IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 15 January 2023, accepted 17 February 2023, date of publication 6 March 2023, date of current version 3 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252559

RESEARCH ARTICLE

Collaborative Coded Distributed Computing Scheme: A Two-Phase Cooperative Game Approach

SUNGWOOK KIM
Department of Computer Science, Sogang University, Mapo-gu, Seoul 04107, South Korea
e-mail: swkim01@sogang.ac.kr

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2022-2018-0-01799; and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2021R1F1A1045472.

ABSTRACT As a modern communication paradigm, Artificial intelligence based Internet of Things (AIoT) can provide an interactive platform across the globe to enrich the quality of networking services. With the AIoT paradigm, coded distributed computing (CDC) has recently emerged to be a promising solution to address the straggling effects in conventional distributed computing systems. In this article, we propose a novel CDC control scheme in the AIoT platform. Based on the cooperative game theory, the main challenges of our scheme are i) the k value decision for the CDC process, and ii) edge node resource allocation for offloading tasks. Using the ideas of coalition game and *weighted Nash social welfare solution* (*WNSWS*), our proposed scheme is developed as a two-phase game model to achieve a mutually desirable solution. At the first phase, a dynamic coalition formation is proceeded to select the most adaptable edge nodes for the offloading subtasks. At the second phase, the *WNSWS* is adopted to effectively share each edge node's computing resource. Based on the jointly design of these two cooperative games, we explore the synergy effect to optimize the CDC process. In the edge assisted distributed computing infrastructure, our reciprocal combinative approach can provide a fair-efficient solution through the sequential interactions of edge nodes and AIoT devices. In the performance evaluation, we provide extensive simulation analyses to show our scheme's superiority by comparing with the existing baseline protocols.

INDEX TERMS Artificial Internet of Things, coded distributed computing, dynamic coalition game, weighted nash social welfare solution, cooperative game theory.

I. INTRODUCTION

Functioning of the Internet is persistently transforming from the Internet of computers (IoC) to the Internet of things (IoT). The IoT paradigm has created a ubiquitously connected world while generating a variety of heterogeneous data in a myriad of fields and applications. It is expected that the total installed IoT-based devices will be projected to amount to approximately 41.6 billion, and nearly 79.4 Zettabytes (ZBs) of data may be generated and consumed in 2025. Major goal of IoT technology is to make the world smart. To satisfy this goal, we will need artificial intelligence (AI). In recent years, there is an increasing demand for the convergence of AI and IoT to tackle high-performance data processing issues in IoT-driven engineering applications. The collective integration of AI and the IoT has greatly promoted the rapid development of AI-of-Things (AIoT) systems that evolve the existing IoT standards to form autonomous future communication architectures to support the intelligent exchange of data between millions of devices [1], [2].

With limited computing supplies, AIoT devices can hardly complete their computation-intensive tasks. One of the major

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

AIoT tasks is matrix multiplication for the machine learning process. Therefore, computation offloading may be a solution for this problem. Traditionally, cloud computing plays a crucial role in the IoT paradigm where the vast resources available in the cloud can provide ubiquitous on demand computing and storage capabilities to support IoT devices. For the further computing process, the cloud-centric AIoT platform requires massive data transmissions from AIoT devices to the cloud center. Although the cloud center has unlimited computational capacity, some serious issues, such as the great pressure on network bandwidth, the inherent latency constraints of network communications, go on plaguing cloud based services. The new AIoT trend pushes the frontier of AI from the centralized cloud to the mobile edge nodes, paving the last mile delivery of AI capabilities. Nowadays, edge intelligence has emerged as a promising and enabling paradigm toward materializing the vision of AIoT. Specifically, the edge computing method brings computational resources closer to the data source with a relatively light access burden and a low transmission delay. It is extremely suitable for the AIoT platform, and has attracted a great deal of interest from industry and academia [2], [3].

With the ability to utilize the AIoT devices, distributed computing has recently become a highly-effective approach for large-scale computations in wireless edge networks. Compared to centralized computing methods, distributed computing is more fault-tolerant and scalable. With these outstanding advantages, distributed computing has been widely applied in the edge-based AIoT platform. In particular, an intensive computation task can be partitioned into multiple subtasks, and then they can be transmitted to several edge nodes for the parallel executing. Although distributed edge computing has many advantages for the AIoT paradigm, it has been facing a technical challenge. The performance of distributed computing is greatly affected by the unpredictable computing latency. In distributed computing executions, the total task processing time is determined by the slowest computing edge node, i.e, the straggler. The reason is that a task requires the calculated results from all edge nodes. This phenomenon partially weakens the advantages of distributed computing [4], [5], [6], [7].

To address the straggling problem, coded distributed computing (CDC) has recently emerged to be a highly-effective solution. The main principle of CDC is to inject computation redundancy to edge nodes via advanced coding theoretic techniques. Such redundancy can compensate for the uncertain computation time while improving the distributed computing stability and latency. Unlike the traditional distributed computing method, the CDC technique does not require all assigned edge nodes to send back their computed results. Specifically, the computation latency with CDC technique is determined by a group of the fastest edge nodes. Therefore, the distributed edge computing with CDC technique has been widely adopted in the AIoT platform [4], [5], [6], [7].

Among CDC techniques, the maximum distance separable (MDS) code has been widely used for matrix multiplication which is the most common operation in machine learning algorithms. By using MDS code, a computation task can be divided into $\boldsymbol{k}$ equal-size subtasks. These subtasks are then encoded into $\boldsymbol{n}$ coded subtasks, which are assigned to $\boldsymbol{n}$ edge nodes in a distributed manner where $\boldsymbol{k} \leq \boldsymbol{n}$. As soon as $\boldsymbol{k}$ edge nodes complete their assigned subtasks and send the results to the master node. Finally, the master node can obtain the expected result by decoding any $\boldsymbol{k}$ results from those $\boldsymbol{n}$ subtasks. In this way, the effect of straggling nodes can be significantly mitigated, and the speed up for the distributed matrix multiplication is decided by a factor of $\mathbf{log}\,\boldsymbol{n}$. In the MDS code, the decision for $\boldsymbol{k}$ and $\boldsymbol{n}$ values has a significant impact on the effectiveness and efficiency of the CDC process. When the $\boldsymbol{k}$ value is set to a higher value, there can be many straggling nodes; this can potentially lead to a long processing latency. When the $\boldsymbol{k}$ value is set to a lower value, the computation workload of the remaining $\boldsymbol{n}-\boldsymbol{k}$ edge nodes will be wasteful. Therefore, it is critical to optimize the tradeoff between the effectiveness and efficiency for the AIoT applications [4], [5], [6], [7].

As the study of edge assisted CDC system is still in its nascent stage, and has not received much attention. In this study, we focus on the cooperative game theory to solve the control problem for MDS code operations. Usually, game theory is used as a mathematical tool to understand and model cooperative or competitive situations which have multiple rational and selfish decision-makers. Especially, the central idea of game theory is to model strategic interactions as a game between a set of players. In cooperative games, all game players want to maximize their payoffs by choosing the best strategies, and no one has any incentive to change its own strategy. Several models of cooperative games are designed for different situations with the relationships of players. In contrast to traditional optimization methods which concentrate to achieve only one objective function, cooperative game models select the most acceptable solution for all players [8]. Because of complicated interactions between intelligent edge nodes and AIoT devices, the cooperative game can be a candidate to design an efficient MDS code management algorithm in the edge assisted CDC platform.

The rest of this paper is organized as follows. Section II describes the technical concepts and features of cooperative game theory, which are adopted to design our proposed CDC control scheme. The related work for the coded distributed computing approach is discussed in Section III. Section IV presents the distributed edge network platform, and formulates the CDC system control problem as a two-phase cooperative game model. And then, the proposed algorithm has been explained in a phasewise description manner. Based on the testbed experiments, computer simulation results are discussed to demonstrate the excellence of our approach in Section V. Finally, conclusion of the work and future study are highlighted in Section VI.

II. TECHNICAL CONCEPTS AND MAIN CONTRIBUTIONS

Game theory provides useful insights into the way game players that share a scarce resource may plan their use of the resource under different situations. Cooperative game theory is the part of game theory that pertains when players can sign binding contracts determining their actions and payoffs. Usually, cooperative game theory assumes that groups of players, called coalitions, are the primary units of decision-making, and may enforce cooperative behavior. In cooperative game theory, dynamic coalition formation game is a game that models the formation of coalitions of players when players have preferences over which coalition they belong to. It is specified by giving a finite set of players, and a preference ranking over all coalitions of players that the player belongs to. The outcome of a coalition formation game consists of a partition of the players into disjoint coalitions, that is, each player is assigned a unique group. Such partitions are often referred to as coalition structures [12], [15].

Bargaining solutions may be interpreted as formulas that determine unique outcomes in some class of cooperative game situations. Originally, J. Nash was the first pioneer to use the tools of game theory to propose a strong bargaining theory. As a bargaining solution, the *weighted Nash social welfare solution*(*WNSWS*) is an important welfare criterion that combines efficiency and fairness considerations. Simply, the outcome of *WNSWS*is to maximizes Nash social welfare functions; it maximizes the product of the individual players' payoffs. This idea goes back to J. Nash's famous solution concept to the cooperative bargaining problem. The *WNSWS*is not only to provide an important quality criterion, but also to successfully converge to the Nash-optimal outcome in a distributed negotiation model [13], [14].

In this study, we develop a new CDC control scheme based on the two-phase cooperative games: coalition formation game and resource bargaining game. At the first phase, $\boldsymbol{k}$ edge nodes are formed a group for the MDS code mechanism; it is operated based on the dynamic coalition formation game. At the second phase, the computing resource of each node is adaptively shared to process AIoT devices' offloading tasks according to the idea of *WNSWS*. Our hybrid game approach can dynamically control the trade-off between service latency and system efficiency while alleviating edge nodes' real straggling effects. Moreover, our approach can ensure the fairness among individual edge nodes in the CDC process. Therefore, we can effectively improve the offloading task delay, CDC system throughput and edge fairness. Specifically, the major contributions of this paper can be summarized as follows:

- We propose a reliable CDC control scheme for the AIoT paradigm, which can support the immersive user experiences in the 6G networks. Especially, we adopt the fundamental ideas of cooperative game theory, and develop a new two-phase game model for the MDS code execution.
- First phase game model: we dynamically form an edge node coalition to decide the k value for MDS code operations. Therefore, we can process the offloaded computation tasks of AIoT devices in the distributed computing infrastructure.
- Second phase game model: each individual edge node allocates its computing resource for the assigned offloading tasks. In a distributed fashion, the limited computation capacity is shared by using the concept of *WNSWS*.
- To mitigate the straggler effects of distributed computing, strategies in two cooperative games are adaptively adjusted. Our jointly designed approach enables the sequential interactions of different system agents to achieve a mutually desirable solution.
- Perform extensive simulations show the efficiency of our proposed scheme compared to those of the state-of-the-art CDC control protocols. Numerical results clearly indicate the superiority of our cooperative game approach in terms of the offloading task delay, CDC system throughput and edge fairness.

III. RELATED WORK

Over recent years, edge assisted CDC systems have shown great promise in many applications including big data analytics and machine learning applications. In distributed systems, the straggling effect requires to be managed while the voluntary cooperation of edge nodes is hard to expect due to their selfish nature. To handle these issues, some researchers have investigated CDC control protocols to speed up the distributed computing process.

The paper [7] proposes the *Coded distributed computing Elastic Resource Allocation*(*CERA*) scheme to elastically allocate computing resources for CDC processes. To jointly optimize the CDC control over heterogeneous edge nodes, the *CERA*scheme consists of two stages. In the first stage, a joint coding and node selection optimization problem is formulated to minimize the expected processing time for a CDC task. Since this problem is nonlinear and NP-hard, an effective integer non-linear programming is modeled to quickly obtain the optimal solutions. This approach can significantly reduce the problem's computational complexity. In the second stage, a smart online approach with the Lyapunov optimization method is developed to dynamically turn off straggling nodes based on their actual performance. To greatly improve the resource utilization, this approach enables the server to optimize the trade-off between the total processing time and the system's resource efficiency. Simulation results have demonstrated that the *CERA*scheme yields a significant gain in terms of the total processing time [7].

In [9], the *Optimal Coded Network Service*(*OCNS*) scheme is designed to mitigate the computing effect of straggling nodes. To motivate the participation of mobile devices in the CDC platform, the *OCNS*scheme proposes incentive mechanisms that distribute the incentive based on workload and completion time of mobile devices. This approach captures the needs for discounting values of the processed results. To analyze the interaction among mobile devices and task publisher, their behaviors are modelled to be motivated by economic aspects, and formulated as a Stackelberg game with a hierarchical decision-making structure. In the *OCNS*scheme, mobile devices lead the competition as leaders; they provide the computational resources, and the task publisher rationally determines its response as a follower; it has a limited budget to response to the mobile device's behavior. Through game-theoretic analysis the *OCNS*scheme can achieve a unique Stackelberg equilibrium with the guarantee of convergence [9].

Yu et al. develop the *Optimal Distributed Resource Allocation*(*ODRA*) scheme to minimize the total execution time accounting for the durations of both computation and communication phases [10]. Specially, the *ODRA*scheme considers a general MapReduce-type framework based on the CDC technique. In this approach, overall computation is decomposed to three stages, Map, Shuffle, and Reduce stages. In the Map phase, each input file is processed locally to generate intermediate values. In the Shuffle phase, all intermediate values are transferred to one of the nodes for reduction. In the Reduce phase, all intermediate values are reduced to the final result. To prove the effectiveness of the *ODRA*scheme, a matching information-theoretic converse on the execution time is derived, and the total execution time of the computation tasks is adaptively reduced [10].

The earlier schemes in [7], [9], [10] have been studied and recently published to mitigate the straggling effect in the distributed computing platform. As aforementioned, a few researchers tackled the problems of stragglers to improve the performance of CDC methods. Even though these existing schemes dynamically control the activities of edge nodes to reduce the processing time of the whole CDC system, they did not consider the cooperative mechanism among intelligent system agents. To the best of our knowledge, our proposed scheme is the first in the literature to investigate a two-phase cooperative game model, and guides selfish CDC system agents toward a socially optimal outcome in the distributed edge computing platform.

IV. MDS CODE CONTROL SCHEME FOR THE AIOTPARADIGM

In this section, we first provide an edge assisted CDC system infrastructure, and the basic ideas of dynamic coalition game and *WNSWS*. And then, we present our proposed scheme to efficiently solve the (n, k) MDS code problem, which captures the benefit of adaptive distributed computing approach.

A. EDGE ASSISTED CDC PLATFORM AND A TWO-PHASE COOPERATIVE GAME MODEL

As illustrated in Fig.1, we consider an edge assisted CDC system platform consisting of one edge computing server ($\mathcal{C}$), a set of mobile edge nodes $\mathbb{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$, and a set of AIoT devices $\mathbb{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ in a given geographical area. The edge node $\mathcal{E}_{1 \le i \le n} \in \mathbb{E}$ has his computing power $(\mathfrak{M}_{\mathcal{E}_i})$, and communicates with the $\mathcal{C}$ through wired links.

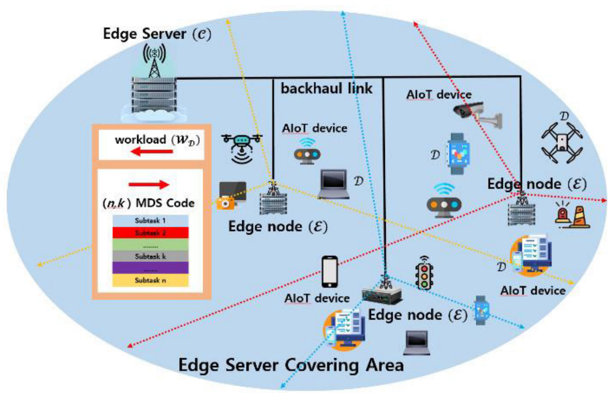


FIGURE 1. The infrastructure of edge assisted CDC system.

Each AIoT device $\mathcal{D}_{1 \le j \le m} \in \mathbb{D}$ generates its computational workload task $(\mathcal{W}_{\mathcal{D}_j})$; $\mathcal{W}_{\mathcal{D}_j}$ is assumed to be delay sensitive, and follows a Poisson process with an average generation rate at each time period. To reduce the computation load, the $\mathcal{D}_j$ can offload its task $\mathcal{W}_{\mathcal{D}_j}$ to the $\mathcal{C}$. Using the (n, k) MDS code, the $\mathcal{C}$ divides the computing task $(\mathcal{W}_{\mathcal{D}_j})$ into k subtasks with an equal size. Then, these subtasks are encoded into n coded subtasks and sent them to n edge nodes. Edge nodes, i.e., $\mathcal{E} \in \mathbb{E}$, locally perform their corresponding subtasks, which are distributed from the $\mathcal{C}$. While performing the distributed computation, each edge node periodically measures the remaining workload and reports this information to the $\mathcal{C}$ [7].

In this paper, the (n, k) MDS code problem for the $\mathcal{D}_j$ and the $\mathcal{E}_i$'sresource sharing problem are formulated as cooperative games $\mathbb{G}_{\mathcal{D}_j}$ and $\mathbb{G}_{\mathcal{E}_i}$, respectively. As a consequence, server, edge nodes, and AIoT devices aresequentially interacted with each other; it is noteworthy that we formulate the $\mathcal{C} - \mathcal{E} - \mathcal{D}$ association in a coordinated manner. Formally, we define our two-phase game $\mathbb{G}$ entities, i.e., $\mathbb{G} = \{\mathbb{G}_{\mathcal{D}_j}, \ \mathbb{G}_{\mathcal{E}_i}\} = \{\mathcal{C}, \ \mathbb{E}, \mathbb{D}, \{\mathbb{G}_{\mathcal{D}_j} | \mathcal{D}_j \in \mathbb{D}, \mathbb{E}, \mathcal{W}_{\mathcal{D}_j}, v(\cdot)\}, \{\mathbb{G}_{\mathcal{E}_i} | \mathcal{E}_i \in \mathbb{E}, \mathfrak{M}_{\mathcal{E}_i}, \mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}, \ S_{\mathcal{D}_j}^{\mathcal{E}_i}, U_{\mathcal{D}_j}^{\mathcal{E}_i}(\cdot)\}, \ T\}$ of gameplay.

- $\mathcal{C}$, $\mathbb{E}$ and $\mathbb{D}$ represent the edge computing server, the sets of edge nodes, and AIoT devices, respectively. They are mutually and reciprocally interdependent, and work together in the edge assisted CDC platform.
- At the first phase, the $\mathbb{G}_{\mathcal{D}_j}$ is designed to form an edge structure to process the subtasks of $\mathcal{W}_{\mathcal{D}_j}$. In the $\mathbb{G}_{\mathcal{D}_j}$, edge nodes are game players, and $v(\cdot)$ is a characteristic function for players' coalition.
- At the second phase, the $\mathbb{G}_{\mathcal{E}_i}$ is developed to share the $\mathfrak{M}_{\mathcal{E}_i}$ for the subtasks of the AIoT devices. Each $\mathbb{G}_{\mathcal{E}_{1 \le i \le n}}$ game is operated in a distributed parallel fashion.
- In the $\mathbb{G}_{\mathcal{E}_i}$, $\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}$ is the $\mathcal{D}_j$'s subtask assigned to the $\mathcal{E}_i$, and it is a game player. The $S_{\mathcal{D}_j}^{\mathcal{E}_i}$ and $U_{\mathcal{D}_j}^{\mathcal{E}_i}(\cdot)$ are the player's strategy and utility function, respectively.
- Discrete time model $T \in \{t_1, \ldots, t_c, t_{c+1}, \ldots\}$ is represented by a sequence of time steps. The length of t_c matches the event time-scale of $\mathbb{G}_{\mathcal{D}_j}$ and $\mathbb{G}_{\mathcal{E}_i}$.

B. FUNDAMENTAL IDEAS OF COALITION GAME MODEL AND WNSWS

Traditionally, forming effective coalitions is a major research challenge in cooperative games; a coalition of players can also do things more efficiently than individual players can do. Therefore, coalition formation entails finding a coalitional structure that maximizes the total payoff. In the coalition formation game, an external factor imposes a certain structure to form stable coalitions, and no player has an incentive to deviate. Formally, a coalition formation game is defined by (N, v) which $N = \{p_1, \ldots, p_n\}$ is the set of players and v is a real-valued characteristic function such that $v(S) : 2^N \rightarrow \mathbb{R}$, s.t., $S \subset N$; a subset S is termed a coalition. Given a coalition game, a coalition structure $\mathcal{C} = \{S_1, \ldots, S_k\}$ is an exhaustive disjoint partition of the space of players into feasible coalitions and $v(\mathcal{C}) = \sum_{S \in \mathcal{C}} v(S)$. By considering game dynamics, the properties from resulting coalitions and its adaptability to environment variable or externalities are important research issues [11], [12].

To implement a dynamic coalition formation algorithm, some definition and rules are necessary. First of all, the coalition composed of all players is referred as a grand coalition $\mathcal{N}$, and a collection of coalitions in the $\mathcal{N}$, denoted $\mathcal{S}$, is defined as the set $\mathcal{S} = \{S_1, \ldots, S_k\}$ of mutually disjoint coalitions $S_{1 \leq i \leq k} \subset \mathcal{N}$ and $\mathcal{N} = \bigcup_{j=1}^{k} S_j$. A*preference relation*$\triangleright$ is an order defined for comparing two coalition collections $\mathcal{S} = \{S_1, \ldots, S_l\}$ and hat $\mathcal{S} = \left\{\hat{S}_1, \ldots, \hat{S}_p\right\}$ that are partitions of the same subset $\mathcal{A} \subseteq \mathcal{N}$. Therefore, game players in $\mathcal{S}$ and hat $\mathcal{S}$ are same. In this case, $\mathcal{S} \triangleright \hat{\mathcal{S}}$ implies that the way $\mathcal{S}$ partitions $\mathcal{A}$ is preferred to the way $\hat{\mathcal{S}}$ partitions $\mathcal{A}$. Based on the concept of *preference relation*, merge and split rules can be defined as follows [11], [12].

Merge Rule: Any set of coalitions $\{S_1, \ldots, S_k\}$ may be merged whenever the merged form is preferred by the players; i.e., where $\{\bigcup_{j=1}^{k}, S_j\} \triangleright \{S_1, \ldots, S_k\}$, therefore, $\{S_1, \ldots, S_k\} \rightarrow \{\bigcup_{j=1}^{k}, S_j\}$.

Split Rule: Any coalition $\bigcup_{j=1}^{k} S_j$ may be split whenever a split form is preferred by the players; i.e., where $\{S_1, \ldots, S_k\} \triangleright \{\bigcup_{j=1}^{k}, S_j\}$, thus, $\{\bigcup_{j=1}^{k}, S_j\} \rightarrow \{S_1, \ldots, S_k\}$.

The basic idea behind the merge-and-split rules is that players enter into a binding agreement to form a coalition through the merge (*or*split) operation if all players are able to improve (*or*not decrease) their individual payoffs from the previous state [11], [12].

The underlying idea of the *WNSWS*comes from the field of bargaining game theory. To get the bargaining solution, all players agree to create a grand coalition for their higher payoffs. To find the optimal grand coalition, we need to optimize the super-criterion on the set of all feasible actions. In the literature of bargaining problems, super-criteria are often referred to as social welfares. Therefore, the main concern of *WNSWS*is to optimize the *Nash Social Welfare*(*NSW*) function, which is a super-criterion over the feasible allocation while ensuring both fairness and efficiency at the same time. To characterize the basic ideas of *WNSWS*, we first start with some definitions. Let $N = \{1 \ldots i \ldots n\}$ be the set of imaginary players, and let $\mathbb{R}$ be the set of all real numbers. $\curvearrowleft \subseteq \mathbb{R}^n$ represents the set of feasible solutions and it is assumed to be bounded where $\mathbb{R}^n$ is the $n-$ fold Cartesian product of $\mathbb{R}$. For all $i \in N, f(\mathbb{x}) := (x \in \mathbb{x} | f_1(x), \ldots, f_n(x))$ is a vector of payoff functions. Traditionally, a multi-objective optimization problem can be stated as [13], [14]:

$$\begin{cases} \max\left\{i \in S^+, x \in \curvearrowleft | f_i(x)\right\} \\ \min\left\{j \in S^-, x \in \curvearrowleft | f_j(x)\right\} \end{cases}, \text{ s.t., } \quad S^+ \cup S^- = N \tag{1}$$

where S^+ and S^- are the index sets of payoff functions that need to be maximized and minimized, respectively. Denote $\mathbb{w} = (\omega_1, \ldots, \omega_n)$ is a vector of players' weights with $\omega_i > 0$ for all $i \in N$; they are players' corresponding degrees of importance. Based on the positive weights, we can aggregate players' payoff functions to form a single-objective optimization problem [13].

$$\max\left\{\sum_{i \in S^+}(\omega_i \times f_i(x)) - \sum_{j \in S^-}\left(\omega_j \times f_j(x)\right)\right\}$$
$$\text{s.t.}, \quad S^+ \cup S^- = N \text{ and } x \in \mathbb{X} \tag{2}$$

Equation (2) is called as the weighted sum optimization in the literature of multi-objective optimization problem. According to the ideas of (1)-(2), the *WNSWS*is mathematically formulated as follows:

$$WNSWS = \max_{x \in \mathbb{X}}\left(\prod_{i \in S^+}(f_i(x) - d_i)^{\omega_i} \times \prod_{j \in S^-}\left(d_j - f_j(x)\right)^{\omega_j}\right)$$
$$\text{s.t.}, \quad f_i(x) \geq d_i, \ \forall i \in S^+ \text{ and } f_j(x) \leq d_j, \forall j \in S^- \tag{3}$$

where d_i is the i's reference point. In the literature of bargaining problems, the reference point is sometimes referred as the disagreement point and it basically indicates the payoff of each player under no coalition. Note that since the player $i \in S^+$ is interested in maximizing its payoff function, $(f_i(x) - d_i)^{\omega_i}$ captures the benefit that it will obtain as the result of creating a coalition. Similarly, the player $j \in S^-$ is interested in minimizing its payoff function. Therefore, $\left(d_j - f_j(x)\right)^{\omega_j}$ captures the benefit that it obtains as the result of creating a coalition. Considering players' weights, the main goal of *WNSWS*is to maximize the product of benefits of players with respect to the reference point. The *WNSWS*in equation (3) can be transformed as a *local-benefit-scale-free WNSWS*, i.e., *lbsf-WNSWS*, by replacing the payoff function of the *WNSWS*with the arbitrary positive constants $\alpha_{1\le i\le n}$ [13], [14].

$$
\begin{aligned}
& lbsf - WNSWS \\
&= \max_{x\in\mathbb{X}} \left(\begin{array}{l} \prod_{i\in S^+} ((\alpha_i \times f_i(x)) - (\alpha_i \times d_i))^{\omega_i} \\ \times \prod_{j\in S^-} \left((\alpha_j \times d_j) - (\alpha_j \times f_j(x))\right)^{\omega_j} \end{array} \right) \\
&= \max_{x\in\mathbb{X}} \left(\begin{array}{l} \prod_{k=1}^{n} (\alpha_k)^{\omega_k} \\ \times \prod_{i\in S^+} (f_i(x) - d_i)^{\omega_i} \\ \times \prod_{j\in S^-} \left(d_j - f_j(x)\right)^{\omega_j} \end{array} \right) \\
& \text{s.t.,} \begin{cases} \alpha_k = \dfrac{1}{\max_{x\in\mathbb{X}} (f_k(x) - d_k)}, & \forall k \in S^+ \\ \alpha_k = \dfrac{1}{\max_{x\in\mathbb{X}} (d_k - f_k(x))}, & \forall k \in S^- \end{cases}
\end{aligned} \tag{4}
$$

Since $\prod_{k=1}^{n} (\alpha_k)^{\omega_k}$ is a positive constant, it has no impact on the optimization and it can be dropped. Therefore, the maximum benefit that each player can obtain will be precisely one in the equivalent problem. This implies that an equivalent *WNSWS*with unit-maximum-benefit can be constructed from the original *WNSWS*formula [13], [14].

C. THE PROPOSED EDGE ASSISTED CDC CONTROL SCHEME FOR THE AIoT PARADIGM

To develop our edge assisted CDC control scheme for AIoT devices, we formulate a new two-phase cooperative game model. At the first phase, we define the n and k values for the MDS code. In our distributed computing scenario, individual devices contact the $\mathcal{C}$ for their offloading services, and have their corresponding edge nodes in the $\mathcal{C}$'s coverage area. Simply, we set the n value as the number of corresponding edge nodes. And then, the decision for k value is made based on the dynamic coalition formation game. For example, the $\mathcal{D}_j$ generates the $\mathcal{W}_{\mathcal{D}_j}$, and the $\mathbb{G}_{\mathcal{D}_j}$ is operated to form a coalition with efficiently workable edge nodes. The characteristic function of each individual edge node is defined as the worth of its contribution. For the $\mathcal{E}_i$, its characteristic function $v(\mathcal{E}_i)$ is given by:

$$
\begin{aligned}
& v\left(\mathcal{E}_i, \mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}\right) \\
& \quad = \eta - \left(\mathcal{G}\left(C_{\mathcal{E}_i}\right) \times log \left(\frac{\left(C_{\mathcal{E}_i} + f\left(\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}\right)\right)}{\mathfrak{M}_{\mathcal{E}_i}} + \theta \right) \right) \\
& \text{s.t. } \mathcal{G}\left(C_{\mathcal{E}_i}\right) = \begin{cases} \varphi, & \text{if } f\left(\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}\right) \ge \left(\mathfrak{M}_{\mathcal{E}_i} - C_{\mathcal{E}_i}\right) \\ \exp\left(\dfrac{f\left(\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}\right)}{\left(\mathfrak{M}_{\mathcal{E}_i} - C_{\mathcal{E}_i}\right)} \right), & \text{otherwise} \end{cases}
\end{aligned} \tag{5}
$$

where η, θ, and φ are control parameters for the $v(\cdot)$. $C_{\mathcal{E}_i}$ is the currently used computing power of $\mathcal{E}_i$, and $f\left(\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}\right)$ is the needed computing capacity for the $\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}$ where $\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}$ is the $n-$divided $\mathcal{W}_{\mathcal{D}_j}$ and assigned to the $\mathcal{E}_i$. $v(S)$ is the characteristic function that associates to each edge node coalition S.

$$
v(S) = \prod_{\mathcal{E}_i \in S} v\left(\mathcal{E}_i\right), \quad \text{s.t., } S \subseteq \mathbb{E} \tag{6}
$$

According to (6), we can form the best coalition S, which consists of adaptable edge nodes to process the $\mathcal{W}_{\mathcal{D}_j}$. Finally, the k value is given by:

$$
\begin{aligned}
& k = \left| S_{\mathcal{D}_j}^{*} \right| \\
& \text{s.t., } v\left(S_{\mathcal{D}_j}^{*}\right) = \max_{S_{\mathcal{D}_j}^{*} \subseteq \mathbb{E}} \left(\mathcal{E}_i \in \mathbb{E} \middle| \prod_{\mathcal{E}_i \in S_{\mathcal{D}_j}^{*}} v\left(\mathcal{E}_i\right) \right)
\end{aligned} \tag{7}
$$

where $\left|S_{\mathcal{D}_j}^{*}\right|$ is the cardinality of $S_{\mathcal{D}_j}^{*}$. Based on the $S_{\mathcal{D}_j}^{*}$ formation, the k subtasks of $\mathcal{W}_{\mathcal{D}_j}$ are assigned to the edge nodes in the $S_{\mathcal{D}_j}^{*}$. In the $\mathbb{G}_{\mathcal{D}}$ for each AIoT device, we can create its coalition $S_{\mathcal{D}}^{*}$. Based on this information, each edge node also creates two subtask coalitions, i.e., $S_{\mathcal{E}}^{+}$ and $S_{\mathcal{E}}^{-}$, where these subtasks are offloaded from its corresponding AIoT devices. For example, the $\mathcal{E}_i$ has his two coalitions such as the $S_{\mathcal{E}_i}^{+}$ and $S_{\mathcal{E}_i}^{-}$. They are defined as follows:

$$
\begin{aligned}
& S_{\mathcal{E}_i}^{+} = \bigcup \left\{ \mathcal{D}_j \in \mathbb{D} \middle| \mathcal{E}_i \in S_{\mathcal{D}_j}^{*} \right\} \quad \text{and} \\
& S_{\mathcal{E}_i}^{-} = \bigcup \left\{ \mathcal{D}_k \in \mathbb{D} \middle| \mathcal{E}_i \notin S_{\mathcal{D}_k}^{*} \right\}
\end{aligned} \tag{8}
$$

At the second stage, individual edge nodes allocate their computing powers $\left(\mathfrak{M}_{\mathcal{E}_{1\le i\le n}}\right)$ for the assigned subtasks. For the $\mathcal{E}_i$, the $\mathbb{G}_{\mathcal{E}_i}$ game model is developed in a distributed manner, and the $\mathfrak{M}_{\mathcal{E}_i}$ is shared by the corresponding AIoT devices. If the $\mathcal{D}_j$ ask its offload task to the $\mathcal{E}_i$, his utility function, i.e., $U_{\mathcal{D}_j}^{\mathcal{E}_i}(\cdot)$, is given by equation (9), as shown at the bottom of the next page, where γ, $\delta$and Γ are control parameters for the $U_{\mathcal{D}_j}^{\mathcal{E}_i}(\cdot)$, which is differently defined whether the $\mathcal{D}_j$ in $S_{\mathcal{E}_i}^{+}$ or $S_{\mathcal{E}_i}^{-}$. $\Delta_{\mathcal{D}_j}^{\mathcal{E}_i}$ is the remaining workload of $\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}$. Therefore, the $\Delta_{\mathcal{D}_j}^{\mathcal{E}_i}$ is needed to be computed by the $\mathcal{E}_i$ for each offload task. To process the $f\left(\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}\right)$, $R_{\mathcal{D}_j}^{\mathcal{E}_i}$ and $S_{\mathcal{D}_j}^{\mathcal{E}_i}$ are the requested and assigned computing powers at the next time period. Therefore, the $S_{\mathcal{D}_j}^{\mathcal{E}_i}$ is thought as the strategy for the $\mathcal{W}_{\mathcal{D}_j}^{\mathcal{E}_i}$. $\mathfrak{P}_{\mathcal{E}_i}$ represents the resource assignment ratio for services in $S_{\mathcal{E}_i}^{+}$. In this study, we adopt the *WNSWS* to decide the $S_{\mathcal{D}_j}^{\mathcal{E}_i}$. According to (3)-(4), the *WNSWS* for the $\mathcal{E}_i$, i.e., $WNSWS_{\mathcal{E}_i}$, is obtained as follows:

$$WNSWS_{\mathcal{E}_i} = \max_{\left\langle \ldots S^{\mathcal{E}_i}_{\mathcal{D}_j}, S^{\mathcal{E}_i}_{\mathcal{D}_k}, \ldots \right\rangle} \left(\begin{array}{c} \prod_{\mathcal{D}_j \in S^+_{\mathcal{E}_i}} \left(U^{\mathcal{E}_i}_{\mathcal{D}_j}(\cdot) - d^+_{\mathcal{D}_j} \right)^{\omega_{\mathcal{D}_j}} \\ \times \prod_{\mathcal{D}_k \in S^-_{\mathcal{E}_i}} \left(d^-_{\mathcal{D}_k} - U^{\mathcal{E}_i}_{\mathcal{D}_k}(\cdot) \right)^{\omega_{\mathcal{D}_k}} \end{array} \right)$$
$$\text{s.t., } \mathcal{D}_j, \mathcal{D}_k \in \mathbb{D}^{\mathcal{E}_i}, \ S^+_{\mathcal{E}_i} \cup S^-_{\mathcal{E}_i} = \mathbb{D}^{\mathcal{E}_i}, \text{ and}$$
$$\sum_{\mathcal{D}_l \in \mathbb{D}^{\mathcal{E}_i}} S^{\mathcal{E}_i}_{\mathcal{D}_l} \leq \mathfrak{M}_{\mathcal{E}_i} \tag{10}$$

where $d^+_{\mathcal{D}_j}$ is the $\mathcal{D}_j$'s reference point if $\mathcal{D}_j \in S^+_{\mathcal{E}_i}$, and $d^-_{\mathcal{D}_k}$ is the $\mathcal{D}_k$'s reference point if $\mathcal{D}_k \in S^-_{\mathcal{E}_i}$. $\omega_{\mathcal{D}_j}$ and $\omega_{\mathcal{D}_l}$ are the weights of $\mathcal{D}_j$ and $\mathcal{D}_k$, respectively. $\mathbb{D}^{\mathcal{E}_i}$ is the set of AIoT devices, which assign their offload workloads to the $\mathcal{E}_i$.

D. MAIN STEPS OF OUR HYBRID COOPERATIVE GAME BASED CDC CONTROL ALGORITHM

Distributed computing has been rapidly emerging as a popular computing paradigm driven by a growing demand from the massive AIoT devices. It is enabled by innovations and advancements of modern wired and wireless networks. However, distributed computing methods face challenges in meeting the required service quality, and satisfying the complex demands of AIoT devices, especially for latency-sensitive applications. Recently, coding techniques have become a popular approach to solve the challenges of the distributed computing systems. In this paper, we introduce a new CDC control scheme, which shows the effectiveness to mitigate the straggler effects based on the cooperative game features. By a sophisticated combination of coalition formation and bargaining games, our proposed two-phase hybrid approach isvery effective to enhance the performance of edge assisted distributed computing system while adaptively handling among conflicting service requirements.

The primary steps of our proposed scheme are described as follows.

Step 1: Based on the simulation scenario, control factors and parameters in our proposed scheme are determined by the Section V and Table 1.

Step 2: At each time period, individual AIoT devices in the $\mathbb{D}$ generates their computation-intensive tasks ($\mathcal{W}_{\mathcal{D}}$), which are offloaded to edge nodes in their coverage area.

Step 3: At the first phase, individual edge nodes form two coalitions to execute the (n, k)MDS code for each $\mathcal{D}$. This process is formulated as a dynamic coalition formation game ($\mathbb{G}_{\mathcal{D}}$) in a coordinated manner.

Step 4: In the $\mathbb{G}_{\mathcal{D}}$, characteristic function for each edge node and coalition (S) are defined as (5),(6) and (7), respectively. Finally, the k value is given from the equation (7).

Step 5: At the second phase, each edge node share its limited $\mathfrak{M}_{\mathcal{E}}$ for the assigned offloaded computation tasks in a distributed manner. For each $\mathcal{E}$, it is developed as a bargaining game ($\mathbb{G}_{\mathcal{E}}$).

Step 6: As the solution concept of $\mathbb{G}_{\mathcal{E}}$, the *WNSWS*is employed based on the equations (1)-(4), each $\mathcal{W}^{\mathcal{E}}_{\mathcal{D}}$ in $\mathcal{E}$ works together as a game player.

Step 7: In the of $\mathbb{G}_{\mathcal{E}}$, utility function for game player is defined as (9), and the $\mathfrak{M}_{\mathcal{E}}$ is shared according to (10).

Step 8: During a sequence of time steps, the $\mathbb{G}_{\mathcal{D}}$ and $\mathbb{G}_{\mathcal{E}}$ games sequentially interact each other to reach an efficient consensus. This interactive feedback process continues in a step-by-step manner to achieve a mutually desirable solution.

Step 9: Individual system agents are constantly self-monitoring the current edge assisted distributed computing environments. At each time period, a new two-phase cooperative game process is re-triggered; proceeds to Step 2 for the next iteration.

$$U^{\mathcal{E}_i}_{\mathcal{D}_j}\left(\mathcal{W}^{\mathcal{E}_i}_{\mathcal{D}_j}, \Delta^{\mathcal{E}_i}_{\mathcal{D}_j}, R^{\mathcal{E}_i}_{\mathcal{D}_j}, S^{\mathcal{E}_i}_{\mathcal{D}_j}\right) = \begin{cases} exp\left(\gamma \times \left(\dfrac{f\left(\mathcal{W}^{\mathcal{E}_i}_{\mathcal{D}_j}\right) - \Delta^{\mathcal{E}_i}_{\mathcal{D}_j}}{f\left(\mathcal{W}^{\mathcal{E}_i}_{\mathcal{D}_j}\right)}\right)\right) \times \dfrac{\min\left(S^{\mathcal{E}_i}_{\mathcal{D}_j}, R^{\mathcal{E}_i}_{\mathcal{D}_j}\right)}{R^{\mathcal{E}_i}_{\mathcal{D}_j}}, \text{if} \mathcal{D}_j \in S^+_{\mathcal{E}_i} \\ \left(\mathfrak{P}_{\mathcal{E}_i} \times \log\left(\left(\dfrac{f\left(\mathcal{W}^{\mathcal{E}_i}_{\mathcal{D}_j}\right) - \Delta^{\mathcal{E}_i}_{\mathcal{D}_j}}{f\left(\mathcal{W}^{\mathcal{E}_i}_{\mathcal{D}_j}\right)}\right) + \delta\right) + \Gamma\right) \times \dfrac{\min\left(S^{\mathcal{E}_i}_{\mathcal{D}_j}, R^{\mathcal{E}_i}_{\mathcal{D}_j}\right)}{R^{\mathcal{E}_i}_{\mathcal{D}_j}}, \ \text{if} \mathcal{D}_j \in S^-_{\mathcal{E}_i} \end{cases}$$
$$\text{s.t.,} \mathfrak{P}_{\mathcal{E}_i} = -\left(\frac{\sum_{\mathcal{D}_l \in S^+_{\mathcal{E}_i}} S^{\mathcal{E}_i}_{\mathcal{D}_l}}{\sum_{\mathcal{D}_l \in S^+_{\mathcal{E}_i}} R^{\mathcal{E}_i}_{\mathcal{D}_l}}\right) \tag{9}$$

V. PERFORMANCE EVALUATION

This section provides simulation results to verify the effectiveness of the proposed CDC control scheme. To outline the benefits of our approach, we show a detailed comparative analysis with other competing protocols of *CERA* [7], *OCNS* [9] and *ODRA* [10]. The evaluations are carried out using the MATLAB, which is an interactive programming environment for scientific computing. To ensure a fair comparison, the following assumptions and system scenario are used.

- Simulated the edge assisted CDC platform consists of one edge server ($\mathfrak{C}$), ten edge nodes and one hundred AIoT devices, i.e., $|\mathbb{E}| = 10$, and $|\mathbb{D}| = 100$.
- Each AIoT device $\mathcal{D}_{1\le j\le 100}$ generates different computation-intensive tasks $\left(\mathcal{W}_{\mathcal{D}_j}\right)$ where the arrival process of $\mathcal{W}_{\mathcal{D}_j}$ is the rate of Poisson process (ρ). The offered range is varied from 0 to 3.0.
- Ten edge nodes are deployed in the $\mathfrak{C}$'s coverage area, and individual AIoT devices are randomly distributed over there.
- Each individual AIoT device can directly contact with the $\mathfrak{C}$, and it communicates with the edge nodes through wired backhaul link.
- The reference point for $\mathcal{D}\in S_{\mathcal{E}}^{+}$ $\left(d_{\mathcal{D}}^{+}\right)$ is zero, and the reference point for $\mathcal{D}\in S_{\mathcal{E}}^{-}$ $\left(d_{\mathcal{D}}^{-}\right)$ is zero, too.
- The weights ($\omega_{\mathcal{D}}$) of all devices are simply assumed as 1.• We assume the absence of physical obstacles in the $\mathfrak{C}$'s coverage area.
- The total computation power of each edge node is randomly decided; the range is varied from 50 GHz to 150 GHz.
- To reduce the computation complexity, the offloading service amount is specified in terms of basic unit ($u_{\mathfrak{M}}$) where one $u_{\mathfrak{M}}$ is 100 MHz in this study. For practical implementations, the computing resource allocation through the *WNSWS*is negotiated discretely by the size of one $u_{\mathfrak{M}}$.
- The edge assisted CDC system performance measures obtained on the basis of 100 simulation runs are plotted as functions of the Poisson process (ρ).

To evaluate the proposed solution, we compare its performance in terms of normalized service latency, CDC system throughput and edge fairness. Table 1 shows the control parameters and system factors used in the simulation.

Fig. 2 depicts the results of service latency for offloaded tasks; they are normalized for a fair comparison. As the AIoT device workload ratio increases, the service latency of all schemes grows polynomially. However, it may easily be seen that our proposed scheme outperforms than all other CDC control protocols such as *CERA*, *OCNS*and *ODRA*schemes. Based on the idea of *WNSWS*, edge nodes in our distributed computing algorithm effective share their computing resources through the interactive negotiation process between two subtask coalitions. Therefore, we can fully exploit the limited computing resource of each edge node while reducing the service delay for offloading tasks. It means that our cooperative game approach can significantly mitigate the straggling effects under the dynamic changing edge assisted distributed computing environments.

To further compare the performance of all schemes, we evaluate their CDC system throughput with the different task load average. In the viewpoint of system operators, it is a main performance criterion to evaluate the system efficiency. As can be observed, the system throughputs of all protocols are improved gradually while increasing average workload rate. This is a common-sense result. But, our proposed scheme adopts a dynamic formation game to find an

TABLE 1. System parameters used in the simulation experiments.

Parameter	Value	Description	
n	10	total number of edge nodes	
m	100	total number of AIoT devices	
$\mathfrak{M}_{\mathcal{E}}$	50GHz~ 150GHz	total computation power of each edge device	
$u_{\mathfrak{M}}$	100 MHz	the minimum amount of computing resource allocation	
η , θ	2 , 1	control parameters for the $v(\cdot)$	
ϕ	7	an adjustment factor for the $v(\cdot)$	
γ	0.5	an adjustment factor for the $U_{\mathcal{D}}^{\mathcal{E}}(\cdot)$	
δ , Γ	1 , 1	control parameters for the $U_{\mathcal{D}}^{\mathcal{E}}(\cdot)$	
$d_{\mathcal{D}}^{+}$	0	a reference point for $\mathcal{D} \in S_{\mathcal{E}}^{+}$	
$d_{\mathcal{D}}^{-}$	0	a reference point for $\mathcal{D} \in S_{\mathcal{E}}^{-}$	
$\omega_{\mathcal{D}}$	1	the weights of $\mathcal{D}$	
Workload ($\mathcal{W}_{\mathcal{D}}$) type	**$\mathcal{W}_{\mathcal{D}}$ amount**	**$f(\mathcal{W}_{\mathcal{D}}^{\mathcal{E}})$ amount**	**Service duration average /*t***
$\mathcal{W}_{\mathcal{D}} \in$ I	0.8GHz	0.1GHz	90 time-periods
$\mathcal{W}_{\mathcal{D}} \in$ II	1.7GHz	0.2GHz	100 time-periods
$\mathcal{W}_{\mathcal{D}} \in$ III	2GHz	0.2GHz	50 time-periods
$\mathcal{W}_{\mathcal{D}} \in$ IV	2.8GHz	0.3GHz	80 time-periods
$\mathcal{W}_{\mathcal{D}} \in$ V	3.8GHz	0.4GHz	60 time-periods
$\mathcal{W}_{\mathcal{D}} \in$ VI	4.5GHz	0.5GHz	45 time-periods

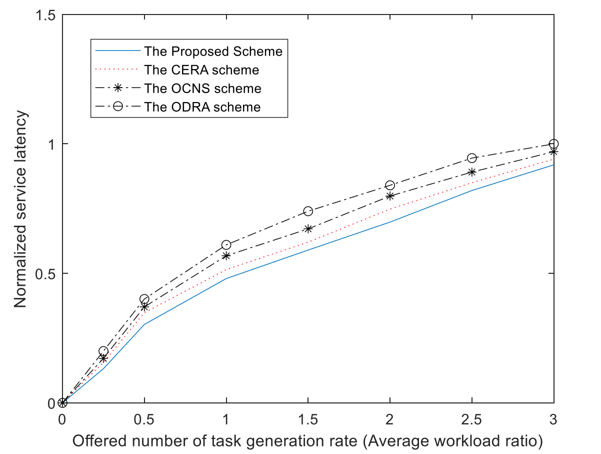


FIGURE 2. Normalized service latency for offloaded tasks.

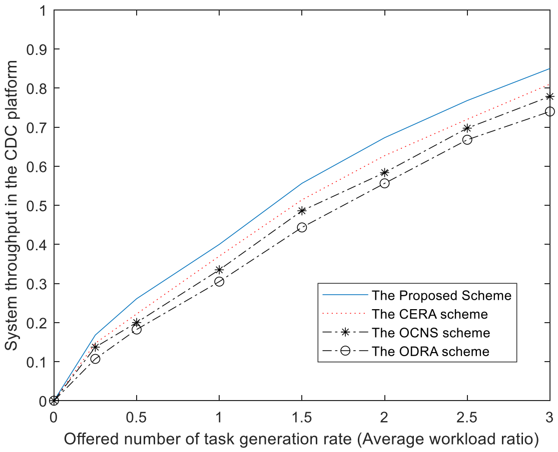


FIGURE 3. System throughput in the CDC platform.

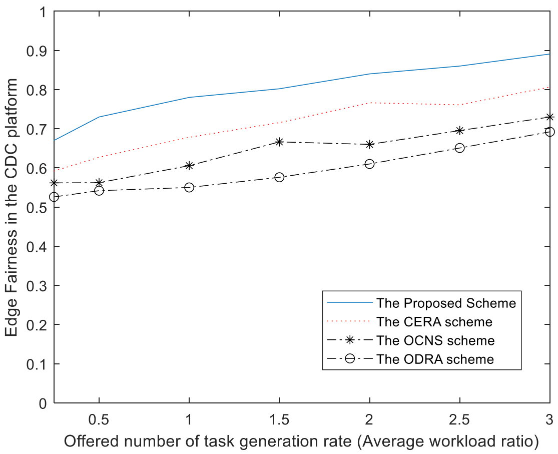


FIGURE 4. Edge Fairness in the CDC platform.

optimal coalitional structure that maximizes the CDC system efficiency. Compare to the static coalition-formation game model, our dynamic coalition formation approach can adaptively select best-effort workers, i.e., edge nodes, behind the merge-and-split rules. Therefore, we can get a Pareto-optimal solution for edge nodes based on the current CDC system condition.

Fig.4 depicts the fairness among edge nodes in the CDC platform. Simulation results clearly indicate the superiority of our proposed scheme about the fairness issue. This is mainly because the feature of cooperative game theory. Classically, the main challenge of cooperative game solutions is to ensure the fairness among game players while generating maximum efficiency. This feature is directly implied in our proposed method. Moreover, this policy can deal with the dynamics and uncertainty of the CDC network environment. Therefore, the $\mathfrak{M}_{\mathcal{E}}$ of each edge node is shared in a fair-efficient manner. Therefore, we can attain an excellent edge node fairness while effectively handle dynamic AIoT devices' requests.

From the simulation results in Fig.2 to Fig.4, we can see that our hybrid cooperative game approach can capture dynamic interactions among AIoT devices and edge nodes to achieve a mutually desirable solution. By employing a coordination paradigm, we reciprocally combine two different cooperative game concepts; the coalition formation game and bargaining game. Therefore, the synergy effect can be obtained while converging to a fair-efficient solution through the negotiative interactions.

VI. SUMMARY AND CONCLUSION

This study has developed a highly-effective approach to jointly optimize the MDS code and edge node selection, thereby significantly enhancing the efficiency of CDC system in the AIoT paradigm. With cooperative game ideas, the CDC control problem is modeled as a coalition formation game and a resource bargaining game between AIoT devices and edge nodes. Particularly, the most adaptable edge nodes are selected through the coalition formation process, and they effectively operate the CDC process to mitigate the straggler effects in the distributed computing network. To satisfy this goal, the computing resource of each edge node is adaptively shared based on the concept of *WNSWS*. To achieve a mutually desirable solution, individual AIoT devices and edge nodes work together and act cooperatively with each other. By employing a coordination paradigm, our approach can strike an appropriate performance balance under dynamically changing CDC system environments. Therefore, we can achieve a 'win-win' solution to ensure, i) system throughput maximization, ii) computation latency minimization, and, iii) fairness provisioning among edge nodes. Finally, our proposed scheme is simulated and analyzed with comparing to that of the state-of-the-art protocols. Simulation results clearly indicate that our method is suitable for the decentralized CDC network infrastructure than other existing *CERA*, *OCNS* and *ODRA*schemes.

From a future-oriented perspective, a hierarchical game-theoretic CDC framework can be developed for the metaverse services, especially for vehicular metaverse. Therefore, we will minimize the idle resources from vehicles to handle intensive computation tasks. In addition, we will reformulate the CDC control problem as a Markov decision process, and design a novel deep reinforcement learning algorithm. The reason is that Markov decision process can find the best set of edge nodes for different learning tasks with edge nodes' straggling parameters. Furthermore, blockchain technology can be explored to improve our proposed scheme. Usually, the information related to AIoT devices is strictly private and confidential. Therefore, we should guarantee AIoT devices' security and privacy.

COMPETING OF INTERESTS

The author Sungwook Kim declares that there are no competing interests regarding the publication of this article.

AUTHOR CONTRIBUTION

The author is a sole author of this work and ES, participated in the design of the study and performed the statistical analysis.

REFERENCES

[1] A. Ghosh, D. Chakraborty, and A. Law, "Artificial intelligence in Internet of Things," *CAAI Trans. Intell. Tech.*, vol. 3, no. 4, pp. 208–218, Dec. 2018.

[2] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13849–13875, Sep. 2021.

[3] L. Jia, Z. Zhou, F. Xu, and H. Jin, "Cost-efficient continuous edge learning for artificial intelligence of things," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7325–7337, May 2022.

[4] C. T. Nguyen, D. N. Nguyen, D. T. Hoang, H.-A. Pham, and E. Dutkiewicz, "Optimize coding and node selection for coded distributed computing over wireless edge networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2022, pp. 1–6.

[5] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Joint coding and scheduling optimization for distributed learning over wireless edge networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 2, pp. 484–498, Feb. 2022.

[6] S. Zhao, "A node-selection-based sub-task assignment method for coded edge computing," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 797–801, May 2019.

[7] C. T. Nguyen, D. N. Nguyen, D. T. Hoang, K. T. Phan, D. Niyato, H.-A. Pham, and E. Dutkiewicz, "Elastic resource allocation for coded distributed computing over heterogeneous wireless edge networks," *IEEE Trans. Wireless Commun.*, early access, 2022, doi: 10.1109/TWC.2022.3213256.

[8] B. T. Tinh, L. D. Nguyen, H. H. Kha, and T. Q. Duong, "Practical optimization and game theory for 6G ultra-dense networks: Overview and research challenges," *IEEE Access*, vol. 10, pp. 13311–13328, 2022.

[9] N. Kim, D. Kim, J. Lee, D. Niyato, and J. K. Choi, "Incentive-based coded distributed computing management for latency reduction in IoT services—A game theoretic approach," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8259–8278, May 2021.

[10] Q. Yu, S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "How to optimally allocate resources for coded distributed computing?" in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

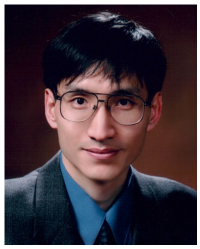
[11] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[12] S. Kim, *Game Theory Applications in Network Design*. Hershey, PA, USA: IGI Global, 2014.

[13] H. Charkhgard, K. Keshanian, R. Esmaeilbeigi, and P. Charkhgard, "The magic of nash social welfare in optimization: Do not sum, just multiply!" *ANZIAM J.*, vol. 64, pp. 119–134, Aug. 2022.

[14] P. G. Saghand and H. Charkhgard, "A cooperative game solution approach for intensity modulated radiation therapy design: Nash Social Welfare optimization," *Phys. Med. Biol.*, vol. 66, no. 7, 2021, Art. no. 075011.

[15] A. Bogomolnaia and M. O. Jackson, "The stability of hedonic coalition structures," *Games Econ. Behavior*, vol. 38, no. 2, pp. 201–230, 2002.



SUNGWOOK KIM received the B.S. and M.S. degrees in computer science from Sogang University, Seoul, Republic of Korea, in 1993 and 1995, respectively, and the Ph.D. degree in computer science from Syracuse University, Syracuse, NY, USA, in 2003, under the supervision of Prof. Pramod K. Varshney. He has held Faculty Member position with the Department of Computer Science, Chung-Ang University, Seoul. In 2006, he returned to Sogang University, where he is currently a Professor with the Department of Computer Science and Engineering and also the Research Director of the Network Research Laboratory. His research interests include resource management, online algorithms, adaptive quality-of-service control, and game theory for network design.