Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1377225, 8 pages
https://doi.org/10.1155/2022/1377225

Research Article

Resource Optimization in MEC-Assisted Multirobot Cooperation Systems

Lanxin Qiu,[1,2] **Yi Zhang**,[1] **Yanbo Wang**,[1] **Yineng Shen**,[2] **Jiantao Yuan**,[3] and **Rui Yin**[3]

[1]*State Grid Zhejiang Electric Power Company Information & Telecommunication Branch, Hangzhou 310000, China*
[2]*Dept. of Information and Electrical Engineering, Zhejiang University, Hangzhou 310000, China*
[3]*Dept. of Information and Electrical Engineering, Zhejiang University City College, Hangzhou 310000, China*

Correspondence should be addressed to Yineng Shen; shenyineng@zju.edu.cn

Received 10 January 2022; Accepted 11 April 2022; Published 9 May 2022

Academic Editor: Carles Gomez

With prevalent utilization of *multirobot cooperation* (MRC) systems, people pay more attention to improve the system performance. Among them, the energy consumption and implementation latency of MRC systems are major concerns, and *mobile edge computing* (MEC) provides a potential way to solve these problems. Therefore, how to leverage MEC to get the balance between computing and communication consumption in MRC systems needs to be investigated urgently. In this paper, a MRC system deployed to accomplish multiple time-critical tasks by MEC technology is studied. The proposed MRC system includes a powerful *master robot* (MR) and several *slave robots* (SRs). As a scheduler, MR is responsible for allocating tasks to SRs and has more computing power. SRs are robots with sensors that interact with the environment. In this paper, we propose a strategy for task allocation and resource management in MRC systems. The results show that the proposed scheme can effectively reduce the total energy consumption in SRs.

1. Introduction

At present, there are many practical applications of multirobot systems, such as floor mopping robots, rescue robots after nature disasters, surgery by surgical robots, and manufacture by mechanical arms [1]. Obviously, multirobot collaboration can greatly improve efficiency compared to a single robot. Therefore, a very hot research direction is the MRC system [2]. However, in a MRC system, the difficulty in accomplishing the lantency-sensitive and computation-intensive tasks comes from diverse computation, communication and sensing capacities of the individual robots, and the limited battery budget [3]. On the one hand, multirobots need cooperate to accomplish the time-critical jobs. On the other hand, efficient cooperation between multirobot is definitely inseparable from efficient communication. Fortunately, MEC technology has emerged as a promising solution to enable the task offloading via wireless communication, which perfectly fits the MRC system structure.

As driven by the explosive growth on data traffic, *artificial intelligence* (AI) has revolutionized science and social life. However, AI also brought tremendous computation workloads [4]. Inevitably, robots also undertake more and more AI tasks in various applications, which imposes a certain computational burden on robot with limited size. To ease these burdens, many research have been done to accelerate computing efforts in the following three areas [2]: different types of robots, control architectures, and communication technologies.

In order to solve the above problems, both the field of robotics and the field of communication are making contributions. In robotics, the solution is to collaborate by adding more robots to overcome the resource constraints of a single robot. Firstly, in [5], the author introduces the controlled mobility to enable the sparse sensing, which can be exploited for energy-efficient and nonredundant sensing in MRC networks. Secondly, through efficient wireless communication technology, robots can share data and working status. In

[6], the authors come up an ad hoc based MRC system in which p-persistent real-time ALOHA is used as the channel competation strategy. By doin this, they achieve ultrareliability and ultralow latency communication. Furthermore, some researchers have noticed that with the development of wireless communication technology, it is a feasible way to deploy the computing tasks of robots to the cloud [7]. For instance, in [8], the authors utilize a distributed framework to build a visual SLAM robotic system, where cloud servers are responsible for map optimization and storing information to reduce computational workload. Furthermore, UAV is a segment of robotics research that has received great attention, and in [9], a joint offloading and trajectory design scheme that minimizes the sum of maximum delay is proposed in a MEC-based UAV system.

As an emerging computation offloading and wireless communication combination technology, MEC deploys cloud-like functions closer to the edge of networks to reduce latency. MEC trades off computational offloading and wireless communication, which plays an important role in reducing energy consumption and computing latency [10]. MEC has also been extensively used in many areas including *vehicle to vehicle* (V2V) [11], *unmanned aerial vehicle* (UAV) [12], and augmented reality [13]. In [14], an air-ground integrated multiaccess edge computing system has been investigated. The interaction among mobile users has been modelled as a stochastic game, which is transformed into a single-agent Markov decision process at each user. Then, an online deep reinforement learning scheme has been proposed to approximate the Q-factor and postdecision Q-factor, respectively, which is used to find the optimal solution at each user. In [15], computation offloading in beyond the fifth generation networks has been studied, where the combination of the wireless communications and multiaccess edge computing is considered. Multiagent Markov decision process has been applied to model the computation offloading problem, where a distributed learning framework is developed. Authors in [16] have extended the MEC technology to the unlicensed bands, where a context-aware communication approach is to efficiently integrate licensed and unlicensed spectrums by MEC technologies.

Based on above investigation, we can observe that MEC is suitable for computation task offloading in MRC systems. In [17], the authors propose an energy-efficient task offloading scheme for multiple devices for TDMA and OFDMA systems. Moreover, in [18], an algorithm based on *Alternating Direction Method of Multipliers* (ADMM) is proposed to maximize the revenue of *MEC system operator* (MSO), which is comprehensive considering offloading, resource allocation, and content caching strategies. To overcome network congestion and long latency in cloud computing, a multilevel resource management algorithm is designed in [19], in which cloud and edge servers cooperate to complete tasks. In the scenario of multiserver serving multiuser, [20, 21] have proposed centralized and distributed methods to derive the optimal task offloading strategies, respectively. In [22], we have propose a MEC-based MRC system and an algorithm to offload task and allocate the resource which focus on fairness and robustness. In [23], we have developed the optimal resource scheduling scheme to ensure that the task can be accomplished in time. Therefore, in our previous work [22, 23], there have two different models, where the optimal model guarantees the performance, and the fairness model guarantees that the SR is always “online.” However, how to achieve the optimality and fairness is still an urgent problem to be solved.

This paper is devoted to proposing a resource allocation scheme applied to an MRC system, which minimizes the total energy consumption of SRs under a task implementation latency constraint. First, we consider a complicated realistic scenario where the SRs are in charge of the sensing and data collection and a MR is responsible for the task offloading and wireless communication resource management. Accordingly, a task implementation strategy in MEC-based MRC system is proposed, in which we divide each task into three parts. Then, an optimal and fair resource scheduling scheme is proposed to minimize the energy consumption of SR. The results show that the proposed scheme can effectively reduce the total energy consumption in SR.

2. System Model

In the paper, a time-critical task implementation process is proposed for MRC systems, and an MEC-based resource allocation scheme is studied. As shown in Figure 1, a MR M cooperates with K SRs, which denoted by a set of $\mathcal{S} = \{s_1, s_2, \cdots, s_K\}$. In addition, a powerful base station will handle some computing tasks for MR. The MR has more powerful computing ability than SRs, but SR has the ability to interact with the environment. Then, we let the MR lead multiple SRs on latency-sensitive computationally intensive tasks, such as AI applications and path planning.

As shown in Figure 2, the proposed task implementation process is divided to three stages. The first stage is called the sensing stage where each SR collects data required by the task under a duration of $T^{(ss)}$. The second is named as SR offloading stage. The SRs will process a certain amount of data locally and the rest of data will be offloaded to the MR with a time limitation $T_s^{(co)}$. In the MR offloading stage, which is the third stage, the MR also preform computation and offloading trade-offs to meet the latency $T_M^{(co)}$.

On the one hand, the MR needs to decide the amount of data each SR needs to collect, the transmit power of each SR, and the amount of data offloaded from each SR. The MR makes decisions based on the channel status of the SR and the remaining battery power, so the SR will feed back these information to the MR through the control channel at the beginning to ensure reliable communication. The MR then feeds back the decision to each SR via the control channel. Similarly, at the beginning of the third stage, the MR first measures and estimates the channel state information between the MR and the BS.

On the other hand, general working robots cannot be very large, so their battery and computing device capabilities are also limited. So we assume that SR has much less battery capacity and computing power than MR, and most of the energy of SR will be spent on sensing and collecting data

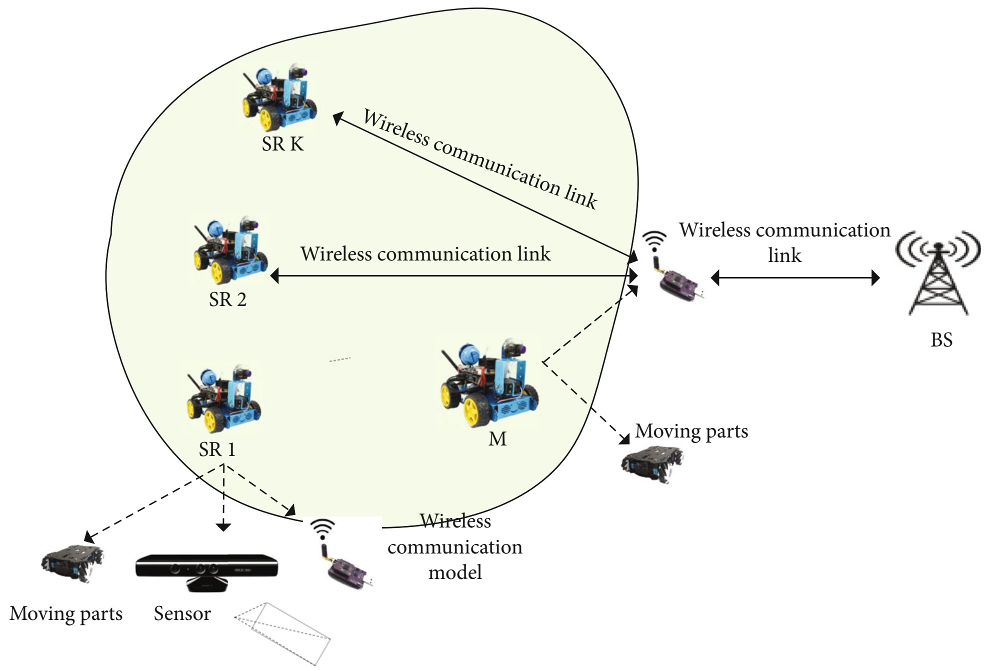

FIGURE 1: Multirobot cooperation system model.

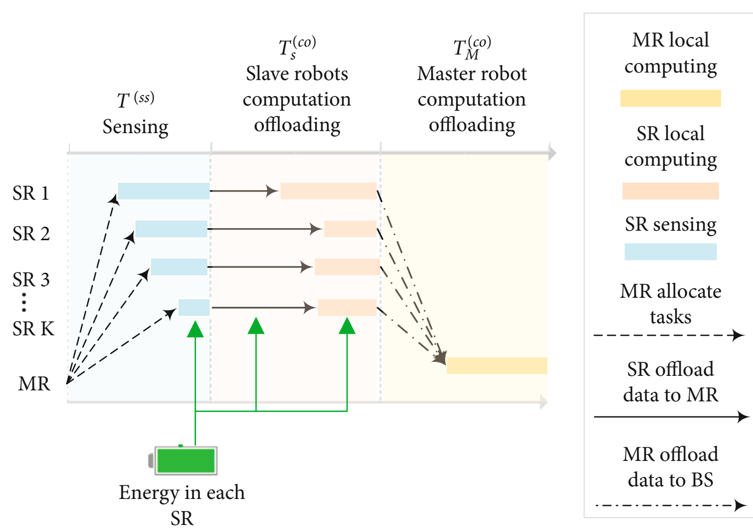

FIGURE 2: The time division structure and energy flow for system model.

in the first stage. In order to maintain the functionality of the MRC system for as long as possible, the power consumption of the SR should be minimized and balanced. To achieve this goal, the mathematical model of the system will be described in detail in the following subsections.

2.1. Data Collection Model. Firstly, MR will allocate different sensing time $t_k^{(ss)}$ to each SR based on the information provided by the SR, which is a variable. The amount of collected data bits at the k-th SR is denoted as

$$d_k = \frac{p_k^{(ss)} t_k^{(ss)}}{H_k}, \qquad (1)$$

where H_k represents the energy consumption used to sense one bit [24], and $p_k^{(ss)}$ is a constant which denote as the sensing power consumption at the kth SR.

2.2. Computation-Offloading Model. Figure 2 also shows the direction of data flow and energy consumption in the MRC system. Let $d_k^{(ol)}$ denotes the offloaded data bits from the kth SR to the MR, and $d_M = \sum_{k\in\mathcal{S}} d_k^{(ol)}$ as the total received data from SRs.

Then, $(d_k - d_k^{(ol)})$ bits are the data that need to be processed locally. Moreover, let W_k represents the computation capability of the kth SR, which is the number of *central processing unit* (CPU) cycles executed per second, and W_M as the computation capacity of the MR. Moreever, C_k as the number of CPU cycles is required at the kth SR to calculate 1-bit data, and C_M as the number of CPU cycles is required for computing 1-bit of data at the MR. To guarantee the latency constraint, the time spent on local computing should be less than a threshold $T_s^{(co)}$.

2.3. Communication Model. Consider the TDMA system. According to Shannon formula, the transmission data rate from the k-th SR to the MR is given by

$$r_k = B_k \log_2\left(1 + \frac{p_k^{(tr)} h_k}{N_k}\right), \tag{2}$$

where B_k is the channel bandwidth allocated to the kth SR, $p_k^{(tr)}$ is the transmission power consumed at SR k, h_k denotes the mean channel gain on the wireless channel between SR k and the MR, and N_k is the Gaussian channel noise, which is fixed. It is noteworthy that the channels between the SRs and MR experience both large fading due to the path loss and shadowing and fast fading due to the reflection and diffraction. Based on Equation (2), the time spent on offloading data from the kth SR to the MR can be written as

$$t_k^{(ol)} = \frac{d_k^{(ol)}}{r_k}. \tag{3}$$

Similarly, when the MR offloads the data to the BS, the transmission rate is given by

$$r_M = B_M \log_2\left(1 + \frac{p_M^{(tr)} h_M}{N_M}\right), \tag{4}$$

where B_M is the bandwidth allocated to the MR, $p_M^{(tr)}$ is transmission power allocation at the MR, h_M is the mean channel gain between the MR and the BS, and N_M is the noise power, which is fixed. According to Equation (4), the time spent on the transmission at the MR is given by $t_M^{(ol)} = D_M^{(ol)}/r_M$.

2.4. Energy Consumption Model. As shown in Figure 2, energy needs to be consumed in the three stages of the system. It is worth noting that we assume that the BS has sufficient computing and communication capabilities and energy. In this paper, we focus on the energy consumption of SRs, hence the computation time spent on the feedback from the BS and the energy cost at the BS is neglected.

First, According to Section 2.1, the energy consumption on sensing is given by $E_k^{(ss)} = p_k^{(ss)} t_k^{(ss)}$.

Next, the energy consumption on local computation at the kth SR is given by

$$E_k^{(co)} = \left(d_k - d_k^{(ol)}\right) C_k p_k^{(co)}, \tag{5}$$

where $p_k^{(co)}$ is the power consumption per CPU computing cycle at the kth SR. Similarly, when the MR executes the computation locally, the energy consumption is $E_M^{(co)} = (d_M - d_M^{ol}) C_M p_M^{(co)}$, where $p_M^{(co)}$ is the power consumption per CPU cycle of computing at the MR.

Finally, for convenience, we define a function $f(x) = N_k (2^{x/B} - 1)$, and according to Equation (3), the energy consumed on data transmission at the kth SR and the MR is given by $E_k^{(tr)} = p_k^{(tr)} t_k^{(ol)} = t_k^{(ol)}/h_k f(d_k^{(ol)}/t_k^{(ol)})$ and $E_M^{(tr)} = p_M^{(tr)} t_M^{(ol)} = t_M^{(ol)}/h_M f(D_M^{(ol)}/t_M^{(ol)})$, respectively.

Based on the above analysis, the total energy consumption at the kth SR and the MR during the task implementation is given by $E_k = E_k^{(ss)} + E_k^{(co)} + E_k^{(tr)}$ and $E_M = E_M^{(co)} + E_M^{(tr)}$, respectively.

3. Problem Formulation

The aim of the proposed MEC-based resource allocation scheme in MRC system is to accomplish time-critical tasks while maintaining the function of the system as long as possible. Then, the key problem is how to management sensing data and offload data in each SRs and MR. Furthermore, we define a weighted factor α_k associated with the kth SR, which can be expressed as $\alpha_k = \min (E_k^{\mathrm{Re}})/E_k^{\mathrm{Re}}, k \in \mathcal{S}$. The factor takes into account the fairness among all SRs. Accordingly, the optimization problem (P1) for the SRs is formulated as

$$\min_{\left\{\mathbf{t}_K^{(ss)}, \mathbf{d}_K^{(ol)}, \mathbf{t}_K^{(ol)}\right\}} \sum_{k=1}^{K} \alpha_k E_k \tag{6}$$

$$s.t. \sum_{k=1}^{K} d_k \geq D \tag{7a}$$

$$\frac{\left(d_k - d_k^{(ol)}\right)}{W_k} \leq T_s^{(co)}, k \in \mathcal{S} \tag{7b}$$

$$E_k^{(\mathrm{Re})} - E_k \geq 0, k \in \mathcal{S} \tag{7c}$$

$$T^{(ss)} \geq t_k^{(ss)} \geq 0, k \in \mathcal{S} \tag{7d}$$

$$T_s^{(co)} \geq t_k^{(ol)} \geq 0, k \in \mathcal{S} \tag{7e}$$

$$d_k \geq 0, k \in \mathcal{S} \tag{7f}$$

$$d_k^{(ol)} \geq 0, k \in \mathcal{S} \tag{7g}$$

where $E_k^{(\mathrm{Re})}$ denotes the remaining battery energy at the kth SR, $\mathbf{t}_K^s$ is the sensing time vector of SRs which denoted as $\mathbf{t}_K^{(ss)} = [t_i^{(ss)}]_{i=1}^K$, $\mathbf{d}_K^{(ol)}$ is the allocated offloading data vector of SRs which represented as $\mathbf{d}_K^{(ol)} = [d_i^{(ol)}]_{i=1}^K$, and $\mathbf{t}_K^{(ol)}$ is the time vector spent on data offloading with $\mathbf{t}_K^{(ol)} = [t_i^{(ol)}]_{i=1}^K$.

In problem (P1), the objective function is set to a min-max problem to improve fairness among SRs. Constraint (7a) is to guarantee the SRs collect the required number of

An optimal resource allocation scheme.
Initialize: System parameters, MR and BS; set $\eta_k = 0$; set $R = 0$ as the number of executed tasks.
While: MRC system can complete task, i.e., problem (P1) has a feasible solution: The Interior Point Method applied to the (SP1) to obtain the globally optimal solution $\{d^*_{R,k}, t^{(ss)*}_{R,k}\}$. And BCD method is used to solve the (SP2) to obtain the globally optimal solution $\{d^{(ol)*}_{R,k}, t^{(ol)*}_{R,k}\}$; *Gradient Descent (GD)* method is applied to update parameters η_k; jump to step 4 until convergence; compute total energy consumption of SRs, and update $E^{(\mathrm{Re})}_{R,k} = E^{(\mathrm{Re})}_{R,k} - E_{R,k}$, which represents the remaining energy of each SR after processing the current task; similarly, get the optimal solution $\{d^{(ol)*}_{R,M}, t^{(ol)*}_{R,M}\}$ by solving the (P2) using BCD method; compute energy consumption $E_{R,M}$ at the MR, and $E^{(\mathrm{Re})}_{R,M} = E^{(\mathrm{Re})}_{R,M} - E_{R,M}$; update $R = R + 1$.

Algorithm 1

data, (7b) ensures that each SR has an upper limit that spend on the local computing. Similarly, (7d) and (7e) represent the time constraints for the first and second stages, respectively. (7c) is to guarantee that the total power consumption at the kth SR is less than its remaining power.

Next, in the third stage, it needs to decide how much data should be offloaded to the BS to minimize the energy consumption. Accordingly, the problem (P2) is written as

$$\min_{\left\{t^{(ol)}_M, d^{(ol)}_M\right\}} \quad E_M \tag{8}$$

$$s.t. E^{(\mathrm{Re})}_M - E_M \geq 0 \tag{9a}$$

$$T^{(co)}_M \geq t^{(ol)}_M \geq 0 \tag{9b}$$

$$\frac{\left(d_M - d^{(ol)}_M\right)}{W_M} \leq T^{(co)}_M \tag{9c}$$

$$d_M \geq d^{(ol)}_M \geq 0 \tag{9d}$$

where (9a) is to guarantee that the total power consumption at the MR is less than its remaining battery power, (9b) represents the time constraints for the third stages, (9c) ensures that the time MR spend on the local computation should be less than $T^{(co)}_M$, and (9d) is to guarantee that the offloaded data bits are less than the total received data bits from SRs.

4. Proposed Algorithm

Through the analysis of the (P1), we find the energy consumption of SRs is coupled with constraints (7a) and (7c), which makes it complicated to solve the problem. To decouple these two constraints, the Lagrangian dual method is applied. Then, Lagrangian dual function corresponding to (6) is given by

$$L = \sum_{k=1}^{K} \alpha_k E_k + \sum_{k=1}^{K} \eta_k \left(E_k - E^{(\mathrm{Re})}_k\right), \tag{10}$$

where $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_n]^T$ is Lagrangian multipliers.

According to (10), the dual problem is defined as

$$\max_{\{\boldsymbol{\eta} \geq 0\}} \left\{ \min_{\left\{\mathbf{t}^{(ss)}_K, \mathbf{D}^{(ol)}_K, \mathbf{t}^{(ol)}_K\right\}} L \right\}, \tag{11}$$

which is convex on $\boldsymbol{\eta}$.

After decoupling, we can split the problem (P3) into three subproblems. First, the subproblem (SP1) is to minimize the weighted energy consumption of SRs during the first stage, which is given by

$$\begin{aligned} \min_{\left\{\mathbf{t}^{(ss)}_K, \mathbf{d}_K\right\}} \quad & \sum_{k=1}^{K} (\alpha_k + \eta_k) E^{(ss)}_k \\ s.t. \quad & (6a), (6d), (6f) \end{aligned} \tag{12}$$

Next, (SP2) is to minimize the weighted energy consumption of SRs during the local computation and task offloading at the second stage, which is given by

$$\begin{aligned} \min_{\left\{\mathbf{d}^{(ol)}_K, \mathbf{t}^{(ol)}_K\right\}} \quad & \sum_{k=1}^{K} (\alpha_k + \eta_k) \left(E^{(co)}_k + E^{(tr)}_k\right) \\ s.t. \quad & (6b), (6e), (6g) \end{aligned} \tag{13}$$

With Lagrange multipliers, $\boldsymbol{\eta}$, (SP1), and (SP2) are all convex optimization problems. In particular, (SP1) is linear programming problems, which can be solved by the Interior Point Method. The *Block Coordinate Descent* (BCD) [25] optimization technique can be used to obtain the optimal solution of (SP2). After achieving the optimal solution of the subprime problem $E^*, E^{(ss)*}_k, E^{(co)*}_k, E^{(tr)*}_k$, the Lagrange dual problem is a linear programming problem, which is formulated as

$$\begin{aligned} \max_{\{\boldsymbol{\eta}\}} L \quad & \left(\mathbf{t}^{(ss)}_K, \mathbf{d}^{(ol)}_K, \mathbf{t}^{(ol)}_K, \eta\right) \\ s.t. \quad & \eta_k \geq 0, k \in \mathcal{S}. \end{aligned} \tag{14}$$

The updating rule in the following Algorithm 1 can be applied to derive the optimal η.

Table 1: Simulation Parameters.

Number of SR			3
Number of MR			1
The initial battery energy of each SR			[3, 4, 1] J
The initial battery energy of MR			10 J
Low power mode threshold			0.2 J
T_s	800 ms	$T_s^{(co)}$	40 ms
B_k, B_M	10 MHz	$T_M^{(co)}$	10 ms
e_k^s	$\in[18, 22]$ nJ/bit	D	5×10^6 bits
C_k	$\in[100, 1000]$ cycles/bit	$p_k^{(ss)}$	$\in[0.05, 0.13]$ W
F_k	$[100, \cdots, 800]$ MHz	F_M	2.4 GHz
$p_M^{(co)}$	1000×10^{-11} J/cycle	W_M	2.4×10^9 cycles/s
N_k	$\in[-86, -96]$ dBm	C_M	100 cycles/bit
W_k	$\in[100, \cdots, 800] \times 10^6$ cycles/s		
$p_k^{(co)}$	$\in[100, 500] \times 10^{-11}$ cycles/bit		
Pass loss model	$15.3 + \alpha \times 10 \log_{10}(Distance)$, $\alpha = 3.75$, $\alpha = 5$,distance $\in [1,100]$ m		

Based on the Interior Point Method, the solution of (P2) can be achieved and the optimal task offloading and resource management scheme can be derived for the MR since it is a convex optimization problem. According to the above analysis, the optimal joint computation and communication resource scheme is concluded in Algorithm 1.

5. Numerical Results

In this section, the performance of proposed task and resource allocation scheme is investigated using computer simulation on MATLAB. We will refer to our scheme, i.e., Algorithm 1 as "ROOP." In order to reflect the performance of the algorithm, the robust scheme, which named as "MRC-RP," we proposed in [23], is considered. The key parameters used in the simulations are listed in Table 1.

5.1. ROOP Scheme Performance. Figure 3 shows the sensing data size allocation and offloading data size versus the number of accomplished tasks in ROOP. On the one hand, since we define a fairness factor α_k, the SR3 with the least battery will hardly participate in the task to save the battery energy. On the other hand, due to good channel conditions of SR2 in this simulation scenario, their transmission energy consumption per bit is lower than local computing. Therefore, SR2 are more inclined to offload more data to the MR. Furthermore, the time limit has a great constraint on the offloading of the SR1, since the SR1 has a bad channel state. Then, the results show that SR1 is not willing to offload.

Figure 4 demonstrates the remaining energy at each SR versus the number of implementing tasks. It can be observed that, in ROOP, since SR3 is assigned more tasks, it consumes more energy than MRC-RP. However, SR3 consumes less

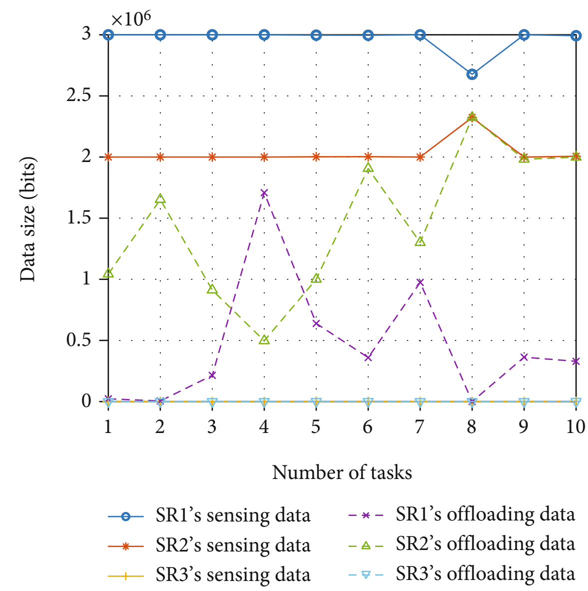

Figure 3: . The sensing data size allocation and data offloading.

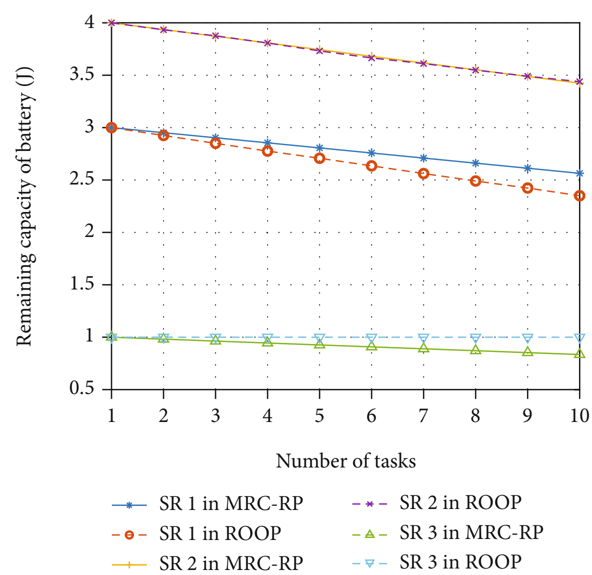

Figure 4: . The remaining battery energy corresponding to the three SRs.

energy to performing tasks than SR1 and SR2, which means the ROOP scheme is more friendly to the energy-less robots. We can observe in Figure 3 that the overall energy consumption in ROOP is smaller than MRC-RP. And as the number of accomplished tasks increases, more energy would be consumed and the remaining energy at the SRs decreases as well.

5.2. Performance Comparison. Figure 5 depicts the total energy consumption of SRs that complete different numbers of tasks. We consider the baseline *greedy offloading policy* (GOP) for comprehensive performance comparisons. The specific strategy of the GOP is that the SR will offload as many task bits as time permits. We observe that the total

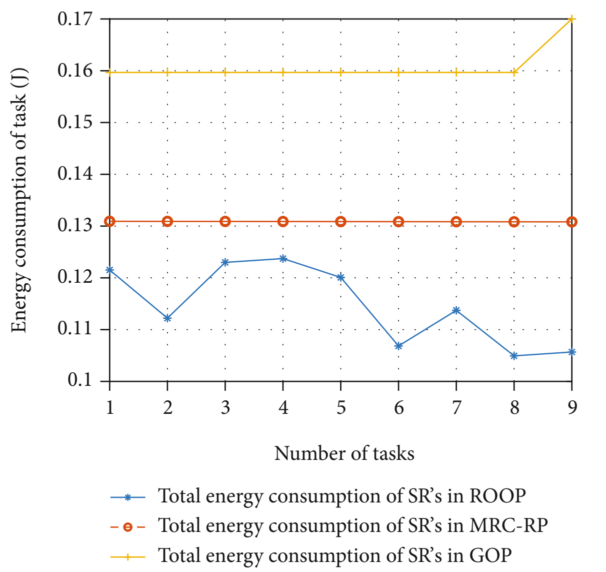

Figure 5: The comparison of energy consumption by ROOP, MRC-RP, and GOP.

energy consumption of SR for GOP and MRC-RP schemes is always much larger than ROOP when the number of tasks is accumulated, which verifies the effectiveness of our task and resource allocation scheme.

6. Conclusion

This work studies task and resource allocation for MRC system. First, a task implementation framework is proposed for MRC system based on MEC. Next, aiming to save the energy of SRs and MR and prolong the MRC system function time, we proposed a time-critical and computation-intensive resource allocation scheme for MEC-based MRC system, in which an MR acts as an edge server to provide computation and communication services to SRs. Simulation results revealed that proposed scheme greatly outperforms the baseline.

Data Availability

No data were used to support the study.

Conflicts of Interest

There is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Science and Technology Project of State Grid Zhejiang Electric Power Co., Ltd. (2021ZK17).

References

[1] Y. Liu, X. Liu, X. Gao et al., "Robotic communications for 5g and beyond: challenges and research opportunities," 2020, http://arxiv.org/abs/2012.05093.

[2] Z. H. Ismail and N. Sariff, "A survey and analysis of cooperative multi-agent robot systems: challenges and directions," *Applications of Mobile Robots*, pp. 8–14, 2018.

[3] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K.-C. Wang, "Review of internet of things (IoI) in electric power and energy systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 847–870, 2018.

[4] H. Kim, H. Nam, W. Jung, and J. Lee, "Performance analysis of CNN frameworks for gpus," *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, vol. 2017, 2017, pp. 55–64, Santa Rosa, CA, USA, April 2017.

[5] Y. Mostofi, "Cooperative wireless-based obstacle/object mapping and see-through capabilities in robotic networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 817–829, 2013.

[6] K.-C. Chen and H.-M. Hung, "Wireless robotic communication for collaborative multi-agent systems," in *International Conference on Communications (ICC)*, pp. 1–7, Shanghai, China, May 2019.

[7] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[8] L. Riazuelo, J. Civera, and J. M. Montiel, "C2TAM: a cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401–413, 2014.

[9] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879–1892, 2018.

[10] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *Journal of Network and Computer Applications*, vol. 169, p. 102781, 2020.

[11] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[12] M. Zhao, W. Li, L. Bao, J. Luo, Z. He, and D. Liu, "Fairness-aware task scheduling and resource allocation in UAV-enabled mobile edge computing networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2174–2187, 2021.

[13] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[14] X. Chen, C. Wu, T. Chen et al., "Information freshness-aware task offloading in air-ground integrated edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 243–258, 2022.

[15] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: a distributed learning framework and applications," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 56–62, 2021.

[16] C. Wu, X. Chen, T. Yoshinaga, Y. Ji, and Y. Zhang, "Integrating licensed and unlicensed spectrum in the internet of vehicles with mobile edge computing," *IEEE Network*, vol. 33, no. 4, pp. 48–53, 2019.

[17] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2016.

[18] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[19] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[20] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[21] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[22] Y. Shen, R. Yin, H. Zhu, X. Chen, and C. Wu, "Resource management in MEC based muti-robot cooperation systems," in *International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 30–37, Hangzhou, China, Dec. 2021.

[23] R. Yin, Y. Shen, H. Zhu, X. Chen, and C. Wu, "Time-critical tasks implementation in MEC based multi-robot cooperation systems," 2021, http://arxiv.org/abs/2111.11038.

[24] J. Zhu and S. Papavassiliou, "On the energy-efficient organization and the lifetime of multi-hop sensor networks," *IEEE Communications Letters*, vol. 7, no. 11, pp. 537–539, 2003.

[25] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2016.