A Joint Offloading and Energy Cooperation Scheme for Edge Computing Networks

Jieyi Zhang[1], Biling Zhang[1], Jiahua Liu[1], Zhu Han[2,3]
1. School of Network Education, Beijing University of Posts and Telecommunications, P. R. China
2. Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA
3. Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701.

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9839191

***Abstract*—For edge computing (EC) network, one critical problem is how to process computation-intensive task in time with efficient energy usage. However, existing works mainly study one aspect of the problem by computation offloading or energy cooperation. Considering that the computation offloading strategy and the energy cooperation strategy are coupled with each other, in this paper, we propose a new information-energy collaboration model for the EC network powered by renewable energy and stored energy. In this new model, an EC node can collaborate with the other EC nodes and the cloud for computation offloading. At the same time, the EC is also able to store energy and share energy with other EC nodes. In such a case, since the EC nodes can have a stable power supply to finish the computing tasks within the latency limit, we formulate the problem of deriving the information-energy collaboration strategy as the optimization problem of minimizing the cloud computing cost and the power purchase cost. To find the optimal offloading and energy cooperation strategy, we first analyze the sixteen cases of the offloading strategy depending on the computing tasks and the renewable energy. Then we further summarize four cases of energy collaboration strategy according to the different offloading strategies. To derive the collaboration strategy with low complexity, we propose the practical Hybrid Greedy Iterative Algorithm (HGIA) to the optimization problem. Finally, the simulation results demonstrate that our approach is effective and stable.**

***Index Terms*—Smart grid communication, computing offload, energy collaboration, renewable energy, greedy strategy.**

I. Introduction

With the development of information and communication technology, the demand for processing computation-intensive tasks is overgrowing. The conflict between the intensive task and scarce resources has become increasingly significant due to a single device's limited computing power and energy. A practical solution is to offload the computing task to a nearby place with more powerful computing power and sufficient energy supply. For example, [1], [2] proposed a game-based strategy for offloading computational tasks from the mobile to the edge network, which improved the performance of the system. [3] considered the task allocation between the edge nodes and the cloud as well as the users, and proposed a multi-layer data flow processing system to achieve a reasonable utilization of the computational power of the cloud-edge-end network. However, the above article did not consider energy collaboration. The performance of edge computing (EC) networks can be affected by the energy supply. If there is not enough energy support, the offloading strategy is challenging to implement.

In order to efficiently utilize energy, energy collaboration among different nodes is very important. Especially when some renewable energy sources, such as wind or solar energy, are used for the edge systems. For example, [4] proposed a heuristic strategy to achieve the coordination of electric vehicle charging and wind power generation in microgrids. For the nodes with communication and computing capabilities such as base station (BS), [5] achieved stable energy collaboration by proposing a linear program-based energy collaboration scheme. While [6] achieved the improvement by the iterative algorithm-based energy utilization efficiency of energy collaboration. However, the above article does not consider the upper-level applications. If the energy demand does not consider the demand of upper-layer applications (such as computing offloading), the system performance cannot reach the optimal level.

A few works have been done by jointly consider computing offloading and energy cooperation [7]-[11]. Based on the smart grid and coordinated multipoint (COMP) energy collaboration model, joint optimization of the energy and power of collaborative BS was proposed by [7] to maximize the throughput of the system. While [11] proposed a BS-centric deployment strategy for the edge computing networks based on centralized and distributed approaches to achieve continuous and low-cost computing and energy services. However, the previous work has not considered inter-EC energy collaboration but considered mostly edge collaboration with the cloud for computing, which leads to the underutilization of resources of the edge.

Therefore, in this paper, we mainly consider the information and energy collaboration among EC nodes, besides the collaboration between EC nodes and the cloud and traditional grid. We propose a new edge network collaboration model, allowing the EC nodes with renewable energy capabilities to provide efficient computing and energy services. Due to the limited computing power of a single EC, some tasks can be offloaded to other nodes or the cloud to complete. Moreover, when renewable energy and storage power are insufficient, electricity needs to be transferred from the neighbor node purchased from the grid to make up. We propose a joint optimization problem, in which the computing cost of offloading edge network calculations to the cloud and the cost of traditional power grid purchase are jointly optimized. In order to find the optimal offloading and energy cooperation strategy, we first analyzed sixteen scenarios of offloading strategies based on computing tasks and renewable energy. Then we further summarized four energy synergy strategy cases based on different offloading strategies. A hybrid greedy iterative algorithm is proposed to solve the problems, which minimize the cloud-computing resource consumption and power grid energy consumption, and meet the service requirements.

II. System Model

A. Network Model

Fig. 1 shows an edge computing network composed of multiple edge nodes. Each EC node is a BS which equipment with an edge server to provide computing and communication services for mobile terminal equipment. Due to limited computing power, computing offloading can be performed between edge nodes or from the edge to the cloud servers. BSs are powered by renewable energy, and the excess renewable energy can be stored in the energy storage device. When the renewable energy is insufficient, it can be powered by the energy storage device or the traditional power grid. The aggregator here collects information on the computing requirements and electricity demand from multiple BSs, and exchanges the information with the cloud and conducts energy cooperation with the power grid.

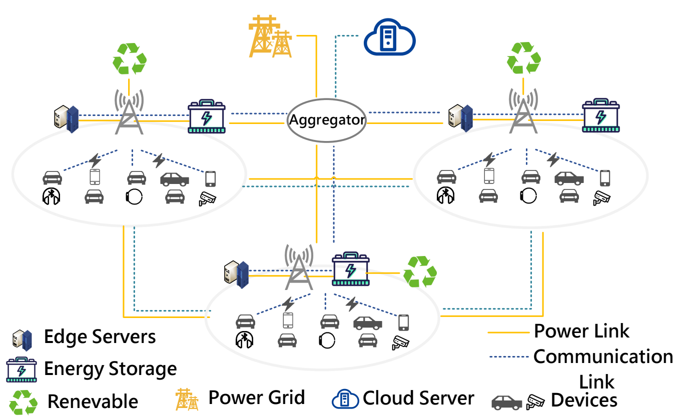

Fig. 1: Information-Energy collaboration in the edge computing network

For the convenience of analysis, as in Fig. 2, we will discuss the collaboration between two edge nodes, i.e., BS_i and BS_j, and let C represent the cloud. In the following, we will formulate the computation offloading model, the energy consumption model and the collaboration model in section II.B, II.C and II.D. Respectively, from the perspective of BS_i. Nonetheless, we can have the aforementioned models for BS_j in a similar way.

B. Computation Offloading Model

The time is divided into execution time slots. The total computational tasks arrive at BS_i at each time slot $t(1 \leq t \leq N)$ are denoted as $o_i(t)$. Base stations can perform computation tasks by local computation and computation offloading. Thus, the computation task can be divided into three parts: the amount of data to be computed locally by BS_i, (i.e., $o_i^l(t)$), the task offload to BS_j, (i.e., $o_i^{ij}(t)$), and the task offload to the cloud server (i.e., $o_i^{ic}(t)$). Hence we will have

$$o_i(t) = o_i^l(t) + o_i^{ij}(t) + o_i^{ic}(t). \tag{1}$$

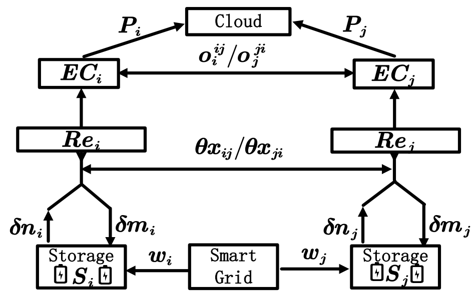

Fig. 2: Schematic diagram of the energy flow of the BS.

When the computing task at slot t are performed locally, the calculation time is

$$\tau_i^l(t) = \frac{o_i^l}{v_i}, \tag{2}$$

where v_i is the processing speed of BS_i.

When the computing offloading occurs, computing tasks can be offloaded between edge servers or between edge servers and cloud servers. Assume orthogonal frequency spectrum is used for communication between BSs. Therefore, the transmission data rate between BS_i and BS_j or BS_i and cloud can be written as

$$r_{ix} = B_{ix} \log_2 \left(1 + \frac{p_{ix} g_{ix}}{\sigma_{ix}^2}\right). \tag{3}$$

Here, $x \in (j, C)$, B_{ix} is the system bandwidth between BS_i and BS_j or cloud, p_{ix} is the transmission power when transferring data, and g_{ix} is the channel power gain, and σ_{ix}^2 is the power of additive Gaussian white noise at the receiver. Then the offloading time can be calculated as

$$\tau_i^{ix}(t) = \frac{o_i^{ix}(t)}{r_{ix}} = \frac{o_i^{ix}(t)}{B_{ix} \log_2 \left(1 + \frac{p_{ix} g_{ix}}{\sigma_{ix}^2}\right)}. \tag{4}$$

And the calculation time at x can be calculated as

$$\tau_i^x(t) = \frac{o_i^{ix}}{v_x}, \tag{5}$$

where v_x denotes the computation rate of the server.

Then the total time delay for the computing task o_i is

$$\tau_i(t) = \max \left(\tau_i^l(t), \tau_i^{ij}(t) + \tau_i^j(t), \tau_i^{ic}(t) + \tau_i^c(t)\right). \tag{6}$$

To ensure the service requirements, a constraint is added to the total time delay τ_i for BS_i, i.e.,

$$0 \leq \tau_i \leq \tau_{\max}, \tag{7}$$

where $\tau_{\max}$ is the maximum delay allowed for the execution of this computational task.

C. Energy Consumption Model

For the BS, the energy consumption results from the local computation and offloading processes. The energy consumption of local computation can be expressed as

$$e_i^l(t) = \kappa \left(v_i\right)^2 o_i^l(t), \tag{8}$$

where κ is an effective switching capacitor for servers. While the energy consumption in the calculated offloading case is mainly caused by the offloading process to the neighboring BSs or cloud which can be calculated as

$$e_i^{ix}(t) = p_{ix}\tau_i^{ix}(t) = \frac{p_{ix}o_i^{ix}(t)}{B_{ix}\log_2\left(1+\frac{p_{ix}g_{ix}}{\sigma_{ix}{}^2}\right)}, \tag{9}$$

where p_{ix} is the transmitted power from BS_i to BS_j or BS_i to the cloud when offloading.

In addition, the computational energy consumption of BS_i to help the neighboring BS with computational tasks can be given as

$$e_i^h(t) = \kappa\left(v_i\right)^2 o_j^{ji}(t), \tag{10}$$

where $o_j^{ji}(t)$ is the amount of task offloading from BS_j.

Consequently, the total energy consumed by BS_i during time slot t is

$$Ce_i(t) = e_i^l(t) + e_i^{ij}(t) + e_i^{ic}(t) + e_i^h(t). \tag{11}$$

It can be seen from the above formula that when BS_i does not offload to BS_j, $e_i^{ij}(t)=0$; when BS_i does not offload to the cloud, $e_i^{ic}(t)=0$. When BS_j does not offload to BS_i, $e_i^h(t)=0$.

D. Energy Collaboration Model.

Assume that BS_i generates renewable energy in time t is $Re_i(t)$. It is usually a stochastic process that represents the variability of the renewable energy produced from onetime step to another. Then the net energy generated by BS_i in time can be given as

$$Le_i(t) = Re_i(t) - Ce_i(t). \tag{12}$$

Note that $Le_i(t)$ can be positive or negative, indicating a surplus or a deficit.

Let the energy stored in BS_i at time t is $s_i(t)\geq 0$. To model the finite storage constraint, we further assume that $s_i(t)\leq S_{max}$. We assume that $s_i(1)=0$. That is, there is no energy at the initial time. Morever, let $m_i(t)$ and $n_i(t)$ be the energy BS_i charged and discharged from the memory at time t, and there is $n_i(t)\leq s_i(t)$. Intuitively, for a given BS_i and time t, at least one of $m_i(t)$ and $n_i(t)$ is zero, i.e., $m_i(t)\cdot n_i(t)=0$. In order to keep their energies neutrality at each time t. The following storage dynamics equations is required to be satisfied

$$s_i(t+1) = s_i(t) + \delta m_i(t) - n_i(t). \tag{13}$$

Here, $0\leq\delta\leq 1$ denotes the energy transfer efficiency of energy access in storage.

The energy transfer from BS_i (or BS_j) to BS_j (or BS_i) is $x_{ij}(t)\geq 0$ (or $x_{ji}(t)\geq 0$). For a given time t, at least one of the $x_{ij}(t)$ and $x_{ji}(t)$ is 0, i.e., $x_{ij}(t)\cdot x_{ji}(t)=0$.

BS_i energy obtained from conventional energy sources at time t is: $w_i(t)\geq 0$.

Thus, it can be formulated as

$$w_i(t) = m_i(t) - Le_i(t) - \delta n_i(t) + x_{ij}(t) - \theta x_{ji}(t), \tag{14}$$

Here, $0\leq\theta\leq 1$ denotes the efficiency of energy transfer between BSs. (14) indicate the energy balance requirements of the BS_i.

III. Task Offloading and Energy Allocation Scheme

A. Problem Formulation

Let $\mathbf{s}(t)=[s_i(t),s_j(t)]^T$ be the energy storage state of the system at time t. Similarly, let

$$\mathbf{m}(t) = [m_i(t), m_j(t)]^T, \tag{15}$$

$$\mathbf{n}(t) = [n_i(t), n_j(t)]^T, and \tag{16}$$

$$\mathbf{o}(t) = [\{o_i^l(t), o_i^{ij}(t), o_i^{iC}(t)\}, \{o_j^l(t), o_j^{ji}(t), o_j^{jC}(t)\}]^T. \tag{17}$$

Then the control strategy π is the sequence of the above control variables, i.e.,

$$\pi(t) = \{(\mathbf{o}(t), \mathbf{m}(t), \mathbf{n}(t), x_{ij}(t), x_{ji}(t)), 1\leq t\leq N\} \tag{18}$$

Given π, the conventional energy BS_i obtain from grid

$$\begin{aligned} w_i(\pi(t)) &= m_i(t) - \delta n_i(t) + x_{ij}(t) - \theta x_{ji}(t) + Re_i \\ &- \left(\kappa\left(v_i\right)^2\left(o_i^{ij}(t) + o_j^{ji}(t)\right) + \frac{p_{ij}o_i^{ij}(t)}{B_{ij}\log_2\left(1+\frac{p_{ij}g_{ij}}{\sigma_{ij}^2}\right)}\right. \\ &+ \left.\frac{p_{iC}(t)}{B_{iC}\log_2(1+\frac{p_{iC}g_{iC}}{\sigma_{iC}^2})}\right). \end{aligned} \tag{19}$$

Then the total electricity of the EC network purchased from the grid is

$$H = w_i(\pi(t)) + w_j(\pi(t)), \tag{20}$$

and the total cost of this electricity is λH, where λ denotes the grid energy pricing factor.

Since the cloud does not provide computational offload services to the BS at no cost, when the BS_i uses of the cloud computing services, it should pay to the cloud

$$P_i(t) = \gamma\kappa(v_c)^2 o_i^{ic}(t), \tag{21}$$

where γ is the payoff factor, considering the cost of computing in the cloud is proportional to the energy consumption of data processing.

Then the cost of the EC network when using computational offload services in the cloud can be given as

$$P_i(\pi(t)) + P_j(\pi(t)) = \gamma\kappa\left(v_C\right)^2\left(o_i^{iC}(t) + o_j^{jC}(t)\right). \tag{22}$$

Consequently, the optimal policy to minimize the ECs network cost, i.e., the cost of purchasing conventional energy plus the cost of providing computational offload services in the cloud, is formulated as

$$\min_{\pi(t)}\sum_{t=1}^{N}\left(\lambda\left(w_i(\pi(t)) + w_j(\pi(t))\right) + P_i(\pi(t)) + P_j(\pi(t))\right) \tag{23}$$

s.t.

$$\begin{gathered} \mathbf{n}(t)\leq\mathbf{s}(t), \\ m_i(t)\cdot n_i(t) = 0, \\ [0,0]^T\leq\mathbf{s}(t)\leq[S_{\max}, S_{\max}]^T, \\ 0<o_i^l(t)<o_i(t), \\ x_{ij}(t)\cdot x_{ji}(t) = 0, \\ w_i(t)\geq 0, \\ 0\leq\tau_i\leq\tau_{\max}. \end{gathered}$$

B. Hybrid greedy iterative algorithm with random distribution of energy and computational tasks.

From (19), we can see that the proposed problem is a multi-parameter linear programming problem. Moreover, the cloud computing offloading strategy and energy cooperation strategy are coupled through Le_i, i.e., changes in the calculation offloading strategy will directly affect the energy collaboration strategy. In this case, solving through traditional algorithms will increase the complexity and calculation time, which is very unfriendly to continuous short-slot tasks. Greedy online planning is equivalent to the following algorithm, which can be understood as accumulating the calculation results of N time slots. Each time slot is independent of each other. Although the energy storage of the previous time slot will be used in the next time slot, the algorithm equivalent strategy will not change, and each time slot can select a strategy according to its computing task. In the following discussions, all are discussed between a time slot, so the effect of time t is not considered for the time being, and the result is universal in any time slot.

The greedy energy scheme that uses all available energy in the energy storage to find the best offloading strategy is an effective method for the information energy and collaboration. In the proposed algorithm, we first discuss the offloading strategy in different RE situations in Subsection 1, and then discuss the energy coordination strategy under the given offloading strategy in Subsection 2.

Proposition 1. For the offloading between ECs, if more tasks are offloaded to the party with more Re, the total cost can be reduced.

Proof: There is supposed BS_i computes data volume o. When it has insufficient Re, offload o to the neighboring EC_j with more Re, and the transmission energy consumption is $e_i^{ij} = p_{ij}\tau_i^{ij}$, j helps i to compute energy consumption as $e_j^h = \kappa(v_j)^2 o$, the total energy consumption is $e_i^{ij} + e_j^h$. If i does not offload, it needs j to support its computational task by transmitting excess power to i through transmission lines by energy collaboration, and at this time BS_i consumes energy locally as $e_i^l = \kappa(v_i)^2 o$, BS_j transmits the power to BS_i as $\frac{e_i^l}{\beta}$, due to the small difference in computing power between EC $v_i \approx v_j$, so $e_i^l \approx e_j^h$, and the actual transmission energy consumption is very small $e_i^{ij} \ll e_j^h$, so $\frac{e_i^l}{\beta} > e_i^{ij} + e_j^h$. So offloading more tasks to the BS with more Re remaining computation is beneficial to reduce the total cost. ■

1) Computation offloading strategy

Let $\tau_i' = \frac{o_i}{V_i}$, $\tau_j' = \frac{o_j}{V_j}$ denote the latency of the computing task process complete locally. $\tau_{\text{imax}}, \tau_{\text{jmax}}$ is the maximum time delay allowed for o_i, o_j. $\tau_{imax} \geq \tau_i'$ means that i has the ability to finish the task alone under the latency constraint. Let $Le_i' = Re_i - Ce_i'$, where Re_i is the renewable energy generated by i, Ce_i' is the energy consumpted task o_i that is completed by BS_i under the latency constraint. Thus, Ce_i' can be formulated as

$$Ce_i' = \min\left\{\kappa(v_i)^3 \tau_{imax}, \kappa(v_i)^3 \tau_i'\right\}, \tag{24}$$

and Ce_j' is expressed in the same way as Ce_i'.

According to the relationship between τ_i' and $\tau_{i\max}$, the positive and negative of Le_i' can be discussed in various cases. Here we will analyze one of the cases and summarize the conclusion into Lemma 1. The other fifteen cases will be omitted here due to page limit.

Lemma 1. For any o_k, τ_{kmax} and Le_j', when $\tau_{i\max} \geq \tau_i', \tau_{j\max} \geq \tau_j'$, and $Le_i' < 0, Le_j' \geq 0$, there is $\tau_{i\max} \geq \tau_j$, the offload strategy is

$$\begin{cases} o_i^{ij} = \min\left\{\frac{Re_j}{K(V_j)^2} - o_j, (\tau_{i\max} - \tau_j)\cdot V_j, o_i - \frac{Re_i}{K(V_i)^2}\right\} \\ o_i^l = o_i - o_i^{ij} \\ o_j^l = o_j \end{cases} \tag{25}$$

when $\tau_{i\max} < \tau_j$, $o_i^l = o_i, o_j^l = o_j$.

Proof: In this case, both BS_i and BS_j have the ability to complete all calculation tasks independently. According to Proposition 1, after consuming Re_i, use BS_j to calculate, and consume as much Re_j as possible, that is, to split the data to more Re side.

When $\tau_{i\max} \geq \tau_j$, BS_j can accept the offloading of BS_i. There are three uninstall strategies, the minimum value of the three is o_i^{ij}:

① EC_i helps j calculate until Re_j is used up, from this, it can be follow as $Re_j - K(V_j)^2 o_j = K(V_j)^2 o_i^{ij'}$, simplifying to get $o_i^{ij'} = \frac{Re_j}{K(V_j)^2} - o_j$.

② Regardless of Re, before the maximum delay limit of the o_i calculation task is reached, the amount of tasks that EC_j can help EC_i calculation can be written as $o_i^{ij''} = (\tau_{i\max} - \tau_j)\cdot V_j$.

③ If the Re_i is consumed, all the remaining calculation tasks are completed at EC_j. At this time, the carrying capacity from EC_i to EC_j is $o_i^{ij'''} = o_i - \frac{Re_i}{K(V_i)^2}$.

When $\tau_{i\max} < \tau_j$, EC_j can't help EC_i perform the calculation, all calculation tasks are done locally. ■

From section II, we can obtain $Ce(\boldsymbol{o}(\boldsymbol{t}))$. After completing the analysis of the computational offloading strategy, we iterate the obtained energy consumption into Eq: $Le_i = Re_i - Ce_i$,$Le_j = Re_j - Ce_j$.

2). Energy collaboration strategy

In this Subsection, we discuss the issue of energy cooperation under the given offload strategy. According to Le_i, we can determine the direction of power sharing and the change of stored energy, and obtain the strategy of energy cooperation. The net remaining energy of the BS can be obtained. Then, based on the energy model, in Fig.2, we propose a greedy algorithm similar to [5] for the best energy collaboration stratage as follows.

Here we will analyze one of the cases and summarize the conclusion into Lemma 2. The other three cases will be omitted here due to page limit.

Lemma 2. According to the result of **1)**, if $Le_i \geq 0, Le_j \geq 0$, then $W_k = 0$ $(k \in (i,j))$,

$$m_k = \min\left\{\frac{(S_{\max} - s_k), Le_k}{\delta}\right\}, \tag{26}$$

and when $s_i = S_{max}, s_j = S_{max}$ or $s_i, s_j < S_{max}$ or $s_j < S_{max},\ s_i = S_{max}$, there is

$$s_k \leftarrow s_k + m_k, \tag{27}$$

and when $s_i < S_{max},\ s_j = S_{max}$, there is

$$s_i \leftarrow s_i + \min\left((S_{max} - s_i - m_i)\,,\ \theta\left(Le_i - \frac{m_i}{\delta}\right)\right). \quad (28)$$

Proof: In this case, in accordance with the principle of maximum storage, the remaining energy is stored locally first, i.e., $m_k = \frac{Le_k}{\delta}$, and if the local energy storage is full, i.e., $Le_k \geq S_{max} - s_k$, the excess energy will be transferred to the storage of another BS as $x_{ji} = Le_j - m_j$. The storage capacity of j, i.e., $m'_j = \min\left\{\frac{\theta x_{ij}, (S_{\max} - s_j)}{\delta}\right\}$, according to the (13) we can get s_i. ■

3) Hybrid greedy iterative algorithm

Based on the analysis of subsection **1)** and **2)**, a hybrid greedy iterative algorithm is proposed for the solution to the problem in (23).

Algorithm 1 Hybrid greedy iterative algorithm (HGIA)

Input: $o_i(t), o_j(t), Re_i(t), Re_j(t), \tau_{imax}(t), \tau_{jmax}(t)$
Output: $W, \pi(t), s(t), t = \{1, \ldots, N\}$
 Initialize: $s_i(1) = s_j(1) = 0$
 for t=1:N **do**
  Compare $(\tau_{i\max}, \tau_{j\max}, Le'_i, Le'_j)$ with $(\tau'_i, \tau'_j, 0, 0)$ and select the lemma in Subsection 1)
  Calculate $Le_i(t), Le_j(t)$ according to the corresponding equations in the **Lemma** 1. and the other fifteen cases.
  Compare $(Le_i(t), Le_j(t))$ with $(0, 0)$ and select the lemma in Subsection 2)
  Calculate $W(t)$ according to the corresponding equations in the **Lemma** 2. and the other three cases.
 end for
 return $W = \sum_{t=1}^{N} W(t), \pi(t), s(t), t = \{1, \ldots, N\}$

From Algorithm 1, we can see that in each time slot, HGIA first judges the size of the calculation task, the local calculation amount within the maximum delay, and whether the renewable energy meets the local calculation consumption. Based on the judgments, HGIA gets the calculation offloading strategy and obtains the $Le_i(t)$ and $Le_j(t)$. Then it selects the energy cooperation strategy according to the size of the $Le_i(t)$ and $Le_j(t)$. The last remaining stored energy is used as the initial storage for the next time slot.

IV. NUMERICAL RESULTS

In this section, we verify the algorithm's ability. Where the total amount of tasks o_i received by each device is a randomly generated amount from 10 to 100 Mbit. The total output renewable energy power RE_i is a random amount between $2 \sim 20$ kWh. The duration of each time slot is set to 1 second. For the cost parameter from the cloud side, we set $\gamma = 0.2$, and the cost factor at the grid side is $\lambda = 0.8$. Set the data processing speed of EC to 200 Mbits/s, 250 Mbits/s, $1,000$ Mbits/s, respectively, the data transmission power $p_{ij} = 1$ kW. The signal-to-noise ratio of the transmitted data channel $\sigma^2 = 10^{-9}$ W. Energy transfer efficiency of storage process is set to $\delta = 0.8$, and the energy transfer efficiency between BSs is set to $\theta = 0.7$.

In order to further understand the advantages of HGIA and prove the applicability of the algorithm to the computational offloading and energy collaboration joint model, we developed two variants of the HGIA energy collaboration strategy with the same energy coordination strategy but different computational offloading strategies. The Cloud Priority Offloading Algorithm (CPOA) prioritizes computing. When uninstalling the cloud, the Hybrid Offloading Algorithm (HOA) is the priority Offloading between ECs. From Fig. 3, we can see that HGIA has a lower cost. CPOA generates many cloud computing costs in varied situations. Although HOA reduces the cost of cloud computing, it does not consider the transmission loss caused by energy synergy. According to the amount of renewable energy produced, HGIA with greedy offloading is adopted, which reduces the cost of energy coordination loss at the offloading calculation point. Therefore, for offloading between ECs, if more tasks are offloaded to the party with more Re, the total cost can be reduced.

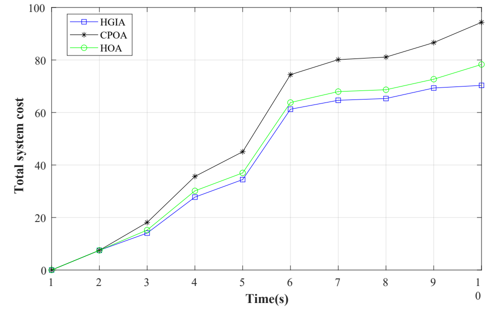

Fig. 3: Cost comparison under different offloading methods.

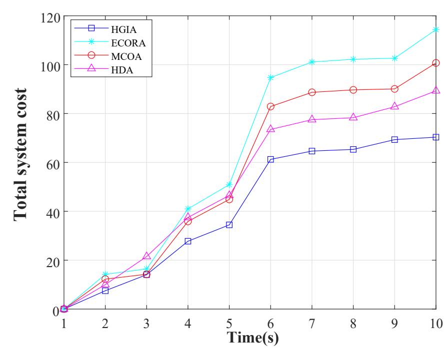

Fig. 4: System cost with different resource allocation schemes over time.

Fig. 4 represents the comparison of mobile edge network computing costs between the different schemes. The offloading calculation algorithm based on matching (MCOA) performs calculation offloading by matching the calculation task delay and the amount of calculation. Energy-saving computing offloading and resource allocation (ECORA) aims to minimize computing energy consumption for computing offloading. The hybrid distributed algorithm (HDA) minimizes electricity purchase costs through a distributed approach.

The results show that HGIA's leads to a lower system cost compared to multiple schemes. Compared with HGIA, MCOA and ECORA lack the cooperation of energy and the application of energy storage. HGIA not only reduces cloud costs by maximizing the computing power of the edge network but considers the storage and use of renewable energy and the collaboration between systems in terms of energy supply.

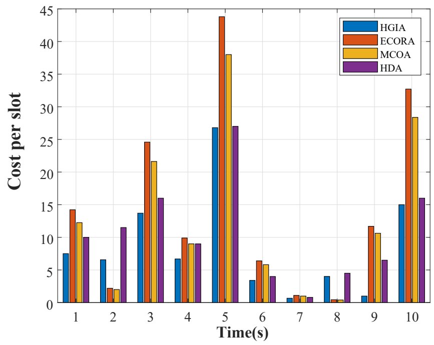

Fig. 5: System cost of single time slot mobile edge network.

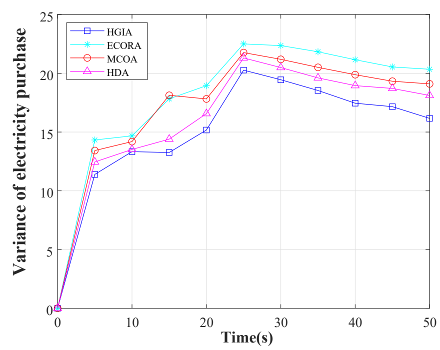

Fig. 6: Variance of power demand from the grid.

Compared with the overall analysis of HGIA, HAD obtains the optimal solution for a single BS and does not achieve the maximum storage energy, so HAD obtains only the overall sub-optimal solution.

Fig. 5 represents the computational cost comparison of the offline mobile edge network for a single time slot. We also note that the cost of HGIA in each time slot is relatively low because the proposed scheme is not only able to schedule the computational operations of the EC, and the power transfer behavior between energy stores according to the system conditions in each time slot, but also able to maximize the energy storage, since the stored energy from the previous time slot will become one of the energy supplies for the next time slot. Compared to other schemes, HGIA ensures the minimization of cost for a single time slot while maximizing the stored energy, using the stored energy to reduce the cost of power purchase for the next time slot, and achieving the minimization of cost in multiple consecutive time slots.

Fig. 6 represents the variance of power demand from the grid over time. We can see that the variance of HGIA is significantly smaller compared to the other schemes. HGIA's energy storage scheme can store the excess renewable energy and do not supply energy for the next time slot BS work, which can reduce the power purchase of the grid and provide it with a more stable energy supply. In contrast, the other three methods have poor energy collaboration, so they cannot collaborate and use of energy in the energy storage of the two systems.

V. CONCLUSIONS

In this paper, we propose a new information energy cooperation model for EC networks driven by renewable energy and energy storage. In the proposed model, EC nodes can cooperate with other EC nodes to perform computational offloading. At the same time, EC can store energy and share energy with other EC nodes. In this case, the EC node can have stable power to complete the calculation task within the delay limit, thereby minimizing the total cost. In order to find the optimal offloading and energy cooperation strategy, we first analyzed sixteen scenarios based on computing tasks and renewable energy offloading strategies. Then we further summarized four energy synergy cases based on the different offloading strategies and then proposed a low-complexity and practical HGIA algorithm. Finally, the simulation results prove that our method is effective and stable in stochastic computing tasks and stochastic renewable energy supply environments.

VI. ACKNOWLEDGMENT

This work is partially supported by National Natural Science Foundation of China (No. 62171060), NSF CNS-2128368, CNS-2107216, Toyota and Amazon.

REFERENCES

[1] Y. Gu, Z. Chang, M. Pan, L. Song and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7475-7484, Aug. 2018.

[2] Y. Hui, Z. Su, T. H. Luan, C. Li, G. Mao and W. Wu, "A Game Theoretic Scheme for Collaborative Vehicular Task Offloading in 5G HetNets," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16044-16056, Dec. 2020.

[3] P. Wang, C. Yao, Z. Zheng, G. Sun and L. Song, "Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2872-2884, April 2019.

[4] Y. Yang, Q. Jia, G. Deconinck, X. Guan, Z. Qiu and Z. Hu, "Distributed Coordination of EV Charging with Renewable Energy in a Microgrid of Buildings," in *Proc. IEEE Power & Energy Soc. General Meeting (PESGM)*, Atlanta, GA, USA, 2019, pp. 1-1.

[5] Y. Chia, S. Sun and R. Zhang, "Energy Cooperation in Cellular Networks with Renewable Powered Base Stations," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 6996-7010, Dec. 2014.

[6] F. K. Ojo, D. O. Akande and M. F. M. Salleh, "Optimal Power Allocation in Cooperative Networks With Energy-Saving Protocols," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5079-5088, May 2020.

[7] J. Xu and R. Zhang, "CoMP Meets Smart Grid: A New Communication and Energy Cooperation Paradigm," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2476-2488, June 2015.

[8] F. Wang and X. Zhang, "Dynamic Computation Offloading and Resource Allocation over Mobile Edge Computing Networks with Energy Harvesting Capability," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1-6.

[9] Y. Liu, C. Yang, L. Jiang, S. Xie and Y. Zhang, "Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111-117, March/April 2019.

[10] J. Zhang, J. Du, C. Jiang, Y. Shen and J. Wang, "Computation Offloading in Energy Harvesting Systems via Continuous Deep Reinforcement Learning," in *Proc. ICC 2020 - 2020 IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, 2020, pp. 1-6.

[11] Y. Liu, S. Xie, Q. Yang and Y. Zhang, "Joint Computation Offloading and Demand Response Management in Mobile Edge Network With Renewable Energy Sources," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15720-15730, Dec. 2020.