Game-theoretic Methods for Edge/Fog/Cloud Computation Offloading: A Systematic Review

Hassan Pirhosseini

Department of Computer and Information Technology Engineering, Qa. c., Islamic Azad University, Qazvin, Iran, *ORCID* 0009-0004-7185-3913, *E-mail*: *h.pirhosseini@iau.ac.ir*

Edris Zarei

Department of Computer and Information Technology Engineering, Qa. c., Islamic Azad University, Qazvin, Iran, *ORCID* 0000-0003-0363-3789, *E-mail*: *edriszarei@qiau.ac.ir*

Mohammad Hossein Rezvani*

Department of Computer and Information Technology Engineering, Qa. c., Islamic Azad University, Qazvin, Iran, ORCID 0000-0002-0292-7008, *Email: mh.rezvani@iau.ac.ir*

Azadeh Pourkabirian

Department of Computer and Information Technology Engineering, Qa. c., Islamic Azad University, Qazvin, Iran, *ORCID* 0000-0001-6153-6819, *E-mail: pourkabirian@iau.ac.ir*

Achyut Shankar

School of Computer Science Engineering and Technology, Bennett University, Greater Noida, 201310, Uttar Pradesh, India, *ashankar2711@gmail.com*

Nebojsa Bacanin [a,b]

a. Faculty of Informatics and Computing, Singidunum University, Serbia, Email: *nbacanin@singidunum.ac.rs*

b. Department of Mathematics, Saveetha School of Engineering, SIMATS Thandalam, Chennai - 602105, Tamilnadu, India

Carsten Maple

Secure Cyber Systems Research Group (SCSRG), WMG, University of Warwick, Coventry, United Kingdom, Email: *cm@warwick.ac.uk*

Wattana Viriyasitavat

Chulalongkorn Business School, Faculty of Commerce and Accountancy, Chulalongkorn University, Thailand, Email: *wattana@cbs.chula.ac.th*

Compliance with ethical standards

Informed consent: Informed consent was obtained from all participants included in the study.

Conflict of interest: On behalf of all authors, the corresponding author states no conflict of interest.

Funding: The authors did not receive support from any organization for the submitted work.

Author contributions: All authors contributed to the concept and design of the study. Hassan Pirhosseini and Edris Zarei collected and analyzed the data. The survey of the project and the validation of the results were done by Mohammad Hossein Rezvani. Azadeh Pourkabirian, Achyut Shankar, Nebojsa Bacanin, Carsten Maple, and Wattana Viriyasitavat were remote scientific advisors. Hassan Pirhosseini and Edris Zarei wrote the first draft of the manuscript, and all authors commented on earlier versions. All authors read and approved the final manuscript. This article reports part of the scientific findings of an academic Ph.D. thesis. The thesis was presented by Hassan Pirhosseini and Edris Zarei as students and Mohammad Hossein Rezvani as the thesis supervisor.

Abstract: Due to the distance of users from remote cloud data centers, cloud computing (CC) may not be a good solution for latency-sensitive applications. Fog computing (FC) and Mobile Edge Computing (MEC) are two complementary paradigms for CC that can reduce energy consumption and delay. In FC, if the cloud resources are not enough to process the task, it is offloaded to the remote cloud. On the other hand, the main advantage of MEC is to reduce latency and availability of data and services for end users. If the edge resources are insufficient to process the work, it may be offloaded to remote cloud servers or fog. One of the suitable tools for modeling edge/fog/cloud discharge is Game Theory (GT). Game theory tries to predict the behaviors and decision results of agents who have the right to choose in interaction with each other by using scenario design and analysis. Here, the goal is to achieve a stable allocation of network resources to meet user requirements. Game-theoretic modeling can lead to agents located at the edge of the network acting more reliably. In this paper, we present an extensive systematic review of GT-based task offloading in different computational paradigms such as MEC, FC, and MCC. In addition to classical game theory, we pay special attention to evolutionary game theory. Generally, these methods are more scalable than classical methods. Instead of considering the user as a player, they consider network characteristics such as the number of CPU cycles. This article applies game theory from different perspectives such as the type of offloading, performance criteria, algorithm type, system components, evaluation tools, and computing environment.
Keywords: computation offloading; Game Theory; fog computing; mobile cloud computing; mobile edge computing; 5G/6G Networks;

1 INTRODUCTION

Over the past two decades, several paradigms have been employed to process user tasks and data. Figure 1 shows the three-layer architecture of the examined paradigms in this review. The cloud layer consists of various servers that can store and process tasks. It is usually assumed that there are sufficient resources in this layer [1, 2]. Cloud Computing (CC) reduces local power consumption and in turn, increases battery life. However, the cloud is usually geographically distant from Mobile Devices (MDs). Therefore, CC may not be a good solution for delay-sensitive applications [3, 4].

Fog computing (FC) and edge computing (EC) are two complementary paradigms for CC that can reduce energy consumption and delay [5]. The fog layer consists of heterogeneous fog servers with relatively limited processing capabilities, including access points, switches, routers, gateways, base stations, and roadside units (RSUs). Similarly, the end device layer includes smart devices (such as smartphones and tablets), IoT devices (such as sensors and cameras), and smart cars [6]. These devices usually have limitations in power consumption and battery life. They are also heterogeneous in that they have different hardware specifications, communication protocols, and architectures.

In FC, if the cloud resources are not enough to process the task, it is offloaded to the remote cloud. On the other hand, the main advantage of Mobile Edge Computing (MEC) is to reduce latency and availability of data and services for end users [7]. In addition, the backhaul link overhead is reduced. This occurs because most user requests are satisfied by data already stored at the edge of the network. MEC can also reduce network congestion and increase data privacy. If the edge resources are insufficient to process the work, it may offload to remote cloud servers or fog.

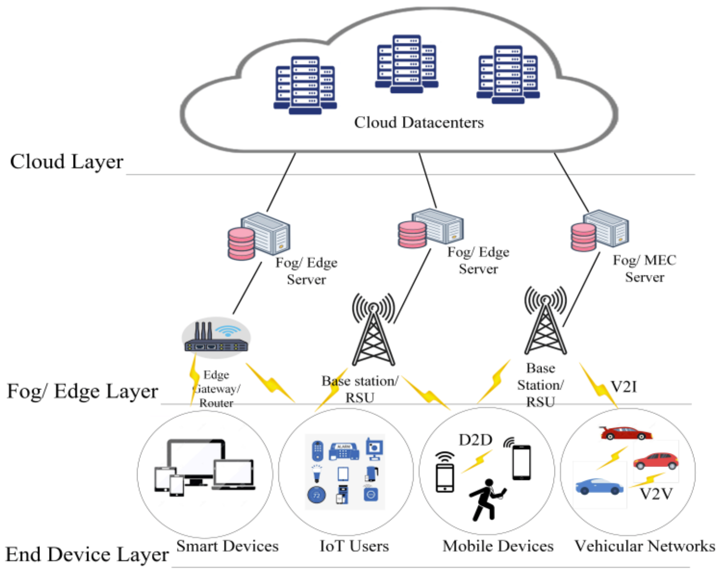


Figure 1: The three-layer architecture of the examined paradigms

1.1 Motivation

Computation offloading is a multi-objective and NP-hard optimization problem [8], meaning it cannot be solved using classical optimization methods in polynomial time. Our literature review has shown that the research conducted in this field is very diverse in system architecture, modeling, computational strategies, and problem formulation. In a broad sense, they can be classified into three categories: 1) mathematical optimization algorithms such as mixed integer programming (MIP), heuristics, metaheuristics, and Game theory (GT) [9, 10] intelligence-based optimization algorithms. Synthetic, and 3) control algorithms such as Lyapunov optimization approaches or a combination thereof. Meanwhile, a major contribution of research belongs to game theory.

Game-theoretic studies how rational players, faced with shared and scarce network resources, can interact with each other. The goal here is to achieve a stable allocation of network resources to meet user requirements. GT analyzes the interaction between autonomous and selfish players [10, 11]. Most of the network devices in the edge and fog layers are wireless. In recent years, considerable research has been done in the field of wireless network technologies [12, 13, 14]. Some of the most important GT research areas are cognitive radio [15], sensor networks [16], and mobile social networks [17, 18]. Game theory tries to predict the behaviors and decision results of agents who have the right to choose in interaction with each other by using scenario design and analysis.

GT, potentially supported by learning algorithms. Therefore, it can play an important role in modeling the interactions of nodes in the edge/fog/cloud. Such modeling can lead to agents deployed at the edge of the network acting more reliably. They may even react appropriately to unexpected situations [1]. Some such situations include load changes, conflicting strategies, incomplete information, latency, jitter, or cyber-attacks.

1.2 Comparison with Previous Studies

Our literature review shows that researchers have studied computation offloading in different paradigms with different approaches. Some studies have studied computation offloading methods in general [19]. In some articles [20], an overview of computation offloading in the MEC paradigm is provided [21, 22, 23]. Some other studies focus on scheduling [24] and load balancing [25, 26]. As a key application, some research addresses vehicular networks [27, 28]. Also, in some research, the problem of offloading is divided into two branches: "code offloading" and "data offloading". Some studies have investigated the problem from the perspective of optimization and decision theory. For example, in [29] a classification is presented in four fields: classical game theory, auction theory, evolutionary games, and hybrid games. Also, the authors in [25] examine various aspects of computational offloading and resource allocation. The methods of offloading based on mathematical optimization [30, 31, 32, 33, 34, 35, 36], Artificial Intelligence (AI) [37, 38], and control theory are examined in [39]. In another study [40], Stochastic strategies are reviewed.

None of the above works have been specifically and exclusively focused on offloading mechanisms in all three layers of edge/fog/cloud from the perspective of game theory. However, some research has been done in some related fields. For example, in [41], game theory has been investigated for offloading computational load and resource allocation in edge computing. Also, in [42], a review of computation offloading approaches in MEC calculations from the point of view of game theory has been discussed. In another research [43], adaptation theory for distributed computation offloading in IoT, fog, and cloud networks has been investigated. In [44], computation offloading and service placement in fog-based Internet of Things networks have been investigated. Also in [45], there is a review of the motivation mechanisms for offloading mobile data in heterogeneous networks. Unlike previous research, we conduct a systematic review regarding the application of game theory techniques and algorithms in computation offloading for fog and edge networks. Our study includes both "classic GT" and "evolutionary GT" branches. Due to the expansion of the use of evolutionary games, this emerging trend must be further scrutinized. Table 1 provides a comprehensive comparison between this article and previous related research.

Table 1: Comprehensive comparison between this article and previous related research

Ref.	Year	Paradigm	Focus on Offloading	Focus on GT	GT Components
[25]	2019	EC	✓	X	X
[26]	2020	EC	X	✓	✓
[40]	2021	CC/FC	✓	✓	✓
[46]	2020	EC	✓	✓	**X**
[47]	2022	CC/FC/IoT	✓	X	X
[48]	2021	EC/CC	✓	X	X
[49]	2020	MEC	✓	✓	✓
[50]	2022	VEC	✓	✓	✓

1.3 Contributions

The main contributions of this article are summarized as follows:

- We present an extensive systematic review of GT-based task offloading in different computational paradigms such as MEC, FC, and MCC. In addition to classical game theory, we pay special attention to evolutionary game theory. Generally, these methods are more scalable than classical methods. Instead of considering the user as a player, they consider network characteristics such as the number of CPU cycles.
- Following the systematic review research method, we first pose key questions to satisfy the requirements of computational offloading. Then, the application of each technique in task offloading schemes in different computing paradigms is described. In addition, for each GT technique, salient features of previous offloading research are reviewed. This review is done from viewpoints such as the type of offloading, performance criteria, type of algorithm, system components, evaluation tools, and computing environment.

The organization of this paper is shown in Figure A-1. Section 2 provides an overview of computational paradigms, computational offloading, and the fundamentals of GT; Section 3 describes the research methodology and its basic questions; Section 4 describes critical GT techniques used for offloading computational paradigms; Sections 5 and 6 are devoted to the discussion of results and future directions and research challenges; Finally, Section 7 concludes the paper.

2 BACKGROUND

This section first provides an overview of computational paradigms, especially computational offloading. Then, it introduces a taxonomy for game theory in offloading. Table A-1 in the appendices summarizes the abbreviations used in this paper.

2.1 An Overview of Computing Paradigms

With the expansion of mobile computing (MC) in the last decade, a paradigm called mobile cloud computing (MCC) has been developed [51]. It provides processing tasks with lower cost and more flexibility [52]. Here, the tasks are delivered to the remote cloud after being transferred to the service provider's server over the Internet. In recent years, with the proliferation of latency-sensitive IoT applications, MCC alone is unable to meet real-world requirements. The main reason is that cloud resources are centralized and far from users. Processing a task in MCC also requires a significant amount of energy [40]. As explained earlier, in the last decade, complementary paradigms such as FC and MEC [53, 54, 55, 56] have emerged to minimize latency and energy consumption.

In general, the unloading operation can be divided into two categories: "delay-sensitive unloading" and "delay-tolerant unloading". Also, from the task point of view, the offloading decision can be divided into two categories “binary offloading” and “partial offloading” [25, 57, 58]. In binary offloading, the task is either processed locally on the end device or offloaded to a cloud server. Conversely, in partial offloading, part of a task is processed locally and the remaining part is offloaded. As an example, suppose the end device is a policeman with a laptop [59]. He wants to find a criminal from a large video file. Here, partially uploading the video to the remote cloud can lead to a significant reduction in processing time. Alternatively, using partial offloading, the video file can be split into multiple parts, each offloaded to different servers. This leads to a significant reduction in processing latency. Usually, task interdependence is modeled with a directed acyclic graph (DAG) [19, 48, 60, 61] or with a task call graph [59].

There are various metrics to evaluate the performance of CC/mec networks. Some of the most important metrics are energy consumption, latency, response time, security, Quality of Service (QoS), Quality of Experience (QoE), provider cost, migration cost of Virtual Machines (VMs), etc. Therefore, computational load shedding is inherently a multi-objective problem.

2.2 Fundamentals of Game Theory

2.2.1 Classic Game Theory

Game theory (GT) is the study of strategic decision-making. It was first developed in the 1960s as a branch of microeconomics [62, 63, 64, 65, 66, 67, 68]. Classic GT is a set of tools or a set of tools that help the agent understand how economic factors behave in strategic situations. GT agents are not price takers but instead, decide the price themselves. In addition to economics, G has been used in other fields such as evolutionary biology, sociology, psychology, political science, project management, financial management, and computer science [66, 69, 70]. GT provides insight into idiosyncratic behavioral interactions in animal groups [69, 71], bargaining, and exchange [72, 73, 74].

Generally, a game defined in GT has two or more players. In addition to being a function of his actions, a player's benefit is also a function of the actions of the rival player. This assumption is summarized under the title of Rationality. Actors are rational. Therefore, the player's optimal behavior depends on what his opponent is doing. Therefore, there is an interaction between the two parties. The set of strategies available to the player and his utility is presented as a value matrix [75, 76]. The most well-known classic games are Matching Pennies, rock-paper-scissors, tic-tac-toe, Meeting in New York, Money Sharing game, Battle of Sexes, chicken Game, and Duopoly. Undoubtedly, prisoner's dilemma is the most famous classic game mentioned in many textbooks. This game includes two thief actors. They committed a crime last night. The police believe that the previous robberies in that neighborhood were also the work of these two people. The police interrogate the thieves separately in two independent rooms. Every thief has two choices: confess silence. The initially expected prediction is that neither of the two will confess. On the contrary, what happens in practice is that both people confess [63]. Next, we will explore other well-known games in Classic GT. The components of a classic game are:

Players *(p1, p2)*: are agents who do a series of actions to increase their utility (profit).

Strategy (action: each of the players can do a series of things. For example, players *p1* and *p2* in the prisoner's dilemma game may or may not confess.

Strategy Profile: It is a list that specifies the value of each house of the profit matrix. The actions that each of the two players *p1* or *p2* can do form a set. If the game consists of only two players, the strategy profile is an ordered pair where the first element is a member of the first set and the second element is a member of the second set.

Order by play: It specifies the order of play of the players.

The information Set: It shows what information each player has at any point in time. Usually, the information set of players is different from each other.

Outcome: It is the outcome of the game. Each Strategy Profile corresponds to an output.

Utility: It is the benefit of the player which can be defined in different ways.

Now we briefly explain the basic concepts of Classic GT.

One of the most important principles of GT is "Rational choice". According to this principle, the action a player chooses based on his preferences should be at least as good as any other available action. In other words, each player, knowing what the other party is doing, chooses something that is at least as good as other actions [63].

Another important concept in GT is Nash Equilibrium. It can provide a prediction as to what the outcome of the game could be. Nash equilibrium specifies a strategy for each player. A Nash equilibrium is an action profile such as a^*. At such a point, no individual will have the incentive to violate that strategy, and no player can gain more profit. A strategy profile is a Nash equilibrium if, for each player, the payoff obtained from this strategy, $u_i(a^*)$, is greater than or equal to the payoff obtained for this player if the others adhere to this strategy. For each player i, let us denote all other players by

i . In other words, sticking to the Nash strategy profile brings the highest payoff for the whole society [77]. Formally, the Nash equilibrium can be expressed as follows:

$$u_i(a^*) \geq u_i(a_i, a^*_{-i}) \quad (1)$$

Another important concept in classic GT is the best response function. It determines for each player what is the best action against the actions of other players. There is a theorem that states that for a player i , an action profile a* is a Nash equilibrium if and only if what is in a^* is the best solution to a^*_{-i} . In other words, a strategy profile is a Nash equilibrium if the action that the player takes in that profile strategy is the best action that the others take.

Now let's define the important types of strategy. The two most common types of strategies in classic GT are *"pure strategies"* and *"mixed strategies"*. A pure strategy provides a complete definition of how a player should play a game. A player's strategy set is a set of pure strategies available to that player [78]. A mixed strategy is a combination of pure strategies where a probabilistic value p , where $0 < p < 1$, is associated with each of these pure strategies. Since the probabilities are continuous, there are an infinite number of possible combinations of strategies available to a player. A completely mixed strategy is a mixed strategy in which the player determines a completely positive probability p for each pure strategy. Therefore, each pure strategy is a special case of a mixed strategy, where the particular strategy is chosen with probability 1 and every other strategy with probability 0. A strategic game can have more than one Nash equilibrium [79, 80]. In the mixed strategy space, instead of defining the Nash equilibrium on the pure strategy, it is defined on the mixed strategy. A mixed strategy profile a^* , for each player, has a vector of probabilities as follows:

$$a^* = (a^*_1, a^*_2, ..., a^*_N) \quad (2)$$

At the Nash equilibrium point, the utility resulting from the mixed strategy profile a^* for each player is greater than or equal to the utility obtained by the other players. In fact, at such a point, no player has the incentive to violate the Nash strategy profile. Finally, the best response function can be used to calculate the Nash equilibrium. In general, nothing has changed. Until now, our discussion consisted of two strategies, and now we have infinite strategies! A Mixed Strategy Profile is a Nash equilibrium if each player's best response to the others is a^*_i . Now we briefly describe the most common game types in classic GT:

● **Normal-form Games versus Extensive-form Games**

In a normal game, all players make decisions simultaneously and this decision happens only once. On the contrary, in the form-extended game, we have a repetition of decisions [81, 82]. Each time, players can make decisions simultaneously or in a predetermined order. An extended-form game is often represented by a game tree. Here, each node (except terminal nodes) is a decision point, and each link represents a decision or set of decisions made by the player/players at a particular point in time. The terminal nodes for a form-extended game are represented by the payoffs earned by each associated player. It is worth mentioning that two methods, Normal Form and Extensive From, are used to express games. The Normal Form method is used to express simultaneous games, while the Extensive Form method is used to express sequential games.

● **Non-cooperative Games versus Cooperative Games**

Games are played for the personal benefit of the players. Even when they cooperate, they think about maximizing their profits. These games can be called "non-cooperative games" or "competitive games". Non-cooperative game theory is a branch of classical GT. Another class of games is cooperative games. Players are not only concerned with a set of financial payoffs, but their goal may be not to violate a set of norms. Therefore, players aim to maximize their expected payoff or utility. In such games, any cooperative behavior is caused by altruistic goals. There is a type of cooperative game called the coalition game, in which players form coalitions to cooperate. Note that there may be some competition between these coalitions [80, 83, 84]. Here, a group of actors cooperate to achieve a result. It is possible to imagine different distributions of benefits (Output) in the coalition. If the sum of all those distributions is constant, the Payoff is said to be transferable. Here, actions may cause Payoff to shift from one player to another. One of the interesting research areas in this field is to predict which coalitions will be formed and how efficient these coalitions will be [85]. Four very important types of non-cooperative games are strategic games with perfect information, strategic games with incomplete information, extended games with complete information, and extended games with incomplete information. Also, important types of cooperative games are coalition games, evolutionary equilibrium, and auctions [63].

●Zero-sum Games

These are a class of competitive games in which the sum of all players' payoffs is zero. In two-player mode, one player's loss equals the other player's gain. Zero-sum games can be solved by the mini-max theorem [86]. In this game, there is a set of strategies that minimize the maximum loss of each player. It can be argued that the stock market is a prime example of a zero-sum game. In contrast, most valid economic transactions are nonzero because each party thinks that what it receives is worth more than what it gives. [87]

● Complete Information Games Versus Incomplete Information Games

who is playing when does he play How many players are there? When is everyone's turn? These are examples of common knowledge. In strategic games with complete information, the structure of the game is common knowledge. It is assumed that during the game, the payoff values are constant. In a game with complete information, each player knows the complete history of previous actions of other players as well as the initial state of the game. In incomplete information games, some or all players do not have access to all information related to the previous actions of other players. Strategy games are games that are expressed in either Strategy form or Normal form. Here, the time parameter is removed in the sense that the players necessarily play at the same time [63].

● Simultaneous Games versus Sequential Games

A simultaneous game is a regular game or an extended-form game in which all players make decisions simultaneously in each iteration. Therefore, in each iteration, each player has to decide without knowing the decisions of other players. On the other hand, a sequential game is a form-extended game in which players make their decisions (strategies) in a predefined order [81, 82]. In a sequential game, some players can observe the actions of other players before making their decisions. Otherwise, the game becomes a simultaneous game, even if the players' actions are not performed simultaneously. Such a consecutive game has "perfect information" if a player can observe the previous player's move. Otherwise, the game has "incomplete information". Sequential games are often used to model bargaining or negotiation mechanisms [87].

● Differential Games

Differential games are often extended games that are modeled in a continuous time domain instead of having discrete decision points [86]. In such games, each variable evolves continuously over time based on a differential equation. These games are ideal for modeling rapidly evolving defensive scenarios, where each player seeks to selfishly optimize some parameter. For example, in the missile tracking problem, the pursuer and the target both try to control the distance. While the pursuer constantly tries to minimize this distance, the target constantly tries to increase it [87].

● Stackelberg Games

Stackelberg is a two-player sequential game commonly used in economics [88]. In a Stackelberg game, there is a leader and a follower, usually firms operating in the same market. The leader company has some kind of competitive advantage in the market that allows it to move first and make the first decision. Here, the optimal decision of the followers depends on the decision of the leader. If a follower chooses a suboptimal action given the leader's performance, it will not only affect his/her profit but also the leader's profit. Therefore, the leader's optimal decision is made with the assumption that the follower can see the leader's performance and maximize his profit according to the leader's performance. Stackelberg games may be used in defensive applications where there are leaders and followers, such as naval formations, fighter planes, or tanks [87]. They have also been used in computer networks in recent years.

● **Common--interest Games**

Common-interest games are another class of non-cooperative games where there is one action that all players strongly prefer over all other profiles [89]. In other words, in common-interest games, the interests of the players are perfectly aligned. It can be argued that common-interest games are the opposite of zero-sum games, in which players' interests are in stark conflict so that any increase in wealth for one player must necessarily lead to a decrease in collective wealth for others. Common interest games were first studied during the Cold War between the US and the Soviet Union to apply strategies to manage international relations [90, 91, 92].

● **Bargaining**

In a Nash bargaining game [93, 94], two players can choose from a set of similar alternatives. However, each alternative has different benefits for players. Typically, some alternatives pay off better for one player, while other alternatives pay off better for the opposing player. If both players choose the same alternative, each of them receives the payoff corresponding to that alternative. If they choose different alternatives, there is no agreement, and each receives a fixed payoff that corresponds to the cost of non-agreement, which is usually negative. Therefore, players have an incentive to choose an alternative that may not necessarily be the best for a player. If there is complete information, that is, the complete set of alternatives and payoffs is known to both players, then there is an equilibrium solution to the Nash bargaining game [95].

● **Sub-games**

A sub-game is a subset of a sequential game such that initially, each player has knowledge of all the players' actions [63]. In simpler terms, a sub-game is part of a sequential game tree in which the first node has complete information [87].

● **Sub-game Perfect Nash Equilibrium**

In a sequential game, a complete sub-game Nash equilibrium represents the set of strategies of each player such that they form a Nash equilibrium for each sub-game [63]. Therefore, in a sequential game tree, it is possible to identify a set of strategies that establish a Nash equilibrium for each branch of the tree starting from a source node. At such points, each player is aware of all previous actions. For all players, such a set of strategies represents the complete subgame Nash equilibrium for that sequential game. A well-known example is the bargaining game [87].

2.2.2 Evolutionary Game Theory

Evolutionary GT was first used by Fisher†† to explain why sex ratios are equal in mammals. He was faced with the puzzle of why, when most of the male animals lack a placenta, the ratio of the sexes is almost equal. He found that the ratio of genders in equal numbers of males and females in the population has a dynamic evolution [96]. This has made Mohin hopeful that maybe this method can cover the shortcomings that exist in the traditional GT. In summary, the most important disadvantages of traditional GTs can be introduced as follows [44, 45]:

a) Difficulty in Choosing an Equilibrium

The concept of Nash equilibrium was first introduced by John Nash in 1950 for traditional GT. The strategies chosen by a group are called Nash equilibrium if the strategies of each individual have the best match with the strategies chosen by others. This means that no individual can improve his efficiency by changing his strategy alone unless at least one other person also changes his strategy. Therefore, it can be concluded that the efficiency of each individual is not optimal in the state of excellence. Also, a major disadvantage of traditional GT is that if players are restricted to using a pure strategy, then some games may not have a Nash equilibrium point. To explain this, consider the coin game in Figure 3. In such a non-cooperative game, players may use a mixed strategy to reach a Nash equilibrium. But the important question is: "For a player, how meaningful is this strategy in practice?" Considering that the cost of compound strategies is usually high, limiting players to pure strategies may lead to not finding a solution. Another disadvantage of traditional GT is the existence of multiple Nash equilibrium points in a game. In this case, choosing one of such equilibrium points is not an easy task at all.

	Head	**Tail**
Head	(0,1)	(1,0)
Tail	(1,0)	(0,1)

Figure 3: Nash equilibrium in a non-cooperative game

b) The Lack of Dynamics in Traditional GTs

Contrary to traditional GTs, in which a dynamic behavior does not have insight and wisdom, here in the evolutionary GT, there is complete dynamism. In 1961, Levintin proposed the first specific application of GTs in the field of biological evolution. Later in 1972, the concept of evolutionary sustainable strategy (ESS) was proposed by Smith [51]. Now, we briefly explain the two main approaches to obtaining ESS:

The first approach:

This approach to obtaining ESS was first proposed by Smith [51] in the form of the Hawk-Dove problem. In this game, two birds compete with each other to get resources worth V. In biological categories, the value V is related to Darwin's survival index, and in cultural categories, V has various interpretations depending on the intended application context. Each organism follows one of the following strategies [51].

In an ESS, there must be properties that, if followed by any member of the population, do not cause a significant deviation from the sustainable strategy. Suppose $\Delta F = (s_1, s_2)$ denotes the change in the value of the player who chose strategy s_1 when the other player followed strategy s_2. Also, $F(s)$ denotes the total value for the player who follows the strategy s.

Furthermore, let each player in the population have an initial value denoted by F_0. We denote an ESS by δ. Denoting the strategy of invasion and change by μ we write

$$F(\Delta) = F_0 + (1-p)\Delta F(\delta,\delta) + p\Delta F(\Delta,c) \tag{3}$$

$$F(\mu) = F_0 + (1-p)\Delta F(\mu,\delta) + p\Delta F(\mu,\mu) \tag{4}$$

, where p is the proportion of the population that follows the invasion strategy μ. To establish an ESS, the value for each player who follows this strategy must be greater than the value of another player who follows the strategy μ. Otherwise, the player who follows the μ strategy can move away from the ESS. So,

$$F(\delta) > F(\mu) \tag{5}$$

If p is very close to zero, then one of the following conditions must be met:

$$\Delta F(\delta,\delta) > \Delta F(\mu,\delta) \tag{6}$$

or

$$\Delta F(\delta,\delta) = \Delta F(\mu,\delta), \Delta F(\delta,\mu) > \Delta F(\mu,\mu) \tag{7}$$

The above expression is a definition provided by Smith. A strategy δ is an ESS if one of the following two conditions is true:

- The value of the utility function of the first player is strictly higher if the first and second players play δ than if the second player plays δ and the first player does not play δ.
- Or that the utility of the first player is equal in the two cases mentioned, but the utility of the first player in the case that he plays δ and the second player does not play δ is strictly higher than in the case that both play δ.

The second approach:

In this approach, the dynamic changes of the strategy are studied to investigate the evolutionary properties of the population. To study the details of these methods, you can refer to [97, 98]. It is worth noting that the above two approaches may produce different outputs.

c) Coalition Equilibrium Point In n-player Cooperative Evolutionary Games

Suppose we represent the set of players with $N = \{1, 2, ..., n\}$. Also, let $e(i)$ be the i-th vector in $\mathbb{R}^N$. For a subset $S \subseteq N$, assume that

$$e(S) = \sum_{i \in S} e(i) \tag{8}$$

For each subset $S \subseteq N$ and vector $X \in \mathbb{R}^N$, we define

$$X(S) = \sum_{i \in S} e(i).x_i \tag{9}$$

Here, $X = \{x_1, x_2, ..., x_N\}$. In this game, we can consider $2^N - 1$ individuals or groups. An n-player game with transferable utility can be defined by $G = (N, V)$ where $V : 2^N \rightarrow \mathbb{R}^N$.

For each coalition S, we denote by $V(S)$ a vector of the set of receipts of each player in the coalition S that can cooperate. It is also assumed that for every coalition S, there is a non-negative real number $v(S)$ such that

$$V(S) = \{x \in \mathbb{R}^N \mid \sum_{i \in S} x_i \leq v(S)\} \tag{10}$$

In the following, we use the expression (N, v) instead of the expression (N, V) [97]. A basic concept in cooperative games is the concept of the core. The core is an n-player game with transferable utility (N, V) is a set $X \in V(N)$ such that no coalition like S can be found that has for each $y_i \geq x_i$, $y \in V(S)$ and for each $i \in S$ and at least for a $j \in S$ we have

$$y_j > x_i \tag{11}$$

Also, a game $G = (N, V)$ is called a balanced game [99] if the following relation holds

$$\{\exists \partial_S \geq 0 \rightarrow \forall S \in \beta, \sum_{i \in S \in \beta} \partial_S = 1; \quad \forall i \in N\} \tag{12}$$

And for each balanced family β of the set of coalitions, let us have

$$\sum_{S \in \beta} \partial_S \, v(S) \leq v(N) \tag{13}$$

Also, a balanced game is called fully balanced if there is an equilibrium for all sub-games as well [100]. At the end of this section, we show the taxonomy of GT-based computation offloading in Figure 4.

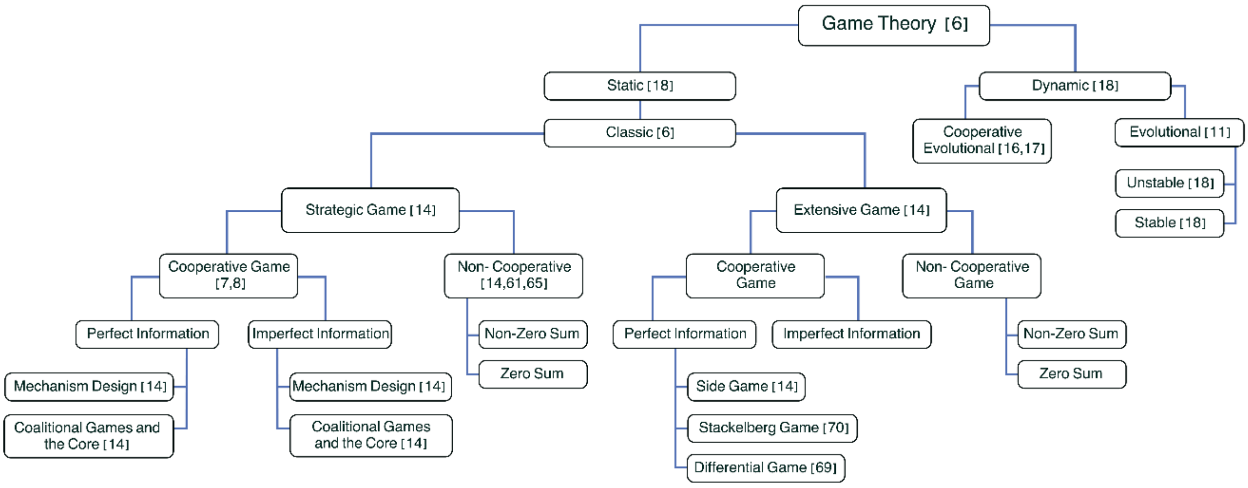


Figure 4: Classification of GT-based used in computation offloading

3 RESEARCH METHODOLOGY

This study aims to systematically investigate GT algorithms in computation offloading. The methodology of this research was based on the guidelines for preparing Systematic Literature Reviews (SLR) [46]. This section includes research questions, a search process, inclusion/exclusion criteria, and quality assessment.

● Research Questions

To plan the review process, first, it is necessary to derive research questions. These questions determine the motivation of the study. Our study faces eight key questions that are the basis of the search strategy for the literature review. On the other hand, a comparative analysis should be done for the interoperability of computation paradigms. The main research questions and the motivations for raising them are as follows:

RQ1: Which GT algorithms are used in computation offloading?

Motivation: By finding the answer to this question, we can identify how to use each GT algorithm in offloading. It helps us to identify the popularity and acceptability of each GT technique in computation offloading.

RQ2: Which offloading scheme, binary or partial, is used for each GT-based approach?

Motivation: The answer to this question can not only find the long-term trend of research in offloading but also determine the direction of research for at least 5 years.

RQ3: What performance criteria are used in GT-based offloading approaches?

Motivation: The literature review shows that researchers have considered different optimization goals and criteria. The answer to this question can determine the importance of each metric. As explained earlier, in general, the criteria chosen in the papers are directly related to the type of tasks, delay-sensitive vs. energy-sensitive. The answer to this question can also determine which classes of programs are important.

RQ4: What are the principles of each offloading approach on GT? Is it implemented as a single-agent or multi-agent?

Motivation: The answer to this question determines whether the environment is considered a Single-agent Game Theory (SAGT) or a Multi-agent Game Theory (MAGT). It specifies how agents interact with the environment and with each other.

RQ5: What simulators do researchers use to evaluate the performance of GT-based approaches?

Motivation: This is one of the vital questions for students and researchers. Unfortunately, the literature review shows that due to a lack of familiarity with the characteristics and capabilities of simulators, wrong choices are often made by young researchers in academic environments.

RQ6: What GT-based approaches are used to meet the needs of each complementary paradigm?

Motivation: As mentioned earlier, complementary paradigms (MCC, MEC, FC, VEC [101], etc.) are components of an ecosystem called distributed computing. Therefore, the GT-based solution proposed for one paradigm should not conflict with the others.

RQ7: What are the main features of each GT-based approach? What are the main features of hybrid approaches?

Motivation: The answer to this question shows what ideas or specific features have been used in previous articles.

RQ8: What are the future research trends and open issues in GT-based computation offloading?

Motivation: The answer to this question will help the researchers to choose a more suitable path to save resources and time.

We will answer the above questions in detail in the following sections.

● **Search Process**

The most important scientific databases in the world used in this survey are

• Wiley Interscience (www3.interscience.wiley.com)
• Springer (www.springerlink.com)
• ACM Digital Library (www.acm.org/dl)
• ScienceDirect (or Elsevier) (www.sciencedirect.com)
• IEEE eXplore (ieeexplore.ieee.org)
• mdpi (www.mdpi.com)

Keywords used to describe the search string included terms such as "GT", "evolutionary GT", "computation offloading", etc. Finally, using keywords and combining them with Boolean operators "AND" and "OR", the search strings were defined as follows:

("Game Theory" or "GT"), ("Evolutionary Game Theory" or "Evolutionary GT"), and ("Computation Offloading" or "Offloading") and (" " or "Blockchain " or "Digital twin" or "AI" or " IoT in Industry 5.0" or " Deep Reinforcement Learning" or " hybrid AI game theoretic" or "Autonomous vehicles" or "5G/6G networks" or "Markov decision processes" or "Convex optimization" or "particle swarm optimization" or "Genetic Algorithms" or "Heuristic")

● Inclusion and Exclusion Criteria

Our literature review shows that after 2018, research in computation offloading accelerated. For this reason, we put our main focus on the beginning of 2018 to the end of 2025. The article selection process is shown in Figure 5. In the quality assessment step, some conference papers that were not well-known or were not highly relevant to the topic were excluded. Table A-2 in the appendices shows the features we finally extracted from the articles. The distribution of articles published from 2018 to 2025 along with their analysis is given in Section 5.

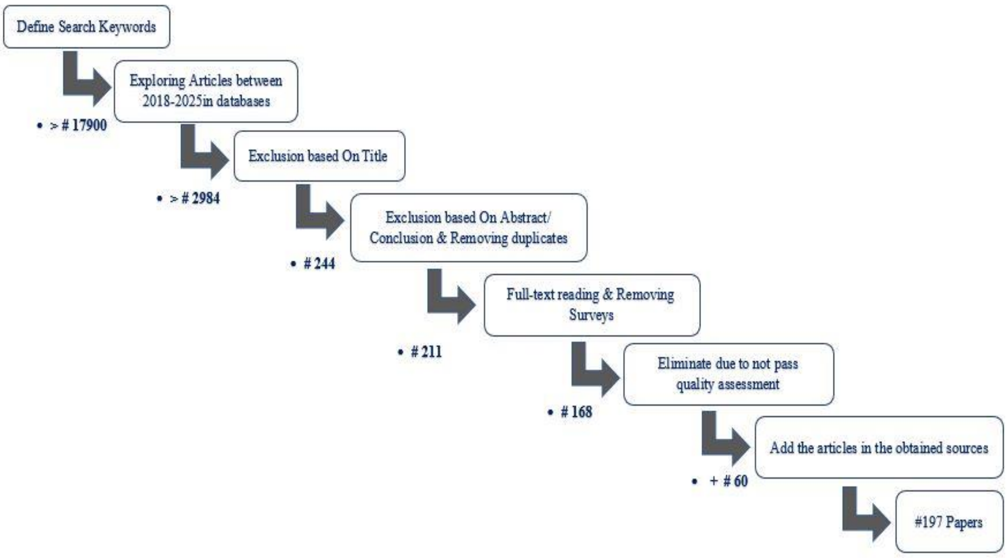


Figure 5: The process of selecting articles in our systematic review

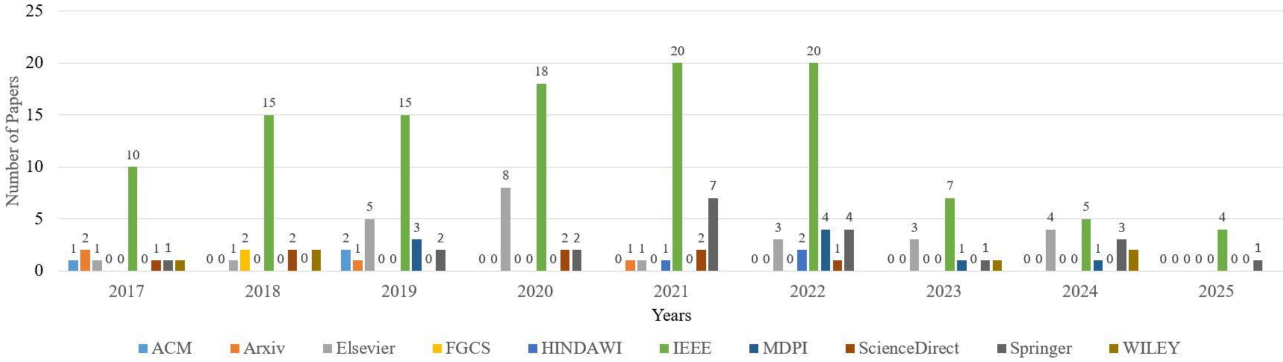
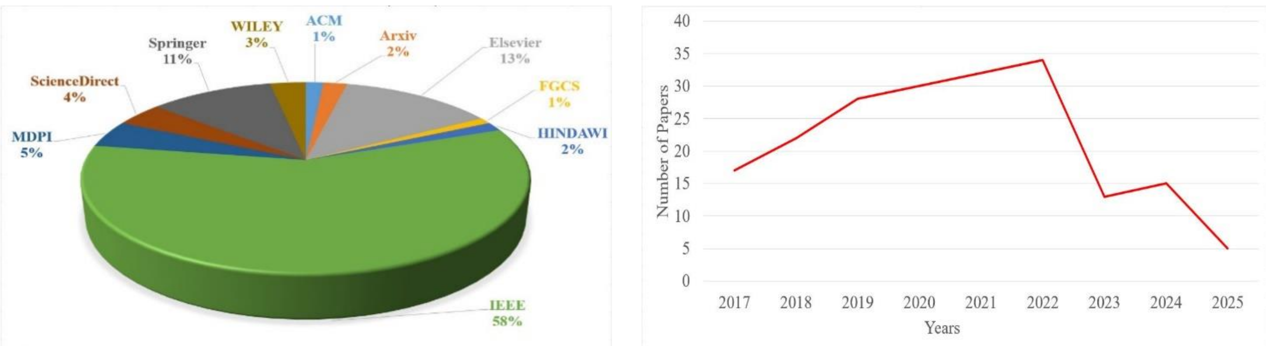


Figure 6: Distribution of articles by publishers during different years



a) Distribution of articles based on the name of the publisher

b) Number of publications per year

Figure 7: Statistics of articles in this survey

● **Distribution of Articles**

Figure 6 shows the distribution of articles in different years. Also, Figure 7-a shows the distribution of articles based on the name of the publisher. As shown in the figure, by the end of 2025, IEEE has the largest share compared to other publishers with 58%, followed by Springer with 11%. Figure 7-b shows the number of publications each year. It represents a significant increase in GT-based computation offloading research since 2018. Also, our literature review shows that a total of 178 journal articles and 18 conference articles have been published. To answer the research questions, we read the full text of selected articles and analyzed them. The results of the analysis will be presented in the following sections.

4 COMPUTATION OFFLOADING MODELING USING GAME THEORY

The environmental information needed to design optimal offloading policies may not be available in dynamic systems. For example, an IoT device may make decisions based on only part of the information available from other devices. Therefore, it may not be possible to run complete information games in such a system. This is just one example of the complexities of offloading modeling with GT. In this section, we first examine the most critical performance criteria in computation offloading. Then, we review GT-based methods that have been used in previous studies.

4.1 Major Performance Metrics

Now, we examine the most important performance metrics in offloading. For each metric, a brief formulation is mentioned for clarification. A complete list of performance measures in different computational paradigms is shown in Table A-3 in the appendices. We advise interested readers to study the table carefully.

- **Delay:** In almost all previous studies, the delay metric, in different ways, has been addressed in performance evaluation [102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121]. To calculate the computation offloading delay, we must add the delay of transferring a task to the server, the delay of execution on the server, and the delay of receiving the results from the server [52]. Let B_n^s and B_n^r denote the bandwidth allocated to send and receive requests issued by a mobile device *n* regarding a server, respectively [77, 81]. Also, t_{nm}^s and t_{nm}^r represent the transmission delay of sending and receiving the *m*-th task from mobile device *n*, respectively. Therefore, the following calculations can be made:

$$t_{nm}^s = \frac{L_{nm}^s}{B_n^s} \tag{14}$$

$$t_{nm}^r = \frac{L_{nm}^r}{B_n^r} \tag{15}$$

In the above formulas, L_{nm}^s and L_{nm}^r denote the size of the *m*-th task issued by the mobile device *n* during sending and receiving, respectively. Now, we calculate the edge processing delay as follows:

$$t_{nm}^{server} = \frac{L_{nm}^s . N_{nm}^c}{f^{server}} \tag{16}$$

In the above formula, f^{server} and N_{nm}^c denote the processing rate of the edge server and the number of processor cycles required for each bit, respectively. Now the offloading delay is calculated as follows:

$$t_{nm}^{offload} = t_{nm}^s + t_{nm}^r + t_{nm}^{server} + t^{wait} \tag{17}$$

, where t^{wait} is the total waiting time of a task [82, 122]. The formula for the local execution of the task on the end device is as follows:

$$t_{nm}^{local} = \frac{L_{nm}^s . N_{nm}^c}{f^{local}} + t^{wait} \tag{18}$$

, where f^{local} is the local processing rate of a task on the mobile end device.

- **Response Time:** Response time is the difference between the time taken to offload a task and the time it takes to receive a response on the mobile end device. It is very similar to the concept of the Ping command. It is obtained as follows [40]:

$$t_{nm}^{response} = p\, t_{nm}^{offload} + (1-p)\, t_{nm}^{local} \tag{19}$$

, where $p \in \{0, 1\}$ is the probability of offloading to the edge server.

- **Energy Consumption:** In almost all previous studies, energy and delay metrics have been considered together in performance evaluation [123, 124, 125, 126]. The energy consumption of offloading, $E^{offload}$, is equal to the sum of the energy consumed to transfer the task from the end device to the edge server, E^{s}, the energy consumed to execute the task on the server, E^{exe}, and the energy consumed to return the result to the device, E^{r} [43]. So we write:

$$E^{offload} = E^{s} + E^{exe} + E^{r} \tag{20}$$

The local execution energy consumption, El, is obtained as follows [85, 86]:

$$E^{local} = P_n^{cpu} . c_n \tag{21}$$

, where P_n^{cpu} and c_n denote the energy consumption per processor cycle and the number of processor cycles required for each bit in the end device *n*, respectively. The energy consumption of remote execution, E^{r}, is obtained as follows:

$$E^{r} = \frac{P_n^{t} . d_n}{r_n} \tag{22}$$

$$r_n = B . \log_2 (1 + \frac{P_n^{t} . g_n^{2}}{N_0 . B}) \tag{23}$$

, where P_n^{t}, d_n, r_n, g_n, B, and N_0 denote the transmission power, task data size, transmission rate, channel gain, wireless channel bandwidth, and white noise power spectral density, respectively.

Now, we calculate the energy used for partial offloading. In this case, part of the task is executed locally and the other part is offloaded. Energy consumption is obtained from the following formula:

$$E_n = p . E^{offload} + (1-p) . E^{local} \tag{24}$$

- **Cost:** The cost of executing each task on the server depends on the location of the end device, response time, and workload [93, 94, 95, 127, 128]. For this purpose, it is necessary to first calculate the expected delay in the local execution as follows:

$$t_k^{L}(x_k) = \frac{1}{(\mu_m - \lambda_a . \bar{x}_k)}, \qquad k = 1, 2, ..., N \tag{25}$$

$$\bar{x}_k \triangleq 1 - x_k$$

$$\mu_m \triangleq \frac{f_m}{\mu_a}$$

$$\mu_a \triangleq L_a . B_a$$

, where x_k, m, a, $\bar{x}_k$, f_m, L_a, and B_a denote the offloading frequency of user k, the computing service rate of user k (tasks per second), task arrival rate, local execution probability, the CPU cycle frequency of the end device, the average CPU cycles required for the task, the task size (in bytes), and the processing density (in cycles/bits), respectively. Now, the local execution energy is calculated as follows:

$$E_k^{L} = k_m . f_m^{2} . \mu_a, \qquad k = 1, 2, ..., N \tag{26}$$

, where f_m^{2} is the power consumption of the end device during the execution of one CPU cycle. Also, k_m is the energy consumption factor that depends on the processor architecture. Now, the cost of local computing is calculated as follows:

$$C_k^{L}(x_k) = c_k^{L} . E_k^{L} + c_k^{d} . t_k^{L}(x_k), \qquad k = 1, 2, ..., N \tag{27}$$

$$0 < c_k^L < 1, \qquad 0 < c_k^d < 1,$$

, where c_k^L and c_k^d are energy weight and delay weight, respectively. Now, the offloading delay to the access point and the energy required for offloading are calculated as follows:

$$t_{k,1}^R(x_k) = \frac{l_a.\mu_a}{\beta_k(x_k)}, \qquad k = 1, 2, ..., N \tag{28}$$

$$E_k^R(x_k) = P_{trans}.t_{k,1}^R(x_k)$$

, where β_k and P_{trans} are the transmission rate and transmission power of user *k*, respectively. Finally, the total computation cost is obtained as follows:

$$t_{k,1}^R(x_k) = \frac{l_a.\mu_a}{\beta_k(x_k)}, \qquad k = 1, 2, ..., N \tag{29}$$

$$E_k^R(x_k) = P_{trans}.t_{k,1}^R(x_k)$$

Now, the computation offloading cost is calculated as follows:

$$C_k^R(x_k) = c_k^R.E_k^R + (1 - c_k^R)t_{k,1}^R(x_k) \tag{30}$$

- **Migration of VM to Execute the Task:** Migration of VMs may be necessary at a lower level after the offload destination is determined. The migration of VMs can be classified from different perspectives. Here we simply divide it into two categories: *VM-based migration* and *container-based migration*. Here, offloading may be done both online (at run time) and offline. In the former, the task is offloaded during the execution, while in the latter, it is offloaded before the execution starts [59, 57]. The migration of VMs during task execution is one of the hot topics in cloud/fog computing and is one of the open research issues for the future.
- **Replication Scale:** In latency-sensitive applications, it is possible to offload the task or its computationally intensive parts to more than one remote server. This is especially used for partial offloading. Task replication can be very complex. To accomplish this, the task must first be analyzed to identify the separable parts of the code from the non-separable parts. Therefore, identifying the Amdahl speedup factor for the task is of great importance. During the last two decades, it has been one of the open issues not only in cloud/fog computing but also in other computing paradigms.
- **Quality of Experience:** With subtle differences, QoE and QoS metrics can complement each other in some cases. QoE is a user-oriented measurement of service with an emphasis on aspects such as requirements, goals, and perceptions. To measure the user's QoE, methods such as the average and standard deviation of his/her satisfaction are used [19]. It can be said that all studies conducted in edge/cloud/fog computing are directly or indirectly QoE-aware.

4.2 Major Game-theoretic Offloading Studies

This section reviews major GT-based research for edge/cloud/fog offloading. Figure 8 shows the most important perspectives that we adopt in our analysis. Also, Table A-4 in the appendices shows the most important abbreviations of game theory algorithms. A summary of computation offloading with evolutionary, greedy, Stackelberg, and metaheuristic algorithms is given in Table A-5 in the appendices. Also, Table A-6 shows a summary of Bayesian, non-cooperative, and multi-user approaches. We strongly recommend studying these tables before reading the rest of this section.

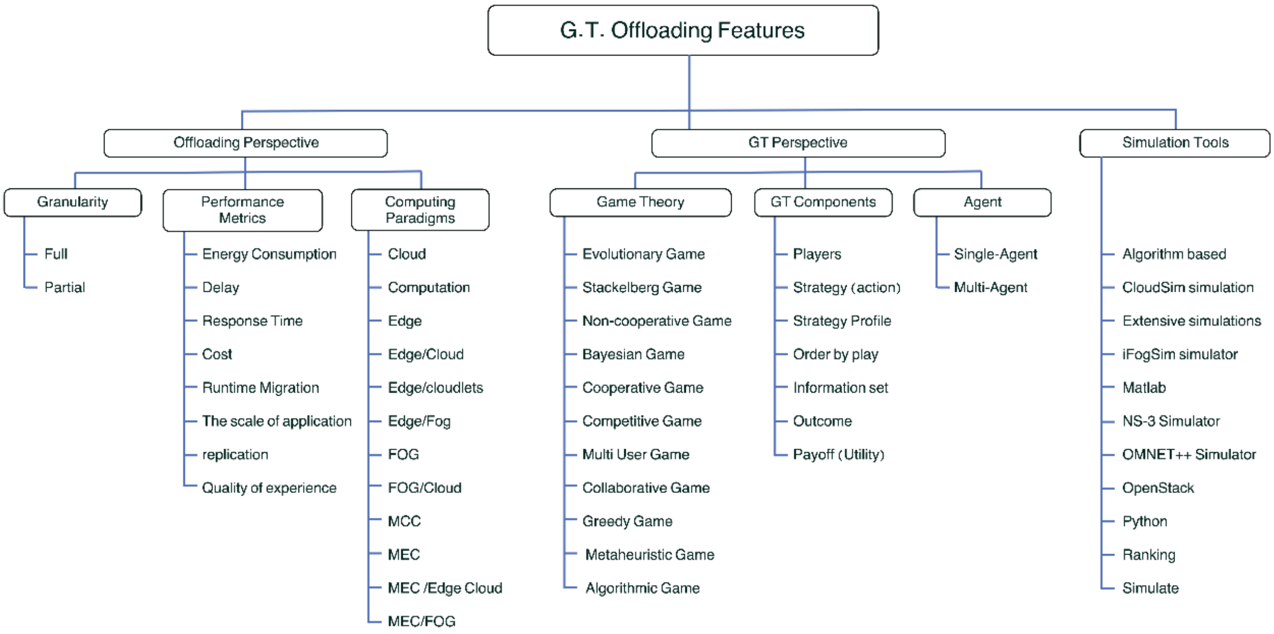


Figure 8: The most important GT perspectives in our analysis

4.2.1- Evolutionary Game

As explained before, in the last decade, evolutionary games have expanded to be used in computer networks due to their scalability. In [129], a decentralized evolutionary game-theoretic algorithm is proposed for offloading in MEC considering the QoS and battery life constraints. One of the advantages of this algorithm is providing a stable strategy with minimum operating cost. It can converge to an ESS with a limited number of iterations. However, it does not take into account the preferences of mobile users. A similar study has been conducted in [130]. The proposed iterative strategy can correctly determine a suitable server for each user. The proposed strategy is stable and can converge to an ESS in a limited number of iterations. In [131], an evolutionary game-theoretic algorithm based on Reinforcement Learning (RL) is designed and its convergence is analyzed. The authors analyze the reaching of stable equilibrium concerning population distribution, resource consumption, and average delay of IoT devices. Also, in [132], the offloading problem is modeled using replicator dynamics, and the achievement of a unique ESS is proved. Similarly, in [133], an evolutionary game is designed to assign weights to cloud robots with different computing powers. It finds the best offloading strategy with the lowest computational cost.

Focusing on improving the performance of partial offloading in fog environments, Khoobkar et al. [134] propose a two-population model based on replicator dynamics, in which offloading strategies and free capacity of the fog device are considered as players. The goal of the system is to achieve a ratio of full offloading to partial offloading. Unlike classical games that treat users as players, here each offloading strategy is considered as a population. Then, the authors model the interaction between the volume of offloaded cycles and free fog resources with two iterative equations. The authors obtain the conditions in which the system reaches a unique equilibrium. Simulations using iFogSim show that the proposed model achieves significant reductions in response time and energy consumption compared to state-of-the-art methods. Lu et al. [135] focus on optimizing the offloading of smart grid transmission line inspection robots. The goal is to enable robots to distribute heavy computational tasks between local processors and edge servers. By modeling this process as a distributed game and using a game theory mechanism, the offloading strategy is dynamically optimized to minimize the offloading cost, communication latency, and server load. Simulation results show that the proposed strategy significantly reduces latency compared to state-of-the-art strategies while improving system efficiency and resource utilization. Asghari et al. [136] address the problem of optimal placement of edge servers in the IoT architecture using the Whale Optimization

Algorithm (WOA) and game theory. The main challenge is to find suitable locations for deploying edge servers to reduce latency and increase coverage and energy efficiency. The authors propose a multi-objective approach that combines the WOA and game theory. First, using game theory, the competition between users and edge servers is analyzed to model user satisfaction and demand. Then, WOA is used to search the solution space and find optimal server locations. Simulation results show that the proposed method has lower latency and energy consumption compared to well-known algorithms such as the genetic algorithm and Particle Swarm Optimization (PSO). It also increases the quality of user experience. Other advantages of the proposed method are high stability and adaptability to complex and dynamic networks.

4.2.2 Stackelberg Game

In [137], a two-step Stackelberg game is designed to reduce delay and complexity. The authors prove that the system can reach Nash equilibrium. Similar research has been conducted in [121, 7] to offload the computation between MD and ECS. In [138], a pricing method based on the Stackelberg game is proposed with the aim of equal allocation of resources among mobile users. Such pricing methods have also been applied in [139, 140, 141]. In some studies [142, 143], offloading is addressed in UAV-MEC and V2V. The authors in [142] model the interaction between the UAV-MEC server and mobile phone users as a Stackelberg game and prove the existence of Nash equilibrium both theoretically and experimentally. In a similar research [144], the Stackelberg game is used to create a mechanism between the three entities of the BS, UAV, and user. Also, according to some requirements for Device-to-Device (D2D) communications, especially in Delay-tolerant Networks (DTNs), some research has been done in this field [145].

The use of the Stackelberg game is not limited to offloading. It has also been used in other areas such as blockchain [146]. In [146], a Stackelberg game is proposed to coordinate the needs of blockchain users and miners in e-health. Here, e-health observers, as blockchain users, can validate data related to health transactions with the help of miners. Khawam et al. [147] propose a two-level game-based framework for offloading tasks from roadside stations (RSUs) to vehicles in vehicular fog networks. At the first level, a two-stage Stackelberg game is defined between the RSU as the leader and the vehicles within the coverage area as followers. Then, at the second level, the RSU delegates the received tasks to the vehicles. Their results show that the proposed mechanism reduces the average waiting time and improves the probability of deadline expiration and blocking of requests. Wu et al. [148] design a two-level Stackelberg game for optimal resource allocation and computing node motivation: a) in the first level, intermediaries as leaders determine the purchase price of resources, and computing nodes as followers determine the supply volume, and b) in the second level, users (leaders) propose the price to pay and the intermediary (followers) allocates the service volume based on that. They use a three-way adaptive game for modeling. The proposed method leads to an increase in the total profit of the players and a decrease in their energy consumption. In a similar study by Chen et al. [149], a two-stage Stackelberg game is used between edge/cloud servers (leaders) and requesting vehicles (followers). Also, Duan et al. [150] model offloading in cloud-robot systems using a multi-stage Stackelberg game. The proposed algorithm, with strategy updates based on maximizing the change in the potential function, ensures that a unique Nash equilibrium is achieved after a finite number of iterations. The Stackelberg game-based incentive mechanism has also been used in other research. For example, Wang et al. [151] address the edge-UAV network as mobile servers. The problem is solved in two steps. First, a server selection game determines which tasks each drone accepts from which users to minimize energy consumption. Then, to encourage drones to provide services, a Stackelberg game is modeled between each drone (leader) and the users under its coverage (followers). Simulation results show that this Stackelberg game succeeds in converging in less than 200 iterations and increases the average user satisfaction. A similar study by Chen et al. [152] was also conducted at two levels of modeling using the Stackelberg game. Recently, transfer learning has attracted much attention from researchers. Zhang et al. [153] aim to improve the efficiency of solving two-stage offloading game models using machine learning. Here, a neural network is used to reduce the time complexity of solving the Stackelberg game. It approximates the relationship between the input characteristics (power, delay, number of cars, etc.) and the equilibrium behavior of the cars. Then, a transfer learning algorithm is used to generalize the trained model to new environments with different data distributions. Their results show that transfer learning reduces the convergence time by more than 50%.

4.2.3 Non-cooperative Games

In [154], interactions of non-cooperative users in competition over heterogeneous resources are modeled. Using queuing theory, the authors analyze the strategies as well as the payoff functions of all mobile users. It finds the best performance of each mobile user at the Nash equilibrium point. A similar study was conducted in [155], which requires only a limited amount of user-system information. However, the proposed method cannot be applied in the real world because the authors do not pay any attention to the user's workload. In [156], using the NBS approach, a non-cooperative game framework with low complexity is presented. As an advantage, the proposed method can be applied to other resource allocation models. Its most important weakness is not paying attention to the migration of containers in different administrative domains. Other similar modelings have been done in [31, 157, 158]. The authors define task execution delay and energy consumption as computational overhead for offloading.

Part of the research in this field is dedicated to vehicle networks [159, 160, 161]. For example, the proposed method in [159] allows each vehicle to adjust its offloading probability. Also, in [160] the goal is to use idle resources of parked vehicles. Xu et al. [162] address offloading optimization in vehicular edge networks for Web3 architecture. They divide the network space into smaller parts according to the capacity of each RSU to obtain the best offloading policy and resource allocation. To this end, they define a dynamic potential game in which vehicles (as players) try to satisfy the quality of service (QoS) by choosing offloading options. The proposed algorithm has log-linear complexity and converges in a finite number of iterations. Their results show that this method reduces energy consumption and power consumption compared to local processing or simple binary offloading methods. Another study by Wu et al. [163] uses a potential game to model offloading, in which each player (user) tries to minimize his/her cost. They prove that this game always reaches a pure Nash equilibrium. The proposed algorithm slows down the interference by sending tasks on OFDMA channels and feedback on the received power from the base stations and computes the best response at each iteration. Complexity and chaos cost analysis shows that the algorithm converges in polynomial time. Simulations on 16-cell networks with 50 users prove that this method reduces the total system cost by up to 30% compared to purely local execution and improves latency and energy consumption. Xu et al. [164] formulate the serviceability ratio maximization. They divide the problem into two sub-problems: a) task offloading, which is modeled as an exact potential game and reached a Nash equilibrium by the distributed algorithm MAD4PG, and b) resource allocation, which is solved by two independent convex optimization problems with gradient-based method and KKT conditions. Simulation results show that the proposed approach outperforms state-of-the-art methods in terms of service ratio and processing time metrics. In another study [165], a smart broker architecture is introduced that manages load distribution and competition among Edge Providers (EPs) based on price, service delay, and security. They present a non-cooperative duopoly model to analyze the influence of these parameters on offloading decisions and evaluate the effect of EPs' selfish behavior on system efficiency by using the "Price of Chaos" (PoA) metric. Their simulation results show that this method has relatively fast convergence. Li et al. [166] propose a two-level framework for task offloading. First, each user decides which tasks to delegate to edge servers using a non-cooperative game and which ones to perform locally. Then, through an auction game, which takes place between edge servers, the servers' profit and resource efficiency are maximized. The authors show using theoretical analysis that the proposed games can reach equilibrium. In a similar study [167], offloading in a vehicular network is formulated as a multi-player non-cooperative game. Here, each player (vehicle) tries to minimize its total delay and energy consumption by choosing the most appropriate handover option. Simulations show that the proposed algorithm increases the probability of handover success by about 25% and reduces the overall delay, compared to reference methods.

4.2.4 Bayesian Games

One of the real-world problems is that end devices have incomplete information about each other and the network load. This causes a decrease in the capacity of the wireless links connected to the local BSs and mutual interference, and in turn, degrades the offloading efficiency. Bayesian games may be a suitable tool for modeling such real-world requirements [168, 169]. A Bayesian scenario with incomplete information may lead the user to achieve better social welfare than scenarios with perfect knowledge. This is due to the competitive nature of the game, where selfish rational users try to offload tasks even when it is not efficient for the entire network.

4.2.5 Coalition games

An important part of GT is dedicated to coalition [170, 171]. In [172], coalition games are used for offloading in MEC and a distributed algorithm is obtained. The authors prove that the obtained solution is convergent and stable. Their results show that the proposed method improves the offloading ratio. In some studies, non-cooperative coalition mechanisms have been used. In [173], context-aware offloading is targeted. The proposed method reduces energy consumption by optimal channel allocation between users and at the same time meets the deadline of each sub-task. To solve the problem, the authors design a coalition sub-game. Another reason for the different delays of each sub-task is the limitation of the computing capacity of end devices. In [174], based on the characteristics of each sub-task, the content-aware offloading problem is modeled as a sub-game. The goal is to minimize the total energy and improve the channel allocation. The authors model the problem as a cooperative sub-game based on the coalition.

In some studies, the combination of coalition games with other games has been addressed. For example, in [175] the combination of coalition and bargaining is used. By using collaborative GT, the proposed method maximizes the synergy between users. This improves the system's efficiency and brings mutual advantages to the parties. In [176], a hedonic coalition formation game is proposed in which each user chooses the optimal coalition according to his/her utility. Here, an auction is implemented in which several cluster heads try to encourage their covered users to share more resources.

4.2.6 Multi-player Games

To satisfy real-world requirements, GT-based methods must consider a large number of users [33, 131, 177, 178, 179]. On the other hand, previous studies show that if the number of MUs exceeds a critical number, the offloading performance decreases [3, 180, 181, 182, 183, 184]. In [129] multi-user-multi-server (MUMS) offloading is addressed for MEC in ultra-dense 5G cellular networks. First, users are grouped based on their workload and their distance from the BS. Then the main problem is divided into several sub-problems and each one is solved in parallel using a genetic algorithm. One of the advantages of this modeling is the reduction of delay and energy consumption. In [185], computation offloading in OFDMA is addressed using GT. The designed offloading strategy can achieve a Nash equilibrium and maximize the number of users. It includes algorithms to determine the offloading thresholds that are necessary in the real world. The authors in [179] aim to minimize the end-to-end computational delay. They first try to solve the problem with a Mixed Integer Nonlinear Programming (MINLP) method, which is inefficient on a large scale. Then, to overcome this challenge, they propose a heuristic offloading scheme that requires little time to converge. In [186] offloading in D2D-VEC is addressed with hybrid strategies. The players in this game are SDN controllers who manage the allocation of computing resources at the edge of the network. The authors not only prove the existence of Nash equilibrium but also prove the uniqueness of the equilibrium point.

Recently, the combined use of GT and RL methods has become popular. Here, RL is used to exploit the experiences acquired by the agent from the environment. In [187], the combination of GT and DQN is considered to solve the offloading problem by considering the limitations of multi-channel interference. In such studies, GT is used to determine the optimal task offloading strategy to improve QoS. Meanwhile, RL is applied to implement dynamic resource allocation [188]. In [189], major challenges in MEC for offloading are investigated. In this research, GT and RL are used. It formulates the energy-efficient edge server activation problem using minority games. In [190], Online RL is used to handle scenarios with unknown future information and incomplete local task information. The most important achievement of this research is the optimization of task completion time. Unfortunately, it ignores the end device's battery limitations.

Interested readers can refer to [155] to study other research. Zhu et al. [191] propose a stochastic game framework to optimize the task offloading of agricultural robots. Robots can choose the most suitable node to offload tasks based on latency requirements, packet loss rate, and energy consumption. To learn the optimal strategies, a Q-learning Nash algorithm based on a stochastic game is designed. Robots achieve collective Nash superposition by cooperating. Gao et al.

[192] model three games in the architecture of 6G networks: a) a game between users and edge nodes to delegate tasks, b) a game between edge nodes to select cacheable content, and c) a game between resource providers to determine service prices. The highlights of this research are its high adaptability to dynamic conditions and scalability for implementation in future high-performance networks. Another study [193] uses deep reinforcement learning (DDPG) to make optimal decisions in data outsourcing. Then, game theory is used to encourage user participation.

4.2.7 Cooperative Games versus Competitive Games

Our literature review shows that few studies have been conducted using competitive GT for computation offloading. In [2], an algorithm with polynomial complexity is proposed for the balanced allocation of resources. The experiments show that the system's equilibrium points have a good performance against different workload scenarios. In [29], the problem of offloading in very dense IoT environments (for example, a hyper-dense multi-user edge server scenario) is addressed. The authors aim to minimize the overall computational overhead considering the limitations of the wireless channel. They solve the problem using a greedy heuristic method.

In contrast to competitive games, the number of studies based on collaborative GT is impressive [194, 195]. The authors in [196] propose a dynamic offloading strategy in the presence of geographically distributed edge servers with time-varying conditions. One of the major problems with traditional GT is the lack of scalability. The reason is that here the user is considered as a player. As a result, when the number of users increases, it becomes very time-consuming to reach the equilibrium point. To resolve this issue, in [160], a partial offloading method based on replicator dynamics is proposed. Here, unlike traditional GT, strategy is used as a player. This makes the proposed method scalable. In [197], a MEC architecture is designed for a dense network with several BSs and several UEs. The authors formulate the computation offloading problem for decomposable tasks to minimize the completion time. They use replicator dynamics to model the offloading decision.

Game theory models often suffer from high computational complexity, making them challenging for real-time offloading decisions in IIoT environments. In this regard, some research has proposed interesting tricks to mitigate this problem. Li et al. [198] propose a two-stage framework for task offloading. In the first stage, end nodes decide which tasks to send to which edge gateways through a potential game. In the second stage, each edge gateway solves a convex optimization problem based on the supply-demand relationship to determine the optimal price of a unit task. The authors prove that the potential game achieves a unique and optimal equilibrium. In [199], the authors model task offloading in 5G networks using cooperative game theory. Each user (player) decides which computing node to delegate their tasks to. In contrast, the servers aim to distribute their resources among users based on the cost-benefit. Their results show that the proposed mechanism leads to reduced latency and increased efficiency. Wang et al. [200] use a combination of game theory and deep reinforcement learning. First, service provider and service consumer vehicles are modeled in a cooperative game. The goal of this system is to minimize the sum of processing delay and energy consumption. Then, a Markov problem with a reward function minus the sum of delay and energy is modeled. In another study, Cao et al. [201] proposed a cooperative system for task offloading. In this model, each cloud robot is considered a player and tries to optimize its offloading strategy.

4.2.8 Algorithmic Games

Traditional offloading schemes such as classical and learning-based optimization algorithms lead to high overhead [202, 203]. In [204], a GT-based scheduling algorithm is proposed for offloading in MEC. In [205], the Particle Swarm Optimization (PSO) algorithm is used in combination with GT to obtain offloading strategies. In some studies, unknown games have been used [204, 206, 207, 208, 209]. For example, in [190], several potential games are developed. In [57, 210, 211], the multi-user offloading problem is addressed, and theoretically proven that there is at least one Nash equilibrium strategy. In [212], a non-cooperative game is used to obtain the best response offloading strategy. A mobility-aware computation offloading framework is proposed in [213]. Its major parts include a) a user mobility prediction module; b) selection of a set of edge/fog devices based on the predicted mobility and current location of the user; c) selection of edge/fog devices availability according to the current load; and d) offloading calculations to the available device. In [214], a non-cooperative game between MDs in the Industrial Internet of Things (IIoT) environment is designed based on MEC. The purpose of this study is to integrate the combined benefits of energy consumption, time delay, and resource cost. In [143], end users can intentionally reserve some time for local preprocessing before requesting an offloading service. Thus, the offloading cost may be reduced due to less demand on edge resources, while the delay cost may be increased due to later reporting of offload requests.

Most studies only consider wireless channel congestion without considering the influence of edge nodes [159]. In addition, centralized offloading strategies lead to heavy computational complexity in core nodes. In [215], joint wireless channel congestion and offloading among multiple edge nodes are considered. First, the offloading problem is formulated as a potential multi-user game, and then a distributed algorithm is presented to reach an equilibrium. In [216], edge computations for two-layered satellite networks are addressed. Any Low Earth Orbit (LEO) satellite can offload computation workloads to Geostationary Orbit (GEO) satellites. Here the offloading problem is modeled as a two-way matching game with LEO satellites and GEO satellites as participants. In [217], several computation offloading schemes for homogeneous/heterogeneous users are presented. The proposed algorithms can effectively minimize system cost while scaling well as the number of users increases. In [218], the random dynamic game model is used to obtain the equilibrium point. The proposed algorithm can quickly obtain the user's equilibrium strategy after a limited number of iterations. One of the disadvantages of this research is that it does not address the dynamics of the environment and how to allocate computing resources. In [203], obtaining the appropriate data size and pricing is formulated as a two-step optimization problem.

Mobility support is one of the key features for users. One of the major challenges is how to manage battery power for fog devices. Scheduling of shared edge/fog resources is another problem that needs to be considered in these computing paradigms [204]. In [219], a GT-based method is proposed to minimize response time and energy consumption for each user. In [181], a fuzzy neural model is used to remove invalid sources. In this study, the PSO is used to select an appropriate fog server. One of the disadvantages of this architecture is that it can only be used on a small scale. Decentralization, privacy, and reliability are major challenges of this research.

In IoV, due to the high speed of the vehicles, the driving assistance services must be performed on time. Combining MEC with IoV resolves the issue of insufficient local computing resources and thus improves QoS. Here, minimizing the processing delay of users' tasks on limited edge server resources is still an open challenge. In [133], a fuzzy offloading scheme in combination with GT is presented. In [220], a MEC architecture is presented that exploits the computing capacity of parked vehicles. There are few studies in which Software-defined Networks (SDNs) have been used. In [221] the SDN controller manages the D2D pairing of vehicles. It combines D2D and VEC communications to further increase the computing capacity of the vehicle network. The goal is to find the maximum number of vehicles that VEC can support under communication and computational constraints. Here, the offloading and resource allocation problem is formulated as a mixed-strategy game. The two major players in this game are the SDN controller and the vehicles. The authors prove the existence and uniqueness of Nash equilibrium theoretically.

One of the goals of Unmanned Aerial Vehicles (UAVs) is to reduce energy consumption while meeting the desired QoS during the mission. In [222], an offloading game is presented to minimize the total cost. Experiments show that the proposed algorithm is scalable and reaches a Nash equilibrium. One of the shortcomings of this research is not taking into account the impact of the mobility of drones. In the literature, coalitions of UAVs have been widely used in urgent missions. In such a situation, coalition leaders act as servers to help other members. In [219], joint optimization of scheduling and resource allocation is addressed. In [223, 224], the joint minimizing delay and maximizing channel access in UAVs is addressed in MEC.

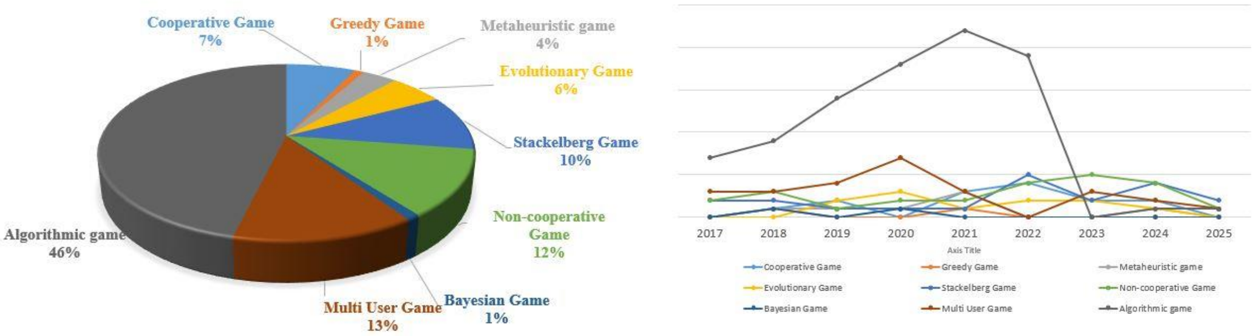
There are very few studies in which partial offloading is addressed. In [179, 178], an offloading scheme with low time complexity is presented to achieve Nash equilibrium. In [225], partial offloading is addressed in a cloud-side cooperative scenario. First, through GT, the optimal user-edge offloading decision is found. Then, the edge-cloud optimal offloading decisions are obtained. The authors in [226] use a combination of deep reinforcement learning and Nash superposition game theory. First, the vehicle motion patterns are extracted with a convolutional neural network. Then, the offloading delay is reduced using the Nash superposition game algorithm. Duan et al. [150] address the task offloading problem with a potential game. Here, each vehicle acts as a player and tries to minimize its total cost by choosing a target server. Simulation results show that the proposed mechanism reduces the cost by about 20%.

5 DISCUSSION AND ANALYSIS OF RESULTS

We now answer questions RQ1 to RQ8 that were previously formulated in Section 3.

RQ1: Which GT algorithms are used in computation offloading?

Analysis Result: As shown in Figure 9-a, algorithmic games have the largest share of GT-based modeling with 46%. After that, multi-player games with 13%, non-cooperative games with 12%, and Stackelberg games with 10% are in order. Figure 9-b also reveals that the growth of evolutionary games has accelerated in the last few years, especially from 2021-2022 onwards. It has attracted the attention of researchers more than classic games. In total, the results show that offloading policy, environment specifications, number of agents, and their learning method are the most important factors in choosing the GT technique. The computing environment mainly includes both discrete and continuous variables. Some GT techniques can only be used with continuous variables and some with discrete ones. Due to the NP-hard nature of the offloading problem and the large action/state space, the use of evolutionary game techniques is expanding. In recent years, thanks to parallel/remote computing, the convergence delay for calculating the Nash equilibrium has become much smaller. This is one of the main reasons for the development of evolutionary GT schemes.

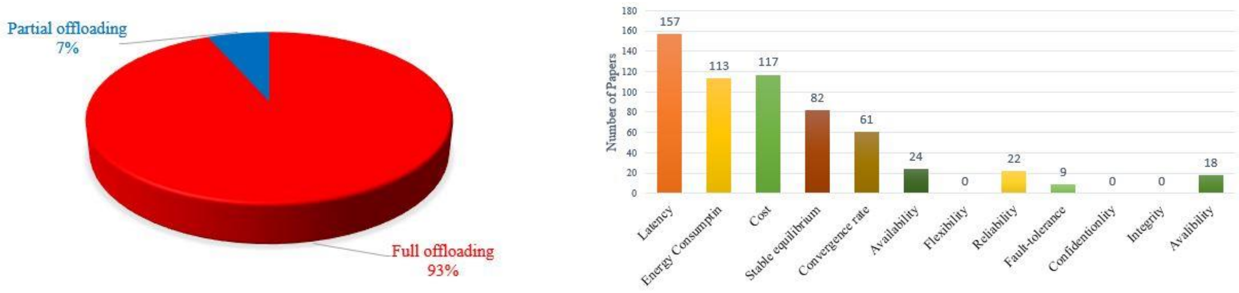


a) Distribution of GT-based methods **b)** Trends in GT-based techniques

Figure 9: Distribution and trend of GT-based offloading research

RQ2: Which offloading scheme, binary or partial, is used for each GT-based approach?

Analysis Result: It was explained earlier in Section 1 that there are two main types of offloading: full (binary) offloading, and partial offloading. As shown in Figure 10-a, 93% of the studies used a binary offloading scheme and the remaining 7% used partial offloading. However, our literature review shows an increasing trend in partial offloading studies. Due to the proliferation of parallel processing applications, especially video streaming, the contribution of partial offloading research is expected to increase dramatically in the near future.



a) Distribution of binary/partial offloading research **b)** Distribution of performance metrics

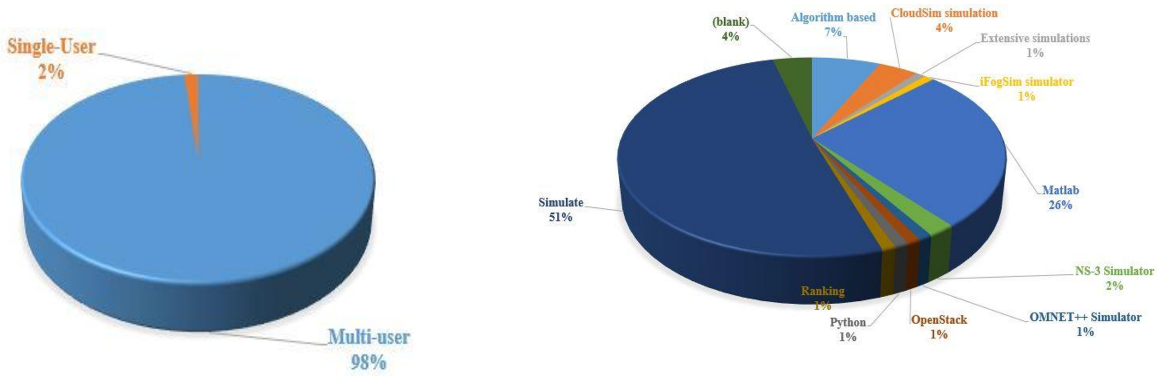
Figure 10: Research distribution and performance criteria in GT-based research

RQ3: What performance criteria are used in GT-based offloading approaches?

Analysis Result: As shown in Figure 10-b, in most GT-based offloading studies, the performance measures of delay, cost, Nash equilibrium, and convergence rate have attracted the most attention of researchers. Our literature review showed that the delay metric was used in 157 references, cost in 117 papers, Nash equilibrium in 82 references, and convergence rate in 61 references. Nash equilibrium is a key metric in GT-based schemes. According to the results, delay, and cost are the most common performance evaluation criteria in most computing paradigms.

RQ4: What are the principles of each offloading approach on GT? Is it implemented as a single-agent or multi-agent?

Analysis Result: As shown in Figure 11-a, 98% of studies are based on multi-agent, and only 2% of them are single-agent. Single-agent methods are not able to capture the complex interactions of users in the network. This is also true in other computing paradigms.



a) SAGT/MAGT distribution **b)** distribution of simulation tools

Figure 11: SAGT/MAGT distribution along with simulation tools in GT-based offloading research

RQ5: What simulators do researchers use to evaluate the performance of GT-based approaches?

Analysis Result: As shown in Figure 11-b, about 51% of the researchers did not provide any information about the simulation tool. Among the rest, 26% have used MATLAB toolboxes and 7% have simulated their work based on mathematical algorithms. Approximately, 4% of researchers have used CloudSim and 2% have used the NS tool. A few of them, who were not interested in having simulation problems, programmed themselves with Python language. The validation of the research of this group of researchers is subject to serious doubts.

RQ6: What GT-based approaches are used to meet the needs of each complementary paradigm?

Analysis Result: As shown in Figure 12, most GT-based studies have been conducted in the MEC paradigm, which includes 95 papers. Among the rest, 46 studies have been conducted in EC, 14 in FC, and 12 in CC. Due to the ever-increasing growth of FC infrastructures, it is expected that the contribution of studies in this paradigm will increase. In this regard, topics such as mobility management, reliability, security, and node heterogeneity may attract more attention from researchers.

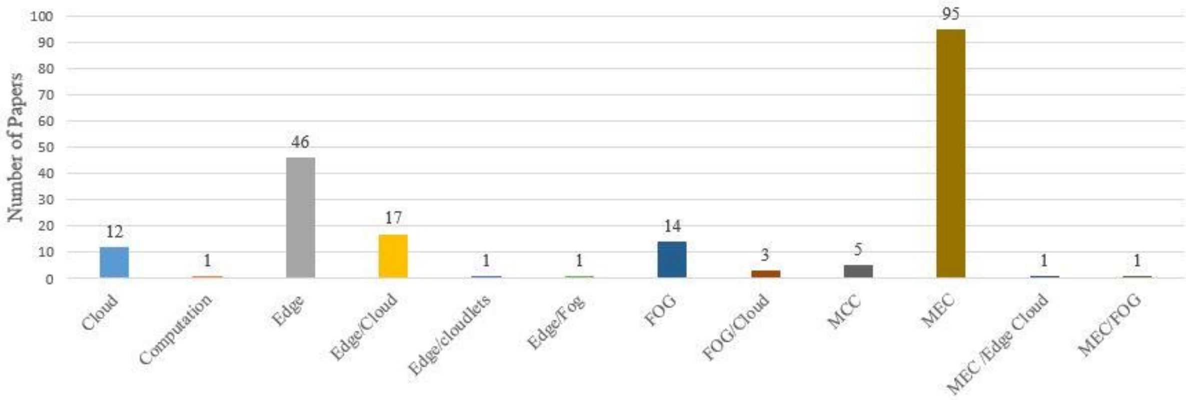


Figure 12: Distribution of paradigms in GT-based offloading

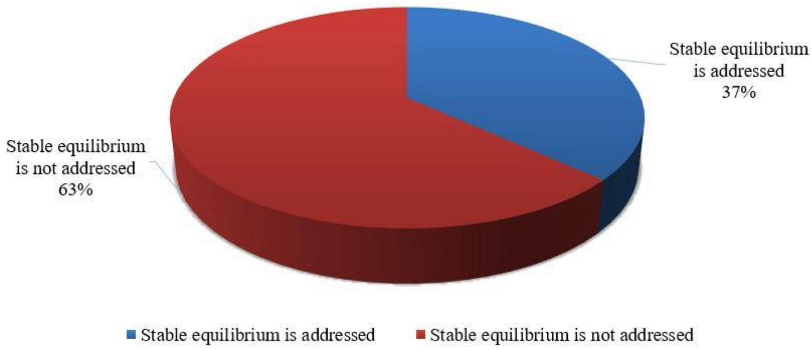


Figure 13: Addressing stable equilibrium in game theoretical studies

RQ7: What are the main features of each GT-based approach? What are the main features of hybrid approaches?

Analysis Result: As shown in Figure 13, about 63% of previous studies have not addressed reaching the stable equilibrium point. This reveals that achieving stable equilibrium in engineering applications, especially in dynamic environments such as edge/fog/cloud, is a challenging issue. Tuning GT hyperparameters is not a trivial task at all. For this reason, it is difficult to design game-theoretic methods for real-world applications. Also, the "Characteristics" column in the appendices tables describes the main idea of each research. It shows that recently special attention has been paid to the combination of GT and new techniques such as RL. These combinations can impose additional complexity in tuning GT hyperparameters in real-world applications.

RQ8: What are the future research trends and open issues in GT-based computation offloading?

Analysis Result: Based on the literature review, the following future directions can be identified:

- **Partially Observable Environments and Incomplete Information:** Unfortunately, computation offloading environments are not always fully observable. Not only the user cannot access all the information of others, but also the devices do not have information about each other. As explained earlier, Bayesian GT can be used to resolve this issue. Using evolutionary GT in combination with Bayesian methods can be an attractive idea. Here, the Bayesian method is used to exploit incomplete information. On the other hand, evolutionary GT is used to provide scalability. Recall that in evolutionary GT, the user is not considered a player. Instead, it considers the strategy as the player. This causes the solution not to be dependent on the number of users and thus provides scalability.

  Another future direction may be to combine GT with RL. Here, RL is used to exploit the experiences gained by the agent from the environment [187], while GT is used to determine the optimal offloading strategy. Considering recent advances in computing methods, especially quantum computing, it is expected that in the near future, time complexity will no longer be a problem. Cao et al. [201] address security and trust in the computational offloading of IoT devices. They model trust evaluation at two levels: a) direct trust between the requester and the resource provider and b) trust of resources based on the quality of service received from servers. They propose a distributed decision-making mechanism based on game theory to avoid the central point of failure. They also secure offloading transactions using blockchain. Shakkeera et al. [227] propose an actor-critic mechanism and an auction mechanism for offloading decision-making. Their evaluation results show that the proposed scheme significantly reduces service latency compared to several state-of-the-art algorithms. Gu et al. [228] present a hybrid model that has three main steps. First, a trust score is calculated for each requester and resource provider. Then, using game theory, distributed decision-making is implemented to avoid the central weakness problem. In the next step, a blockchain-based mechanism is introduced to make transactions immune to free-riding attacks. The proposed method reduces energy consumption and latency and improves network reliability.

Recently, a new paradigm called Industry 5.0 has emerged after Industry 4.0. The key feature of Industry 5.0 is to highlight and mainstream human-centric approaches, sustainability, and resilience in industrial operations [229]. Here, special importance is given to human-machine interaction and the use of artificial intelligence. This, in turn, targets social and environmental issues. Edge computing has a prominent place in Industry 5.0 because it reduces latency at the edges of the network and improves decision-making. Sharma et al. [229] examine the role of edge computing in Industry 5.0 and address research gaps in this area by analyzing the fundamentals, applications, and challenges. They demonstrate that key technologies in this area include 6G, blockchain, quantum computing, digital twins, and collaborative robots. Also, the most important applications in this area include predictive maintenance, smart health, supply chain, and industrial transportation. Their proposed architecture consists of three layers: industrial sensors, near-edge computing, and cloud computing. Their results show that the IoT with a 30% share and edge computing with a 20% share are the most important dominant technologies, respectively.

- **NP-hard Multi-objective Optimizations:** As explained in Section 4, most previous studies have focused on the joint optimization of delay and energy consumption [2, 89]. As artificial intelligence develops, new multi-objective problems may be defined that better match real-world requirements. Scheduling, mobility management, reliability, fault tolerance, availability, and security are among the most important single objectives. Although achieving an optimal combination of these goals is an NP-hard issue, recent advances in computing power are promising. Also, with the emergence of distributed trust technologies such as blockchain, establishing security in FC will be facilitated. Among the most important research topics related to this issue is the simultaneous optimization of two sub-problems of task offloading and service caching in VEC [230]. Here, the task offloading sub-problem may be optimized through a many-to-one adaptive game between requesters and edge devices (RSUs or volunteer vehicles). Another related research topic is the joint optimization of two sub-problems of resource allocation and task offloading. For example, in [231], a bargaining game is used between the vehicle (resource buyer) and the server (resource seller) to agree on the optimal value of the resource unit price. As another example, in [232], to overcome the NP-hardness of the problem, it is transformed into a non-cooperative game, and the existence of a Nash equilibrium is guaranteed using potential game theory. Then, an iterative game-theoretic algorithm is applied, which converges to equilibrium at each step by minimizing the cost (delay and energy) and updating the strategies through global controllers. Another research topic is the simultaneous attention to the movement-aware strategies and resource allocation in the MEC based on game theory. For example, in [233], two sub-problems are solved: a) the Lagrangian multiplier method is used to find the optimal distribution of resources among users; b) then, a non-cooperative game is formed among users for offloading decision-making.

- **Partial Offloading and Emerging IoT Applications:** A new wave of applications is emerging that make the use of partial offloading inevitable. The most important examples are augmented reality, metaverse, computer games, smart city, e-health, IIoT, industrial and agricultural monitoring, etc. MEC and FC require a lot of interaction with the CC due to resource limitations. In addition, partial offloading itself has prerequisites such as determining parallelizable parts of the task. It seems that part of future research is to provide complementary architectures to resolve these problems, especially for IoT. For example, in [234], multi-agent stochastic learning with entropy boosting is used to accelerate convergence. It broadens the search space by adding an entropy component to the reward function and prevents the algorithm from getting stuck in local optima. In [235], offloading is modeled for delay-constraint tasks. First, the base station selection by IoT devices is performed using a many-to-one matching game. Then, resource pricing and offloading of tasks are designed as a two-stage Stackelberg game. AI and IoT play a crucial role in advancing Industry 5.0 by combining human cognition with intelligent automation [236]. Also, the Industrial Internet of Things (IIoT) as the backbone of this transformation, by connecting sensors and intelligent machines, increases efficiency through AI-based decision-making. The IoT architecture includes perception, communication, and application layers, supported by real-time feedback loops for continuous improvement. Standards such as ISO/IEC 30141 and IEEE P2413, along with protocols such as MQTT and CoAP, ensure the secure and interoperable integration of AI and the IoT. Key challenges in this area are the scalability of decentralized systems and the resource constraints of devices [237]. For this purpose, DRL approaches are increasingly being used [238]. The most important challenges of DRL methods are buffer and CPU consumption. Future directions include upgrading the single-agent solutions to multi-agent ones to increase productivity.

6 CONCLUSION AND FUTURE TRENDS

This paper presented a systematic review of current research on game-theoretic (GT) techniques in computation offloading. We analyzed the advantages and disadvantages of each GT technique for different performance metrics. Ultimately, the survey answered eight basic questions. It showed that algorithmic GT with an approximate share of 46% and multi-player games with 13% are the most popular GT techniques. 93% of the studies used a binary offloading scheme and the remaining 7% used a partial design. In most GT studies, performance metrics such as delay, response time, energy consumption, convergence rate, QoE, and cost have attracted the most attention of researchers. Almost 4% of the articles did not mention their evaluation tool. Among the others, 5% have used a combination of Python and CloudSim, 4% CloudSim, 1% Python and 26% Matlab. Also, our results showed that most GT-based studies have been conducted on MEC.

We concluded that the most important research trends for the next decade are: exploiting the partially observable environments, multi-objective optimizations, partial offloading, and emerging IoT applications. It is expected that the combined evolutionary GT and RL will be expanded to resolve incomplete information issues. Here, RL is used to exploit incomplete information and evolutionary GT to guarantee scalability.

REFERENCES

[1] Panwar, N., Sharma, S., & Singh, A. K. (2016). A survey on 5G: The next generation of mobile communication. Physical Communication, 18, 64-84.

[2] Cardellini, V., De Nitto Personé, V., Di Valerio, V., Facchinei, F., Grassi, V., Lo Presti, F., & Piccialli, V. (2016). A game-theoretic approach to computation offloading in mobile cloud computing. Mathematical Programming, 157, 421-449.

[3] Liang, S., Wan, H., Qin, T., Li, J., & Chen, W. (2020, October). Multi-user computation offloading for mobile edge computing: A deep reinforcement learning and game theory approach. In 2020 IEEE 20th International Conference on Communication Technology (ICCT) (pp. 1534-1539). IEEE.

[4] Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. Future generation computer systems, 56, 684-700.

[5] Hossain, E., & Hasan, M. (2015). 5G cellular: key enabling technologies and research challenges. IEEE Instrumentation & Measurement Magazine, 18(3), 11-21.

[6] Hsu, T. C., Yang, H., Chung, Y. C., & Hsu, C. H. (2020). A Creative IoT agriculture platform for cloud fog computing. Sustainable Computing: Informatics and Systems, 28, 100285.

[7] Li, N., Yan, J., Zhang, Z., Martinez, J. F., & Yuan, X. (2020, December). Game theory based joint task offloading and resource allocation algorithm for mobile edge computing. In 2020 16th International Conference on Mobility, Sensing and Networking (MSN) (pp. 791-796). IEEE.

[8] Akyildiz, I. F., Nie, S., Lin, S. C., & Chandrasekaran, M. (2016). 5G roadmap: 10 key enabling technologies. Computer Networks, 106, 17-48.

[9] Yang, L., Zhang, H., Li, X., Ji, H., & Leung, V. C. (2018). A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing. IEEE/ACM Transactions on Networking, 26(6), 2762-2773.

[10] R. Gibbons, A primer in game theory. Harvester, 1992.

[11] Dutta, P. K. (1999). Strategies and games: theory and practice. MIT press..

[12] Han, Z., Niyato, D., Saad, W., Başar, T., & Hjørungnes, A. (2012). Game theory in wireless and communication networks: theory, models, and applications. Cambridge university press.

[13] Zhang, Y., & Guizani, M. (Eds.). (2011). Game theory for wireless communications and networking. CRC press.

[14] MacKenzie, A. B., & DaSilva, L. A. (2006). Game theory for wireless engineers. Synthesis Lectures on Communications, 1(1), 1-86

[15] Wang, W., Kwasinski, A., Niyato, D., & Han, Z. (2016). A survey on applications of model-free strategy learning in cognitive wireless networks. IEEE Communications Surveys & Tutorials, 18(3), 1717-1757

[16] Abdalzaher, M. S., Seddik, K., Elsabrouty, M., Muta, O., Furukawa, H., & Abdel-Rahman, A. (2016). Game theory meets wireless sensor networks security requirements and threats mitigation: A survey. Sensors, 16(7), 1003.

[17] Chen, X., Proulx, B., Gong, X., & Zhang, J. (2014). Exploiting social ties for cooperative D2D communications: A mobile social networking case. IEEE/ACM Transactions on Networking, 23(5), 1471-1484.

[18] Song, L., Niyato, D., Han, Z., & Hossain, E. (2014). Game-theoretic resource allocation methods for device-to-device communication. IEEE Wireless Communications, 21(3), 136-144.

[19] Islam, A., Debnath, A., Ghose, M., & Chakraborty, S. (2021). A survey on task offloading in multi-access edge computing. Journal of Systems Architecture, 118, 102225.

[20] Zaman, S. K. U., Jehangiri, A. I., Maqsood, T., Haq, N. U., Umar, A. I., Shuja, J., ... & Alsharekh, M. F. (2022). LiMPO: lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. Cluster Computing, 1-19.

[21] Chen, R., Cui, L., Zhang, Y., Chen, J., Yao, K., Yang, Y., ... & Han, H. (2020, October). Delay optimization with FCFS queuing model in mobile edge computing-assisted UAV swarms: A game-theoretic learning approach. In 2020 International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 245-250). IEEE.

[22] Vailati Riboni, F. C. P., Comazzi, B., Castelnuovo, G., Molinari, E., & Pagnini, F. (2018). Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST. LECTURE NOTES OF THE INSTITUTE FOR COMPUTER SCIENCES, SOCIAL INFORMATICS AND TELECOMMUNICATIONS ENGINEERING, 253, 115-118.

[23] Asheralieva, A. (2019). Optimal computational offloading and content caching in wireless heterogeneous mobile edge computing systems with hopfield neural networks. IEEE Transactions on Emerging Topics in Computational Intelligence, 5(3), 407-425.

[24] Liu, D., Khoukhi, L., & Hafid, A. (2017, May). Decentralized data offloading for mobile cloud computing based on game theory. In 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC) (pp. 20-24). IEEE.

[25] Shakarami, A., Shahidinejad, A., & Ghobaei-Arani, M. (2020). A review on the computation offloading approaches in mobile edge computing: A g ame-theoretic perspective. Software: Practice and Experience, 50(9), 1719-1759.

[26] Gasmi, K., Dilek, S., Tosun, S., & Ozdemir, S. (2022). A survey on computation offloading and service placement in fog computing-based IoT. The Journal of Supercomputing, 78(2), 1983-2014.

[27] Wang, W., & Cao, L. (2019). Negative sequence analysis: A review. ACM Computing Surveys (CSUR), 52(2), 1-39..

[28] De Souza, A. B., Rego, P. A., Carneiro, T., Rodrigues, J. D. C., Reboucas Filho, P. P., De Souza, J. N., ... & Sikdar, B. (2020). Computation offloading for vehicular environments: A survey. IEEE Access, 8, 198214-198243.

[29] Lin, H., Zeadally, S., Chen, Z., Labiod, H., & Wang, L. (2020). A survey on computation offloading modeling for edge computing. Journal of [21] Chen, R., Cui, L., Zhang, Y., Chen, J., Yao, K., Yang, Y., ... & Han, H. (2020, October). Delay optimization with FCFS queuing model in mobile edge computing-assisted UAV swarms: A game-theoretic learning approach. In 2020 International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 245-250). IEEE.

[30] Huang, B., Li, Z., Tang, P., Wang, S., Zhao, J., Hu, H., ... & Chang, V. (2019). Security modeling and efficient computation offloading for service workflow in mobile edge computing. Future Generation Computer Systems, 97, 755-774.

[31] Su, Y., Fan, W., Liu, Y. A., & Wu, F. (2021). Game-based pricing and task offloading in mobile edge computing enabled edge-cloud systems. arXiv preprint arXiv:2101.05628.

[32] Adhikari, M., & Gianey, H. (2019). Energy efficient offloading strategy in fog-cloud environment for IoT applications. Internet of Things, 6, 100053.

[33] Chu, S., Fang, Z., Song, S., Zhang, Z., Gao, C., & Xu, C. (2020). Efficient multi-channel computation offloading for mobile edge computing: A game-theoretic approach. IEEE Transactions on Cloud Computing, 10(3), 1738-1750.

[34] Wang, Y., Yang, J., Guo, X., & Qu, Z. (2019). A game-theoretic approach to computation offloading in satellite edge computing. IEEE Access, 8, 12510-12520.

[35] Guo, H., Liu, J., Zhang, J., Sun, W., & Kato, N. (2018). Mobile-edge computation offloading for ultradense IoT networks. IEEE Internet of Things Journal, 5(6), 4977-4988.

[36] Guo, H., Liu, J., & Qin, H. (2018). Collaborative mobile edge computation offloading for IoT over fiber-wireless networks. IEEE Network, 32(1), 66-71.

[37] Li, Y., & Jiang, C. (2020). Distributed task offloading strategy to low load base stations in mobile edge computing environment. Computer Communications, 164, 240-248.

[38] Hong, Z., Chen, W., Huang, H., Guo, S., & Zheng, Z. (2019). Multi-hop cooperative computation offloading for industrial IoT–edge–cloud computing environments. IEEE Transactions on Parallel and Distributed Systems, 30(12), 2759-2774.

[39] Hassanzadeh, M., & Shahrrava, B. (2022). Linear version of Parseval's theorem. IEEE Access, 10, 27230-27241.

[40] Saeik, F., Avgeris, M., Spatharakis, D., Santi, N., Dechouniotis, D., Violos, J., ... & Papavassiliou, S. (2021). Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions. Computer Networks, 195, 108177.

[41] Masdari, S. A. G. A. M. (2020). M Hosseinzadeh M A survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective. J Grid Comput, 18(4), 639.

[42] Bauso, D., & Timmer, J. (2009). Robust dynamic cooperative games. International Journal of Game Theory, 38, 23-36.

[43] Bennis, M., Guruacharya, S., & Niyato, D. (2011, December). Distributed learning strategies for interference mitigation in femtocell networks. In 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011 (pp. 1-5). IEEE.

[44] Semasinghe, P., Hossain, E., & Zhu, K. (2014). An evolutionary game for distributed resource allocation in self-organizing small cells. IEEE Transactions on Mobile Computing, 14(2), 274-287.

[45] Doebeli, M., Hauert, C., & Killingback, T. (2004). The evolutionary origin of cooperators and defectors. science, 306(5697), 859-862.

[46] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. Information and software technology, 51(1), 7-15.

[47] Maray, M., & Shuja, J. (2022). Computation offloading in mobile cloud computing and mobile edge computing: survey, taxonomy, and open issues. Mobile Information Systems, 2022.

[48] Jiang, C., Cheng, X., Gao, H., Zhou, X., & Wan, J. (2019). Toward computation offloading in edge computing: A survey. IEEE Access, 7, 131543-131558.

[49] Zamzam, M., El-Shabrawy, T., & Ashour, M. (2020, October). Game theory for computation offloading and resource allocation in edge computing: A survey. In 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES) (pp. 47-53). IEEE.

[50] Tran-Dang, H., & Kim, D. S. (2022). A Survey on Matching Theory for Distributed Computation Offloading in IoT-Fog-Cloud Systems: Perspectives and Open Issues. IEEE Access.

[51] Cai, G., & Kock, N. (2009). An evolutionary game theoretic perspective on e-collaboration: The collaboration effort and media relativeness. European Journal of Operational Research, 194(3), 821-833.

[52] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

[53] Sadatdiynov, K., Cui, L., Zhang, L., Huang, J. Z., Salloum, S., & Mahmud, M. S. (2022). A review of optimization methods for computation offloading in edge computing networks. Digital Communications and Networks.

[54] Zhang, J., Xia, W., Zhang, Y., Zou, Q., Huang, B., Yan, F., & Shen, L. (2017, December). Joint offloading and resource allocation optimization for mobile edge computing. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-6). IEEE.

[55] Zhang, Y., Dong, X., & Zhao, Y. (2020, March). Decentralized computation offloading over wireless-powered mobile-edge computing networks. In 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS) (pp. 137-140). IEEE.

[56] Guo, H., Liu, J., & Zhang, J. (2018, May). Efficient computation offloading for multi-access edge computing in 5G HetNets. In 2018 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[57] Khoobkar, M. H., Dehghan Takht Fooladi, M., Rezvani, M. H., & Gilanian Sadeghi, M. M. (2022). Partial offloading with stable equilibrium in fog-cloud environments using replicator dynamics of evolutionary game theory. Cluster Computing, 25(2), 1393-1420.

[58] Lu, H., Gu, C., Luo, F., Ding, W., & Liu, X. (2020). Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning. Future Generation Computer Systems, 102, 847-861.

[59] Chen, X., & Liu, G. (2021). Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks. IEEE Internet of Things Journal, 8(13), 10843-10856.

[60] Shakarami, A., Ghobaei-Arani, M., & Shahidinejad, A. (2020). A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. Computer Networks, 182, 107496.

[61] Pan, S., Zhang, Z., Zhang, Z., & Zeng, D. (2019). Dependency-aware computation offloading in mobile edge computing: A reinforcement learning approach. IEEE Access, 7, 134742-134753.

[62] Piraveenan, M. (2019). Applications of game theory in project management: a structured review and analysis. Mathematics, 7(9), 858.

[63] Berry, R. A., & Johari, R. (2013). Economic modeling in networking: A primer. Foundations and Trends® in Networking, 6(3), 165-286.

[64] Osborne, M. J. (2004). An introduction to game theory (Vol. 3, No. 3). New York: Oxford university press.

[65] Barough, A. S., Shoubi, M. V., & Skardi, M. J. E. (2012). Application of game theory approach in solving the construction project conflicts. Procedia-Social and Behavioral Sciences, 58, 1586-1593.

[66] Kasthurirathna, D., & Piraveenan, M. (2014, December). Topological stability of evolutionarily unstable strategies. In 2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS) (pp. 35-42). IEEE.

[67] Myerson, R. B. (1991). Game theory: analysis of conflict. Harvard university press.

[68] Von Neumann, J. (1959). On the theory of games of strategy. Contributions to the Theory of Games, 4, 13-42.

[69] Asher, C. C., Mahoney, J. M., & Mahoney, J. T. (2005). Towards a property rights foundation for a stakeholder theory of the firm. Journal of Management & Governance, 9(1), 5-32.

[70] Chang, Y. L. (2007). Who should own access rights? A game-theoretical approach to striking the optimal balance in the debate over Digital Rights Management. Artificial Intelligence and Law, 15, 323-356.

[71] Bshary, R., & Oliveira, R. F. (2015). Cooperation in animals: toward a game theory within the framework of social competence. Current Opinion in Behavioral Sciences, 3, 31-37.

[72] Hadjichrysanthou, C., & Broom, M. (2012). When should animals share food? Game theory applied to kleptoparasitic populations with food sharing. Behavioral Ecology, 23(5), 977-991.

[73] Lundberg, S., & Pollak, R. A. (1994). Noncooperative bargaining models of marriage. The American Economic Review, 84(2), 132-137.

[74] Manser, M., & Brown, M. (1980). Marriage and household decision-making: A bargaining analysis. International economic review, 31-44.

[75] Von Neumann, J., & Morgenstern, O. (2007). Theory of games and economic behavior. Princeton university press.
[76] Hamilton, D., Goldsmith, A. H., & Darity Jr, W. (2009). Shedding “light” on marriage: The influence of skin shade on marriage for black females. Journal of Economic Behavior & Organization, 72(1), 30-50.
[77] Kasthurirathna, D., Nguyen, H., Piraveenan, M., Uddin, S., & Senanayake, U. (2014, August). Optimisation of strategy placements for public good in complex networks. In Proceedings of the 2014 International Conference on Social Computing (pp. 1-8).
[78] Shubik, M. (1970). Game theory, behavior, and the paradox of the prisoner's dilemma: Three solutions. Journal of Conflict Resolution, 14(2), 181-193.
[79] Polak, B. (2007). Econ 159: Game theory. Open Yale Courses.
[80] Kasthurirathna, D., Piraveenan, M., & Harre, M. (2013, August). Evolution of coordination in scale-free and small world networks under information diffusion constraints. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 183-189).
[81] Kasthurirathna, D., Piraveenan, M., & Uddin, S. (2015). Evolutionary stable strategies in networked games: the influence of topology. Journal of Artificial Intelligence and Soft Computing Research, 5(2), 83-95.
[82] Amadae, S. M. (2016). Prisoners of reason: Game theory and neoliberal political economy. Cambridge University Press.
[83] Martin, B. (1978). The selective usefulness of game theory. Social studies of science, 8(1), 85-110.
[84] Thedchanamoorthy, G., Piraveenan, M., Uddin, S., & Senanayake, U. (2014). Influence of vaccination strategies and topology on the herd immunity of complex networks. Social Network Analysis and Mining, 4, 1-16.
[85] Ritzberger, K. (2002). Foundations of non-cooperative game theory. OUP Catalogue.
[86] Peleg, B., & Sudhölter, P. (2007). Introduction to the theory of cooperative games (Vol. 34). Springer Science & Business Media.
[87] Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. Econometrica: Journal of the Econometric Society, 97-109.
[88] Kamien, M. I., & Schwartz, N. L. (2012). Dynamic optimization: the calculus of variations and optimal control in economics and management. courier corporation.
[89] Long, N. V. (2011). Dynamic games in the economics of natural resources: a survey. Dynamic Games and Applications, 1, 115-148.
[90] Schelling, T. C. (1980). The Strategy of Conflict: with a new Preface by the Author. Harvard university press.
[91] Calcagno, R., Kamada, Y., Lovo, S., & Sugaya, T. (2014). Asynchronicity and coordination in common and opposing interest games. Theoretical Economics, 9(2), 409-434.
[92] Schelling, T. C. (1958). The strategy of conflict. Prospectus for a reorientation of game theory. Journal of Conflict Resolution, 2(3), 203-264.
[93] Lewis, D. (2008). Convention: A philosophical study. John Wiley & Sons.
[94] Nash Jr, J. F. (1950). The bargaining problem. Econometrica: Journal of the econometric society, 155-162.
[95] Myerson, R. B. (1999). Nash equilibrium and the history of economic theory. Journal of Economic Literature, 37(3), 1067-1082.
[96] Lehrer, E. (2003). Allocation processes in cooperative games. International Journal of Game Theory, 31, 341-351.
[97] Altman, E., El-Azouzi, R., Hayel, Y., & Tembine, H. (2009). The evolution of transport protocols: An evolutionary game perspective. Computer Networks, 53(10), 1751-1759.
[98] Hu, Z., Zheng, Z., Wang, T., Song, L., & Li, X. (2016). Game theoretic approaches for wireless proactive caching. IEEE Communications Magazine, 54(8), 37-43.
[99] Bauso, D. (2016). Game theory with engineering applications. Society for industrial and applied mathematics.
[100] Aghassi, M. L. (2005). Robust optimization, game theory, and variational inequalities (Doctoral dissertation, Massachusetts Institute of Technology).
[101] Wang, X., Ning, Z., & Guo, S. (2020). Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm. IEEE Transactions on Parallel and Distributed Systems, 32(2), 411-425.
[102] Liu, Y., Xu, C., Zhan, Y., Liu, Z., Guan, J., & Zhang, H. (2017). Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach. Computer Networks, 129, 399-409.
[103] Wang, K., Wang, X., & Liu, X. (2021). A high reliable computing offloading strategy using deep reinforcement learning for iovs in edge computing. Journal of Grid Computing, 19, 1-15.
[104] Jiang, W., & Lv, S. (2021). Hierarchical deployment of deep neural networks based on fog computing inferred acceleration model. Cluster Computing, 24(4), 2807-2817.
[105] Zhang, G., Shen, F., Liu, Z., Yang, Y., Wang, K., & Zhou, M. T. (2018). FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks. IEEE Internet of Things Journal, 6(3), 4388-4400.
[106] Keshavarznejad, M., Rezvani, M. H., & Adabi, S. (2021). Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms. Cluster Computing, 1-29.
[107] Sharma, M., Sharma, S., & Singh, G. (2020). Remote monitoring of physical and mental state of 2019-nCoV victims using social internet of things, fog and soft computing techniques. Comput Methods Programs Biomed, 105609-105609.
[108] Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. Future Generation Computer Systems, 104, 187-200.
[109] Yin, L., Luo, J., & Luo, H. (2018). Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing. IEEE Transactions on Industrial Informatics, 14(10), 4712-4721.

[110] De Maio, V., & Kimovski, D. (2020). Multi-objective scheduling of extreme data scientific workflows in fog. Future Generation Computer Systems, 106, 171-184.

[111] Elashri, S., & Azim, A. (2020). Energy-efficient offloading of real-time tasks using cloud computing. Cluster Computing, 23(4), 3273-3288.

[112] Dinesh Subramaniam, E. V., & Krishnasamy, V. (2021). Energy aware smartphone tasks offloading to the cloud using gray wolf optimization. Journal of Ambient Intelligence and Humanized Computing, 12, 3979-3987.

[113] Ning, Z., Dong, P., Wang, X., Hu, X., Liu, J., Guo, L., ... & Leung, V. C. (2020). Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks. IEEE Transactions on Mobile Computing, 21(4), 1319-1333.

[114] Liu, Z., Yang, X., Yang, Y., Wang, K., & Mao, G. (2018). DATS: Dispersive stable task scheduling in heterogeneous fog networks. IEEE Internet of Things Journal, 6(2), 3423-3436.

[115] Swain, C., Sahoo, M. N., Satpathy, A., Muhammad, K., Bakshi, S., Rodrigues, J. J., & de Albuquerque, V. H. C. (2020). METO: Matching-theory-based efficient task offloading in IoT-fog interconnection networks. IEEE Internet of Things Journal, 8(16), 12705-12715.

[116] Abualigah, L., Diabat, A., & Elaziz, M. A. (2021). Intelligent workflow scheduling for Big Data applications in IoT cloud computing environments. Cluster Computing, 24(4), 2957-2976.

[117] Shakhovska, N. (2017). Advances in intelligent systems and computing. Springer International Pu.

[118] Wang, J., Wu, W., Liao, Z., Sherratt, R. S., Kim, G. J., Alfarraj, O., ... & Tolba, A. (2020). A probability preferred priori offloading mechanism in mobile edge computing. IEEE Access, 8, 39758-39767.

[119] Huang, X., Yang, Y., & Wu, X. (2019, September). A meta-heuristic computation offloading strategy for IoT applications in an edge-cloud framework. In Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control (pp. 1-6).

[120] Adhikari, M., Srirama, S. N., & Amgoth, T. (2019). Application offloading strategy for hierarchical fog environment through swarm optimization. IEEE Internet of Things Journal, 7(5), 4317-4328.

[121] Ranadheera, S., Maghsudi, S., & Hossain, E. (2018). Computation offloading and activation of mobile edge computing servers: A minority game. IEEE Wireless Communications Letters, 7(5), 688-691.

[122] Hart, S. (1992). Games in extensive and strategic forms. Handbook of game theory with economic applications, 1, 19-40.

[123] Yao, J., & Ansari, N. (2019). Task allocation in fog-aided mobile IoT by Lyapunov online reinforcement learning. IEEE Transactions on Green Communications and Networking, 4(2), 556-565.

[124] Liao, Z., Peng, J., Xiong, B., & Huang, J. (2021). Adaptive offloading in mobile-edge computing for ultra-dense cellular networks based on genetic algorithm. Journal of Cloud Computing, 10(1), 1-16.

[125] He, X., Liu, J., Jin, R., & Dai, H. (2017, December). Privacy-aware offloading in mobile-edge computing. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-6). IEEE.

[126] Pourkabirian, A. and Haghighat, A.T., 2007, September. Energy-aware, delay-constrained routing in wireless sensor networks through genetic algorithm. In *2007 15th International Conference on Software, Telecommunications and Computer Networks* (pp. 1-5). IEEE.

[127] Liu, F., Huang, Z., & Wang, L. (2019). Energy-efficient collaborative task computation offloading in cloud-assisted edge computing for IoT sensors. Sensors, 19(5), 1105.

[128] Psomas, C., & Krikidis, I. (2020). Wireless powered mobile edge computing: Offloading or local computation?. IEEE Communications Letters, 24(11), 2642-2646.

[129] Noreen, S., & Saxena, N. (2020). A review on game-theoretic incentive mechanisms for mobile data offloading in heterogeneous networks. IETE Technical Review, 34(sup1), 15-26.

[130] Ranadheera, S., Maghsudi, S., & Hossain, E. (2017). Mobile edge computation offloading using game theory and reinforcement learning. arXiv preprint arXiv:1711.09012.

[131] Dong, C., & Wen, W. (2019). Joint optimization for task offloading in edge computing: An evolutionary game approach. Sensors, 19(3), 740.

[132] Chen, Y., Zhang, S., Xiao, M., Qian, Z., Wu, J., & Lu, S. (2020, December). Multi-user edge-assisted video analytics task offloading game based on deep reinforcement learning. In 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS) (pp. 266-273). IEEE.

[133] Mensah, R. N., Zhiyuan, L., Okine, A. A., & Adeke, J. M. (2021, May). A game-theoretic approach to computation offloading in software-defined D2D-enabled vehicular networks. In 2021 2nd Information Communication Technologies Conference (ICTC) (pp. 34-38). IEEE.

[134] Khoobkar, M.H., Fooladi, M.D.T., Rezvani, M.H. and Sadeghi, M.M.G., 2023. Joint optimization of delay and energy in partial offloading using dual-population replicator dynamics. Expert Systems with Applications, 216, p.119417.

[135] Lu, X., Yuan, S., Nian, Z., Mu, C. and Li, X., 2024. Optimizing Task Offloading for Power Line Inspection in Smart Grid Networks with Edge Computing: A Game Theory Approach. Information, 15(8), p.441.

[136] Asghari, A., Azgomi, H. and Darvishmofarahi, Z., 2023. Multi-objective edge server placement using the whale optimization algorithm and game theory. Soft Computing, 27(21), pp.16143-16157.

[137] Alli, A. A., & Alam, M. M. (2019). SecOFF-FCIoT: Machine learning based secure offloading in Fog-Cloud of things for smart city applications. Internet of Things, 7, 100070.

[138] Hsieh, C. Y., Ren, Y., & Chen, J. C. (2023). Edge-cloud offloading: Knapsack potential game in 5G multi-access edge computing. IEEE Transactions on Wireless Communications.

[139] Roostaei, R., Dabiri, Z., & Movahedi, Z. (2021). A game-theoretic joint optimal pricing and resource allocation for mobile edge computing in NOMA-based 5G networks and beyond. Computer Networks, 198, 108352.

[140] Chen, Y., Zhao, J., Wu, Y., Huang, J., & Shen, X. S. (2022). Qoe-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach. IEEE Transactions on Mobile Computing.

[141] Pham, X. Q., Huynh-The, T., Huh, E. N., & Kim, D. S. (2022). Partial computation offloading in parked vehicle-assisted multi-access edge computing: A game-theoretic approach. IEEE Transactions on Vehicular Technology, 71(9), 10220-10225.

[142] Lu, H., Xu, G., Sung, C. W., Mostafa, S., & Wu, Y. (2022, July). A Game Theoretical Balancing Approach for Offloaded Tasks in Edge Datacenters. In 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) (pp. 526-536). IEEE.

[143] Chen, R., Fan, Y., Jia, M., & Yuan, S. (2022). A Game-Theoretic Scheme for Parked Vehicle-Assisted MEC Computation Offloading. Scientific Programming, 2022.

[144] Wang, M., Zhang, L., Gao, P., Yang, X., Wang, K., & Yang, K. (2023). Stackelberg Game-Based Intelligent Offloading Incentive Mechanism for a Multi-UAV-Assisted Mobile Edge Computing System. IEEE Internet of Things Journal.

[145] Hossain, M. D., Sultana, T., Hossain, M. A., Layek, M. A., Hossain, M. I., Sone, P. P., ... & Huh, E. N. (2022). Dynamic Task Offloading for Cloud-Assisted Vehicular Edge Computing Networks: A Non-Cooperative Game Theoretic Approach. Sensors, 22(10), 3678.

[146] Liwang, M., Wang, J., Gao, Z., Du, X., & Guizani, M. (2019). Game theory based opportunistic computation offloading in cloud-enabled IoV. Ieee Access, 7, 32551-32561.

[147] Khawam, K., Khabbaz, M. and Saad, J., 2025. A Game Theoretical Priority-Aware R2V Task Offloading Framework For Vehicular Fog Networks. IEEE Open Journal of the Communications Society.

[148] Wu, H., Nie, J., Xiong, Z., Cai, Z., Zhou, T., Yuen, C. and Niyato, D., 2023. A game-based incentive-driven offloading framework for dispersed computing. IEEE Transactions on Communications, 71(7), pp.4034-4049.

[149] Chen, Y., Wu, J., Han, J., Zhao, H. and Deng, S., 2024. A game-theoretic approach based task offloading and resource pricing method for idle vehicle devices assisted VEC. IEEE Internet of Things Journal.

[150] Duan, Y. and Jiang, C., 2024. Binary task offloading strategy for cloud robots using improved game theory in cloud-edge collaboration. The Journal of Supercomputing, 80(10), pp.14752-14772.

[151] Wang, M., Zhang, L., Gao, P., Yang, X., Wang, K. and Yang, K., 2023. Stackelberg-game-based intelligent offloading incentive mechanism for a multi-UAV-assisted mobile-edge computing system. IEEE Internet of Things Journal, 10(17), pp.15679-15689.

[152] Chen, Z., Yang, Y., Xu, J., Chen, Y. and Huang, J., 2024. Task Offloading and Resource Pricing Based on Game Theory in UAV-Assisted Edge Computing. IEEE Transactions on Services Computing.

[153] Zhang, J., Wu, Y., Min, G. and Li, K., 2024. Neural network-based game theory for scalable offloading in vehicular edge computing: A transfer learning approach. IEEE Transactions on Intelligent Transportation Systems, 25(7), pp.7431-7444.

[154] Liu, Y., Wang, S., Huang, J., & Yang, F. (2018, May). A computation offloading algorithm based on game theory for vehicular edge networks. In 2018 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[155] Li, M., Wu, Q., Zhu, J., Zheng, R., & Zhang, M. (2018). A computing offloading game for mobile devices and edge cloud servers. Wireless Communications and Mobile Computing, 2018, 1-10.

[156] Kuang, Z., Shi, Y., Guo, S., Dan, J., & Xiao, B. (2019). Multi-user offloading game strategy in OFDMA mobile cloud computing system. IEEE Transactions on Vehicular Technology, 68(12), 12190-12201.

[157] Qin, A., Cai, C., Wang, Q., Ni, Y., & Zhu, H. (2019, March). Game theoretical multi-user computation offloading for mobile-edge cloud computing. In 2019 IEEE conference on multimedia information processing and retrieval (MIPR) (pp. 328-332). IEEE.

[158] Xu, F., Yang, W., & Li, H. (2020). Computation offloading algorithm for cloud robot based on improved game theory. Computers & Electrical Engineering, 87, 106764.

[159] Li, S., Hu, X., & Du, Y. (2021). Deep reinforcement learning and game theory for computation offloading in dynamic edge computing markets. IEEE Access, 9, 121456-121466.

[160] Ng, J. S., Lim, W. Y. B., Xiong, Z., Niyato, D., Leung, C., Kim, D. I., ... & Yang, Q. (2021). A game-theoretic approach towards collaborative coded computation offloading. arXiv preprint arXiv:2102.08667.

[161] Lakshmi, A. S., & Vijayalakshmi, M. (2021). An adaptive multi-cloud offloading using hierarchical game-theoretic approach. International Journal of Intelligent Networks, 2, 7-17.

[162] Xu, X., Tang, S., Qi, L., Zhou, X., Dai, F. and Dou, W., 2023. Cnn partitioning and offloading for vehicular edge networks in web3. IEEE Communications Magazine, 61(8), pp.36-42.

[163] Wu, L., Sun, P., Wang, Z., Li, Y. and Yang, Y., 2023. Computation offloading in multi-cell networks with collaborative edge-cloud computing: A game theoretic approach. IEEE Transactions on Mobile Computing, 23(3), pp.2093-2106.

[164] Xu, X., Liu, K., Dai, P., Jin, F., Ren, H., Zhan, C. and Guo, S., 2023. Joint task offloading and resource optimization in NOMA-based vehicular edge computing: A game-theoretic DRL approach. Journal of Systems Architecture, 134, p.102780.

[165] Ait Temghart, A., Marwan, M. and Baslam, M., 2025. Optimization of computational offloading in the mobile edge: a game theoretic approach. Cluster Computing, 28(3), pp.1-15.

[166] Li, F., Ge, E., Hu, W. and Xia, R., 2024. A two-level game theoretic approach for task offloading in mobile edge computing. Engineering Applications of Artificial Intelligence, 136, p.108819.

[167] Zhang, K., Yang, J. and Lin, Z., 2023. Computation Offloading and Resource Allocation Based on Game Theory in Symmetric MEC-Enabled Vehicular Networks. Symmetry, 15(6), p.1241.

[168] Ma, W., Liu, X., & Mashayekhy, L. (2019, July). A strategic game for task offloading among capacitated UAV-mounted cloudlets. In 2019 IEEE international congress on internet of things (ICIOT) (pp. 61-68). IEEE.

[169] Jiang, Q., Xu, X., He, Q., Zhang, X., Dai, F., Qi, L., & Dou, W. (2021, September). Game theory-based task offloading and resource allocation for vehicular networks in edge-cloud computing. In 2021 IEEE International Conference on Web Services (ICWS) (pp. 341-346). IEEE.

[170] Wu, B., Zeng, J., Ge, L., Su, X., & Tang, Y. (2019). Energy-latency aware offloading for hierarchical mobile edge computing. IEEE Access, 7, 121982-121997.
[171] Chen, R., Cui, L., Wang, M., Zhang, Y., Yao, K., Yang, Y., & Yao, C. (2021). Joint computation offloading, channel access and scheduling optimization in UAV swarms: A game-theoretic learning approach. IEEE Open Journal of the Computer Society, 2, 308-320.
[172] Ma, X., Lin, C., Xiang, X., & Chen, C. (2015, November). Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing. In Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 271-278).
[173] Mondal, S., Das, G., & Wong, E. (2018). Computation offloading in optical access cloudlet networks: A game-theoretic approach. IEEE Communications Letters, 22(8), 1564-1567.
[174] Wang, E., Dong, P., Xu, Y., Li, D., Wang, L., & Yang, Y. (2021, October). Distributed game-theoretical task offloading for mobile edge computing. In 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) (pp. 216-224). IEEE.
[175] Zhou, H., Wang, Z., Cheng, N., Zeng, D., & Fan, P. (2022). Stackelberg-Game-Based Computation Offloading Method in Cloud–Edge Computing Networks. IEEE Internet of Things Journal, 9(17), 16510-16520.
[176] Luo, J., Qian, Q., Yin, L., & Qiao, Y. (2020, December). A game-theoretical approach for task offloading in edge computing. In 2020 16th International Conference on Mobility, Sensing and Networking (MSN) (pp. 756-761). IEEE.
[177] Chen, S., Chen, X., Chen, Y., & Li, Z. (2019, August). Distributed computation offloading based on stochastic game in multi-server mobile edge computing networks. In 2019 IEEE International Conference on Smart Internet of Things (SmartIoT) (pp. 77-84). IEEE.
[178] Fang, T., Chen, J., & Zhang, Y. (2021). Content-aware multi-subtask offloading: A coalition formation game-theoretic approach. IEEE Communications Letters, 25(8), 2664-2668.
[179] Xu, J., Hu, Z., & Zou, J. (2020). Computing offloading and resource allocation algorithm based on game theory for IoT devices in mobile edge computing. International Journal of Innovative Computing, Information and Control, 16(6), 1895-1914.
[180] Li, K. (2018). A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing. IEEE Transactions on Sustainable Computing.
[181] Cui, Y., Zhang, D., Zhang, T., Chen, L., Piao, M., & Zhu, H. (2020). Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices. AEU-International Journal of Electronics and Communications, 118, 153134.
[182] Liu, J., Wang, X., Shen, S., Yue, G., Yu, S., & Li, M. (2021). A Bayesian Q-learning game for dependable task offloading against DDoS attacks in sensor edge cloud. IEEE Internet of Things Journal, 8(9), 7546-7561.
[183] Zhou, J., Tian, D., Sheng, Z., Duan, X., & Shen, X. (2021). Distributed task offloading optimization with queueing dynamics in multiagent mobile-edge computing networks. IEEE Internet of Things Journal, 8(15), 12311-12328.
[184] Yang, Y., Long, C., Wu, J., Peng, S., & Li, B. (2021). D2D-enabled mobile-edge computation offloading for multiuser IoT network. IEEE Internet of Things Journal, 8(16), 12490-12504.
[185] Jošilo, S., & Dán, G. (2017, May). A game theoretic analysis of selfish mobile computation offloading. In IEEE INFOCOM 2017-IEEE conference on computer communications (pp. 1-9). IEEE.
[186] Liu, W., Cao, B., Zhang, L., Peng, M., & Daneshmand, M. (2020, June). A distributed game theoretic approach for blockchain-based offloading strategy. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.
[187] Wang, Y., Lang, P., Tian, D., Zhou, J., Duan, X., Cao, Y., & Zhao, D. (2020). A game-based computation offloading method in vehicular multiaccess edge computing networks. IEEE Internet of Things Journal, 7(6), 4987-4996.
[188] Chen, Z., & Wang, X. (2020). Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach. EURASIP Journal on Wireless Communications and Networking, 2020(1), 1-21.
[189] Yang, B., Li, Z., & Liu, W. (2018). Non-cooperative game approach for task offloading in edge clouds. arXiv preprint arXiv:1812.07781.
[190] Zhou, S., & Jadoon, W. (2020). The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment. Computer Networks, 178, 107334.
[191] Zhu, A., Zeng, Z., Guo, S., Lu, H., Ma, M. and Zhou, Z., 2023. Game-theoretic robotic offloading via multi-agent learning for agricultural applications in heterogeneous networks. Computers and Electronics in Agriculture, 211, p.108017.
[192] Gao, L., Chen, X., Yin, B., Cui, L. and Xing, H., 2023, March. Game theory based task offloading, content caching and resource pricing under edge-cloud collaboration in 6g network. In 2023 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.
[193] You, L., Dawoud, H., Imran, M. and Popoola, O., 2024, October. JVECORA: A Joint Game Theory-Based Algorithm for Efficient Offloading and Load Balancing in Vehicular Networks. In 2024 IEEE 24th International Conference on Communication Technology (ICCT) (pp. 626-633). IEEE.
[194] Zamzam, M., Elshabrawy, T., & Ashour, M. (2021). A minimized latency collaborative computation offloading game under mobile edge computing for indoor localization. IEEE Access, 9, 133861-133874.
[195] Guo, H., Liu, J., & Lv, J. (2019). Toward intelligent task offloading at the edge. IEEE Network, 34(2), 128-134.
[196] Wang, H., Lin, Z., Guo, K., & Lv, T. (2021, June). Computation offloading based on game theory in MEC-assisted V2X networks. In 2021 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.
[197] Xia, C., Jin, Z., Su, J., & Li, B. (2023). Mobility-Aware Offloading and Resource Allocation Strategies in MEC Network Based on Game Theory. Wireless Communications and Mobile Computing, 2023.
[198] Li, S., Li, W., Liu, H. and Sun, W., 2024. A two-stage service-oriented task offloading framework with edge-cloud collaboration: a game theory approach. Journal of Systems Science and Systems Engineering, 33(5), pp.521-551.

[199] Yuan, Y. and Su, W., 2023, August. A game theory-based strategy for allocating and offloading computing resources in 5G networks. In 2023 International Conference on Networking and Network Applications (NaNA) (pp. 92-97). IEEE.

[200] Wang, L., Zhou, W., Xu, H., Li, L., Cai, L. and Zhou, X., 2023. Research on task offloading optimization strategies for vehicular networks based on game theory and deep reinforcement learning. Frontiers in Physics, 11, p.1292702.

[201] Cao, D., Liang, P., Wu, T., Zhang, S. and Zhou, L., 2024. Enhancing trust and security in IoT computing offloading through game theory and blockchain-based control strategy. Concurrency and Computation: Practice and Experience, 36(15), p.e7956.

[202] Fang, T., Wu, D., Chen, J., & Liu, D. (2022). Cooperative task offloading and content delivery for heterogeneous demands: A matching game-theoretic approach. IEEE Transactions on Cognitive Communications and Networking, 8(2), 1092-1103.

[203] Kim, S. (2022). Bargaining Game Based Offloading Service Algorithm for Edge-Assisted Distributed Computing Model. IEEE Access, 10, 63648-63657.

[204] Wang, S., Hu, Z., Deng, Y., & Hu, L. (2022). Game-Theory-Based Task Offloading and Resource Scheduling in Cloud-Edge Collaborative Systems. Applied Sciences, 12(12), 6154.

[205] Xu, F., Xie, Y., Sun, Y., Qin, Z., Li, G., & Zhang, Z. (2022). Two-stage computing offloading algorithm in cloud-edge collaborative scenarios based on game theory. Computers & Electrical Engineering, 97, 107624.

[206] Zhang, T. (2017). Data offloading in mobile edge computing: A coalition and pricing based approach. IEEE Access, 6, 2760-2767.

[207] Li, Q., Zhao, J., & Gong, Y. (2019, May). Cooperative computation offloading and resource allocation for mobile edge computing. In 2019 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.

[208] Zhan, Y., Guo, S., Li, P., & Zhang, J. (2020). A deep reinforcement learning based offloading game in edge computing. IEEE Transactions on Computers, 69(6), 883-893.

[209] Wang, Z., Wu, T., Zhang, Z., & Zhou, H. (2021, July). A game theory-based computation offloading method in cloud-edge computing networks. In 2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-6). IEEE.

[210] Wang, M., Yang, K., Zhao, Y., Wang, Y., Guo, J., & Ma, X. (2021, December). Multi-Server Offloading Based on Game Theory in Cloud-Edge Collaborative Computing. In Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications (pp. 128-133).

[211] Huang, Y. Y., & Wang, P. C. (2023). Computation Offloading and User-Clustering Game in Multi-Channel Cellular Networks for Mobile Edge Computing. Sensors, 23(3), 1155.

[212] Xu, X., Jiang, Q., Zhang, P., Cao, X., Khosravi, M. R., Alex, L. T., ... & Dou, W. (2022). Game theory for distributed iov task offloading with fuzzy neural network in edge computing. IEEE Transactions on Fuzzy Systems, 30(11), 4593-4604.

[213] Hu, H. C., & Wang, P. C. (2022). Computation Offloading Game for Multi-Channel Wireless Sensor Networks. Sensors, 22(22), 8718.

[214] Zhang, Z., Wu, L., & Zeng, F. (2022, November). Optimal Task Offloading Strategy in Vehicular Edge Computing Based on Game Theory. In Wireless Algorithms, Systems, and Applications: 17th International Conference, WASA 2022, Dalian, China, November 24–26, 2022, Proceedings, Part III (pp. 554-562). Cham: Springer Nature Switzerland.

[215] Mebrek, A., & Yassine, A. (2021). Intelligent resource allocation and task offloading model for IoT applications in fog networks: a game-theoretic approach. IEEE Transactions on Emerging Topics in Computational Intelligence.

[216] Mukherjee, A., Ghosh, S., De, D., & Ghosh, S. K. (2022). Mcg: mobility-aware computation offloading in edge using weighted majority game. IEEE Transactions on Network Science and Engineering, 9(6), 4310-4321.

[217] Guglielmi, A. V., Levorato, M., & Badia, L. (2018, May). A Bayesian game theoretic approach to task offloading in edge and cloud computing. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.

[218] Pham, Q. V., Nguyen, H. T., Han, Z., & Hwang, W. J. (2019). Coalitional games for computation offloading in NOMA-enabled multi-access edge computing. IEEE Transactions on Vehicular Technology, 69(2), 1982-1993.

[219] Asheralieva, A., & Niyato, D. (2019). Hierarchical game-theoretic and reinforcement learning framework for computational offloading in UAV-enabled mobile edge computing networks with multiple service providers. IEEE Internet of Things Journal, 6(5), 8753-8769.

[220] Teymoori, P., & Boukerche, A. (2022, May). Dynamic Multi-user Computation Offloading for Mobile Edge Computing using Game Theory and Deep Reinforcement Learning. In ICC 2022-IEEE International Conference on Communications (pp. 1930-1935). IEEE.

[221] Ding, Y., Li, K., Liu, C., & Li, K. (2021). A potential game theoretic approach to computation offloading strategy optimization in end-edge-cloud computing. IEEE Transactions on Parallel and Distributed Systems, 33(6), 1503-1519.

[222] Nowak, D., Mahn, T., Al-Shatri, H., Schwartz, A., & Klein, A. (2018, March). A generalized nash game for mobile edge computation offloading. In 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) (pp. 95-102). IEEE.

[223] Xiao, Z., Dai, X., Jiang, H., Wang, D., Chen, H., Yang, L., & Zeng, F. (2019). Vehicular task offloading via heat-aware MEC cooperation using game-theoretic method. IEEE Internet of Things Journal, 7(3), 2038-2052.

[224] Hui, Y., Su, Z., Luan, T. H., Li, C., Mao, G., & Wu, W. (2020). A game theoretic scheme for collaborative vehicular task offloading in 5G HetNets. IEEE Transactions on Vehicular Technology, 69(12), 16044-16056.

[225] Luo, Z., & Huang, A. (2021, November). Joint Game Theory and Greedy Optimization Scheme of Computation Offloading for UAV-Aided Network. In 2021 31st International Telecommunication Networks and Applications Conference (ITNAC) (pp. 198-203). IEEE.

[226] He, C., Jiang, W., Wang, X., Wang, W., Wang, J. and Xie, X., 2025. Deep Reinforcement Learning and Nash Equilibrium Game based task offloading optimization strategy for the IoV. IEEE Sensors Journal.

[227] Shakkeera, L., 2024, December. Securing Mobile Cloud Data Offloading in Blockchain Using Deep Deterministic Policy Gradient with Game Theory. In 2024 Fourth International Conference on Multimedia Processing, Communication & Information Technology (MPCIT) (pp. 115-122). IEEE.

[228] Gu, L., Cui, M., Xu, L. and Xu, X., 2022. Collaborative offloading method for digital twin empowered cloud edge computing on Internet of Vehicles. Tsinghua Science and Technology, 28(3), pp.433-451.

[229] Sharma, M., Tomar, A., & Hazra, A. (2024). Edge computing for industry 5.0: Fundamental, applications and research challenges. *IEEE Internet of Things Journal*.

[230] Cheng, C., Zhai, L., Zhu, X., Jia, Y. and Li, Y., 2024. Dynamic task offloading and service caching based on game theory in vehicular edge computing networks. Computer Communications, 224, pp.29-41.

[231] Sun, Z., Sun, G., Liu, Y., Wang, J. and Cao, D., 2023. BARGAIN-MATCH: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks. IEEE Transactions on Mobile Computing, 23(2), pp.1655-1673.

[232] Liu, L., Mao, W., Li, W., Duan, J., Liu, G. and Guo, B., 2024. Edge computing offloading strategy for space-air-ground integrated network based on game theory. Computer Networks, 243, p.110331.

[233] Xia, C., Jin, Z., Su, J. and Li, B., 2023. Mobility-Aware Offloading and Resource Allocation Strategies in MEC Network Based on Game Theory. Wireless Communications and Mobile Computing, 2023(1), p.5216943.

[234] Zhang, J., Zhang, J., Shen, F., Yan, F. and Bu, Z., 2024. DOGS: Dynamic Task Offloading in Space-Air-Ground Integrated Networks With Game-Theoretic Stochastic Learning. IEEE Internet of Things Journal.

[235] Keshta, I., Soni, M., Deb, N., Saravanan, K. and Khan, I.R., 2024. Game theory-based optimization for efficient IoT task offloading in 6G network base stations. Measurement: Sensors, 33, p.101184.

[236] Sharma, M., Tomar, A., & Hazra, A. (2024). From Connectivity to Intelligence: The Game-Changing Role of AI and IoT in Industry 5.0. *IEEE Consumer Electronics Magazine*.

[237] Sharma, M., Tomar, A., Hazra, A., Akhter, Z., Dhangar, D., & Singh, R. K. (2024, July). Edge Device Selection For Industrial IoT Task Offloading In Mobile Edge Computing. In *2024 IEEE International Conference on Omni-layer Intelligent Systems (COINS)* (pp. 1-4). IEEE.

[238] Sharma, M., Tomar, A., & Hazra, A. (2024, July). Intelligent Task Offloading in IoT Devices Using Deep Reinforcement Learning Approaches. In *2024 IEEE International Conference on Omni-layer Intelligent Systems (COINS)* (pp. 1-4). IEEE.

A APPENDICES

These appendices contain some tables and sub-sections of the original article that were moved here due to space limitations.

A.1 Tables and Figures

This appendix contains some large tables that we moved here from the main article.

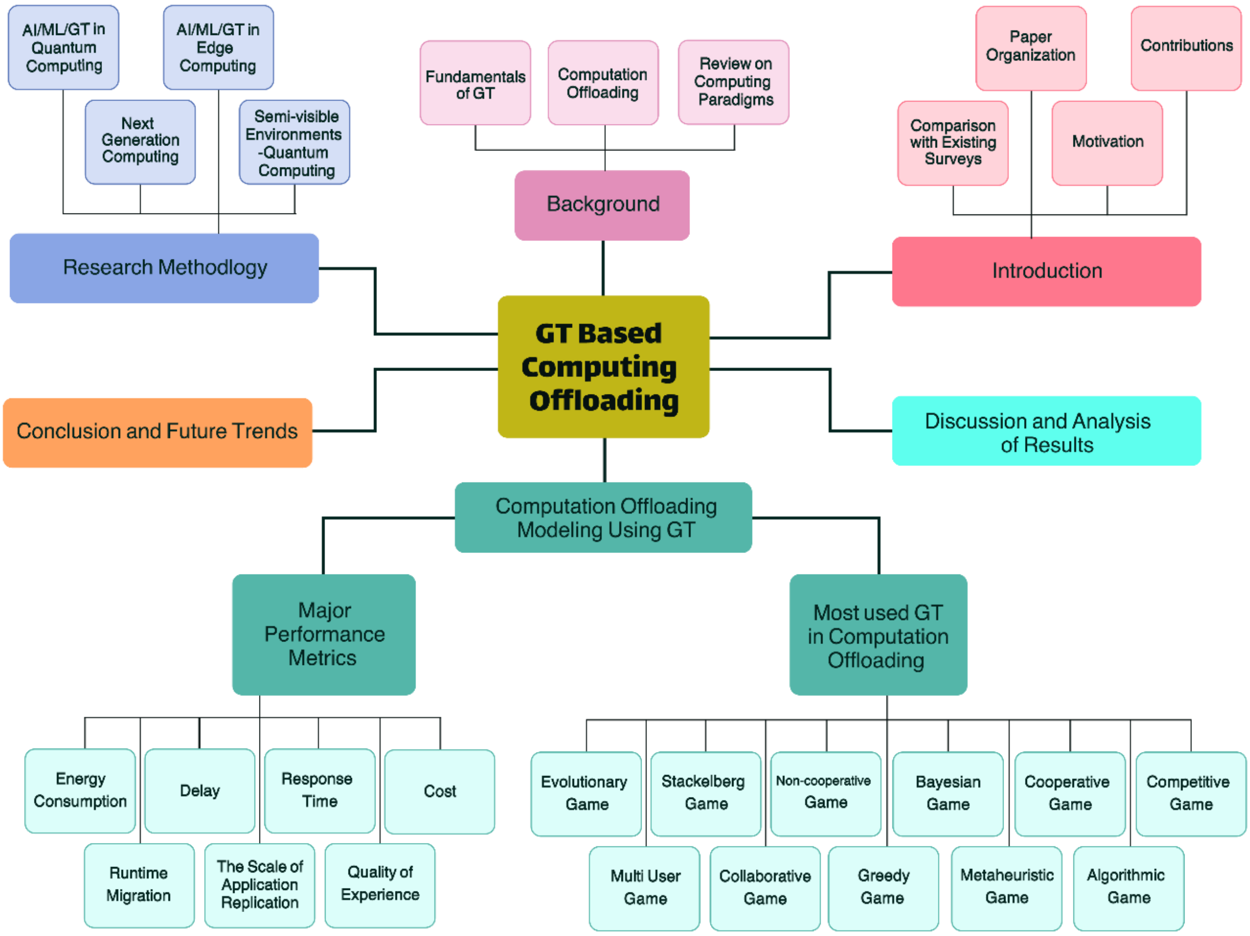


Figure A-1: The organization of this article

Table A-1: List of abbreviations

Acronym	Description	Acronym	Description	Acronym	Description
AI	Artificial Intelligence	ML	Machine Learning	DRL	Deep Reinforcement Learning
QoS	Quality of Service	PIoT	Power Internet of Things	TC	terrestrial cloud
QoE	Quality of Experience	MINLP	Mixed Integer Non-linear Program	RRLO	Realtime Reinforcement Learning-based Offloading
IoT	Internet of Things	MU	mobile user	eNB	evolved Node B
SARSA	State-action-reward-state-action	KNN	K-Nearest Neighbor	UE	user equipment
MDP	Markov Decision Process	CC	Cloud computing	MES	mobile edge servers
SA	Single Agent	MA	Multi-agent	S2S	sequence-to-sequence
DQN	Deep Q-network	UAV	unmanned aerial vehicles	FC	Fog Computing
EC	Edge Computing	MEN	mobile edge networks	IoMT	Internet of Medical Things

MEC	Mobile Edge Computing	LSTM	long short-term memory network	MSE	mean squared error
IoV	Internet of Vehicle	MTC	machine type communication	SWA	stochastic weight averaging
SDN	Software Defined Network	DFG	Dissolution and Formation of Groups	PER	prioritized experience replay
NFV	Network Function Virtualization	DAG	directed acyclic graph	OMA	orthogonal multiple access
C-RAN	cloud radio access networks	MRL	meta reinforcement learning	VNF	Virtual Network Functions
F-AP	Fog-Access Point	EH	energy harvesting	OU	Ornstein-Uhlenbeck
VFC	Vehicular Fog Computing	GCN	graph convolutional networks	MCC	Mobile cloud computing
D3PG	Double-Dueling-Deterministic Policy Gradients	OEC	opportunistic edge computing	NE	Nash Equilibrium
DNC	Differential Neural Computer	CSI	Channel State Information	CAV	Connected and Automated Vehicle
NOMA	Non-orthogonal multiple access	FAP	Fog Access Points	EIoT	Edge-Internet-of-Things
POMDP	Partially observable Markov Decision Process	INLP	integer nonlinear programming	MTCD	Machine-type communication devices
MFC	Mobile Fog Computing	MARL	Multi-agent Reinforcement Learning	MBS	macro base station
FDTMDP	finite-horizon Discrete-time Markov decision process	MADDPG	Multi-agent Deep Deterministic Policy Gradient	A2C	advantage actor-critic
F-RAN	Fog Radio Access Networks	LTE- V	Long-term evolution-vehicle	A3C	asynchronous advantage Actor-critic
DDPG	Deep Deterministic Policy Gradient	DNN	Deep Neural Network	CSI	channel state information
RSU	road-side units	VT	vehicle terminals	DVFS	Dynamic Voltage and Frequency Scaling
VEC	vehicular edge computing	D2D	device-to-device	MIP	mixed-integer programming
PPO	proximal policy optimization	CNN	convolutional neural network	CMDP	constrained Markov decision process
SBS	small cell base stations	V2I	Vehicle-to-Infrastructure	V2V	Vehicle- to-Vehicle

Table A-2: Features extracted from references

Attribute
Date of publication
Bibliographic data
Type of article: conference or journal
DOI
Objectives
Game theory methods that are used
Unloading type: partial vs. full unloading
Metrics used for evaluation
Single-Agent Game Theory (SAGT) /Multi-Agent Game Theory (MAGT)
The method used in simulation or implementation
The computing paradigm that this research is related to
Special considerations applied (user mobility, task interdependence, etc.)

Table A-3: A complete list of performance metrics in different computing paradigms

Paradigm/ MDP Elements	Cloud Computing (CC)	Fog Computing (FC)	Edge Computing (EC) and Mobile Edge Computing (MEC)	Vehicular Environment
Stable equilibrium	- number of VMs that run in PMs - waiting time - dwell zone - number of nodes - queue length - number of nodes connected to AP - input data size - CPU availability - network latency - channel state is different in each time slot - the combination of caching & request status	- Workload state of each fog node - queued tasks - set of arrived tasks - Task attributes - number of tasks in the buffer in progress - priority of the computational task - distributed computing resources - channel state, - power resources - networking bandwidth capability - processing & storage of fog nodes - channel gain - available & needed resources - remaining computing resources - maximum tolerable delay of the task - number of idle computing resources of the fog server	- Task profiles (task size, the security level of the task, task arrival) - CPU cycles, input data size - task queue length - battery power of the target mobile device - processing capacity, memory - network bandwidth - channel gain - channel conditions - available computing resources - downlink & uplink capacity - wireless transmission rate - number of connected user devices - computational capacity of the server - status of each server - network channel distribution - power gain state - available communication mode - power resources - computation capability - SNR - offloading ratio - battery level - state of CPU - transmission rate - available spectrum, cache & computing resources - the quality of the signal	- location/ position of a vehicle -the distance between vehicles and the RSU/ server node - moving speed of the vehicle - remaining resources of vehicles and server - wireless environment - task information (task size, task profile, deadline, task priority, …) - Vehicle's task queue - arrival rate of moving vehicles and offloading task flows - state of CPU - SINR/ SNR - channel gain - bandwidth - computing power of the server - channel allocation state - maximum tolerable delay - number of parked vehicles - system overhead
Convergence rate	- offloading decision - service status - set of PMs in the cloud	- offloading decision (executed locally, offloaded to fog server adjacent fog server, or remote cloud) - power allocation decision	- offloading decision (executed locally on the mobile device, offload to RSU/ edge server or adjacent edge server, or remote cloud) - resource allocation decision - server selection decision - handover policies - bandwidth allocation decision - channel allocation decision - power allocation decision - caching decisions - allocated energy units - technology decision - accuracy decision	- offloading decision (executed locally on the vehicle, offload to edge/ fog servers (V2I), offload to other vehicles (V2V, V2V2I)) - resource allocation decision - power allocation decision - channel selection decision - server selection decision - bandwidth allocation - V2V is required - pricing strategy
Latency, Energy Consumption, Cost	minimizing - energy - delay/ latency - cost - the offloading failure rate - time of execution maximizing	minimizing - energy - delay/ latency maximizing - utility	minimizing - energy - delay/ latency - cost - network usage - bandwidth cost - privacy & security cost - the task failure rate - computational overhead	minimizing - energy - delay/ latency - cost - the offloading failure rate - response time - overhead Maximizing

	- utility		- error probability	- utility - task processing rate - QoE

Table A-4: Abbreviations for game theory algorithms

Co-evolutionary Games	CG
Algorithmic Games	AG
Stackelberg Games	SG
Greedy Games	GG
Non-cooperative Games	NCG
Bayesian Games	BG
Multi-user Games	MUG

Table A-5: A summary of applying evolutionary, greedy, Stackelberg, and metaheuristic algorithms to computation offloading

Ref.	**Method (RQ1)**	**Mode (RQ2)**	**Metrics (RQ3)**	**MDP (RQ4)**	**Eval. Tools (RQ5)**	**Paradigm (RQ6)**	**Characteristics (RQ7)**
[129]	CG	full offloading	• latency • cost • stable equilibrium • convergence rate • availability • reliability • fault-tolerance	multi-user	MATLAB	MEC	• task processing time • resource procurement cost • VM configuration assignment
[131]	CG	full offloading	• energy • consumption • latency • stable equilibrium • convergence rate • availability • reliability • fault-tolerance	multi-user	Algorithmic	MEC	the evolutionary game algorithm based on reinforcement learning, and analyze the convergence of the EGT-QL algorithm
[130]	CG	full offloading	• latency • cost • stable equilibrium • convergence rate	multi-user	OMNET++ Simulator	MEC	• QoS • cost
[132]	CG	full offloading	• latency • cost • convergence rate • availability	multi-user	MATLAB	MEC	a distributed evolutionary game algorithm based on Q-learning • convergence of evolutionary game-theoretic on Q learning (EGT-QL
[133]	CG	full offloading	• latency • cost • convergence rate • availability • reliability • fault-tolerance	multi-user	OpenStack	Cloud/MEC	• system cost of different computation offloading algorithms • convergence trend of game theory algorithm

							• relationship between the number of offloading cloud robots and the distance
[190]	GG	full offloading	• latency • cost • convergence rate • stable equilibrium	multi-user	MATLAB	MEC	the problem was defined to minimize the overall computation overhead of all tasks while satisfying the wireless channel constraint
[30]	MG	full offloading	• cost • convergence rate	multi-user	iFogSim simulator	Fog	• response time • energy consumption • VM placement
[31]	MG	full offloading	• reliability • latency • cost	multi-user	CloudSim simulation	Fog	• namely response time • reliability • financial cost
[32]	MG	full offloading	• cost • convergence rate • latency • stable equilibrium	multi-user	iFogSim simulator	Fog	• energy consumption • latency
[34]	MG	full offloading	• fault-tolerance • availability • latency • cost	multi-user	MATLAB	Edge	a) where to compute the localization task, either locally on the end device or the edge server or the cloud server, b) which localization technique should be used, and finally, c) which transmission technology is recommended to be chosen in combination with the localization technique
[36]	MG	full offloading	• availability • latency • cost	multi-user	MATLAB	MEC	Online Offloading, Scheduling, and Resource Allocation (OOSRA) Algorithm
[137]	SG	full offloading	• latency • cost • convergence rate • stable equilibrium	multi-user	CloudSim simulation	Edge	• formulate the interactions among cloud service operators and edge server
[121]	SG	full offloading	• latency	multi-user	Simulate	Edge	latency
[7]	SG	full offloading	• convergence rate • latency • availability	multi-user	MATLAB	MEC /Edge Cloud	a game for the computing offloading between an MD and ECSs
[143]	SG	full offloading	• convergence rate • latency • cost • stable equilibrium	multi-user	MATLAB	Cloud	1) how to determine the appropriate offloading rate of requestors; 2) how to select the most appropriate computation service provider; 3) how to identify the

							ideal pricing strategy for each service provider
[146]	SG	full offloading	• latency • cost • availability	multi-user	MATLAB	MEC	1) blockchain users, 2) blockchain miners. Wireless devices that need to upload transactions to the blockchain are blockchain users. Both normal edge servers and edge servers with wireless access function in this system are blockchain miners
[139]	SG	full offloading	• cost • convergence rate • latency • stable equilibrium • availability • fault-tolerance	multi-user	MATLAB	MEC	The authors have proposed a hybrid CTMC and queueing model for a mobile cloud offloading system under the specific threat of timing attacks.
[140]	SG	full offloading	• latency • stable equilibrium • cost	multi-user	Simulate	Edge	game-theoretic
[147]	SG	full offloading	• latency	multi-user	Simulate	Fog	game-theoretic
[148]	SG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge	- game-theoretic - Nash equilibrium
[149]	SG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge/Cloud	- game-theoretic - energy consumption -latency
[150]	SG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge/Cloud	- time - energy consumption
[163]	SG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge/Cloud	- cost in terms of energy cost - delay
[135]	SG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge	Nash equilibrium
[235]	SG	full offloading	• cost	multi-user	Simulate	MEC	social consequences, latency limits, and energy consumption
[151]	SG	full offloading	• energy consumption • cost	multi-user	Simulate	MEC	Nash equilibrium through game theory
[152]	SG	full offloading	• energy consumption • cost	multi-user	Simulate	MEC	game-theoretic
[153]	SG	full offloading	• latency	multi-user	Simulate	Edge	The efficiency of game theoretical offloading in VEC

Table A-6: A summary of Bayesian, non-collaborative, and multi-user approaches for computation offloading

Ref.	Method (RQ1)	Mode (RQ2)	Metrics (RQ3)	MDP (RQ4)	Eva. Tools (RQ5)	Paradigm (RQ6)	Characteristics (RQ7)
[168]	BG	full offloading	• cost • latency • stable equilibrium • convergence rate	Multi-user	NA	Edge/Cloud	• Delay • Cost
[190]	BG	full offloading	• reliability • latency • cost • availability	Multi-user	MATLAB	Edge/Cloud	• Greedy Q-Learning Algorithm • resource procurement cost • VM configuration assignment
[169]	BG	full offloading	• latency • stable equilibrium • energy consumption	Multi-user	Algorithm based	MEC	Game-theoretic
[154]	NCG	full offloading	• reliability • latency • cost • availability • stable equilibrium • energy consumption • fault-tolerance	Multi-user	NA	MEC	• M/G/1 queueing model to characterize multiple heterogeneous UEs and MECs • Response time
[155]	NCG	full offloading	• latency	Multi-user	NA	MEC	Evolutionary game theory
[156]	NCG	full offloading	• latency • stable equilibrium • reliability	Multi-user	NA	MEC	Evolutionary game theory
[157]	NCG	full offloading	• reliability • latency • cost • availability • stable equilibrium • energy consumption • fault-tolerance	Multi-user	MATLAB	Edge	Game-based Task Offloading Algorithm(GBTOA
[160]	NCG	full offloading	• reliability • latency • cost • availability • stable equilibrium • energy consumption • fault-tolerance	Multi-user	MATLAB	MEC	• formulation of computation offloading scheme • formulating the computation offloading problem as a noncooperative game proposing a PV-assisted MEC computation offloading scheme that effectively reduces the burden on the MEC server

[158]	NCG	full offloading	• latency • cost • availability • stable equilibrium • energy consumption • fault-tolerance • convergence rate	Multi-user	MATLAB	Cloud	• An interaction among three entities, such as user equipment, root server, and application vendor, is studied for adaptive offloading • Multi-cloud resource allocation
[161]	NCG	full offloading	• latency	Multi-user	MATLAB	MEC	• Evaluation of Cooperative Task Offloading • formulate the MEC grouping as a utility maximization
[173]	NCG	full offloading	• latency • cost • stable equilibrium • energy consumption	Multi-user	MATLAB	Cloud	content-aware task offloading (CATO)
[31]	NCG	full offloading	• cost • stable equilibrium	Multi-user	MATLAB	Edge	Game-theoretic
[180]	MUG	full offloading	• latency • cost • delay	Multi-user	NA	MEC	NA
[177]	MUG	full offloading	• latency • cost • availability • stable equilibrium • energy consumption • fault-tolerance • fault-tolerance	Multi-user	NA	MEC	• Latency • Energy consumption
[185]	MUG	full offloading	• energy • consumption	Multi-user	NA	MCC	NA
[131]	MUG	full offloading	• latency • stable equilibrium	Multi-user	NA	MCC	QoE
[179]	MUG	full offloading	• latency • cost • availability • stable equilibrium • energy • consumption • fault-tolerance	Multi-user	NA	MEC	MEC-Cloud architecture
[33]	MUG	full offloading	• latency • cost • energy consumption • fault-tolerance • availability • reliability	Multi-user	NS-3 Simulator	MEC	1)Convergence of the game 2)Effect of interference 3)Effect of the number of users 4) Effect of task size
[187]	MUG	full offloading	• latency • convergence rate • fault-tolerance • availability • reliability	Multi-user	NS-3 Simulator	MEC	• Reinforcement Learning and Game Theory Approach • Deep Q Network Approach Q-Learning Approach

[3]	MUG	full offloading	• latency • cost	Multi-user	Algorithm based	Edge	video analytics task offloading (MEVAO)
[181]	MUG	full offloading	• latency • cost • energy • consumption • fault-tolerance • availability • reliability • convergence rate • stable equilibrium	Multi-user	NS-3 Simulator	MEC	• The computation overhead model is built based on game theory. • results show that the system computation overhead by the proposed partial computation offloading algorithm is less than that by the MDCBAU algorithm or ECBAU algorithm
[182]	MUG	full offloading	• latency • cost • energy • consumption • fault-tolerance • availability • reliability • convergence rate	Multi-user	MATLAB	MEC	• Impact of Computation Complexity of Video Computing Tasks • Impact of Number of Users • Impact of Data Size of Video • Computing Tasks
[186]	MUG	full offloading	• latency • availability	Multi-user	MATLAB	MEC	analysis of utilities at NASH equilibrium
[178]	MUG	full offloading	• latency • cost • energy consumption • convergence rate • stable equilibrium	Multi-user	MATLAB	MEC	Nash equilibrium
[184]	MUG	full offloading	• cost	Multi-user	MATLAB	MEC	NA
[183]	MUG	full offloading	• cost • energy consumption	Multi-user	MATLAB	MEC	satisfy the QoS of users and respect their behaviors
[231]	NCG	full offloading	• energy consumption • cost	multi-user	Simulate	Edge	optimizing the strategies of resource allocation, resource pricing, and task offloading
[162]	NCG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge	game theory
[163]	NCG	full offloading	• latency • energy consumption	multi-user	Simulate	Edge/Cloud	minimize the total cost in terms of energy cost and delay
[232]	NCG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	Edge	minimizing system consumption cost consisting of delay and energy consumption as a non-cooperative game
[164]	NCG	full offloading	• energy consumption • cost	multi-user	Simulate	Edge	to solve the remaining resource allocation problem
[165]	NCG	full offloading	• latency	multi-user	Simulate	Edge	game theory

[166]	NCG	full offloading	• latency • energy consumption	multi-user	Simulate	MEC	the allocation of computing resources
[233]	NCG	full offloading	• latency • energy consumption • cost	multi-user	Simulate	MEC	optimizing offloading decisions and resource allocation schemes
[167]	NCG	full offloading	• latency • energy consumption	multi-user	Simulate	MEC	game theory
[227]	NCG	full offloading	• latency	multi-user	Simulate	MEC	game-theoretic