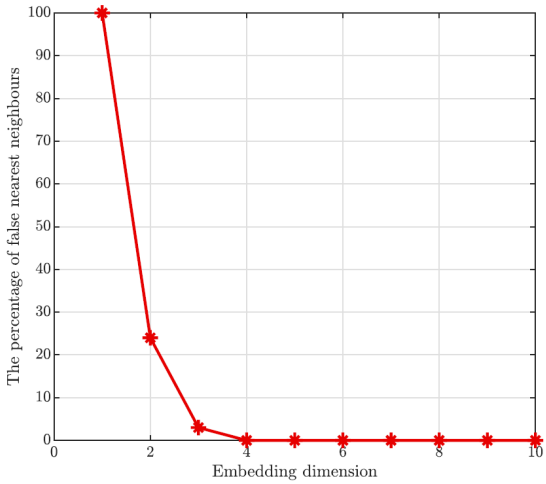

Figure 6.17: The optimal value of m is the one before the false neighbors plot drops to zero.

in our case $m = 3$.

Largest Lyapunov Exponent

In order to apply CT principles to our time series, the chaotic behavior of $\mathcal{S}$ needs to be verified. The positive value of the largest Lyapunov exponent is a strong signature of the presence of chaos, therefore it is a suitable criterion for establishing the chaotic nature of a time series. In this respect, the largest Lyapunov exponent has been analyzed by using the method proposed by Rosenstein [238], based on the estimation of the local divergence rates of trajectories over the whole data set in the phase space. Specifically, the local divergence is estimated on the neighborhood of each point of $\mathcal{S}$ in the phase space. The general idea of the method is the measurement, for each s_i in $\mathcal{S}$, of the expansion rate in a particular time span Δn of the trajectories. In particular, that quantity is given by

$$\mathcal{E}(\Delta n) = \frac{1}{t}\sum_{i=1}^{t} \ln\left(\frac{1}{|\mathcal{N}(\mathbf{v}_i)|}\sum_{\mathbf{v}_z \in \mathcal{N}(\mathbf{v}_i)} |\mathbf{v}_{i+\Delta n} - \mathbf{v}_{z+\Delta n}|\right); \tag{6.26}$$

where $\mathcal{N}(\mathbf{v}_i)$ is the neighborhood of point $\mathbf{v}_i$ in the phase space.
The details of algorithm are reported in [238] and in our case, the resulted largest Lyapunov is $\lambda_{max} = 0.7$. Since $0 < \lambda_{max} < \infty$, the time series exhibits a chaotic behavior [124, 158].

6.2.5 Time Series Forecasting

In order to predict the exact future behavior of $\mathcal{S}$, the knowledge about the m first-order ordinary differential equations is essential, but it is impracticable when real data are involved. However, by means of PSR, the underlying dynamics of the map $\mathbf{F}$, defined as

$$\mathbf{v}_{i+1} = \mathbf{F}(\mathbf{v}_i), \tag{6.27}$$

can be captured. In this regard, the phase space reconstruction has to be performed. In accordance with PSR, the suitable values for time delay τ and embedding dimension m have been calculated.
Given $\mathcal{S}$, in order to predict the value Δn steps ahead of s_t, i.e. $s_{t+\Delta n}$, a local approximation of $\mathbf{F}$ is defined, by considering the neighborhood around $\mathbf{v}_t$, in the phase space, of radius ϵ, hereafter referred to $\mathcal{N}_\epsilon(\mathbf{v}_t)$. Furthermore, for each $\mathbf{v}_y \in \mathcal{N}_\epsilon(\mathbf{v}_t)$, its trajectory at time $y + \Delta n$ is observed, hence its value $s_{y+\Delta n}$, and then the mean value over all the future behavior of the points belonging to $\mathcal{N}_\epsilon(\mathbf{v}_t)$ is considered. With the aim to improve the prediction accuracy, a weight α_y is assigned to each $s_{y+\Delta n}$ value, obtained starting from the points in $\mathcal{N}_\epsilon(\mathbf{v}_t)$. Specifically, the higher is the similarity between s_t and $s_{t+\Delta n}$, the higher is the value of α_i. The weight α_t has been defined as follows

$$\alpha_t = \frac{1}{|s_{t+\Delta n} - s_t|}. \tag{6.28}$$

Let η be the number of points in $\mathcal{N}_\epsilon(\mathbf{v}_t)$. In formal terms, the forecast value of $s_{t+\Delta n}$, that is $\hat{s}_{t+\Delta n}$, is given by

$$\hat{s}_{t+\Delta n} = \frac{1}{\eta} \sum_{\mathbf{v}_y \in \mathcal{N}_\epsilon(\mathbf{v}_t)} \alpha_y s_{y+\Delta n}. \tag{6.29}$$

Hence, considering the chaotic scalar time series $\mathcal{S}$, the whole prediction procedure can be summarized as follows

- select the time delay τ optimal for $\mathcal{S}$ according to subsection 6.2.4;

- select the embedding dimension m optimal for $\mathcal{S}$ according to subsection 6.2.4;
- reconstruct the phase space associated to $\mathcal{S}$ using τ and m;
- select radius ϵ and build $\mathcal{N}_\epsilon(\mathbf{v}_t)$;
- for each $\mathbf{v}_y \in \mathcal{N}_\epsilon(\mathbf{v}_t)$, measure $s_{y+\Delta n}$;
- for each $\mathbf{v}_y \in \mathcal{N}_\epsilon(\mathbf{v}_t)$, calculate α_y;
- forecast $\hat{s}_{t+\Delta n}$ in accordance with (6.29).

It is straightforward to note that a crucial issue in this prediction algorithm is the selection of a proper value for the radius ϵ of the neighborhood of $\mathbf{v}_t$. Obviously, the choice of ϵ is strictly related to the number of points involved in the forecasting process. Indeed, a too small radius implies a poor number of points involved in the forecasting process which may result in an inaccurate prediction, instead, a very large number of neighbors, hence a high value of ϵ, considers not useful information that degrades the prediction outcome. Unfortunately, there not exist specific procedures to determine the optimal value for ϵ or about the number of points that have to be involved in the forecast process [64]. Given the time series $\mathcal{S}$ composed of t scalar values, the estimation of the optimal value of τ requires a computational complexity in the order of $\mathcal{O}(t \cdot \log t)$, while by applying the procedure to find the suitable embedding dimension m, the maximum amount of time taken is $\mathcal{O}(t^2 + 4 \cdot \xi \cdot t + \xi^2 \cdot t)$, where ξ is the number of considered m values. Furthermore, the phase space reconstruction exhibits a computational complexity of $\mathcal{O}(\lambda \cdot m)$, where $\lambda = t - (m-1)\tau$. Finally, the nearest neighbors procedure has a complexity in the order of $\mathcal{O}(t)$. AS consequence, the overall time complexity is

$$\mathcal{O}(t \cdot \log t) + \mathcal{O}(t^2 + 4 \cdot \xi \cdot t + \xi^2 \cdot t) + \mathcal{O}(\lambda \cdot m) + \mathcal{O}(t). \tag{6.30}$$

In order to improve the very short-term forecasting, the map $\mathbf{F}$ can be modeled with a global nonlinear approximation given by the radial basis function [124] defined as follows.

$$\mathbf{F}(\mathbf{v}) = \theta_0 + \sum_{w=1}^{g} \theta_w \phi(||\mathbf{v} - \zeta_w||), \tag{6.31}$$

where ζ_w are the g centers of the attractor, ζ_0 and ζ_w are coefficients, and ϕ is the Lorentzian function expressed by

$$\phi(\psi) = \frac{1}{[1 + (\frac{\pi}{a})^2]}, \tag{6.32}$$

with a constant and $\pi > 0$. The joint utilization of both the approximation models presented in (6.29) and (6.31) has been pursued and, while the local approach ensures a lower long-term forecasting error by considering the recent evolution of the phase points, the global one improves the performance of the very short-term forecasting taking into accounts all the phase points.

6.2.6 Numerical Results

In the case study here discussed, the nonlinear time series $\mathcal{S}$ derives form sampled data collected in the city of Chengdu, from Didi Chuxing, the biggest TNC in China. The dataset contains the passengers requests of one month, from 11/01/2016 to 11/30/2016, and the whole dataset contains more than 6.11 million of passengers requests. The whole area has been divided into 20×20 same-size grids. Every grid is a square with sides equal to 700 meters, and the longitude of the focus area is from 30.60E to 30.73E, the latitude is from 104.00N to 104.15N, while the considered surface is about $207.35\,\mathrm{km}^2$. The proposed CT approach has been evaluated by comparing it with two of the most common state-of-art predictive models, the well known AR and the Auto Regressive Moving Average with Exogeneous Input (ARMAX) [132]. More in depth, AR model is based on the historical behavior of the time series and the predicted value linearly depends on the previous values, while ARMAX combines both moving average and autoregressive terms. Performance is measured in terms of MSA, and mean percent error (MAPE) defined as

$$MAPE = \frac{1}{n} \sum_{k=1}^{n} \left| \frac{\hat{s}_{k+1} - s_{k+1}}{s_{k+1}} \right| \cdot 100. \tag{6.33}$$

In order to provide a complete overview about the performance of the approach adopted, the forecasting results when the prediction horizon is a variable number from 1 to 9 steps ahead has been considered. More in depth, the results shown hereafter are referred to prediction based on the previous 96 hours to forecast the passengers demand 9 hours ahead. As it

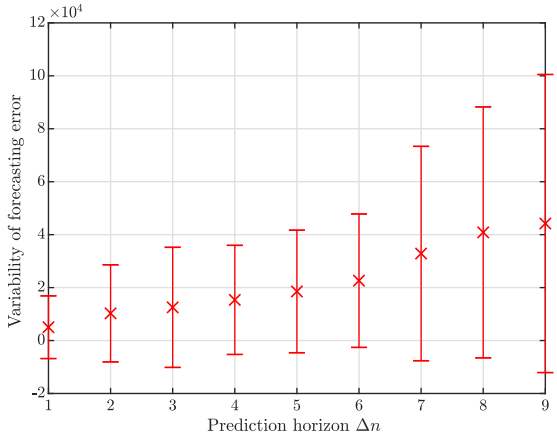

Figure 6.18: Variability of forecasting error.

is evident in Figure 6.18 and Figure 6.19, the forecasting accuracy applying the local CT prediction is better for close prediction horizons. Furthermore, Figure 6.18 depicts the the amount of variation of the forecasting error, expressed in terms of mean forecasting error plus or minus the corresponding standard deviation. It is evident that both the mean error and its variability increase for high values of Δn. It is due to the general difficulty in predicting behavior for long interval times. This trend is confirmed by Figure 6.19, that shows the performance of proposed CT approach in comparison with AR and ARMAX. As Figure 6.19 exhibits, despite the forecasting error increases with Δn, the CT approach achieves better results than the other two predictive models. Figure 6.20 depicts the MAPE metrics for a prediction horizon from 1 to 5 hours ahead. Also in this case, the better behavior of the CT approach is highlighted. The advantages of combining both the local and global prediction approaches are evident in Figure 6.21 and Figure 6.22, where the MAPE and the MSE metrics are respectively represented. In conclusion, the CT approach provides a suitable solution to forecast values in complex and nonlinear dynamical systems, by investigating their dynamics and geometrical structure in the corresponding reconstructed phase space, and chasing the time series behavior with more accuracy than the AR and ARMAX models.

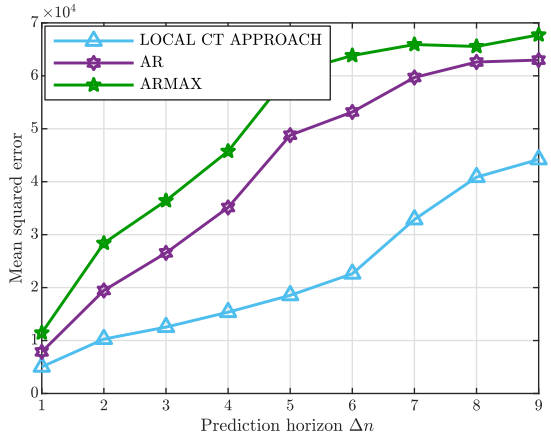

Figure 6.19: Forecasting error expressed in terms of mean squared error.

6.2.7 Conclusions

The problem of the passengers demand forecasting has been addressed, on the basis of on real data from Didi Chuxing, the most famous TNC in China. In order to forecast the future behavior of passenger demands, a CT approach to deal with the corresponding nonlinear scalar time series has been proposed. With the aim at applying the CT principles, the corresponding phase space has been reconstructed and the chaotic behavior studied.The prediction algorithm has been performed observing the trajectories of the points, in the phase space, close to the one that has to be predicted, and by applying a global predictive model to improve the accuracy over the forecast results for very close time horizons. The validity of the CT approach has been confirmed by simulations and comparison with other two state-of-art predictive models, expressed in terms of mean squared and mean percent forecasting error.

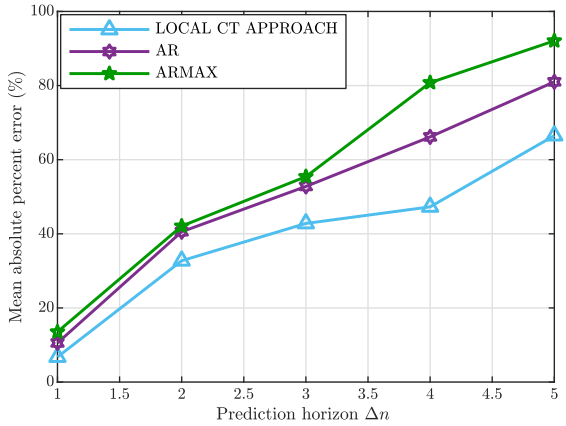

Figure 6.20: Mean absolute percentage error.

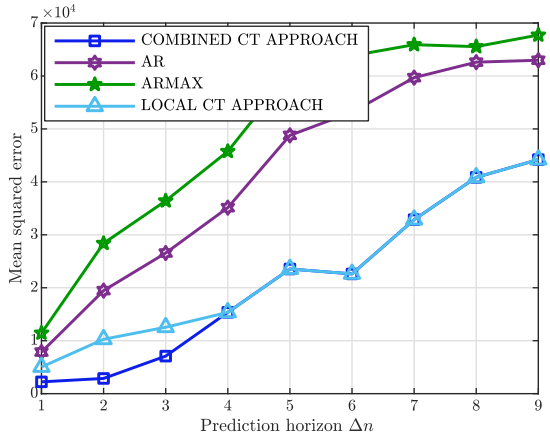

Figure 6.21: Mean squared error adopting the combination of local and global prediction model.

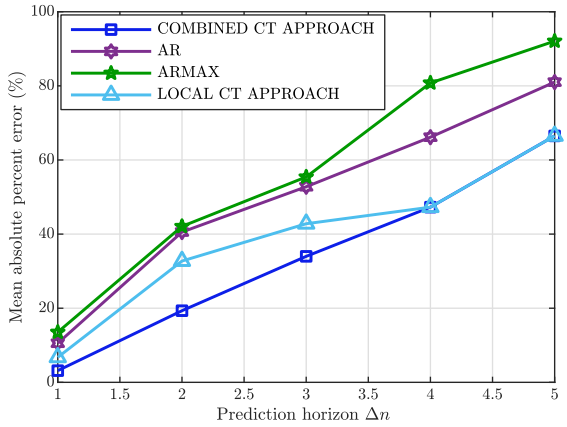

Figure 6.22: Mean absolute percentage error adopting the the combination of local and global prediction model.

Chapter 7

Machine Learning: Towards the Cross-Layer Frameworks

If you have everything under control, you're not moving fast enough

Mario Andretti

7.1 A Federated Learning Framework for Mobile Edge Computing Networks

7.1.1 Motivation

Recently, the ever increasing dissemination in our daily life of intelligent devices such as wearable devices, smartphones, smart cards, sensors and so on, has triggered the proliferation of numerous distributed networks devices generating massive quantity of heterogeneous data to be processed and interpreted [229, 257]. Due to such unprecedented amount of data with exponential growth trend [54] and the typical private nature of these data, sending all the data to a remote cloud is become impracticable, unnecessary and full of privacy concerns [257]. Therefore, all of these factors have contributed to the emergence of the new mobile edge computing (MEC) paradigm [156, 162, 221], which exploits the advancement on storage and computation capacity of modern devices for pushing processing and storing

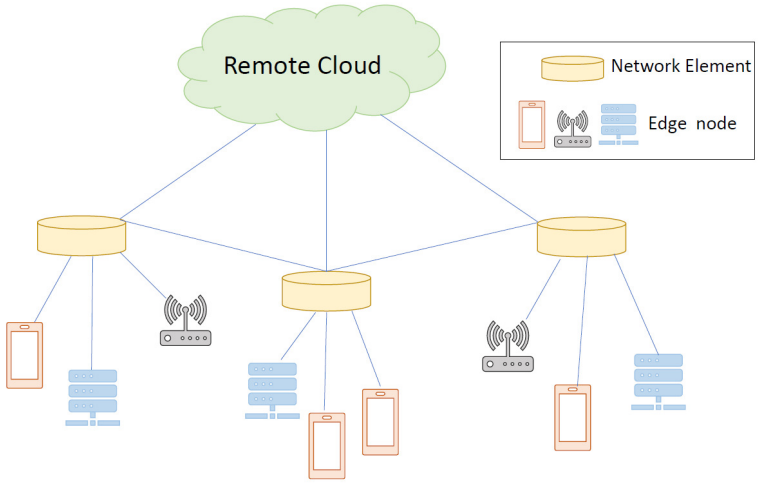

Figure 7.1: Cloud-MEC Network Architecture

procedures locally on the devices themselves. The MEC approach involves the cooperation of edge-nodes with the remote cloud in order to give rise to a computing system able to support a large scale tasks processing and managing environment [229]. Within this context, the efficient and effective handling of big data brings out several information and statistical features hidden in the datasets, useful for many application areas such as resource planning, system conditions forecasting, classification and so on. In this regards, machine learning (ML) [286] techniques have gained momentum to properly catch and interpret data behavior, by providing a wide range of solutions to analyze datasets trend on the cloud site. Despite the dataset characteristics represent an invaluable source of information to be properly exploited, from the other side the manipulation of users sensitive data implies significant responsibilities and risks in keeping them in a centralized site such as the cloud [169]. In order to manipulate big data with respect of users privacy, the federated learning (FL) [169, 229, 257, 281] approach has emerged as a set of ML techniques to perform statistical and mathematical training models directly on devices. FL framework involves ML models locally trained at devices level, hereafter named as clients, and then the aggregation of these results in a central server, for example a base station.

7.1.2 Contributions

The main contributions of this study are

- The contextualization of the FL framework to the VRCs deployment problem, by exploiting the decentralized training data is a powerful tool to pursuit effective results in the VRCs allocation problem;
- The detection and application of basic methods to perform decentralized data training without draining the hardware resources of the EDs;
- The extensive numerical simulations and comparison with the chaos theory approach, performed to validate the remarkable behavior resulting by applying the proposed approach to the VRCs deployment problem.

Furthermore, similarly to paper [129, 169, 206, 256, 280, 285], this work proposes the application of the FL by using straightforward methods belonging to the gradient descent algorithms family. This conservative choice is due to the fact that more complex methods may result in prohibitive consumption of the EDs resources, which represents a crucial point in the decentralized data training research field. Differently than previous literature, this work focuses on the application of the FL framework to the VRCs deployment problem, by exploiting FL to predict the individual edge device (ED) demand in order to perform a proper VRCs planning. Furthermore, for the best of our knowledge, this is the first study to contextualize FL to the VRCs allocation problem. Finally, the goodness of the proposed approach has been tested by resorting to extensive numerical simulation and by comparison with other predictive disciplines.

7.1.3 Related Works

ML techniques constitute a wide branch of big data manipulation literature in MEC networks. Paper [234] investigates the application of various machine learning techniques in order to report the impact of different ML methods on the MEC network. Furthermore, paper [234] analyzes the effectiveness of the ML algorithms to detect the presence of malicious attacks in a MEC network. Authors in [284] proposes a deep supervised learning method aiming at minimizing the overall network cost in performing computational offloading. Differently, a MEC blockchain network has been studied

in paper [161], in which an auction solution based on deep learning is formulated to perform edge resource allocation in order to maximize the edge computing service provider profit. More in depth, paper [161] builds a multilayer neural network based on the optimal auction solution. In paper [295] a multi hidden multi layer convolutional neural network is adopted to perform data authentication in robust mobile crowd sensing problem, aiming at improving sensing reliability and reducing the overall latency. By taking into account a real-time industrial application environment, authors in [216] address the position based confidentiality problem in MEC systems by exploiting the k-nearest neighbor and the decision tree approaches. Paper [48] examines the main classes of ML solutions to measure the benefits deriving from edge caching mechanisms, especially in terms of user satisfaction and energy efficiency evaluation.

Contrary, distributed ML is adopted in paper [133, 146, 248, 249]. Paper [248] uses a distributed version of the support vector machine method within an internet of things (IoT) context to evaluate system performance implementing the distributed ML. The distributed stochastic variance reduced gradient is applied in paper [249], in which the authors aim at optimizing the number of collection points to perform data analysis, considering a fixed target accuracy, in order to minimized the amount of network traffic spent to send all the data towards the collection points. In [146] the crowd sensing problem in an edge computing scenario is treated by proposing a distributed deep learning approach, in order to lower the traffic congestion in the cloud site and balance the traffic. In particular, authors in [146] involve the human in the loop methodology to give a hierarchical structure to the crowd sensing problem, aiming at controlling the whole crowd sensing process. Furthermore, the distributed Q-learning algorithm is applied in paper [133], where the minimization of the users outage is performed by users themselves, selecting the most critical cell on which run the minimization and considering a heterogeneous networks context.

Finally, recently FL has gained momentum and paper [129, 169, 206, 256, 280, 285] constitute the main examples of such branch of literature. Paper [280] proposes a novel aggregation data framework for the over the air computation, by exploiting the signal superposition property of wireless channels. The aim of papers [257, 280] is the maximization of the number of devices involved in the aggregation process, by minimizing the aggregation error. Authors in [256] and in [257] adopt FL in a MEC system, in

which the distributed gradient descent method is applied to determine the best trade off between local updates and global aggregations, taking into account the minimization of the loss function subjected to some resource constraints. In the same way, paper [285] considers as case study the MEC environment, by proposing the application of the hybrid filtering on stacked encoders to predict the fluctuation of files popularity in the contents caching problem. Then, paper [169] combines the federated averaging algorithm proposed in [169] with the stochastic gradient descent algorithm, in order to train data in a distributed fashion avoiding high level of communication costs. Paper [229] addresses the multi task learning problem by resorting to the FL framework based on the novel Mocha context aware optimization algorithm. A blockchained FL architecture is designed in [129], on the basis of which a distributed consensus strategy is provided, by analyzing the blockchain end-to-end delay. Finally, FL is proposed in [206] to face the optimization of the transmission and computation costs in a mixed IoT-MEC network, throughout the application of the multiple deep reinforcement learning agents.

7.1.4 Reference Scenario

The reference system scenario consists of the cloud network architecture mixed with that MEC as depicted in Figure 7.1, where the cloud is located in the remote area of the network, and there is a set of NEs $\mathcal{N} = \{1, \ldots, i, \ldots, n\}$ situated close to the EDs, hereafter represented by the set $\mathcal{D} = \{1, \ldots, j, \ldots, m\}$. Each ED requires computation of one and only one task belonging to the set $\mathcal{T}$, for which both $\mathcal{D}$ and $\mathcal{T}$ have the same number of elements. Each NE is equipped with a central processing unit (CPU), homogeneous in frequency for all the NEs. Differently, cloud is equipped with a higher CPU frequency. Then, each task, in order to be computed, requires a specific application which has to be installed in advanced on the computation site. In this regards, loading the applications on NEs requires the presence of available storage resource blocks (SRBs), since each application needs a fixed number of SRBs. Accordingly, each NE disposes of a number s_i of SRBs.

Each ED requiring task computation primarily looks for one VRC of the required application on a close NEs[1]. In the event that no one NE owns the

[1]Here is assumed that each ED sends the task to the nearest NE which, if it does not contain the application requested by the ED, forwards the task to the nearest NE

VRC of the required application, the task is sent to the cloud on which all the application types are present. Furthermore, the transmission cost among the NEs has been assumed negligible, while a fixed data rate for the wireless link between the EDs and their nearest NE has been set. Hence, the overall computation cost (OCC) experienced by ED j in performing computation is given by

$$T_j = \sum_{i \in \mathcal{N}} (\tau_{j,i} + c_{j,i^*}) x_{j,i} + (1 - x_{j,i})(\tau_{j,C} + c_{j,i^*}), \tag{7.1}$$

where $\tau_{j,i}$ and $\tau_{j,C}$ represent the time spent by task j on NE i and on cloud, respectively. It is important to note that both $\tau_{j,i}$ and $\tau_{j,C}$ are expressed as the sum of the task execution time spent in the CPU of the NE and the cloud, and the queuing time experienced by the task waiting for its execution on these sites[2]. Moreover, c_{j,i^*} represents the transmission cost in sending the task from ED j to its nearest NE i^*. Since the transmission time among NEs and between NEs and cloud have been supposed negligible, only the c_{j,i^*} cost has been we have taken into account. Finally, $x_{j,i}$ is a binary value equals to 1 if the task j is computed on the NE i, 0 otherwise. It is important to highlight that the OCC in (7.1) strongly depends on the queuing time experienced by task on the designated computation site. In fact, a proper deployment of VRCs on the ENs network may drastically reduce the OCC task time.

7.1.5 Problem Formulation

The main objective of this paper is the maximization of the hit rate in finding the VCRs of the requested applications on the NEs. This metric evaluation is due to the fact that the deployment of VRCs, provided on the basis of the forecast EDs applications demand, lowers the tasks OCC. Therefore, in formal terms, the AHP can be expressed as

$$\mathcal{P}(\mathbf{X}) = \frac{\mathcal{H}(\mathbf{X})}{m} 100, \tag{7.2}$$

where $\mathcal{H}(\mathbf{X})$ is the function which, given the VRCs allocation matrix $\mathbf{X}$, whose generic element is $x_{j,i}$, associates the corresponding number of hits

containing that application. Finally, it has been assumed that each NE has knowledge about the VCRs contained by other NEs, and that each NE has stored a routing table in which, for each pair of NEs, the shortest path between those NEs is saved.

[2]The CPU queue has been assumed with the first-in-first-out service policy.

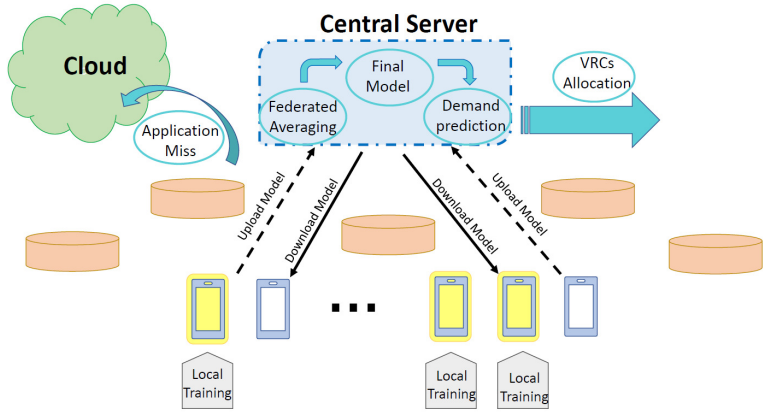

Figure 7.2: FL framework in Cloud-MEC Network Architecture

occurred in finding the VRCs loaded on the NEs.
Furthermore, the main goal of this paper is given by

$$\max_{\mathbf{X}} \mathcal{P}(\mathbf{X}). \tag{7.3}$$

s.t.

$$0 \leq s_i \leq S, \forall i \in \mathcal{N}, \tag{7.4}$$

where S expresses the maximum number of SRBs available on a NE. Hence, constraint (7.4) expresses that each NE i has a maximum limitation about the number of SRBs. It is important to note that $\mathcal{P}(\mathbf{X})$ depends on the allocation matrix $\mathbf{X}$, hence on the deployment of the VRCs on the NEs.

7.1.6 Federated Learning Framework

The Learning Problem

Generally speaking, ML aims at learning parameter models on the basis of some training data. In this regards, a ML model is typically characterized by a loss function depending on the data sample z and a parameter vector $\mathbf{w}$, i.e., $f_z(\mathbf{w})$, which catches the error introduced by the model in relation with the training data [256]. By assuming the presence of m EDs, each of which having local data Δ_j, $j = 1, \ldots, m$, the collective loss function can be

expressed as [169, 256]

$$F_j(\mathbf{w}) = \frac{1}{|\Delta_j|} \sum_{z \in \Delta_j} f_z(\mathbf{w}), \tag{7.5}$$

in which $|\Delta_j|$ indicates the cardinality of Δ_j, i.e., the number of elements in Δ_j. Similarly, from (7.5) follows that the global loss function computed on all the distributed local dataset Δ_j is given by [169, 256]

$$F(\mathbf{w}) = \frac{\sum\limits_{j \in \{1,\ldots,m\}} |\Delta_j| F_j(\mathbf{w})}{\sum\limits_{j \in \{1,\ldots,m\}} |\Delta_j|}. \tag{7.6}$$

As well explained in [256], the direct consequence from (7.5) and (7.6), is searching $\mathbf{w}^\star$ such that

$$\mathbf{w}^\star = \arg\min F(\mathbf{w}). \tag{7.7}$$

Therefore, as several previous state of art works [169, 256, 257], in order to optimize (7.7) with low computational complexity, the gradient descent method is applied.

Federated Learning Scheme

As depicted in Figure 7.2, the proposed FL framework is composed by the clients side, responsible for the local data training, and the server side, typically a Base Station, represented by a central server intended for improving the global learning model, through the merging and aggregation of EDs updated local models. This approach is based on the interaction process between the clients and server sides, and, during each algorithm iteration round u, the EDs involved in the training procedure are a subset of the whole EDs set, whose number of elements is equal to y.

The algorithm acts as follows

- In parallel, each ED j among the y EDs involved in the training procedure, updates its local parameter vector $\mathbf{w}_j(u)$, implicitly built on the basis of its local dataset Δ_j, in accordance with the following rule [257]

$$\mathbf{w}_j(u) = \hat{\mathbf{w}}_j(u-1) - \alpha \nabla F_j(\hat{\mathbf{w}}_j(u-1)), \tag{7.8}$$

  where α is the learning rate and $\hat{\mathbf{w}}_j(u-1)$ represents the term $\mathbf{w}_j(u-1)$ after global aggregation.

- The server side provides the weighted average as proposed in [169] and expressed by

$$\mathbf{w}(u) = \frac{\sum\limits_{j\in\{1,\dots,m\}} |\Delta_j|\mathbf{w}_j}{\sum\limits_{j\in\{1,\dots,m\}} |\Delta_j|}. \tag{7.9}$$

Distributed data training performing the algorithm previously presented implies several advantages in terms of clients privacy, preservation of the computational resources of EDs, and messages exchange. In fact, the data training provided locally on the clients site, allows users to keep protected their sensitive information. In addition, roughly speaking, the uploading of the ED j parameter vector $\mathbf{w}_j$ does not expose the client to any sort of privacy issue, since, given $\mathbf{w}_j$, to retrieve Δ_j is not trivial.
Furthermore, for each algorithm iteration round, the involvement of only a part of the EDs set ensures to reduce the message passing between clients and server sides. Finally, last but not least, it is important to highlight that the usage of the gradient descent algorithm performs optimization without implying an excessive resource consumption, by taking into account the EDs perspective.

VRCs Allocation

As previously detailed, in this study the FL has been used to provide accurate distributed prediction on the future EDs applications demand by considering the historical EDs applications requests mitigated by the correlation with the other EDs involved in the training procedure. Once the prediction has been achieved, the main goal here is the exploitation of these information in order to provide a proper VRCs allocation, in terms of AHP metric.

Given the predicted application demand, practically expressed in terms of predicted application popularity, the VRCs allocation strategy, given the applications popularity vector $\mathbf{p}$ sorted in descending order, consists of the following steps:

1. Start with the NEs empty, hence, with all the SRBs available;

2. Deploy one VRC of each application, starting by that most requested in $\mathbf{p}$, on all the NEs in the network with a number of available SRBs able to host the considered application;

3. If any NE can host the considered application, the VRCs allocation algorithm terminates, otherwise;

4. Consider the next application in $\mathbf{p}$ and repeat steps $2) - 4)$ until there exists at least one NE able to host the processed application.

A very relevant point is that the proposed VRCs allocation algorithm provides an unbalanced VRCs deployment, according with the different popularity level of each application. In particular, step 3) guarantees that the outcome VRCs allocation is not a uniform distribution of the applications. In fact, such point ensures that no VRCs allocation is provided if the most requested application cannot be stored.

Furthermore, as regards the EDs computation strategy, as previously anticipated, each ED j requesting task computation with application r, sends task to its nearest NE, i.e, i^*, which acts in the following three ways:

- The NE i^* computes the task if it contains at least one VRCs of the application r;

- The NE i^* does not contain any application r VRC, but at least one NE, i.e.,v, has loaded at least one application r VRC. Therefore, i^* forwards the task to the NE v;

- Any NE contains one application r VRC and i^* forwards the task to the far cloud.

7.1.7 Numerical Results

In order to extensively test the proposed FL based framework, numerical simulations have been performed in TensorFlow. A simulation scenario constituted by $\mathcal{N} = 6$ NEs, equipped with a CPU frequency equals to 2.4 GHz, and considering a number of uniformly distributed SRBs $s_i \in [50, 300]$, has been considered. Each application occupies a number of SRBs uniformly distributed within the interval $[15, 40]$, while the cloud has been modeled by supposing a CPU frequency equals to 4.6GHz. The application requests have been simulated as in $[149, 176, 285]$, by using MovieLens 1M dataset [107], hereafter referred as Dataset 1, and MovieLens 100K dataset [107], named as Dataset 2. Each task has been supposed composed by a number of 64 bits format instructions uniformly distributed within the integer interval

[250, 800], requiring 8 CPU cycles per instruction. Furthermore, the connection link between the EDs and the nearest set has been considered equal to 100 Mbit/s.

As regards the loss function, the mean squared error (MSE) which, for each data q_l in Δ_j, is defined as

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\hat{q}_l - q_l)^2, \tag{7.10}$$

has been considered. In (7.10) M represents the number of the samples in test data. Then, in order to test the effectiveness of the proposed approach, we compared the accuracy on the predicted values obtained with the chaos theory (CT) approach, through the phase space reconstruction method [159, 239].

The higher accuracy levels in prediction procedure are clearly evident in Figure 7.3 and Figure 7.4, which depict the MSE behavior, by increasing the time prediction horizon. In both Figures 7.3 and 7.4, the accuracy metric trend is showed by varying the prediction horizons in order to test the different approaches. In fact, the accuracy analysis has been performed by taking into account prediction horizons greater than that on which the VRCs placement is actually pursuit, i.e. one hour. As it is straightforward to note, the MSE values grow by increasing the prediction horizon. This is due to the intrinsic difficulty in predicting the series long term behavior.
Figure 7.5 and Figure 7.6 highlight system performance by considering the AHP by varying the algorithm communication rounds, for different number of EDs involved in the learning process. Both the figures confirm that greater is the number of EDs participating to the learning process, higher is the number of EDs involved in the process means more significant and accurate information on which the VRCs allocation strategy can act. Furthermore, by increasing the number of algorithm rounds, by which models are updated, the AHP reaches higher values, by improving system performance.

Finally, Figure 7.7 and Figure 7.8 make evident the system improvement on mean task OCC reached by involving a high number of clients in the learning process. These results confirm those previous exposed in Figure 7.5 and Figure 7.6 and, similarly, better performance, i.e., low values of mean OCC are obtained by increasing communication rounds to the considered two sided framework. All these results validate the goodness of the proposed ap-

proach for the VRCs allocation problem, and highlight the strict correlation between a valuable prediction model and remarkable system performance. Finally, the resulting system performance makes clear the suitability of FL to our problem.

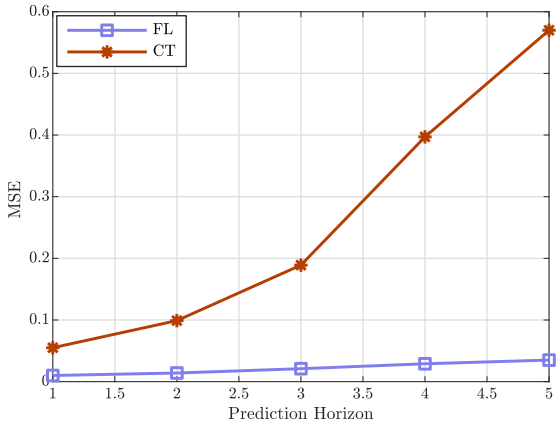

Figure 7.3: MSE by varying the time prediction horizon for Dataset 1

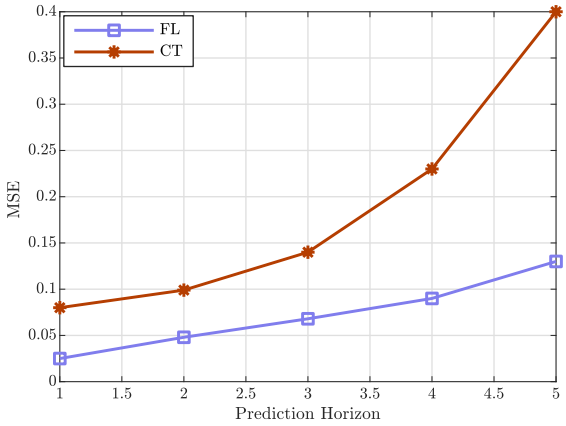

Figure 7.4: MSE by varying the time prediction horizon for Dataset 2

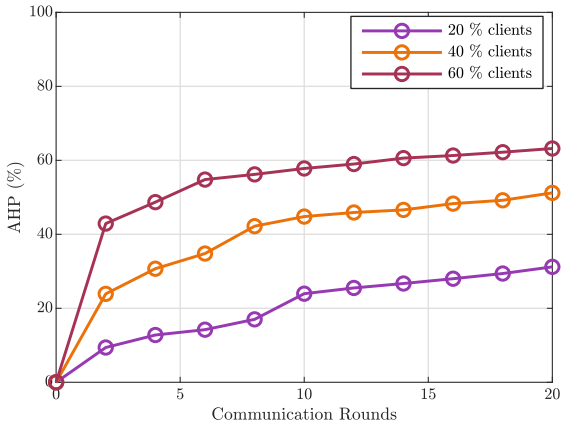

Figure 7.5: AHP by varying communication rounds for Dataset 1

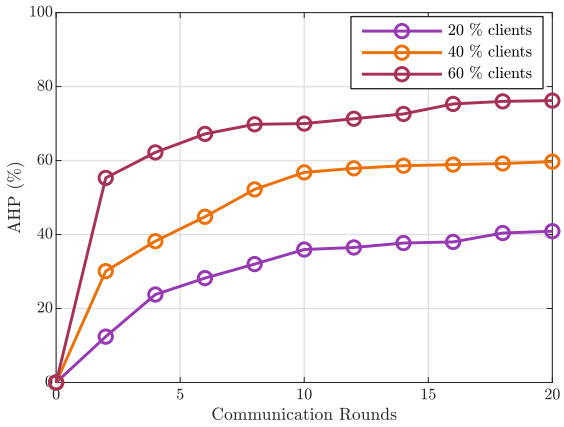

Figure 7.6: AHP by varying communication rounds for Dataset 2

7.1.8 Conclusions

This work addressed the VRCs allocation problem in a hybrid cloud-MEC network, aiming at maximizing the AHP probability, i.e., the odds of finding a VRC of the application requested by the EDs on a NE in the edge of the

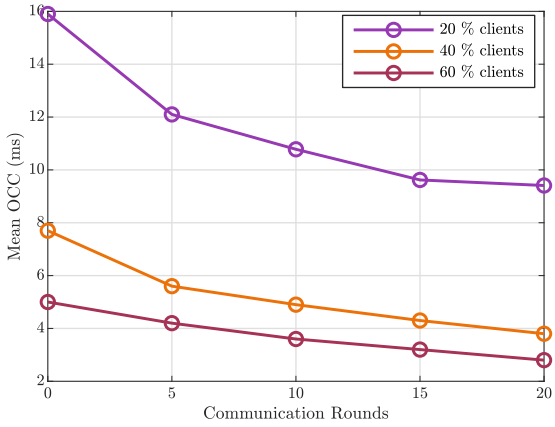

Figure 7.7: Mean OCC by varying communications rounds for Dataset 1

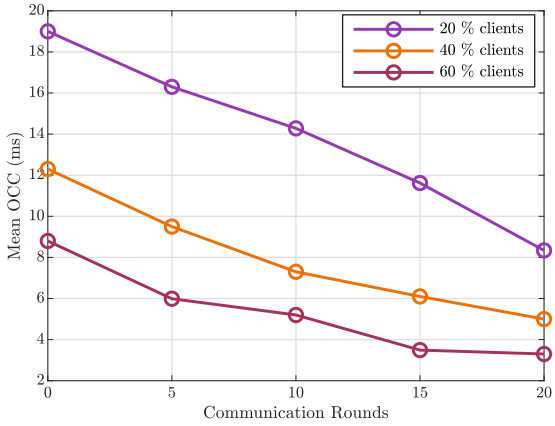

Figure 7.8: Mean OCC by varying communications rounds for Dataset 2

network, instead of in the cloud, typically located in the remote area of the network. The problem has been addressed by applying the FL framework with the gradient descent algorithms family to avoid the excessive exploitation of the EDs hardware resource such as battery lifetime or computational components. Finally, the validity of the proposed framework has been shown

throughout extensive empirical evaluation of system performance in comparison to the chaos theory based predictive approach.

7.2 Federated learning for Cloud-Edge Computing: A Service Provider Revenue Maximization Framework

7.2.1 Motivation

The diffusion of smart devices demanding for real-time data processing is exponentially growing. Therefore, service providers are dealing with the functional integration of the classical cloud computing architecture with edge computing network. However, the intrinsic limited capacity of edge computing nodes implies a proper allocation to improve user satisfaction and service fulfillment. Hence, demand prediction is crucial in the services management and exploitation. The main challenge here consists of the high variability on applications requests that results in inaccurate forecasts. Furthermore, the most popular prediction approaches typically involve sensitive users information gathering at a central unit, leading to privacy concerns. Federated learning is recently emerged as a solution to train mathematical learning models on the users site.

7.2.2 Contributions

The emergence of new network paradigms such as edge computing (EC) [156, 162, 221, 244], for which the limitations typical of the cloud architecture have been bypassed moving computation nodes to the network edges close to the end users, have given rise to a wide range of challenges in many research areas [63, 225]. Consequently, several new issues, such as user mobility, heterogeneity in quality of service (QoS) or service requirements, massive volume of data, users privacy, diversity on data types and so on, have led to numerous efforts from both academia and industry in providing highly effective and efficient solutions [54, 257]. In particular, there exists a significant branch of literature regarding possible solutions to improve the EC networks (ECN) performance in order to guarantee high level of users satisfaction and to provide dynamic and flexible network resource allocation and decision making strategies.

Generally speaking, heterogeneity in data flows types implies different QoS or service requirements. Furthermore, in a service provider (SP) perspective, such diversity triggers new data flows management policies, service provision costs and selling prices. In this respect, the SP revenue maximization is strictly related to the adopted management and administration policy. Indeed, a proper resource exploitation planning is essential to guarantee elevated levels of network efficiency, users satisfaction and consequent high SP revenues. In particular, having an a priori knowledge about the data flows service demand can be properly exploited to perform a suitable resource infrastructure planning with maximum income. In order to pursuit this objective, machine learning (ML) [20, 22, 131, 178, 286] is emerged by providing many techniques to perform data behavior interpretation and analysis. The ability of ML techniques in catching data trend, patterns and hidden features, has ensured its applicability to many problems. However, although the knowledge and extrapolation of users data characteristics positively impacts many application areas, it may result not compliant with some specific user privacy constraints [169]. In this respect, Federated Learning (FL) [169, 229, 245, 257, 262, 281] is recently emerged as promising tool to perform, locally on the users' devices, statistical and mathematical training models based on ML methodologies without loosing users privacy constraints. The FL framework consists of the devices level, generally indicated in literature as clients, and a central server unit which aggregates and merges the data preliminary processed by the clients.

7.2.3 Contributions

This work proposes the application of the FL framework, in order to forecast the service demands, without loosing the users privacy constraints. Moreover, on the basis of the service demands forecasting, this study proposes a suitable virtual functions (VFs) placement both on the ECN and cloud. Summarizing, the contributions are

- Application of the FL strategy to forecast the network VFs demand, in order to take into account the users privacy;
- Formulation of the SP maximum revenue problem, by considering service requests (SRs) with different priority and hence, different cost and price. In particular, the SP can accept the data SRs with low priority if all the high priority flows have been satisfied;

- Proposal of a VFs placement strategy and a suitable matching based SRs allocation algorithm based on the considered FL and the previously provided VFs forecasting scheme;
- Performance evaluation of the proposed approach and the comparisons with a centralized chaos theory (CT) based prediction scheme, by resorting to extensive computer simulation runs.

Moreover, in contrast to papers [121, 147, 148], which provide profit maximization solutions without taking into account the users privacy issues, we propose a revenue maximization framework based on data information elaborated locally on the users devices, avoiding the typical privacy concerns of the other approaches. Hence, as in papers [129, 169, 206, 256, 280, 285], we propose a FL based framework by using the gradient descent algorithm as optimizer. The motivation for this conservative choice resides in the fact that more complex methods may result in prohibitive consumption of the end users (EU) hardware resources, which is a crucial point in the distributed data training problems. Furthermore, in contrast to the previous state of art works, this work contextualizes the application of the FL to the VFs deployment problem, by exploiting the FL framework to properly predict the application network demand, in order to maximize the SP revenue. Furthermore, a VFs placement and a SRs service allocation is provided to evaluate the actual validity of the proposed solution. In fact, the SRs service allocation algorithm, based on the matching theory, does not take into account the SP perspective, but only the users, i.e., the SRs, interests. Finally, for the best of our knowledge, this is the first study which applies the FL to the SP revenue maximization problem, by considering even the users perspective. The proposed approach performance has been evaluated by resorting to extensive numerical simulation and by providing comparison with the centralized CT based predictive method.

7.2.4 Problem Statement

As reference scenario, we consider a single SP featuring an ECN constituted by $\mathcal{N}$ computation nodes (CNs) located at the network edges, and a more powerful cloud located far from the ECN. We suppose that all the CNs are equipped with a central processing unit (CPU) with the same computational capability and number of available storage resource blocks (SRBs) S. Differently, the cloud is assumed having a storage capacity of U SRBs, with

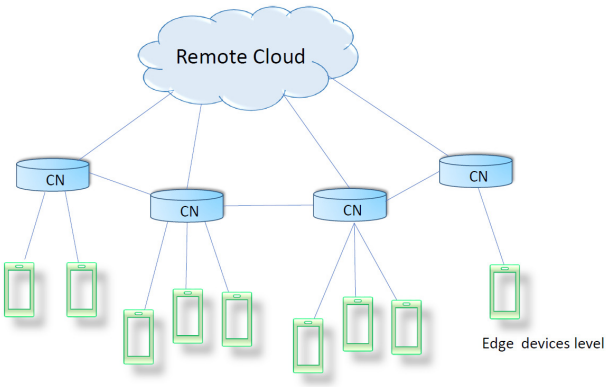

Figure 7.9: Hybrid Cloud-MEC Network Architecture

$S < U$. In addition, we assume the availability of high speed wired links between CNs and from any CN to the cloud[3] Furthermore, we guess that the ECN is able to support $\mathcal{T}$ different high priority service types, which are characterized by different provision costs and selling prices. Each service type $i \in \mathcal{T}$ has associated a QoS level expressed as a time deadline τ_i before which the type i service accomplishment has to be completed. In addition, we consider the presence of $\mathcal{M}$ service types requests with lower priority and without any time deadline constraint. The number of requests belonging to this class is indicated hereafter with y_j, with $j \in \mathcal{M}$.

Periodically, the SP updates the service demand and we assume that does not arrive any new request between two SP updates.

Let x_i be the number of SRs demanding for service i. We suppose that each SR is originated by an end user (EU), and that an EU requires only one SR. Therefore, as direct consequence, hereafter we assume interchangeable the SR and EU terms. Then, as regards the SP, the provision of a service has a cost mainly depending on x_i and following the model given by [258]

$$c(x_i) = \begin{cases} 0, & x_i = 0, \\ \beta_{c,i} + \beta_{l,i}\mu_i^{x_i}, & x_i \geq 0, \end{cases} \tag{7.11}$$

[3]We have assumed that the connection towards the cloud is performed throughout the CN nearest to the SRs needing computation. Consequently, the communication latency cost between SRs and their nearest CN has no impact on the overall SR completion time and hence it has been neglected in defining (7.17).

in which $\beta_{c,i}, \beta_{l,i}, \mu_i$ are real valued parameters whose value changes on the basis of the request type.

Similarly, the provision cost for providing y_j SRs of type j follows the rule [258]

$$b(y_j) = \begin{cases} 0, & y_j = 0, \\ \alpha_{c,j} + \alpha_{l,j}\nu_j^{y_j}, & y_i \geq 0, \end{cases} \tag{7.12}$$

where $\alpha_{c,j}, \alpha_{l,j}, \nu_j$ are, also in this case, real valued parameters.

Moreover, for each service type with high priority, the SP revenue results ruled by the following relation

$$U(x_i, q_i) = \frac{\log(1 + x_i)}{q_i}, \tag{7.13}$$

with $q_i = |x_i - k_i|$, where k_i is the number of SRs for which τ_i has been respected. Then, the SP revenue for the low priority SRs is given by

$$U(y_j, z_j) = \frac{\log(1 + y_j)}{z_i}, \tag{7.14}$$

where z_j is the number of SRs among y_j accepted by network for their service. Hence, the SP revenue, corresponding to the provision of the i-th and the j-th service type, can be expressed as

$$\mathcal{X}(x_i, q_i) = U(x_i, q_i) - c(x_i), \tag{7.15}$$

and

$$\mathcal{Y}(y_j, z_j) = U(y_j, z_j) - b(y_j), \tag{7.16}$$

respectively.

Both the SRs with high and low priority, in order to be accomplished, require the presence of a VF in set $\mathcal{V}$ which has to be preliminary loaded on at least one CN of the network or on the far cloud. The loading process requires the CN or cloud availability in terms of SRBs, since each VF $v \in \mathcal{V}$ requires a number a_v of SRBs, different for each VF. Consequently, the time required for the service accomplishment (TSA) of a generic SR r, independently by its priority, is given by

$$T_r = \sum_{v \in \mathcal{V}} \sum_{h \in \mathcal{N}} (\gamma_z + \omega_{z,h})\rho_{r,h}\theta_{v,h} + (1 - \rho_{r,h})\zeta_{v,C}(\gamma_C + \omega_{z,C}), \tag{7.17}$$

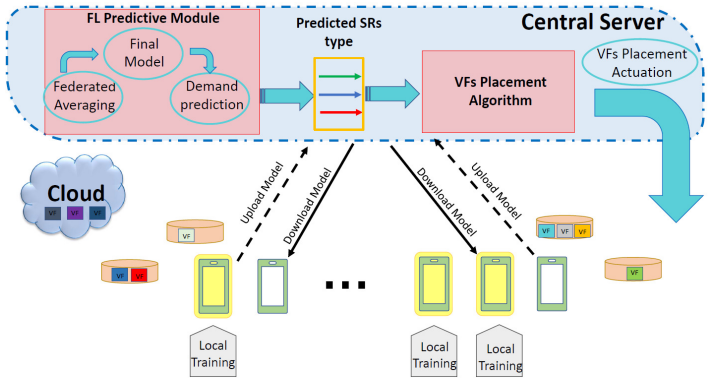

Figure 7.10: FL framework for the VFs placement

where γ_z and γ_C are the execution time spent by the SR z on the CPU of a CN and of the cloud, respectively. It is important to note that both the execution times γ_z and γ_C mainly depend on the size of the SR z, the CPU frequency of the node hosting its elaboration, and the time spent by the SR on that node waiting for the actual computation. Therefore, $\omega_{z,h}$ and $\omega_{z,C}$ represent the queuing time experienced by the the SR z waiting for its execution on the CN h and cloud, respectively[4]. Furthermore, $\rho_{r,h}$ is a binary value equals to 1 if the SR j is executed on the CN h, 0 otherwise. Similarly, $\theta_{v,h}$ is equal to 1 when the VF v is present on CN h, 0 otherwise. Finally, $\zeta_{v,C}$ is equal to 1 if the VF v is loaded on cloud, 0 otherwise. It is important to make evident that the TSA in (7.17) strongly depends on the queuing time experienced by the SR on the service provision site. In fact, a proper deployment of VFs on the ECN may drastically reduce the TSA time.

In formal terms, the aim of this work is the maximization of the SP revenue by providing decision making on the VFs placement, in order to satisfy the SRs. Therefore, the main goal here is given by

$$\min_{\mathbf{q},\mathbf{z}} \sum_{i=1,\ldots,\mathcal{T}} \mathcal{X}(x_i, q_i) + \sum_{j=1,\ldots,\mathcal{M}} \mathcal{Y}(y_j, z_j), \tag{7.18}$$

s.t.

$$T_i \leq \tau_i, \forall i = 1, \ldots, \mathcal{T}, \tag{7.19}$$

[4]The CPU queue has been modeled with the first-in-first-out service policy.

$$\sum_{v \in \mathcal{V}} \theta_{v,h} a_v \leq S, \forall h \in \mathcal{N}, \tag{7.20}$$

$$\sum_{v \in \mathcal{V}} \zeta_{v,C} a_v \leq U. \tag{7.21}$$

In problem (7.18)-(7.20), constraint (7.19) expresses the fact that each SR with high priority has to be served, while constraint (7.20) and (7.21) represent that the VFs allocation has to respect the storage limit of CNs and cloud, respectively.

7.2.5 Federated Learning Framework

7.2.6 The Learning Problem

The aim of ML is the exploitation of some data used for training, to learn models. In order to do that, typically, ML involves the definition of a loss function representing the error implicitly resulting from the model training [256]. The loss function depends on the data sample z and a parameter vector $\mathbf{w}$, and it is named hereafter as $f_z(\mathbf{w})$. As previously introduced, this work supposes the presence of L SRs, with $L = \mathcal{T} + \mathcal{M}$, deriving from an underlying level of EUs, each of which disposes of a local dataset Θ_l, $l = 1, \ldots, L$. Therefore, as assumed in [169, 256], we suppose the collective loss function equals to

$$F_l(\mathbf{w}) = \frac{1}{|\Theta_l|} \sum_{z \in \Gamma_l} f_z(\mathbf{w}), \tag{7.22}$$

where $|\Gamma_l|$ is the number of elements belonging to Γ_l, referred as the cardinality of the Γ_l set. Respectively, the global function evaluated at the central server site, the global loss function, based on the distributed local dataset Θ_l and defined as [169, 256], is expressed by the following relation

$$F(\mathbf{w}) = \frac{\sum\limits_{l=1,\ldots,L} |\Theta_l| F_l(\mathbf{w})}{\sum\limits_{l=1,\ldots,L} |\Theta_l|}. \tag{7.23}$$

Therefore, the objective here is to find $\mathbf{w}^\star$ such that [256]

$$\mathbf{w}^\star = \arg\min F(\mathbf{w}). \tag{7.24}$$

Accordingly with numerous state of art papers [169, 256, 257] recently proposed in literature, the optimization of (7.24) limiting the computational complexity, is pursuit by applying the gradient descent method.

Federated Learning Framework

As represented in Figure 7.22, the proposed FL framework consists of the clients level, responsible for the distributed local data training, and of a server side. The server side is typically represented by a base station or a more general central unit, set up for improving the global learning model, and to merge the locally trained EUs models. The clients and server sides interact to each other, throughout a series of iteration rounds u. It is important to highlight that the number of EUs involved in the training procedure are a subset of the totality of the EUs.

The FL procedure consists of the following steps

- Let $\mathcal{K}$ be the set of the EUs involved in the training process. In parallel, each EU belonging to $\mathcal{K}$, i.e. EU χ, updates its local parameter vector $\mathbf{w}_\chi(u)$, which depends on its local dataset Θ_χ, accordingly with the following rule [257]

$$\mathbf{w}_\chi(u) = \hat{\mathbf{w}}_\chi(u-1) - \xi \nabla F_\chi(\hat{\mathbf{w}}_\chi(u-1)), \tag{7.25}$$

  where ξ is the learning rate and $\hat{\mathbf{w}}_\chi(u-1)$ represents the term $\mathbf{w}_\chi(u-1)$ after global aggregation.

- As detailed in [169], the server side computes the weighted average expressed by

$$\mathbf{w}(u) = \frac{\sum_{\chi \in \mathcal{K}} |\Theta_\chi| \mathbf{w}_\chi}{\sum_{\chi \in \mathcal{K}} |\Theta_\chi|}. \tag{7.26}$$

It is important to make evidence that EUs, in performing distributed data training accordingly with the FL framework, achieve numerous advantages in terms of clients privacy, and limited exploitation of their computational resources. This is directly connected to the fact that training data locally on the clients site, help users to keep their sensitive and personal information reserved, since the uploading of the EU χ parameter vector $\mathbf{w}_\chi$ does not expose the client to any sort of privacy matter. More specifically, from $\mathbf{w}_\chi$, is not elementary to retrieve Θ_χ.

Finally, each algorithm iteration round involves just a part of the whole EUs

set, reducing the message passing between clients and central server entities. Strongly connected with this aspect, the usage of the gradient descent algorithm is able to afford the learning problem without implying an excessive resource consumption, meeting the limited computation capabilities intrinsic of each mobile device.

VFs Placement Planning

Once the FL framework is applied to obtain SRs prediction on the basis of the historical EUs information, properly aggregated by the central server, the VFs placement planning strategy starts. The placement acts on the basis of the VFs popularity, expressed with the popularity vector $\mathbf{p}$.

In order to validate the benefits of the proposed framework to the VFs placement problem, we propose a straightforward placement strategy strictly dependent on $\mathbf{p}$. Supposing that the predicted network SRs are given in terms of VFs popularity and expressed with the popularity vector $\mathbf{p}$, the VFs placement is realized throughout the following steps

1. Process the popularity vector $\mathbf{p}$ starting from the most popular VF in $\mathbf{p}$, i.e., $r^\star$, hence from the most requested VF;
2. Deploy $r^\star$ on the first CN with enough available SRBs to host $r^\star$;
3. Deploy $r^\star$ on the cloud if it has enough available SRBs to host $r^\star$;
4. If $r^\star$ cannot be loaded neither on the CNs nor on the cloud
   (a) if there not exists in $\mathbf{p}$ the VF $\hat{r}$ which can be hosted by a CN or cloud, then terminates placement;
   (b) Otherwise repeat steps $1) - 4)$.

SRs Allocation Planning

The designed SRs allocation policy is based on the matching theory principles [30, 212], and consider the EUs perspective. In order to better explain this point, it is important to highlight that the SRs allocation strategy is based on metrics which do not consider the SP revenue, but only the EUs interests. In this regards, the two parts involved in the matching are the SRs and the computational sites, referred hereafter, for each SR r, as $\mathcal{C}_r$. The set of the computational sites may be different for diverse SRs since, given

the SR r, $\mathcal{C}_r$ consists of the CNs which contain the VF requested by r and of the cloud, if this contains the desired VF. Each SR r expresses the preference in being matched, i.e., in being computed, with each element of $\mathcal{C}_r$ and vice versa. The SRs aim at minimizing their own TSA defined as in (7.17), hence they prefer to be executed on computation sites which lower (7.17). Differently, the computation sites prefer SRs requiring VFs with stringent deadline requirements.

Therefore, the matching algorithm consisting of a modified version of the Gale-Shapley [30] algorithm can be summarized throughout the following steps

1. Each SR builds its preference on the elements belonging to $\mathcal{C}_r$;
2. Each SR r, proposes to be computed on its most preferred computation site;
3. Each computation site, among the received computation proposals, accept the SR requiring the VF type with the closest deadline, and discard the other proposals;
4. Update queuing time on each CN;
5. Update preferences of the unallocated SRs;
6. repeat steps $2) - 6)$ until all the SRs are allocated.

7.2.7 Numerical Results

The proposed FL based framework has been tested by resorting to numerical simulations in the Tensorflow environment. We supposed the network scenario consisting of $\mathcal{N} = 3$ CNs, equipped with a CPU frequency equals to 2.4 GHz, while the cloud has been equipped with a CPU frequency equals to 4.6GHz. Furthermore, we set $S = 70$ and $U = 120$.

The VFs required by SRs have been modeled similarly as in [149,176,285], and we considered the presence of two priority, corresponding to the set MovieLens 1M dataset [107] and MovieLens 100K dataset [107], respectively. We modeled 10 VFs, each of which needs a number of SRBs uniformly distributed in $[50, 80]$.

Each SRs has been modeled as a number of 64 bits format instructions uniformly distributed in $[250, 800]$, needing of 8 CPU cycles per instruction.

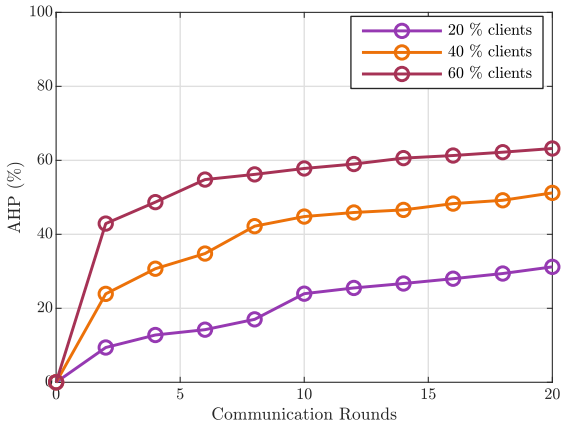

Figure 7.11: SP revenue by varying communication rounds, considering 100 SRs and 20 VFs

Furthermore, as loss function, we adopted the mean squared error (MSE) which, for each data ι_ϕ in Θ_χ, is defined as

$$MSE = \frac{1}{\Phi}\sum_{\phi=1}^{\Phi}(\hat{\iota}_\phi - \iota_\phi)^2, \tag{7.27}$$

where Φ is the number of the samples in test data, and $\hat{\iota}_\phi$ is the predicted value. Then, to test the effectiveness of the proposed approach, we made comparison in terms of accuracy of our strategy, with the prediction scheme based on the application of the CT principles by performing the phase space reconstruction method as explained in [159,239], and by using the predictive model of the k-neighbors discussed in [124]. It is important to note that the CT approach is performed on the central server site, on which all the users data are gathered without considering the preservation of their privacy.

Figure 7.12 and Figure 7.13, which exhibit the MSE behavior by varying the prediction horizon, confirm the greater accuracy of the proposed model in comparison to CT. As it is evident in Figure 7.12 and Figure 7.13, the MSE grows as the prediction horizon increases. This is the direct consequence to the natural difficulty in predicting the long term behavior of the series.

Then, Figure 7.11 makes clear the significant improvement obtained by in-

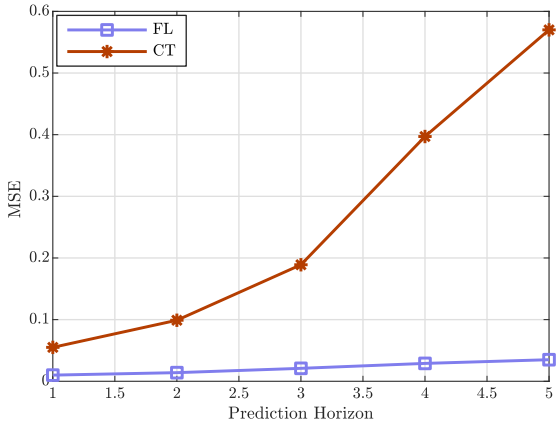

Figure 7.12: MSE by varying the time prediction horizon for type 1 SRs

creasing the number of communication rounds, i.e., information updates, between the server and the clients, for different numbers of EUs involved in the FL process. The direct implication is that higher is the number of the EUs taking part in the learning process, greater is the levels of accuracy on the acquired information on which the VFs placement strategy is based. Moreover, SP revenue improves its trend. Figure 7.14 shows the SP revenue behavior by increasing the number of SRs. As it is straightforward to note, the SP revenue tends to grow by increasing the number of SRs, until the network infrastructure is not saturated and consequently it cannot accept new SRs. Finally, Figure 7.15 depicts the behavior of the percentage of the SRs discarded, i.e., the percentage of the SRs which have not been served by the network infrastructure since their computation is not finished before the expiration of their deadline. In conclusion, the resulting system performance makes clear the validity of the FL application for our problem.

7.2.8 Conclusion

This work has proposed a framework based on the federated learning paradigm to maximize the SP revenue, in a hybrid cloud-MEC network. The framework consists of the FL application to predict the SRs demand, in compliance with the users' privacy. Furthermore, a VFs placement on the basis of the

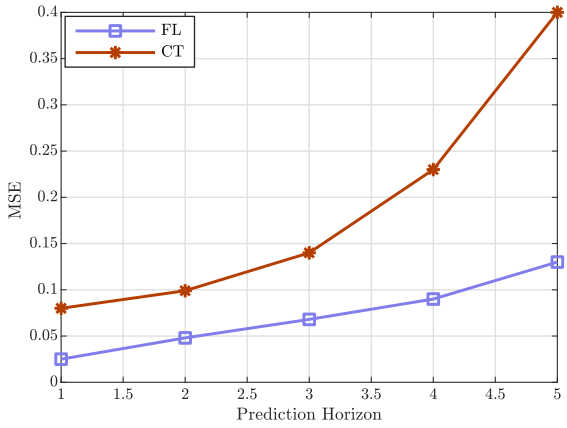

Figure 7.13: MSE by varying the time prediction horizon for type 2 SRs

obtained prediction has been performed, then, a SRs allocation based on matching theory proposed. The VFs placement planning and the SRs allocation strategy have been created to test the actual behavior of the proposed FL framework, in terms of accuracy on the predicted VFs compared to the VFs requested by the SRs. Finally, the framework performance evaluation has been provided, confirming the validity of our proposal, in comparison to an alternative predictive approach based on the CT.

7.3 Price Control for Offloading Services with Chaotic Data

7.3.1 Motivation

Recently, the emergence of next generation network architectures such as mobile edge computing (MEC) [166] and fog computing (FC) [8], in parallel with the exponential growth of users demanding for different classes of services and applications [109, 189], has launched many challenges in a large number of research fields. Heterogeneity on services types can imply not only different quality of service requirements or different service (QoS) strategies, but also, from a service provider perspective, different prices policies. Under

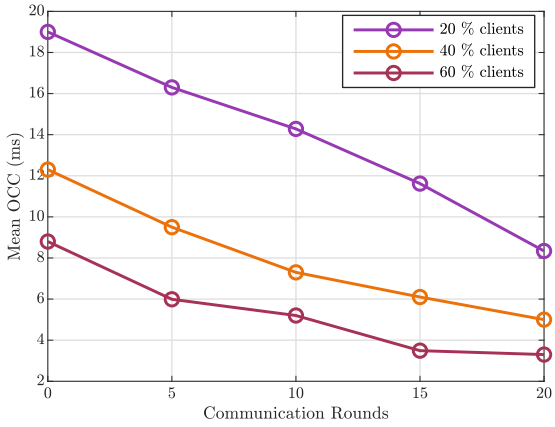

Figure 7.14: SP revenue by varying the number of SRs, considering 10 VFs

this perspective, price control has gained attention as method to influence costumers behaviors through the services price regulation.

7.3.2 Contributions

This study addresses the prices control problem considering the provider perspective and more than one class of service for the computational offloading. Provider has a monopoly on offloading services provision and its price regulation directly impacts the customers demand. For each offloading service type, we consider an associated provision cost, and real sampled data are used to model the service demand. Such time series exhibits a chaotic behavior and an approximation of the state equations is provided by following [293]. Therefore, the main contributions are

- The formulation of dynamic price control problem, with a provider perspective, in terms of finite-horizon optimal control, considering real sampled data to model the equations state;
- The analysis and discussion of a possible analytical approximation of the equations state, by using polynomials with fixed degree;
- The application of the actor-critic NNs based framework proposed in [293], capable of overcoming the unknown system dynamics behind

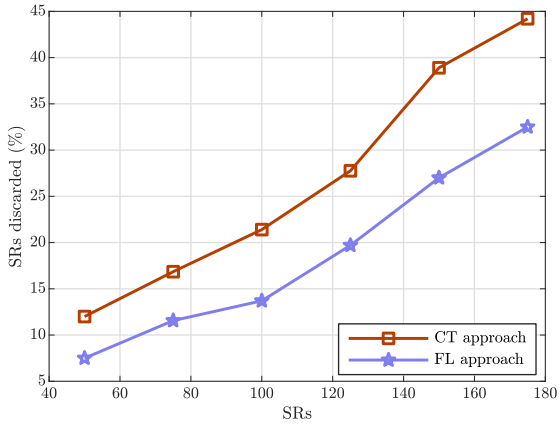

Figure 7.15: Percentage of SRs discarded, by increasing the SRs number

the computational offloading requests samples;

- Numerical simulations to confirm strategy validity. In addition, the price analysis results has been provided for both the analytical and the NNs based approximation of the unknown system dynamics.

7.3.3 Related Works

Maximization of profit by controlling prices is a technique widely used in many business activities [272]. Examples are constituted by power industry, in which higher prices are imposed on customers during hours characterized by massive peaks of electricity consumption, Internet service provider strategy, where the current number of users directly impacts prices. Furthermore, pricing control largely regulates flight tickets and tolls prices [32, 172, 197, 272].

The maximization of the revenue of a system with a M/M/1 queue through the static pricing scheme is pursued in [187], while dynamic programming is applied in [160] and [21], where the long-run average welfare is maximized, considering a M/M/s system in [160], and a M/M/1 system in [21]. In reference to the price control within the queuing networks context, [272] addresses both the dynamic pricing and the service rate control problem, considering an open Jackson network with limited capacity, aim-

ing at maximizing the long-run average welfare. Despite in the literature optimal control is extensively applied to many types of problems, the majority of its application concerns the dynamical system for which the system dynamics is known, and the optimal regulation is reached either by using the Riccati equation [33] or Hamilton-Jacobi-Bellman (HJB) equation [25], or through a backward-in-time and offline strategy. Since the applicability of optimal control strategies is strictly related to the algebraic formulation of the problem, many efforts have been made by control researchers community to provide more flexible resolution tools. Due to the incomplete information about many dynamical systems, the approximate dynamic programming (ADP) has recently gained momentum. Therefore, [79] proposed a constrained ADP framework consisting of an adaptive neural network (NN) controller aiming at guaranteeing a fixed performance level in the formulated linear system. Similarly, ADP is discussed in [224], in which two practical applications to the nonlinear tracking control and the power grid coordination of the heuristic dynamic programming, are presented. In order to propose as much as possible model-free solutions, the Q-learning iteration method is applied for discrete-time linear quadratic optimal control problems in [39] and [263]. Online finite-horizon optimal control problems is discussed in [293], which deals with the NN based optimal control problem in presence of full uncertainty over the system dynamic, and the equations state are approximated with a nonlinear affine function.

7.3.4 Problem Formulation

Let $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^{N}$ be a given scalar time series, in which $\mathbf{d}_i \in \mathbb{R}^c$ and each element $d_{i,j}$ is the number of offloading requests of class j sampled at time i. $\mathcal{D}$ derives from a previous sampling procedure into an urban area modeled as a rectangle R of dimension $A \times B$. Furthermore, each class service request demand r_h in the dataset consists of a quadruple in the form $(r_{h,id}, r_{h,pc}, r_{h,pt}, r_{class})$, in which $r_{h,id}$ is the request ID, $r_{h,pc}$ the origin coordinates, $r_{h,pt}$ is the request time, and $r_{h,class}$ the service request type. In order to evaluate the number of service requests for each service class, requests have been clustered based on the request time. In this respect, time has been partitioned into equal slots. Starting from 0, the k-th time slot is identified by the interval $[k \times \mathcal{F}, (k+1) \times \mathcal{F})$, where $\mathcal{F}$ is the time span of the interval. Hence, since the computational offloading requests outside the city are discarded, the demand at interval k for class type C, i.e., $d_{k,C}$, is

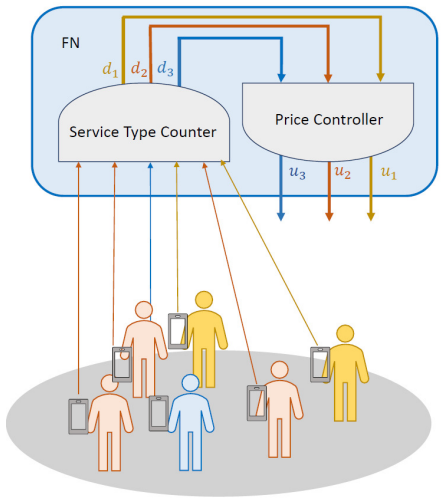

Figure 7.16: Reference scenario

given by

$$d_{k,C} = |\{h \in [0, N) : r_{h,pc} \in R \wedge r_{h,pt} \in \mathcal{T} \wedge r_{h,class} = C\}|, \tag{7.28}$$

where N is the total number of received requests and $\mathcal{T} = [k \times \mathcal{F}, (k+1) \times \mathcal{F}]$. Therefore, $\mathbf{d}_i$ represents the requests for computational offloading for each service class.

The work proposes the study of the maximization of the provider profit in supplying offloading services, by controlling service prices. In this respect, we consider a monopoly condition, in which the effects of market competition on pricing are not taken into account. Consequently, the service provider can influence demand by varying the prices. In a formal term, the study and analysis of the provider profit maximization by controlling the the offloading service prices within a given interval time, can be formulated as a finite-horizon optimal control problem as follows

$$\dot{\mathbf{x}} = F(k, \mathbf{x}_k, \mathbf{u}_k) \tag{7.29}$$

$$\min_{\mathbf{u} \in \mathcal{U}} J(\mathbf{x}_k, k) \tag{7.30}$$

s.t.

$$x_k = \mathbf{d}_k, \quad k = 1, \ldots, N, \tag{7.31}$$

where

$$\begin{aligned} J(\mathbf{x}_k, k) &= \Psi(J(\mathbf{x}_T, T)) + \sum_{k=i}^{T-1} \mathcal{E}(\mathbf{x}_k, \mathbf{u}_k, k) \\ &= \Psi(J(\mathbf{x}_T, T)) + \sum_{k=i}^{T-1} [(\mathbf{u}_k - \mathbf{s}_k)\mathbf{x}_k]^2 dk, \end{aligned} \tag{7.32}$$

$\mathcal{U} = \{u : [0, T] \to \mathbb{R}^J\}$, $\dot{\mathbf{x}} = [d_{k,1}, \cdots, d_{k,J}]^T$, and J is the number of service classes. Furthermore, $\mathbf{u}_k$ represents the prices control variable, where the generic element $u_{k,i}$ expresses the unit price of each service type i at time k, and $\mathbf{u}_k \in \mathbb{R}^J$. Finally, $\mathbf{s}_k \in \mathbb{R}^J$ is the provisioning cost per service, and $\mathbf{s}_{k,i}$ expresses the unit cost of providing service type i. As regards the provision cost $\mathbf{s}_k$, we considered different constant-plus-exponential cost functions [254] for different service types, defined as follows

$$s_{k,i} = \frac{\rho_{i,u} + \rho_{i,v} \nu^{x_{k,i}}}{x_{k,i}}, i = 1, \ldots, J, \tag{7.33}$$

where $\rho_{i,u}, \rho_{i,v}$ and ν are positive parameters, different based on the considered service type. The $\Psi(J(\mathbf{x}_T, T))$ is the terminal term, which basically describes the system behavior while $\mathcal{E}(\mathbf{x}_k, \mathbf{u}_k, k)$ is time-varying and state and control dependent function at the k-th step. Furthermore, since the problem is formulated with a finite horizon, also the control inputs result to be time varying.

Approximation of State Equation

Time series $\mathcal{D}$ exhibits a chaotic behavior, previously verified through the well known phase space reconstruction procedure [124]. In fact, the resulting maximal Lyapunov exponent λ_{max} is equals to 0.5, hence greater than zero, confirming its chaotic behavior which implies nonlinear trend. In Table 7.1 are reported the valued deriving from the analysis of the nonlinear time series $\mathcal{D}$. In particular, the Rosentain's method [209] has been applied to evaluate the largest Lyapunov exponent, while the embedding dimension and the time delay have been found through the application of the false nearest neighbors method and the one based on the study of the mutual information, respectively. It is straightforward to note, from (7.29)-(7.31),

Table 7.1: Time Series Analysis

Maximal Lyapunov exponent	0.5	Rosentain's method
Embedding Dimension	4	False nearest neighbors method
Time Delay	5	Mutual information method

that the lack of knowledge about F dynamics limits the problem resolution. As detailed in [231], we suppose that the ordinary differential equations corresponding to F are approximately expressible in terms of polynomials with a fixed degree. Supposing two polynomial degree, as in [231], and considering $\mathbf{x} = [x_1, x_2, \ldots, x_N]$, a useful representation of F is constituted by

$$f_i(t, \mathbf{x}) = \sum_{j=1}^{N} \sum_{l=j}^{N} e_{i,j,l}(t) x_j x_l + \sum_{j=1}^{N} g_{i,j}(t) x_j + h_i(t). \tag{7.34}$$

The main idea behind the framework proposed in [231] for the approximation of F dynamics, is the chaos synchronization, realized by considering the following system

$$\hat{x}_i = \hat{f}_i(t, \mathbf{x}) = \sum_{j=1}^{N} \sum_{l=j}^{N} \hat{e}_{i,j,l}(t) x_j x_l + \sum_{j=1}^{N} \hat{g}_{i,j}(t) x_j + \hat{h}_i(t), \tag{7.35}$$

and by performing an adaptive approximation to reach a good estimation $\hat{e}_{i,j,l}, \hat{g}_{i,j}, \hat{h}_i$ of real coefficients $e_{i,j,l}, g_{i,j}, h_i$. In order to reach this goal, authors in [231] define the coupling procedure for which

$$\hat{x}_i = \hat{F}(\hat{\mathbf{x}}) + \Gamma(\Theta(\mathbf{x}) - \Theta(\hat{\mathbf{x}})), \tag{7.36}$$

where Θ represent a vector of m values of system state, in our case assumed known. Then, $\Gamma = \gamma \mathbf{I}_n$, with $\gamma \in \mathbb{R}$ and $\mathbf{I}_n$ is the identity matrix. The next step is the synchronization solution procedure, to obtain $x(t) = x'$, by defining the potential

$$\boldsymbol{\xi}_i = \langle M(t) \rangle_\theta, \quad i = 1, \ldots, N, \tag{7.37}$$

where $\langle M(t)\rangle_\theta = \theta \int^t e^{-\theta(t-t')}) M(t')$, $M(t) = [\dot{x}_i - f_i(\hat{x}_1, \ldots, \hat{x}_N)^2]$, and θ represents the bandwidth of the first order filter. It is important to note that system synchronization condition is verified if $\boldsymbol{\xi}_i = 0$, and hence the following relations have to be verified

$$\begin{aligned} \frac{\partial \hat{e}_{ijl}(t)}{\partial t} &= -\delta_e \frac{\partial \boldsymbol{\xi}_i}{\partial \hat{g}_{ij}}, \\ \frac{\partial \hat{g}_{ij}(t)}{\partial t} &= -\delta_g \frac{\partial \boldsymbol{\xi}_i}{\partial \hat{g}_{ij}}, \\ \frac{\partial \hat{h}_i(t)}{\partial t} &= -\delta_h \frac{\partial \boldsymbol{\xi}_i}{\partial \hat{h}_i}, \end{aligned} \tag{7.38}$$

with $\delta_e, \delta_g, \delta_h > 0$. After some algebraic manipulations and, the problem is traced to the resolution of the following $[(N^2/2) + (3N/2) + 1]N$ linear equations

$$\begin{aligned} \sum_{r=1}^{N} \sum_{u=r}^{N} \hat{e}_{i,r,u} \langle \hat{x}_r \hat{x}_u \hat{x}_j \hat{x}_l \rangle_\theta + \sum_{r=1}^{N} \hat{g}_{i,r} \langle \hat{x}_r \hat{x}_j \hat{x}_l \rangle_\theta + \hat{h}_i \langle \hat{x}_j \hat{x}_l \rangle_\theta \\ = \langle \dot{x}_i \hat{x}_j \hat{x}_l \rangle_\theta \end{aligned} \tag{7.39}$$

$$\begin{aligned} \sum_{r=1}^{N} \sum_{u=r}^{N} \hat{e}_{i,r,u} \langle \hat{x}_r \hat{x}_u \hat{x}_j \rangle_\theta + \sum_{r=1}^{N} \hat{g}_{i,r} \langle \hat{x}_r \hat{x}_j \rangle_\theta + \hat{h}_i \langle \hat{x}_j \rangle_\theta \\ = \langle \dot{x}_i \hat{x}_j \rangle_\theta \end{aligned} \tag{7.40}$$

$$\begin{aligned} \sum_{r=1}^{N} \sum_{u=r}^{N} \hat{e}_{i,r,u} \langle \hat{x}_r \hat{x}_u \rangle_\theta + \sum_{r=1}^{N} \hat{g}_{i,r} \langle \hat{x}_r \rangle_\theta + \hat{h}_i \\ = \langle \dot{x}_i \rangle_\theta. \end{aligned} \tag{7.41}$$

Due to the application of the adaptive approximation framework discussed in [231], the reconstruction of an approximated system dynamics is possible, in dependence of state $\mathbf{x}$ and time k. As regards the optimal control formulation problem, the state dynamics requires dependence from also the control $\mathbf{u}$. Unfortunately, a framework capable to retrieve an analytical approximation of F under these conditions, as the best of our knowledge, there not exists. In this respect, we apply the approximated solution proposed in [293] and [274], by supposing F nonlinear with affine dynamics, approximated by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + g(\mathbf{x}_k)\mathbf{u}_k, \tag{7.42}$$

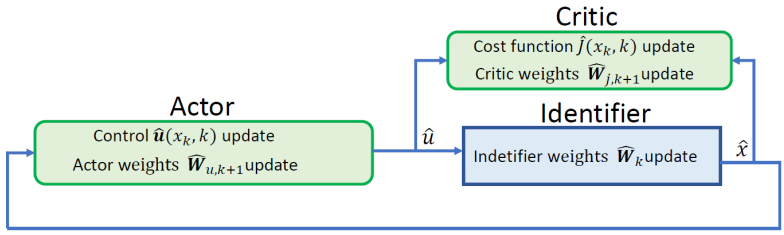

Figure 7.17: NNs scheme flow

in which $f(\mathbf{x}_k) \in \mathbb{R}^n$, $g(\mathbf{x}_k) \in \mathbb{R}^{n \times m}$ are unknown nonlinear dynamics, and $\mathbf{u}_k \in \mathbb{R}^m$ is the control vector. As direct consequence of the Bellman's optimality principle of [142], the optimal control $\mathbf{u}_k^*$ is theoretically obtained from the stationarity condition

$$\frac{\partial J^*(\mathbf{x}_k, k)}{\partial \mathbf{u}_k} = 0, \tag{7.43}$$

where $J^*(\mathbf{x}_k, k)$ in turn is given by

$$J^*(\mathbf{x}_k, k) = \min_{\mathbf{u}_k \in \mathcal{C}} \{[(\mathbf{u}_k - \mathbf{s}_k)\mathbf{x}_k]^2 + J^*(\mathbf{x}_{k+1}, k+1)\}. \tag{7.44}$$

Therefore, from (7.43) and (7.44) we obtain that

$$\mathbf{u}_k^* = \mathbf{s}_k - \frac{1}{2} + g(\mathbf{u}_k)^T \frac{\partial J^*(\mathbf{x}_{k+1}, k+1)}{\partial x_{k+1}}. \tag{7.45}$$

As it is straightforward to note from (7.45), $\mathbf{u}_k^*$ depends on the future state $\mathbf{x}_{k+1}$, that makes (7.45) difficult to tackle. In order to avoid this problem, we apply the online NN based scheme proposed in [293] and [274], based on an actor-critic strategy, consisting of two NNs with the constant weight structure and time-varying activation function. Moreover, the dynamics of the value function is approximated with a critic network. Differently, the actor network generates the control inputs.

7.3.5 Neural Network Solution

NN Identifier

As widely discussed in [293], as consequence of NNs theory [274], the F function can be approximated by

$$\begin{aligned}\mathbf{x}_{k+1} &= f(\mathbf{x}_k) + g(\mathbf{x}_k)\mathbf{u}_k \\ &= \mathbf{W}^T\sigma(x_k)\bar{\mathbf{u}}_k + \bar{\epsilon}_k,\end{aligned} \tag{7.46}$$

where $\mathbf{W} \in \mathbb{R}^{L\times n}$, $\sigma(x_k) = \begin{bmatrix}\sigma_f(x_k) & 0\\ 0 & \sigma_g(x_k)\end{bmatrix}$, $\bar{\mathbf{u}}_\mathbf{k} = \begin{bmatrix}1\\ \mathbf{u}_k\end{bmatrix} \in \mathbb{R}^{m+1}$, and $\bar{\epsilon}_k = \begin{bmatrix}\epsilon_f\\ \epsilon_g\end{bmatrix} \in \mathbb{R}^n$. The number of hidden layers is L, Generally speaking, the k-th state, using the NN identifier and indicating with $\hat{\mathbf{W}}_k^T$ the most recent matrix weights, can be expressed as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{W}}_k^T\sigma(\mathbf{x}_{k-1})\bar{\mathbf{u}}_{k-1}. \tag{7.47}$$

The weights matrix updating rule is given by

$$\hat{\mathbf{W}}_{k+1}^T = \mathbf{\Phi}_k\mathbf{U}_k(\mathbf{U}_k^T\mathbf{\Phi}_k^T\mathbf{\Phi}_k\mathbf{U}_k)^{-1}(\mathbf{X}_{k+1}^T - a\mathbf{\Phi}_k^T), \tag{7.48}$$

in which $0 < a < 1$ and,

$$\mathbf{\Phi}_k = \mathbf{X}_k - \hat{\mathbf{W}}_k^T\mathbf{\Psi}_{k-1}\mathbf{U}_{k-1}, \tag{7.49}$$

where $\mathbf{X}_k = [\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_{k-l}] \in \mathbb{R}^{n\times(l+1)}$, $\mathbf{\Psi}_{k-1} = [\sigma(\mathbf{x}_{k-1}), \sigma(\mathbf{x}_{k-2}), \ldots, \sigma(\mathbf{x}_{k-l-1})]$, and

$$\mathbf{U}_{k-1} = \begin{bmatrix}\bar{\mathbf{u}}_{k-1} & 0 & \ldots & 0\\ 0 & \bar{\mathbf{u}}_{k-2} & 0 & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & \bar{\mathbf{u}}_{k-l-1}\end{bmatrix} \in \mathbb{R}^{(m+1)(l+1)\times(l+1)}$$

.

NN Optimal Controller

As proposed and detailed in [293], the time-varying, finite-horizon optimal control can be realized by approximating the value function and the control inputs with critic NN and an actor one, respectively [274].

Let $\mathbf{W}_J^T$ and $\mathbf{W}_\mathbf{u}^T$ represent the target NN weights, $\sigma_J(\mathbf{x}_k, T-k)$ and $\sigma_\mathbf{u}(\mathbf{x}_k, T-k)$ are the activation functions, $\epsilon_J(\mathbf{x}_k, k)$ and $\epsilon_\mathbf{u}(\mathbf{x}_k, k)$ the NN

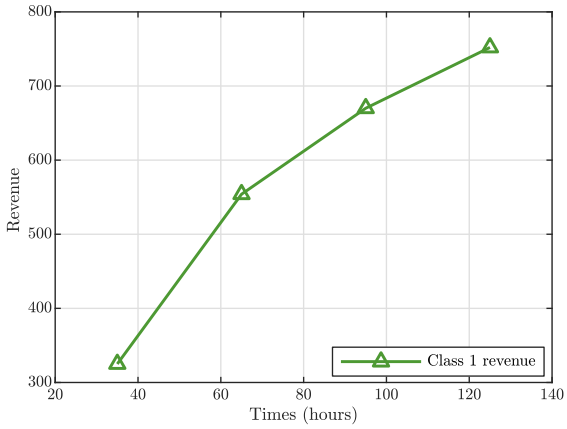

Figure 7.18: The total revenue of service class 1.

corresponding to the reconstruction errors of the critic and actor part, respectively.

Let $\hat{W}$ and $\hat{J}$ be the estimated value function and the target matrix, respectively. Therefore, the value function can be approximated by

$$\hat{J}(\mathbf{x}_k, k) = \hat{\mathbf{W}}_{J,k}^T \sigma_J(\mathbf{x}_k, T-k) \tag{7.50}$$

From [293] results that the critic NN weights are updated by following the relation

$$\hat{\mathbf{W}}_{J,k} - \alpha_J \frac{\sigma_1(\mathbf{x}_k, T-k)\sigma_1^T(\mathbf{x}_k, T-k)\hat{\mathbf{W}}_{J,k}}{1+\sigma_1(\mathbf{x}_k, T-k)\sigma_1(\mathbf{x}_k, T-k)} + \zeta, \tag{7.51}$$

with

$$\zeta = \alpha_J \frac{\sigma_1(\mathbf{x}_k, T-k)(\hat{\sigma}_J^T(\mathbf{x}_T, 0)\hat{\mathbf{W}}_J + \sigma_1(\mathbf{x}_k, k))}{1+\sigma_1^T(\mathbf{x}_k, T-k)\sigma_1(\mathbf{x}_k, T-k)}, \tag{7.52}$$

and α_J defined as in [293].

Optimal Control Feedback

According to [293], the optimal control strategy is designed to minimized (29) and results defined by

$$\hat{\mathbf{u}}(\mathbf{x}_k, k) = \hat{\mathbf{W}}_{\mathbf{u},k} + \sigma_{\mathbf{u}}(\mathbf{x}_k, T-k), \tag{7.53}$$

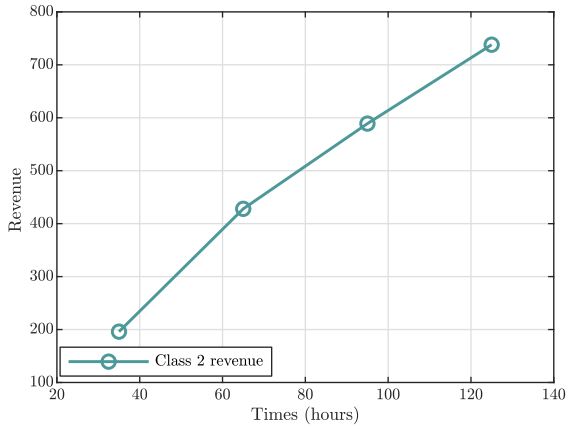

Figure 7.19: The total revenue of service class 2.

in which $\hat{\mathbf{W}}_{\mathbf{u},k}$ represents the estimation of the matrix weight corresponding to the target actor. Then, the updating rule for $\hat{\mathbf{W}}_{\mathbf{u},k}$ is given by

$$\hat{\mathbf{W}}_{\mathbf{u},k+1} = \hat{\mathbf{W}}_{\mathbf{u},k} - \alpha_{\mathbf{u}} \frac{\sigma_{\mathbf{u}}(\mathbf{x}_k, T-k)\breve{\mathbf{u}}^T(\mathbf{x}_k, k)}{1 + \sigma_{\mathbf{u}}^T(\mathbf{x}_k, T-k)\sigma_{\mathbf{u}}(\mathbf{x}_k, T-k)}, \tag{7.54}$$

$\alpha_{\mathbf{u}} > 0$.

7.3.6 Numerical Results

In order to prove the validity of the approach, we show results considering $J = 3$ classes of services, deriving by real data set composed by the samples collected during a period of 130 hours. Furthermore, we have avoided negative provider revenue by imposing that the provider cannot select a service price lower than the production cost necessary for the provision of that service. The service cost has been generated into interval $(0, 1)$. As regards the NN set up, parameters have been set as in [293] for the nonlinear case, while for the $g(\mathbf{x}_k)$ and $f(\mathbf{x}_k)$, we referred to [137]. Figures 7.18, 7.19 and 7.20 depict the revenue trend of each service class, expressed in function of time. As it is evident, its behavior changes based on the service class and time, and it is highly influenced by the number of requests for each type of service. Likewise, Figure 7.21, 7.22 and 7.23 represents the prices behavior.

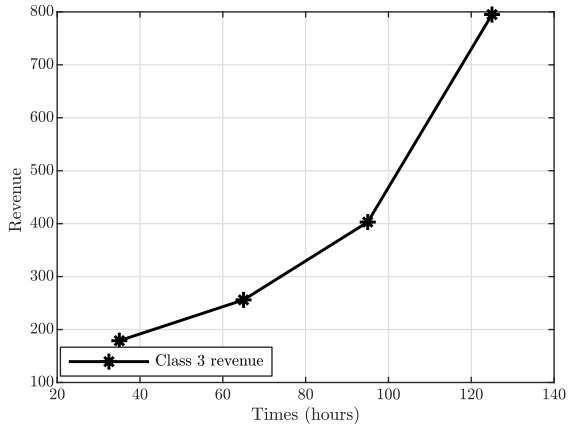

Figure 7.20: The total revenue of service class 3.

Also in this case, prices exhibit a very dynamical nature, that is strictly related to the services demand. In order to provide a complete profit analysis, Figure 7.24 highlights the always increasing cumulative service provider revenue. The figure shows performance comparison among the NNs strategy, the price selection based on the randomized choice of the value price for each service class, and the NNs strategy by considering F analytically approximated as in (7.34). The accuracy in analytically approximating F by following model proposed in (7.34) System performance by using approximation (7.34) is strictly related to the degree of the polynomials chose for (7.34). In this respect, Figure 7.25 expresses, in percentage, the error gap between the approximation of F and its real value, by changing the hypothesis for the reconstruction of its analytical expression. This is expressed in terms of mean absolute percentage error, defined by

$$MAPE = \frac{1}{S}\sum_{i=1}^{S}\left|\frac{\mathbf{d}_i - \hat{\mathbf{d}}_i}{\mathbf{d}_i}\right| \cdot 100, \tag{7.55}$$

where S is the number of simulation trials. As the number of samples increases,the gap grows. Form the other side, higher is the degree of the approximation polynomial, better is the approximation. By employing the F approximation presented in (7.34), overcomes random alternative, the NNs approach based on uncertainty better fits system behavior. This is confirmed

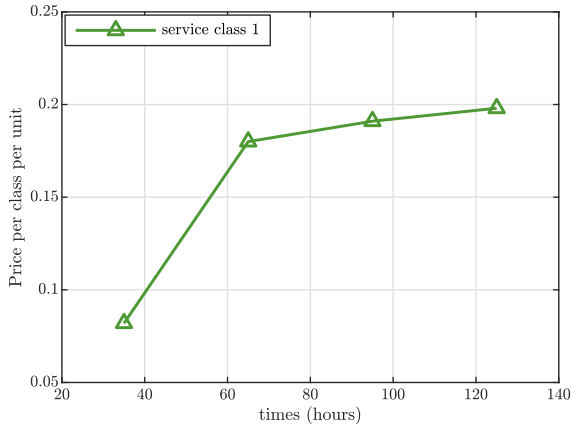

Figure 7.21: Service class 1 price per unit.

by comparing Figures 7.18-7.23 with Figure 7.24. In fact, despite each service revenue curve exhibits a wide value dynamics, the total profit increases. This means that the strategy rules services price in order to improve the provider profit and from Figure 7.24 it is clear the profit improvement by applying the analyzed framework in comparison with the alternatives taken into account.

7.3.7 Conclusion

This work considers the dynamic control prices problem, by supposing a service provider perspective. A market monopoly condition has been assumed, and the provider prices selection directly impacts the customers demand. The prices selection involves different classes of services, and the price regulation is required for each of these classes. The analysis has been conducted on a set of real sampled data with chaotic behavior, and a NN framework has been applied to perform provider revenue maximization. The chaotic dynamics of the sampled data has been approximated through a nonlinear affine state equation. Numerical results exhibit the performance of the approximated strategy and show the prices behavior and an increasing provider revenue trend.

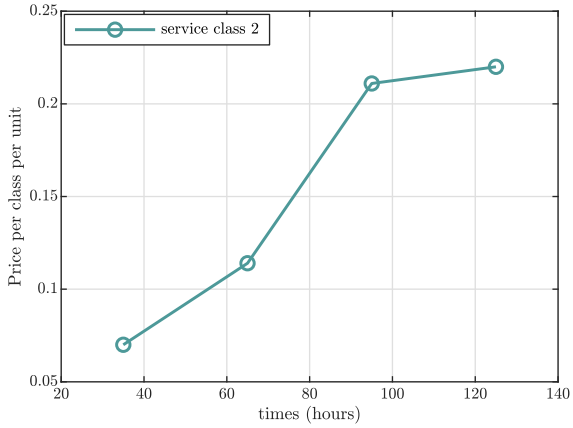

Figure 7.22: Service class 2 price per unit.

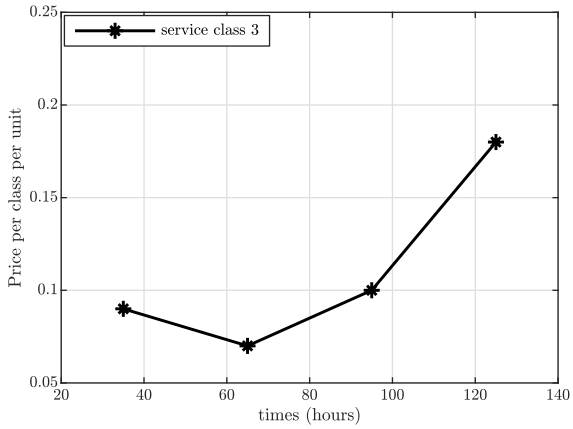

Figure 7.23: Service class 3 price per unit.

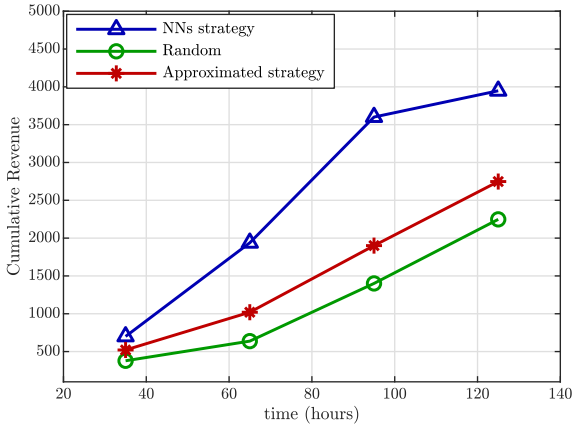

Figure 7.24: Cumulative provider revenue by varying time

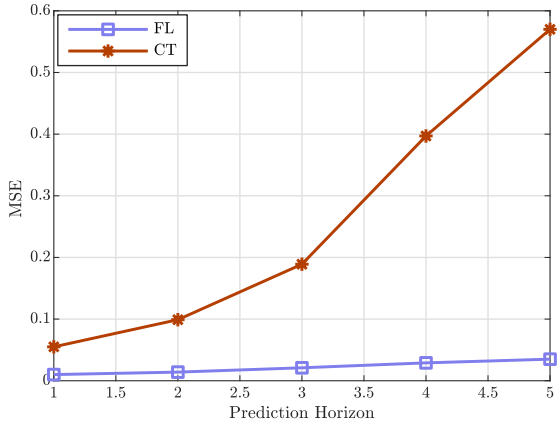

Figure 7.25: MAPE value by increasing the number of considered samples.

Chapter 8

Conclusion and Future Works

Wheresoever you go, go with all your heart.

Chinese Proverb

This dissertation has investigated the limits and potentialities of modern networks, by analyzing some of the major challenges which afflict the today's networks. Such complexity, intrinsic in the nature of novel networks architectures and communication paradigms, has opened the doors towards the re-definition of the term resource. Therefore, if in the past the resource was generally identified by the physical resource as communication channels, central processor units, storage blocks, in the modern wireless networks era the resource assumes a more flexible and heterogeneous meaning, indicating for example the virtual functions, service provider revenue and so on. The wide range of meanings which the term resource may adopt directly impacts on the objectives of the contemporary resource allocation problems. Different goals, even contrasting each other, high system conditions variability, and densification, to name a few, lead towards the substantial necessity for novel or renewed solutions, able to provide good approximations of the emerging multifaceted challenges, powerful strategies, and cross-layer effectiveness to embrace the complexity of modern problems.

During this thesis, several frameworks have been presented and the theoretical insight beyond their formulations provided. Their application to diverse contexts has been suggested, by introducing some adjustments respect to the traditional versions, in order to better meet the ever higher standard performance demand in service provisioning. System performance results

are given and extensively discussed, providing the validity of the proposed frameworks to the problems considered. In addition, a crucial role has been also given to the infrastructure service provider perspective, by taking into account the revenue obtainable from the services supply, in reference to the network infrastructure maintenance costs. In order to perform such a type of analysis, the users perception about received services becomes also a crucial point, since a satisfied user is willing to pay higher prices. In this respect, in the future works, challenge is the integration of the proposed cross-layer frameworks with the users personal perception about service experience. In fact, together with the meaning of the term resource, also the whole network conception is changing, by leaving the too strict approach based on the evaluation of firm QoS constraints, in favor of a more user-centric vision, typical in emerging network paradigms.

Appendix A

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.[1]

International Journals

1. **B. Picano**, R. Fantacci, and H. Zhu, "Nonlinear dynamic chaos theory framework for passenger demand forecasting in city", *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 2019, [DOI: 10.1109/TVT.2019.2930363].
2. F. Chiti, R. Fantacci, and **B. Picano**, "A matching game for tasks offloading in integrated edge-fog computing systems", *EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS*, 2019,[DOI:10.1002/ett.3718].
3. F. Chiti, R. Fantacci, F. Paganelli, and **B. Picano**, "Virtual functions placement with time constraints in fog computing: a matching theory perspective", *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*, 2019, [DOI: 10.1109/TNSM.2019.2918637]
4. G. Bartoli, F. Chiti, R. Fantacci, and **B. Picano**, "An efficient resource allocation scheme for applications in lr-wpans based on a stable matching with externalities approach", *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, vol. 68, no. 6, pp. 5893-5903, June 2019, [DOI: 10.1109/TVT.2019.2909136]
5. F. Chiti, R. Fantacci, and **B. Picano**, "A matching theory framework for tasks offloading in fog computing for iot systems", *IEEE INTERNET OF*

[1]The author's bibliometric indices are the following: H-index = 1, total number of citations = 4 (source: Google Scholar on Month 09, 2019).

THINGS JOURNAL, vol. 5, no. 6, pp. 5089-5096, Dec. 2018 [DOI: 10.1109/JIOT.2018.2871251]

Submitted

1. R. Fantacci, and **B. Picano**, "Performance Analysis of a Delay Constrained Data Offloading Scheme in an Integrated Cloud-Fog-Edge Computing System", to *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*.
2. R. Fantacci, and **B. Picano**, " When Network Slicing Meets Prospect Theory: A Service Provider Revenue Maximization Framework", to *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*.
3. R. Fantacci, and **B. Picano**, "Federated learning for Cloud-Edge Computing: A Service Provider Revenue Maximization Framework", to *IEEE Conference on Computer Communications (INFOCOM)*.
4. R. Fantacci, and **B. Picano**, "A Federated Learning Framework for Mobile Edge Computing Networks", to *CAAI Transactions on Intelligence Technology (CAAI)*.
5. R. Fantacci, and **B. Picano**, "A Matching Game with Discard Policy for Virtual Machines Placement in Hybrid Cloud-Edge Architecture for Industrial IoT Systems", to *IEEE ON INDUSTRIAL INFORMATICS*.

International Conferences

1. **B. Picano**, R. Fantacci, and Z. Han, "Price Control for Computational Offloading Services with Chaotic Data", in *IEEE International Conference on Computing, Networking and Communications (ICNC)*, Hawaii (USA), 2020.
2. R. Fantacci, and **B. Picano**, "Performance Analysis of an Edge Computing System for Real Time Computations and Mobile Users", in *IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, (USA), 2019.
3. **B. Picano**, F. Chiti, R. Fantacci, and Z. Han, "Passengers demand forecasting based on chaos theory", in *IEEE International Conference on Communications (ICC)*, Shanghai (CN), 2019.
4. G. Bartoli, R. Fantacci, D. Marabissi, and **B. Picano**, "Efficient matching for almost blank subframes allocation in ultra dense networks", in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, Marrakesh.
5. F. Chiti, R. Fantacci, **B. Picano**, Y. Gu, X. Du and Z. Han, "A Low Complexity Matching Game Approach for LTE-Unlicensed", *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, ON, 2017.

Bibliography

[1] "Google cloud," *Google Compute Engine Documentation.* [Online]. Available: https://cloud.google.com/compute/docs/

[2] "Book reviews," *IIE Transactions*, vol. 30, no. 9, pp. 855–863, Sep 1998. [Online]. Available: https://doi.org/10.1023/A:1017135818496

[3] "IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) - Amendment 6: TV White Space Between 54 MHz and 862 MHz Physical Layer," *IEEE Std 802.15.4m-2014*, pp. 1–118, Apr. 2014.

[4] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, Apr. 2016.

[5] 3GPP, "Evaluations of RSRP/RSRQ measurement," Third Generation Partnership Project, Standard R4-110284, 2011.

[6] ——, "Lte-advanced," 2013. [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced

[7] ——, 2014. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1603-lte_in_unlicensed

[8] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[9] N. H. Abdullah, R. Adnan, A. M. Samad, and F. Ahmat Ruslan, "Lightning forecasting modelling using artificial neural network (ann): Case study sultan abdul aziz shah airport or skypark subang," in *2018 IEEE Conference on Systems, Process and Control (ICSPC)*, Dec 2018, pp. 1–4.

[10] F. M. Abinader, E. P. L. Almeida, F. S. Chaves, A. M. Cavalcante, R. D. Vieira, R. C. D. Paiva, A. M. Sobrinho, S. Choudhury, E. Tuomaala, K. Doppler, and V. A. Sousa, "Enabling the coexistence of lte and wi-fi in unlicensed bands," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 54–61, Nov 2014.

[11] D. J. Abraham, R. W. Irving, and D. F. Manlove, "Two algorithms for the student-project allocation problem," *Journal of Discrete Algorithms*, vol. 5, no. 1, pp. 73 – 90, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570866706000207

[12] S. Abusayeed, R. Mahbubur, I. Dali, C. Lu, R. Chandra, and J. Liu, "Snow: Sensor network over white spaces," in *ACM SenSys'16*. ACM, November 2016, top 3 Papers. [Online]. Available: https://www.microsoft.com/en-us/research/publication/snow-sensor-network-white-spaces/

[13] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[14] A. Ajith and T. G. Venkatesh, "Qoe enhanced mobile data offloading with balking," *IEEE Communications Letters*, vol. 21, no. 5, pp. 1143–1146, May 2017.

[15] A. Al-Sawaai, I.-U. Awan, and R. J. Fretwell, "Analysis of the weighted fair queuing system with two classes of customers with finite buffer," 05 2009, pp. 218–223.

[16] A. Alheraish, "Design and implementation of home automation system," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 4, pp. 1087–1092, Nov. 2004.

[17] M. Ali, N. Riaz, M. I. Ashraf, S. Qaisar, and M. Naeem, "Joint cloudlet selection and latency minimization in fog networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4055–4063, Sep. 2018.

[18] C. Alippi, R. Fantacci, D. Marabissi, and M. Roveri, "A cloud to the ground: The new frontier of intelligent and autonomous networks of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 14–20, December 2016.

[19] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: an Introduction to Dynamical Systems*. New York, USA: Spinger, 1997.

[20] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1996–2018, Fourthquarter 2014.

[21] B. Ata and S. Shneorson, "Dynamic control of an m/m/1 service system with adjustable arrival and service rates," *Management Science*, vol. 52, no. 11, pp. 1778–1791, Nov. 2006.

[22] S. Athmaja, M. Hanumanthappa, and V. Kavitha, "A survey of machine learning algorithms for big data analytics," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, March 2017, pp. 1–4.

[23] M. Bagher Shahin, A. Doniavi, M. Solimanpur, and M. Shahin, "A novel approach for optimization in a fuzzy finite capacity queuing model with system cost and expected degree of customer satisfaction," *Decision Science Letters*, vol. 4, pp. 487–496, 07 2015.

[24] M. Bahrami, "Cloud computing for emerging mobile cloud apps," in *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, March 2015, pp. 4–5.

[25] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, ser. Modern Birkhäuser Classics. Birkhäuser Boston, 2009. [Online]. Available: https://books.google.it/books?id=dyjrBwAAQBAJ

[26] G. Bartoli, F. Chiti, R. Fantacci, and B. Picano, "An efficient resource allocation scheme for applications in lr-wpans based on a stable matching with externalities approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5893–5903, June 2019.

[27] G. Bartoli, R. Fantacci, D. Marabissi, and B. Picano, "Efficient matching for almost blank subframes allocation in ultra dense networks," April 2019, accepted.

[28] G. Bartoli, R. Fantacci, M. Marabissi, and M. Pucci, "Adaptive muting ratio in enhanced inter-cell interference coordination for lte-a systems," in *IWCMC*, Aug. 2014, pp. 990–995.

[29] A. Basharat and M. Shah, "Time series prediction by chaotic modeling of nonlinear dynamical systems," in *IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, September 2009, pp. 1941–1948.

[30] S. Bayat, Y. Li, L. Song, and Z. Han, "Matching theory: Applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 103–122, Nov 2016.

[31] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997.

[32] G. Bitran and R. Caldentey, "An overview of pricing models for revenue management," *Manufacturing & Service Operations Management*, vol. 5, no. 3, pp. 203–229, Jul. 2003.

[33] S. Bittanti, A. Laub, and J. Willems, *The Riccati Equation*, ser. Communications and Control Engineering. Springer Berlin Heidelberg, 2012. [Online]. Available: https://books.google.it/books?id=6SzvCAAAQBAJ

[34] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, March 2017.

[35] E. Bodine-Baron, C. Lee, B. Chong, Anthonyand Hassibi, and A. Wierman, *Peer Effects and Stability in Matching Markets*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 117–129.

[36] T. E. Bogale and L. B. Le, "Massive mimo and mmwave for 5g wireless hetnet: Potential benefits and challenges," *IEEE Veh. Technol. Mag.*, vol. 11, no. 1, pp. 64–75, Mar 2016.

[37] G. Box, *Box and Jenkins: Time Series Analysis, Forecasting and Control*. London: Palgrave Macmillan UK, 2013, pp. 161–215. [Online]. Available: https://doi.org/10.1057/9781137291264_6

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[39] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. American Control Conference*, Baltimore, MD, USA, Jun. 1994, pp. 3475–3479 vol.3.

[40] L. Brilli, T. Pecorella, L. Pierucci, and R. Fantacci, "A novel 6lowpan-nd extension to enhance privacy in ieee 802.15.4 networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[41] P. M. T. Broersen, "The best order of long autoregressive models for moving average estimation," in *the 8th European Signal Processing Conference (EUSIPCO 1996)*, Trieste, Italy, September 1996.

[42] A. Brogi and S. Forti, "Qos-aware deployment of iot applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, Oct 2017.

[43] M. Cai, F. Cai, A. Shi, B. Zhou, and Y. Zhang, "Chaotic time series prediction based on local-region multi-steps forecasting model," in *Advances in Neural Networks - ISNN*, F.-L. Yin, J. Wang, and C. Guo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 418–423.

[44] A. Castiglione, A. D. Santis, A. Castiglione, F. Palmieri, and U. Fiore, "An energy-aware framework for reliable and secure end-to-end ubiquitous data communications," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, Sep. 2013, pp. 157–165.

[45] D. Celebi, B. Bolat, and D. Bayraktar, "Light rail passenger demand forecasting by artificial neural networks," in *International Conference on Computers Industrial Engineering*, Troyes, France, July 2009, pp. 239–243.

[46] W. Chandramitasari, B. Kurniawan, and S. Fujimura, "Building deep neural network model for short term electricity consumption forecasting," in *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, Aug 2018, pp. 43–48.

[47] K. Chang, J. Gao, M. Chen, and Y. Yuan, "Urban daily water demand short-term forecasts based on the chaos theory," in *Fourth International Conference on Natural Computation*, vol. 4, Jinan, China, October 2008, pp. 696–699.

[48] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, *Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era*, JUNE 2018, vol. 25, no. 3.

[49] 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 Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4590–4602, Oct 2018.

[50] F. Chen, N. Wang, R. German, and F. Dressler, "Performance Evaluation of IEEE 802.15.4 LR-WPAN for Industrial Applications," in *Proc. Fifth Annual Conference on Wireless on Demand Network Systems and Services*, Garnisch-Pertenkirchen, Germany, Jan. 2008, pp. 89–96.

[51] L. Chen, P. Zhou, L. Gao, and J. Xu, "Adaptive fog configuration for the industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4656–4664, Oct 2018.

[52] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, March 2018.

[53] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive d2d collaboration for energy-efficient mobile edge computing," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 64–71, Aug 2017.

[54] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," vol. 3, no. 6, Dec 2016, pp. 854–864.

[55] L. Chisci, F. Papi, T. Pecorella, and R. Fantacci, "An evolutionary game approach to p2p video streaming," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–5.

[56] F. Chiti, R. Fantacci, F. Paganelli, and B. Picano, "Virtual functions placement with time constraints in fog computing: a matching theory perspective," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2019.

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

[58] F. Chiti, R. Fantacci, B. Picano, Y. Gu, X. Du, and Z. Han, "A low complexity matching game approach for lte-unlicensed," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–5.

[59] F. Chiti, R. Fantacci, and A. Tani, "Performance Evaluation of an Adaptive Channel Allocation Technique for Cognitive Wireless Sensor Networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[60] ——, "Performance Evaluation of An Adaptive Channel Allocation Technique for Cognitive Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5351–5363, 2016.

[61] F. Chiti, R. Fantacci, Y. Gu, and Z. Han, "Content sharing in internet of vehicles: Two matching-based user-association approaches," *Vehicular Communications*, vol. 8, pp. 35 – 44, 2017, internet of Vehicles. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214209616300808

[62] F. Chiti, R. Fantacci, and B. Picano, "A matching game for tasks offloading in integrated edge-fog computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 0, no. 0, p. e3718, e3718 ett.3718. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3718

[63] P. Corcoran and S. K. Datta, "Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, Oct 2016.

[64] H. Cui and X. Song, "Research on electricity price forecasting based on chaos theory," in *International Seminar on Future Information Technology and Management Engineering*, Leicestershire, UK, November 2008, pp. 398–401.

[65] Y. Cui, W. Lv, Q. Wang, and B. Du, "Usage demand forecast and quantity recommendation for urban shared bicycles," in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Oct 2018, pp. 238–2388.

[66] R. Dao, G. Wu, G. Zhang, X. Yang, and J. Zhang, "A queueing model without customer waiting applied in flexible production line to optimize the number of servers," in *2011 International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 1, Nov 2011, pp. 105–108.

[67] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, Dec 2016.

[68] J. Ding, X. Xue, and G. Li, "Rss method and fuzzy logic combined vertical handoff decision algorithm," in *IET International Communication Confer-*

ence on Wireless Mobile and Computing (CCWMC 2009), Dec 2009, pp. 137–139.

[69] B. S. Dordevic, S. P. Jovanovic, and V. V. Timcenko, "Cloud computing in amazon and microsoft azure platforms: Performance and service comparison," in *2014 22nd Telecommunications Forum Telfor (TELFOR)*, Nov 2014, pp. 931–934.

[70] Z. Du, Q. Wu, B. Jiang, Y. Xu, and Z. Qin, "Interference-Aware Spectrum Access Self-Organization: A Weighted Graph Game Perspective," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–10, 2017.

[71] A. M. El-Hajj, Z. Dawy, and W. Saad, "A stable matching game for joint uplink/downlink resource allocation in OFDMA wireless networks," in *Proc. IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012, pp. 5354–5359.

[72] S. El Kafhali, K. Salah, and S. Ben Alla, "Performance evaluation of iot-fog-cloud deployment for healthcare services," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, Nov 2018, pp. 1–6.

[73] R. Fantacci, "Performance evaluation of prioritized handoff schemes in mobile cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 485–493, March 2000.

[74] R. Fantacci, T. Pecorella, R. Viti, C. Carlini, and P. Obino, "Enabling technologies for smart building, what's missing?" 10 2013, pp. 1–5.

[75] R. Fantacci and B. Picano, "Performance analysis of an edge computing system for real time computations and mobile users," in *GLOBECOM 2019 - 2019 IEEE Global Telecommunications Conference*, Hawaii, USA, Dec 2019, accepted.

[76] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng, "Device-to-device communications in cellular networks," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 49–55, April 2014.

[77] X. Feng, G. Sun, X. Gan, F. Yang, X. Tian, X. Wang, and M. Guizani, "Cooperative Spectrum Sharing in Cognitive Radio Networks: A Distributed Matching Approach," *IEEE Transactions on Communications*, vol. 62, no. 8, pp. 2651–2664, Aug. 2014.

[78] P. Ferrari, A. Flammini, E. Sisinni, S. Rinaldi, D. Brandão, and M. S. Rocha, "Delay estimation of industrial iot applications based on messaging protocols," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2188–2199, Sep. 2018.

[79] S. Ferrari, J. E. Steck, and R. Chandramohan, "Adaptive feedback control by constrained approximate dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 982–987, Aug. 2008.

[80] H. Flores, P. Hui, P. Nurmi, E. Lagerspetz, S. Tarkoma, J. Manner, V. Kostakos, Y. Li, and X. Su, "Evidence-aware mobile computational offloading," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1834–1850, Aug 2018.

[81] L. Forum, "Lte-u technical report. coexistence study for lte-u sdl," 2015. [Online]. Available: http://www.lteuforum.org/uploads/3/5/6/8/3568127/lte-u_forum_lte-u_technical_report_v1.0.pdf

[82] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A*, vol. 33, no. 2, pp. 1134–1140, February 1986. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.33.1134

[83] K. R. Gadham and T. Ghose, "Importance of social welfare point for the analysis of demand response," in *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, Jan 2016, pp. 182–185.

[84] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962. [Online]. Available: http://www.jstor.org/stable/2312726

[85] P. Gandotra and R. K. Jha, "Device-to-device communication in cellular networks: A survey," *Journal of Network and Computer Applications*, vol. 71, pp. 99 – 117, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516301229

[86] L. Gao, T. H. Luan, B. Liu, W. Zhou, and S. Yu, *Fog Computing and Its Applications in 5G*. Switzerland: Springer, 2017, pp. 571–593.

[87] L. G. U. Garcia, G. W. O. Costa, A. F. Cattoni, K. I. Pedersen, and P. E. Mogensen, "Self-Organizing Coalitions for Conflict Evaluation and Resolution in Femtocells," in *Proc. IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, FL, USA, Dec. 2010, pp. 1–6.

[88] Y. Geng and M. Gao, "Distributed stable marriage with incomplete list and ties using spark," 2015. [Online]. Available: https://stanford.edu/~rezab/classes/cme323/S15/projects/stable_marriage_spark_report.pdf

[89] S. M. T. F. Ghomi and K. Forghani, "Airline passenger forecasting using neural networks and box-jenkins," in *12th International Conference on Industrial Engineering (ICIE)*, Tehran, Iran, January 2016, pp. 10–13.

[90] D. González-Sánchez and O. Hernández-Lerma, "A survey of static and dynamic potential games," *Science China Mathematics*, vol. 59, no. 11, pp. 2075–2102, Nov 2016. [Online]. Available: https://doi.org/10.1007/s11425-016-0264-6

[91] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of Queueing Theory*, 4th ed. New York, NY, USA: Wiley-Interscience, 2008.

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

[93] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: fundamentals and applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, May 2015.

[94] Y. Gu, Y. Zhang, L. X. Cai, M. Pan, L. Song, and Z. Han, "Exploiting Student-Project Allocation Matching for Spectrum Sharing in LTE-Unlicensed," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[95] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Matching and Cheating in Device to Device Communications Underlying Cellular Networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2156–2166, Oct. 2015.

[96] Y. Gu, "Matching Theory Framework for 5G Wireless Communications," Ph.D. dissertation, Faculty of the Electrical and Computer Engineering, Houston, Texas, 2016. [Online]. Available: http://wireless.egr.uh.edu/dissertation.htm

[97] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *CoRR*, vol. abs/1410.6513, 2014. [Online]. Available: http://arxiv.org/abs/1410.6513

[98] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 7, pp. 1645–1660, 2013.

[99] R. A. Guerin, "Channel occupancy time distribution in a cellular radio system," *IEEE Transactions on Vehicular Technology*, vol. 36, no. 3, pp. 89–99, Aug. 1987.

[100] C. Guo, M. Sheng, Y. Zhang, and X. Wang, "A Jain's Index Perspective on $\alpha-$Fairness Resource Allocation over Slow Fading Channels," *IEEE Communications Letters*, vol. 17, no. 4, pp. 705–708, Apr. 2013.

[101] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[102] F. A. Haight, "Queueing with reneging," *Metrika*, vol. 2, no. 1, pp. 186–197, Dec 1959. [Online]. Available: https://doi.org/10.1007/BF02613734

[103] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, Feb 2015.

[104] Z. Han, Y. Gu, and W. Saad, *Matching Theory for Wireless Networks*, ser. Wireless Networks. Springer International Publishing, 2017. [Online]. Available: https://books.google.it/books?id=Uvy_DgAAQBAJ

[105] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjørungnes, *Matching Theory for Wireless Networks*. UK: Springer International Publishing, 2017.

[106] Z. Han, "An optimization theoretical framework for resource allocation over wireless networks," Ph.D. dissertation, 11 2003.

[107] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015. [Online]. Available: http://doi.acm.org/10.1145/2827872

[108] A. M. Hatami, M. Mirmohseni, and F. Ashtiani, "A new data offloading algorithm by considering interactive preferences," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2016, pp. 1–6.

[109] F. He, K. He, P. Liang, and J. Wang, "On-demand service-oriented architecture and standardization," in *Proc. IEEE International Conference on Services Computing*, Washington, DC, USA, Jul. 2011, pp. 739–740.

[110] J. He, Z. Tang, H. H. Chen, and Q. Zhang, "An accurate and scalable analytical model for IEEE 802.15.4 slotted CSMA/CA networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 440–448, Jan. 2009.

[111] Y. He, J. Deng, and H. Li, "Short-term power load forecasting with deep belief network and copula models," in *9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, August 2017, pp. 191–194.

[112] J. L. Hennessy and D. A. Patterson, *Computer Architecture, Fifth Edition: A Quantitative Approach*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[113] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.

[114] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized

handoff procedures," *IEEE Transactions on Vehicular Technology*, vol. 35, no. 3, pp. 77–92, Aug. 1986.

[115] S. Hong, "Optimal server allocation for real time computing systems with bursty priority jobs," in *Proceedings Second International Workshop on Real-Time Computing Systems and Applications*, Oct 1995, pp. 218–223.

[116] T. Hu, G. Zheng, and T. Liao, "Smart mobility: Evaluation of demand-responsive transit systems in chiayi city," in *International Smart Cities Conference (ISC2)*, Wuxi, China, September 2017.

[117] A. Ihler, J. Hutchins, and P. Smyth, "Adaptive event detection with time-varying poisson processes," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06, New York, NY, January 2006, pp. 207–216. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150428

[118] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared systems," Digital Equipment Corporation, DEC-TR-301, tech. rep., 1984.

[119] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 352–357.

[120] L. Jiang, H. Tian, Z. Xing, K. Wang, K. Zhang, S. Maharjan, S. Gjessing, and Y. Zhang, "Social-aware energy harvesting device-to-device communications in 5g networks," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 20–27, August 2016.

[121] Y. Jiao, P. Wang, D. Niyato, M. Abu Alsheikh, and S. Feng, "Profit maximization auction and data management in big data markets," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2017, pp. 1–6.

[122] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, pp. 1–1, 03 2019.

[123] Y. Jin and Y. Wen, "When cloud media meet network function virtualization: Challenges and applications," *IEEE MultiMedia*, vol. 24, no. 3, pp. 72–82, 2017.

[124] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge: Cambridge University Press, 2003.

[125] S. Kawauchi, H. Sugihara, and H. Sasaki, "Development of very-short-term load forecasting based on chaos theory," *Electrical Engineering in Japan*, vol. 148, no. 2, pp. 55–63. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/eej.10322

[126] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A*, vol. 45, no. 6, pp. 3403–3411, March 1992. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.45.3403

[127] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *2011 9th IEEE International Conference on Industrial Informatics*, July 2011, pp. 410–415.

[128] N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt, "Wireless sensors networks for Internet of Things," in *Proc. IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, Apr. 2014, pp. 1–6.

[129] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," 2019, pp. 1–1.

[130] J. Kim and Y. Kwon, "Interference-aware topology control for low rate wireless personal area networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 1, pp. 97–104, Feb. 2009.

[131] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[132] C. Kongcharoen and T. Kruangpradit, "Autoregressive integrated moving average with explanatory variable (arimax) model for thailand export," June 2013.

[133] T. Kudo and T. Ohtsuki, "Cell selection using distributed q-learning in heterogeneous networks," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Oct 2013, pp. 1–6.

[134] J. Kwak, O. Choi, S. Chong, and P. Mohapatra, "Processor-network speed scaling for energy for delay tradeoff in smartphone applications," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1647–1660, June 2016.

[135] J. Kwak, Y. Kim, J. Lee, and S. Chong, "Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, Dec 2015.

[136] N. Lata, *Channel Assignment and Graph Multicoloring*. Wiley-Blackwell, 2002, ch. 4, pp. 71–94.

[137] S. Lavenberg, *Computer Performance Modeling Handbook*, ser. Notes and reports in computer science and applied mathematics. Academic Press, 1983.

[138] P. Law, W. Wu, Y. Zheng, and H. Qu, "VisMatchmaker: Cooperation of the User and the Computer in Centralized Matching Adjustment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 231–240, Jan. 2017.

[139] H. Lee, S. Vahid, and K. Moessner, "A survey of radio resource management for spectrum aggregation in lte-advanced," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 745–760, Second 2014.

[140] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, April 2013.

[141] S. Lee and D. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," vol. 1678, pp. 179–188, November 1999.

[142] F. Lewis, *Optimal control*. Wiley, 1986.

[143] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service popularity-based smart resources partitioning for fog computing-enabled industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4702–4711, Oct 2018.

[144] G. Li, C. Zhu, and X. Li, "Application of chaos theory and wavelet to modeling the traffic of wireless sensor networks," in *International Conference on Biomedical Engineering and Computer Science*, Wuhan, China, April 2010.

[145] J. Li, H. Peng, L. Liu, G. Xiong, B. Du, H. Ma, L. Wang, and M. Zakirul Alam Bhuiyan, "Graph cnns for urban traffic passenger flows prediction," in *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct 2018, pp. 29–36.

[146] L. Li, K. Ota, and M. Dong, "Human in the loop: Distributed deep model for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4957–4964, Dec 2018.

[147] M. Li, Y. Sun, H. Huang, J. Yuan, Y. Du, Y. Bao, and Y. Luo, "Profit maximization resource allocation in cloud computing with performance guarantee," in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, Dec 2017, pp. 1–2.

[148] S. Li, J. Huang, and S. R. Li, "Dynamic profit maximization of cognitive mobile virtual network operator," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 526–540, March 2014.

[149] S. Li, J. Xu, M. van der Schaar, and W. Li, "Popularity-driven content caching," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[150] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in sdn-based industrial internet of things with edge computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, June 2018.

[151] X. Li, Y. Liu, H. Ji, H. Zhang, and V. C. M. Leung, "Optimizing resources allocation for fog computing-based internet of things networks," *IEEE Access*, vol. 7, pp. 64 907–64 922, 2019.

[152] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless Networks*, vol. 23, no. 1, pp. 23–41, Jan 2017. [Online]. Available: https://doi.org/10.1007/s11276-015-1133-7

[153] K. Liang, L. Zhao, X. Zhao, Y. Wang, and S. Ou, "Joint resource allocation and coordinated computation offloading for fog radio access networks," *China Communications*, vol. 13, no. Supplement2, pp. 131–139, N 2016.

[154] F. Liberati, A. Giuseppi, A. Pietrabissa, V. Suraci, A. D. Giorgio, M. Trubian, D. Dietrich, P. Papadimitriou, and F. D. Priscoli, "Stochastic and exact methods for service mapping in virtualized network infrastructures," *International Journal of Network Management*, vol. 27, no. 6, p. e1985, 2017, e1985 nem.1985. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.1985

[155] F. Liu, F. Xu, and S. Yang, "A flood forecasting model based on deep learning algorithm via integrating stacked autoencoders with bp neural network," in *IEEE Third International Conference on Multimedia Big Data (BigMM)*, Laguna Hills, CA, April 2017, pp. 58–61.

[156] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang, "Mobile edge cloud system: Architectures, challenges, and approaches," vol. 12, no. 3, Sep. 2018, pp. 2495–2508.

[157] X. Liu, X. Fang, Z. Qin, C. Ye, and M. Xie, "A short-term forecasting algorithm for network traffic based on chaos theory and svm," *Journal of Network and Systems Management*, vol. 19, no. 4, pp. 427–447, December 2011. [Online]. Available: https://doi.org/10.1007/s10922-010-9188-3

[158] Z. Liu, H. Yang, and M. Lai, "Electricity price forecasting model based on chaos theory," in *International Power Engineering Conference*, Leicestershire, UK, November 2005.

[159] Z. Liu, "Chaotic time series analysis," *Mathematical Problems in Engineering*, February 2010.

[160] D. W. Low, "Optimal dynamic pricing policies for an m/m/s queue," *Operations Research*, vol. 22, no. 3, pp. 545–561, 1974.

[161] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, "Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach," pp. 1–6, May 2018.

[162] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[163] D. Manlove, *Algorithmics Of Matching Under Preferences*, ser. Theoretical computer science. World Scientific Publishing, 2013. [Online]. Available: http://eprints.gla.ac.uk/79820/

[164] Z. A. Mann, "Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 11:1–11:34, Aug. 2015. [Online]. Available: http://doi.acm.org/10.1145/2797211

[165] J. Mao, X. Zhang, and J. Li, "Wind power forecasting based on chaos and wavelet packet theory," in *13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, May 2018, pp. 1604–1608.

[166] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[167] D. Marabissi, G. Bartoli, and A. Stomaci, "Low-complexity distributed cell-specific bias calculation for load balancing in udns," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 1056–1060, Jan 2019.

[168] V. Marbukh, "Systemic risks in the cloud computing model: Complex systems perspective," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, June 2016, pp. 863–866.

[169] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[170] F. Mehmeti and T. Spyropoulos, "Is it worth to be patient? analysis and optimization of delayed mobile data offloading," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 2364–2372.

[171] Y. Mehmood, N. Haider, M. Imran, A. Timm-Giel, and M. Guizani, "M2m communications in 5g: State-of-the-art architecture, recent advances, and

research challenges," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 194–201, 2017.

[172] H. Mendelson, "Pricing computer services: queueing effects," *Commun. ACM*, vol. 28, pp. 312–321, Mar. 1985.

[173] Y. Meng, X. Liu, and M. Zhou, "A novel model to determine the optimal number of servers in finite input source fuzzy queueing system," in *2017 36th Chinese Control Conference (CCC)*, July 2017, pp. 4175–4180.

[174] D. R. Miller, "Matrix-geometric solutions in stochastic models: An algorithmic approach, by marcel r. neuts, the johns hopkins university press, baltimore, 1981, 352 pp. price: $32.50," *Networks*, vol. 13, no. 2, pp. 311–312, 1983. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230130219

[175] D. Mittal, U. N. Kar, and D. K. Sanyal, "A novel matching theory-based framework for computation offloading in device-to-device communication," in *2017 14th IEEE India Council International Conference (INDICON)*, Dec 2017, pp. 1–6.

[176] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Transactions on Wireless Communications*, 2016.

[177] N. Mohamed, J. Al-Jaroodi, S. Lazarova-Molnar, I. Jawhar, and S. Mahmoud, "A service-oriented middleware for cloud of things and fog computing supporting smart city applications," in *IEEE SmartWorld Congress*, San Francisco, CA, August 2017.

[178] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2923–2960, Fourthquarter 2018.

[179] P. Moll, M. Lux, S. Theuermann, and H. Hellwagner, "A network traffic and player movement model to improve networking for competitive online games," in *Proc 16th Annual Workshop on Network and Systems Support for Games (NetGames)*, Jun. 2018, pp. 1–6.

[180] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Time Series Analysis and Forecasting*, 2nd ed. Cambridge, UK: Wiley, 2015.

[181] C. K. Moorthy and B. G. Ratcliffe, "Short term traffic forecasting using time series methods," *Transportation Planning and Technology*, vol. 12, no. 1, pp. 45–56, July 1988.

[182] R. Morabito, I. Farris, A. Iera, and T. Taleb, "Evaluating performance of containerized iot services for clustered devices at the network edge," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1019–1030, Aug 2017.

[183] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, September 2013.

[184] H. Mori and S. Urano, "Short-term load forecasting with chaos time series analysis," in *Proceedings of International Conference on Intelligent System Application to Power Systems*, Jan 1996, pp. 133–137.

[185] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. [Online]. Available: http://www.jstor.org/stable/2098689

[186] E. A. Nadaraya, "On estimating regression," *Theory Pb. Appl.*, 1964.

[187] P. Naor, "The regulation of queue size by levying tolls," *Econometrica*, vol. 37, no. 1, pp. 15–24, 1969. [Online]. Available: http://www.jstor.org/stable/1909200

[188] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed petri nets," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[189] K. Nishimatsu, A. Inoue, and T. Kurosawa, "Service-demand-forecasting method using multiple data sources," in *12th International Telecommunications Network Strategy and Planning Symposium*, New Delhi, India, Nov. 2006.

[190] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multihop device-to-device communications," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 56–65, April 2014.

[191] D. Niyato, P. Wang, H. P. Tan, W. Saad, and D. I. Kim, "Cooperation in Delay-Tolerant Networks With Wireless Energy Transfer: Performance Analysis and Optimization," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3740–3754, Aug. 2015.

[192] E. Ott, *Chaos in Dynamical Systems*. Cambridge: Cambridge University Press, 2002.

[193] A. Outamazirt, K. Barkaoui, and D. Aïssani, "Maximizing profit in cloud computing using m/g/c/k queuing model," in *2018 International Symposium on Programming and Systems (ISPS)*, April 2018, pp. 1–6.

[194] S. Pandi, R. S. Schmoll, P. J. Braun, and F. H. P. Fitzek, "Demonstration of mobile edge cloud for tactile internet using a 5g gaming application," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2017, pp. 607–608.

[195] 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 Communications Conference (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013, pp. 4483–4488.

[196] M. Papadopouli, E. Raftopoulos, and H. Shen, "Evaluation of short-term traffic forecasting algorithms in wireless networks," in *2nd Conference on Next Generation Internet Design and Engineering*, Valencia, Spain, April 2006.

[197] I. C. Paschalidis and J. N. Tsitsiklis, "Congestion-dependent pricing of network services," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 171–184, Apr. 2000.

[198] M. Pečar and G. Papa, "Transportation problems and their potential solutions in smart cities," in *International Conference on Smart Systems and Technologies (SST)*, Osijek, Croatia, October 2017, pp. 195–199.

[199] C. Pham, N. H. Tran, S. Ren, W. Saad, and C. S. Hong, "Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach," *IEEE Transactions on Services Computing*, 2017.

[200] Q. Pham, T. Leanh, N. H. Tran, B. J. Park, and C. S. Hong, "Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach," *IEEE Access*, vol. 6, pp. 75 868–75 885, 2018.

[201] B. Picano, F. Chiti, R. Fantacci, and Z. Han, "Passengers demand forecasting based on chaos theory," in *IEEE International Conference on Communications (ICC)*, Shanghai, CN, May 2019, accepted.

[202] B. Picano, R. Fantacci, and Z. Han, "Nonlinear dynamic chaos theory framework for passenger demand forecasting in smart city," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2019.

[203] L. Pierucci and D. Micheli, "A neural network for quality of experience estimation in mobile communications," *IEEE MultiMedia*, vol. 23, no. 4, pp. 42–49, Oct 2016.

[204] L. Pu, X. Chen, J. Xu, and X. Fu, "D2d fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted d2d collaboration," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3887–3901, Dec 2016.

[205] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, Singapore, December 2014.

[206] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported internet of things," vol. 7, 2019, pp. 69 194–69 201.

[207] S. Rhee, "Catalyzing the internet of things and smart cities: Global city teams challenge," in *1st International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE) in partnership with Global City Teams Challenge (GCTC) (SCOPE - GCTC)*, Vienna, Austria, April 2016.

[208] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, May 2017.

[209] M. Rosenstein, J. Collins, and C. D. Luca, "A practical method for calculating largest lyapunov exponents from small data set," vol. 65, no. 1-2, pp. 117–134, May 1993.

[210] E. Rotem, A. Naveh, D. Rajwan, A. Ananthakrishnan, and E. Weissmann, "Power management architecture of the 2nd generation intel xae; core microarchitecture, formerly codenamed sandy bridge," in *2011 IEEE Hot Chips 23 Symposium (HCS)*, Aug 2011, pp. 1–33.

[211] A. E. Roth, "Deferred acceptance algorithms: history, theory, practice, and open questions," *International Journal of Game Theory*, vol. 36, no. 3, pp. 537–569, Mar 2008. [Online]. Available: https://doi.org/10.1007/s00182-008-0117-6

[212] A. E. Roth and M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, UK, 1990.

[213] S. Sadr and R. Adve, "Hierarchical resource allocation in femtocell networks using graph algorithms," in *Proc. IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012, pp. 4416–4420.

[214] J. Saldana, G. Marfia, and M. Roccetti, "Satisfying the hunger for mobile online games: Providing quality time in vehicular scenarios," in *2012 11th Annual Workshop on Network and Systems Support for Games (NetGames)*, Nov 2012, pp. 1–2.

[215] A. G. Salman, B. Kanigoro, and Y. Heryadi, "Weather forecasting using deep learning techniques," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Depok, Indonesia, October 2015, pp. 281–285.

[216] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm

through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[217] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, "Incremental deployment and migration of geo-distributed situation awareness applications in the fog," in *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, ser. DEBS '16. New York, NY, USA: ACM, 2016, pp. 258–269. [Online]. Available: http://doi.acm.org/10.1145/2933267.2933317

[218] A. B. Sediq, R. H. Gohary, and H. Yanikomeroglu, "Optimal tradeoff between efficiency and Jain's fairness index in resource allocation," in *Proc. IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, Sydney, Australia, Sep. 2012, pp. 577–583.

[219] P. Semasinghe, S. Maghsudi, and E. Hossain, "Game theoretic mechanisms for resource management in massive wireless iot systems," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 121–127, February 2017.

[220] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for iot systems: A computation offloading game," *CoRR*, vol. abs/1710.06089, 2017.

[221] X. Shan, H. Zhi, P. Li, and Z. Han, "A survey on computation offloading for mobile edge computing information," in *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, May 2018, pp. 248–251.

[222] P. Shuai, T. En, J. Huilin, P. Zhiwen, L. Nan, and Y. Xiaohu, "An improved graph coloring based small cell discovery scheme in LTE hyper-dense networks," in *Proc. IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, New Orleans, LA, USA, Mar. 2015, pp. 17–22.

[223] N. Shukla, "Machine learning with tensorflow," 2018.

[224] J. Si and L. Yang, "Approximate dynamic programming for continuous state and control problems," in *Proc. 17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, Jun. 2009, pp. 1415–1420.

[225] S. Singh, "Optimize cloud computations using edge computing," in *2017 International Conference on Big Data, IoT and Data Science (BID)*, Dec 2017, pp. 49–53.

[226] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov 2018.

[227] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards qos-aware fog service placement," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, May 2017, pp. 89–96.

[228] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized iot service placement in the fog," *Serv. Oriented Comput. Appl.*, vol. 11, no. 4, pp. 427–443, Dec. 2017. [Online]. Available: https://doi.org/10.1007/s11761-017-0219-8

[229] V. Smith, C. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," vol. abs/1705.10467, 2017. [Online]. Available: http://arxiv.org/abs/1705.10467

[230] E. S. Sopin, A. V. Daraseliya, and L. M. Correia, "Performance analysis of the offloading scheme in a fog computing system," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Nov 2018, pp. 1–5.

[231] F. Sorrentino and E. Ott, "Using synchronization of chaos to identify the dynamics of unknown systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 3, p. 033108, 2009.

[232] V. B. C. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–5.

[233] S. Sthapit, J. R. Hopgood, and J. Thompson, "Distributed computational load balancing for real-time applications," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 1385–1189.

[234] P. Subramaniam and M. J. Kaur, "Review of security in mobile edge computing with deep learning," pp. 1–5, March 2019.

[235] C. S. Sum, L. Lu, M. T. Zhou, F. Kojima, and H. Harada, "Design considerations of IEEE 802.15.4m low-rate WPAN in TV white space," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 74–82, Apr. 2013.

[236] C. S. Sum, M. T. Zhou, L. Lu, R. Funada, F. Kojima, and H. Harada, "IEEE 802.15.4m: The first low rate wireless personal area networks operating in TV white space," in *Proc. 18th IEEE International Conference on Networks (ICON)*, Singapore, Dec. 2012, pp. 326–332.

[237] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4692–4701, 2018.

[238] M. T. Rosenstein, J. J. Collins, and C. De Luca, "A practical method for calculating largest lyapunov exponents from small data set," vol. 65, no. 1-2, pp. 117–134, May 1993.

[239] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, D. Rand and L.-S. Young, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 366–381.

[240] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, thirdquarter 2017.

[241] M. Taneja and A. Davy, "Resource aware placement of iot application modules in fog-cloud computing paradigm," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*. IEEE, 2017, pp. 1222–1228.

[242] C. Tang, X. Wei, S. Xiao, W. Chen, W. Fang, W. Zhang, and M. Hao, "A mobile cloud based scheduling strategy for industrial internet of things," *IEEE Access*, vol. 6, pp. 7262–7275, 2018.

[243] N. T. Ti and L. B. Le, "Joint resource allocation, computation offloading, and path planning for uav based hierarchical fog-cloud mobile systems," in *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*, July 2018, pp. 373–378.

[244] H. Tianfield, "Towards edge-cloud computing," in *2018 IEEE International Conference on Big Data (Big Data)*, Dec 2018, pp. 4883–4885.

[245] N. H. Tran, W. Bao, A. Zomaya, N. Minh N.H., and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, April 2019, pp. 1387–1395.

[246] K. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Jul. 2016.

[247] ——, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, 07 2016.

[248] T. Tuor, S. Wang, T. Salonidis, B. J. Ko, and K. K. Leung, *Demo abstract: Distributed machine learning at resource-limited edge nodes*, April 2018, pp. 1–2.

[249] L. Valerio, A. Passarella, and M. Conti, "Optimal trade-off between accuracy and network cost of distributed learning in mobile edge computing: An analytical approach," pp. 1–9, June 2017.

[250] G. Wang, L. Xiong, and C. Yuan, "Resource allocation for device-to-device communications based on guard area underlying cellular networks," *Chinese Journal of Electronics*, vol. 26, no. 6, pp. 1297–1301, 2017.

[251] H. Wang and D. Chi, "Short-term load prediction based on chaos time series theory," in *Second International Conference on Intelligent Computation Technology and Automation*, vol. 2, Changsha, China, October 2009, pp. 189–192.

[252] M. Wang, L. Zhu, L. T. Yang, M. Lin, X. Deng, and L. Yi, "Offloading-assisted energy-balanced iot edge node relocation for confident information coverage," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[253] S. Wang, X. Huang, Y. Liu, and R. Yu, "Cachinmobile: An energy-efficient users caching scheme for fog computing," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, July 2016, pp. 1–6.

[254] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," vol. 91, 06 2015, pp. 1–9.

[255] S. Wang, A. Zhou, F. Yang, and R. N. Chang, "Towards network-aware service composition in the cloud," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2017.

[256] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," vol. abs/1804.05271, 2018. [Online]. Available: http://arxiv.org/abs/1804.05271

[257] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. S. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 63–71.

[258] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. S. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," *CoRR*, vol. abs/1506.05261, 2015. [Online]. Available: http://arxiv.org/abs/1506.05261

[259] W. Wang, Y. Zhao, M. Tornatore, A. Gupta, J. Zhang, and B. Mukherjee, "Virtual machine placement and workload assignment for mobile edge computing," in *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, Sept 2017, pp. 1–6.

[260] W. Wang and W. Zhou, "Computational offloading with delay and capacity constraints in mobile edge," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[261] Y. Wang, L. Zhu, and L. Zhu, "Social welfare model of standard competition," in *2010 2nd IEEE International Conference on Information Management and Engineering*, April 2010, pp. 15–18.

[262] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in

IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, April 2019, pp. 2512–2520.

[263] Q. L. Wei, H. G. Zhang, and L. L. Cui, "Data-based optimal control for discrete-time zero-sum games of 2-d systems using adaptive critic designs," *Acta Automatica Sinica*, vol. 35, no. 6, pp. 682–692, Jun. 2009.

[264] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, Mar 2017.

[265] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030 – 1081, October 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207014001083

[266] M. Whaiduzzaman, A. Naveed, and A. Gani, "Mobicore: Mobile device based cloudlet resource enhancement for optimal task response," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 144–154, Jan 2018.

[267] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process : Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, November / December 2003.

[268] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, March 2017.

[269] V. W. S. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, *Mobile Edge Computing*. Cambridge University Press, 2017, p. 76–91.

[270] H. Wu and K. Wolter, "Stochastic analysis of delayed mobile offloading in heterogeneous networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 461–474, Feb 2018.

[271] M. Wu and L. Tan, "An infinite-source m/m/s retrial queuing network model with balking and impatient customers," in *2013 IEEE Global High Tech Congress on Electronics*, Nov 2013, pp. 25–29.

[272] L. Xia and S. Chen, "Dynamic pricing control for open queueing networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3290–3300, Oct. 2018.

[273] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4585–4600, June 2019.

[274] H. Xu, A. Sahoo, and J. Sarangapani, *Neural network control of nonlinear discrete-time systems in affine form in the presence of communication network*, Aug. 2014, pp. 151–192.

[275] H. Xu and B. Li, "Anchor: A versatile and efficient framework for resource management in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1066–1076, 2013.

[276] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2572–2581, Aug 2018.

[277] J. Xu, J. Wang, Y. Zhu, Y. Yang, X. Zheng, S. Wang, L. Liu, K. Horneman, and Y. Teng, "Cooperative distributed optimization for the hyper-dense small cell deployment," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 61–67, May 2014.

[278] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[279] B. Yang, G. Mao, M. Ding, X. Ge, and X. Tao, "Dense Small Cell Networks: From Noise-Limited to Dense Interference-Limited," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4262–4277, May 2018.

[280] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," Dec. 2018.

[281] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," vol. abs/1902.04885, 2019. [Online]. Available: http://arxiv.org/abs/1902.04885

[282] K. Yao, Q. Wu, Y. Xu, and J. Jing, "Distributed ABS-Slot Access in Dense Heterogeneous Networks: A Potential Game Approach With Generalized Interference Model," *IEEE Access*, vol. 5, pp. 94–104, 2017.

[283] S. Yi, H. Wei, L. Xiao-bo, L. Bin, and L. Tao, "Study of two-dimensional motion estimation technique in vehicle tracking," in *the 6th International Conference on ITS Telecommunications*, Chengdu, China, June 2006.

[284] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: A deep learning approach," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.

[285] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, *Federated Learning Based Proactive Content Caching in Edge Computing*, 12 2018.

[286] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," vol. abs/1803.04311, 2018. [Online]. Available: http://arxiv.org/abs/1803.04311

[287] D. Zhang, R. Shen, J. Ren, and Y. Zhang, "Delay-optimal proactive service framework for block-stream as a service," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 598–601, Aug 2018.

[288] 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 of Things Journal*, vol. 4, no. 5, pp. 1204–1215, Oct 2017.

[289] Y. Zhang, J. He, and S. Guo, "Energy-efficient dynamic task offloading for energy harvesting mobile cloud computing," in *2018 IEEE International Conference on Networking, Architecture and Storage (NAS)*, Oct 2018, pp. 1–4.

[290] Y. Zhang, J. Ren, J. Liu, C. Xu, H. Guo, and Y. Liu, "A survey on emerging computing paradigms for big data," *Chinese Journal of Electronics*, vol. 26, no. 1, pp. 1–12, 2017.

[291] J. Zhao, Y. Liu, K. K. Chai, Y. Chen, and M. Elkashlan, "Many-to-many matching with externalities for device-to-device communications," *IEEE Wireless Communications Letters*, vol. 6, no. 1, pp. 138–141, Feb 2017.

[292] L. Zhao and J. Liu, "Optimal placement of virtual machines for supporting multiple applications in mobile edge networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6533–6545, July 2018.

[293] Q. Zhao, H. Xu, and S. Jagannathan, "Neural network-based finite-horizon optimal control of uncertain affine nonlinear discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 486–499, Mar. 2015.

[294] L. Zhirong, Q. Hua, Z. Jihong, and X. Xiguang, "User-oriented graph based frequency allocation algorithm for densely deployed femtocell network," *China Communications*, vol. 10, no. 12, pp. 57–65, Dec. 2013.

[295] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," vol. 32, no. 4, July 2018, pp. 54–60.

[296] Z. Zhou, G. Ma, M. Dong, K. Ota, C. Xu, and Y. Jia, "Iterative Energy-Efficient Stable Matching Approach for Context-Aware Resource Allocation in D2D Communications," *IEEE Access*, vol. 4, pp. 6181–6196, 2016.

[297] Z. Zhou, G. Ma, C. Xu, Z. Chang, and T. Ristaniemi, "Energy-efficient resource allocation in cognitive D2D communications: A game-theoretical and matching approach," in *Proc. IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[298] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[299] M. Zou, C. Zhao, B. Shen, and K. Kwak, "Cooperative Spectrum Sensing Using DF Cooperative Diversity Protocol in Cognitive Radio," in *Proc. In-*

ternational Symposium on Communications and Information Technologies, Vientiane, Laos, Oct. 2008, pp. 604–608.