When UAV Swarm Meets Edge-Cloud Computing: The QoS Perspective

Wuhui Chen, Baichuan Liu, Huawei Huang, Song Guo, and Zibin Zheng

Abstract

In this article, we propose a hybrid computing model, UAV-Edge-Cloud, bringing edge/cloud computing and UAV swarm together to achieve high quality of service (QoS) guarantees. First, we design this novel hybrid computing framework to provide powerful resources to support resource-intensive applications and real-time tasks at edge networks. Next, we discuss some potential applications for smart cities and raise open research issues of the proposed hybrid framework. We then study a joint task placement and routing problem for latency-critical applications as a case study. Finally, the simulation results show that our approach can improve the QoS of UAV swarms effectively.

Introduction

Unmanned aerial vehicle (UAV) swarms are widely applied for service provisioning in many domains such as package delivery, farming, and network connectivity [1, 2]. However, the UAV swarm faces inherent physical constraints as all computation and storage are conducted onboard, and its advancement is restricted by its limited computational, storage, and communication capacities. Due to their poor onboard resources, UAV swarms cannot effectively perform resource-intensive applications such as artificial intelligence (AI)-related tasks for crowd sensing in smart cities. The resource-intensive applications usually have the following characteristics:

- Big data volume: Sensors or cameras on UAV swarms generate big streams of data traffic, which requires huge communication resources [3].
- Huge demand for processing power: Applications such as object recognition, navigation, and AI processing require intensive processing power to accomplish.

Because of these characteristics and limited resources on UAV swarms, it is challenging for UAV swarms to effectively support resource-intensive applications.

Many efforts [4, 5] have been made to relieve the resource constraints of UAV swarms. Among these existing proposals, cloud robotics [5] has been considered as the most promising solution to improve the capacity of UAV swarms. It enables UAV swarms to benefit from the redundant resources of modern data centers. Although cloud-based UAV swarms have obtained some positive results, they have yet to achieve high quality of service (QoS) for resource-intensive applications for the following reasons.

First, big data transfer between UAV swarms and the remote cloud could cause latency issues. UAV swarms are latency-intolerant because they have to make real-time or near-real-time decisions. Even slight delays in avoiding collisions and obstacles could cause dangerous consequences. On the other hand, the generated large amount of streaming data in UAV swarms would cause long communication delay when the data is transferred to remote centralized data centers for processing.

Second, big data transmission could cause energy efficiency issues. A UAV swarm is energy-constrained, and the battery is one of the most precious resources. While computation offloading in cloud robotics could reduce the energy consumption caused by big data processing, it could increase the energy consumption caused by big data transfer. The problem will deteriorate if the applications are communication-intensive or UAV swarms experience poor network connectivity.

In this article, we first propose a novel UAV-Edge-Cloud computing model for the UAV swarm to extend its capacity and support resource-intensive applications. To the best of our knowledge, this is the first attempt to study the hybrid computing model to support resource-intensive applications with QoS guarantees over UAV swarms. We then show the benefits and challenges in providing UAV-Edge-Cloud-based applications. Finally, under our hybrid framework, we study the joint task placement and routing problem for latency-critical UAV swarms using the Markov approximation algorithm for further QoS improvement.

Related Works

The related works on solving the resource constraints of UAV swarms can generally be classified into three categories. The comparisons of these existing works are summarized in Table 1.

Category A: Ad Hoc Cloud UAV Swarms: To overcome the resource constraints of individual UAVs, an ad hoc cloud UAV swarm, which pools together the computing capability of a set of neighboring UAVs to form a virtual ad hoc cloud, has been widely explored [4, 6]. Li *et al.* [6] investigated the fundamental mobile cloudlet (UAV) properties that unfold whether and when a mobile cloudlet can provide mobile application service. However, although a UAV could share its computation workload with neighboring UAVs in the same ad hoc network, the overall effective-

Digital Object Identifier:
10.1109/MNET.2019.1800222

Wuhui Chen, Baichuan Liu, Huawei Huang (corresponding author), and Zibin Zheng are with Sun Yat-sen University; Song Guo is with the Hong Kong Polytechnic University.

Category	Literature	Model	Resource	Latency	Energy	Intelligence
Ad hoc cloud UAV swarms	[4, 6]	Ad hoc cloud model	Weak	Low	High energy consumption caused by computing workloads	Low
Cloud-based UAV swarms	[5, 7]	Cloud computing model	Strong	High	High energy consumption caused by communication workloads	High
Edge-based UAV swarms	[8, 9]	Edge computing model	Medium	Low-medium	Low	Medium
UAV-Edge-Cloud UAV swarms	Our proposal	Hybrid computing model	Strong	Low	Low	High

TABLE 1. Comparison of state-of-the-art approaches.

ness is still constrained by the limited computing and storage resources of individual UAVs.

Category B: Cloud-Based UAV Swarms: To overcome the resource constraints of UAV swarms, the cloud-based UAV swarm has been widely studied [5, 7]. Mohanarajah *et al.* [5] proposed an open source cloud robotics platform, Rapyuta, that allows robots to outsource their onboard computational processes to modern data centers. However, the high latency and low energy efficiency between UAV swarms and remote cloud hamper its real-world applications.

Category C: Edge-Based UAV Swarms: The edge-based UAV swarm, which brings computing resources to the edge of network, is considered as a promising solution to improve the high latency and low energy efficiency [8, 9]. Kwak *et al.* [9] proposed DREAM, a dynamic CPU/network resource and task allocation algorithm, to minimize energy consumption of a mobile device for given delay constraints. However, due to the resource limitation on edge servers, an edge-based UAV swarm may not be able to handle the resource-intensive applications.

In this article, we aim to bring the UAV computing model, cloud computing model, and edge computing model together, and combine their advantages to release the resource constraints of UAV swarms while ensuring the QoS guarantees.

Architecture

To support resource-intensive applications and ensure QoS guarantees for UAV swarms, we propose a hybrid UAV-Edge-Cloud computing model.

Overview

UAV swarms have big data, dynamic, and heterogeneity characteristics, while applications on UAV swarms usually have strict real-time and resource-intensive requirements. Due to these characteristics, the QoS guarantees in UAV swarms become a challenging task. To reduce the latency, energy consumption, and operating cost, we bring the edge computation together with UAV computing and cloud computing to UAV swarms. Figure 1a shows the UAV-Edge-Cloud architecture. It consists of a UAV swarm layer, which is able to form an ad hoc virtual cloud, an edge layer, which is close to the UAV swarm providing services with low latency via distributed edge servers, and a remote cloud layer, which has abundant resources based on centralized cloud data centers. The UAV swarm layer and the edge layer are able to interact with users quickly in a real-time manner, while the cloud data centers can provide high business intelligence by processing large amounts of data. These different layers collaborate with each other and provide services with high QoS guarantees, as shown in Fig. 1b. More details are explained as follows.

UAV swarms have big data, dynamic, and heterogeneity characteristics, while applications on UAV swarms usually have strict real-time and resource-intensive requirements. Due to these characteristics, the QoS guarantees in UAV swarms become a challenging task.

Cross-layer Cooperation

UAV-UAV Cooperation for Sophisticated Tasks: There are three levels of UAV-UAV cooperation:

1. Performing tasks collaboratively: In many scenarios, to accomplish a complex task that is beyond the capacity of one single (even powerful) UAV, UAV swarms are required to collaborate with each other. One example is crowd sensing in smart cities, such as monitoring urban traffic congestion or disaster damage.
2. Processing data collaboratively: Collaborative UAVs could pool together their computational capability to form virtual ad hoc cloud for data processing such as data fusion and data analysis for collision or obstacle avoidance.
3. Communicating collaboratively: UAVs in swarms could communicate with their peers and share information for distributed decision making in various cooperative tasks.

UAV-Edge Cooperation for Real-Time Processing: Real-time applications have very strict latency requirement, but the UAV individual or the ad hoc virtual cloud still have limited computational resources, which produce long data processing times and cannot meet the latency requirement of real-time applications. Fortunately, the edge servers deployed close to the UAV swarms, brought together by the ad hoc virtual cloud, are able to provide computational resources with lower delay, less network pressure, and more efficient data processing for real-time applications. The researchers [10] reduced the latency from 900 ms to 169 ms for a face recognition application by moving the computation from the cloud to the edge. It can be widely used in collisions and obstacle avoidance scenarios for UAV swarms in smart cities.

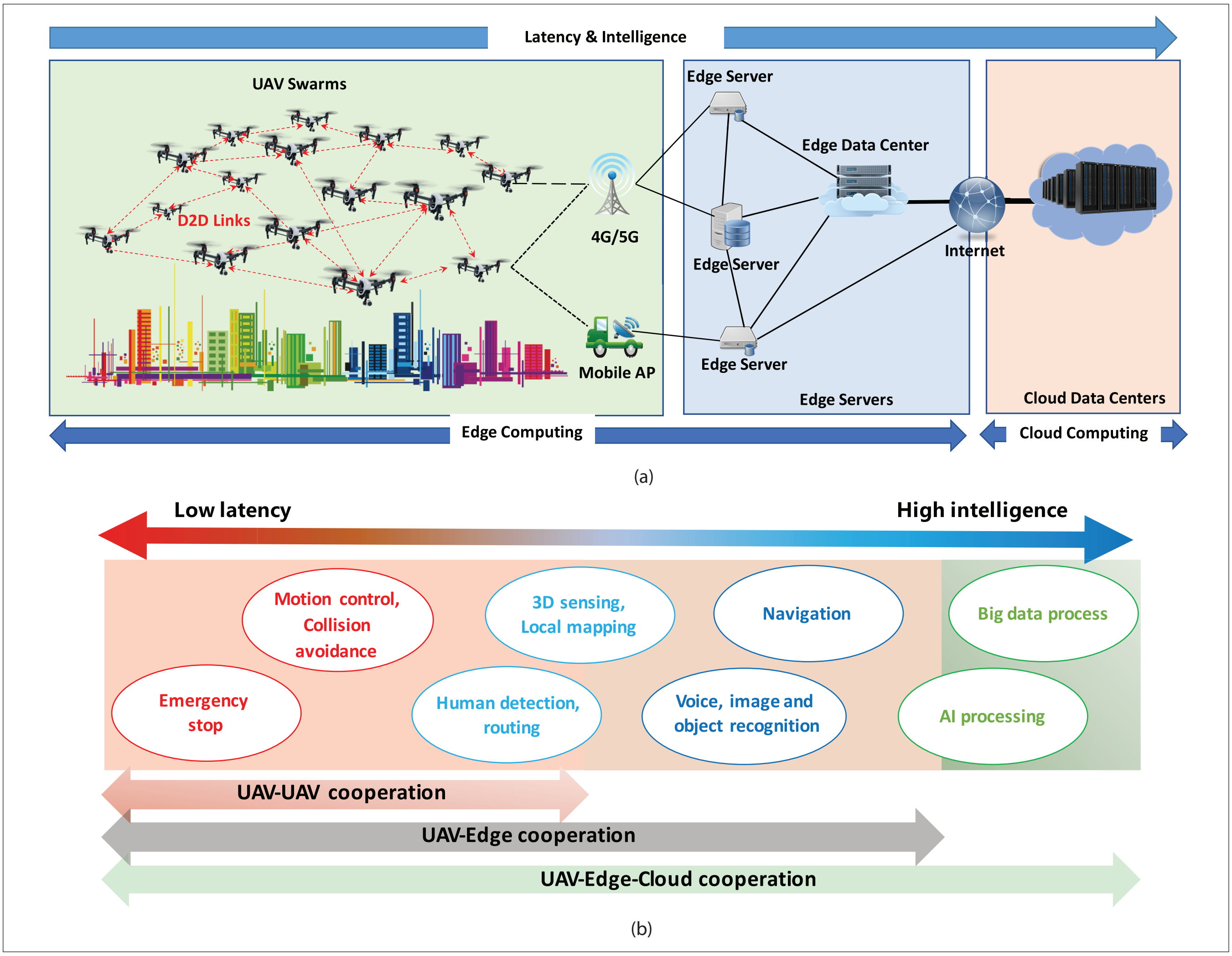

FIGURE 1. The architecture and applications of UAV-Edge-Cloud computing architecture: a) UAV-Edge-Cloud computing architecture; b) applications of crowd sensing in smart cities based on UAV-Edge-Cloud Computing.

UAV-AP-Edge Cooperation for Efficiency Communication: Due to the stochastic characteristics of wireless networks, a UAV will experience sporadic wireless network disconnection caused by irregular UAV mobility or bandwidth fluctuations. Therefore, UAV-access point (AP)-Edge cooperation is required to enhance the stability and reliability of UAV swarm communication. First of all, because a mobile UAV can experience poor or even intermittent connectivity, it is unable to directly connect a cloud access point (e.g., UAVs in a tunnel, or in areas of damage after a disaster). In these cases, UAVs shall collaborate with each other to forward transmitted messages to the AP via neighboring UAVs. Second, because huge data traffic would happen between UAV swarms and edge servers, the collaboration of APs is required to enhance the efficiency. AP collaboration (e.g., using Multipath-TCP) can provide seamless communication channels migration for UAV swarms to reduce the latency, energy consumption, and operating cost.

UAV-Edge-Cloud Cooperation for Resource-Intensive Applications: Big data as well as computation-intensive applications on these big data need UAV-edge-cloud cooperation. It cannot perform as effectively if one of them is missing. First, an infrastructure-based cloud can provide abundant cloud resources to help process and store large-scale gathered data for the UAV swarms. Second, an ad hoc UAV cloud can provide a solution for UAV swarms to sporadic wireless network connectivity, or in scenarios where UAV swarms are unable to directly connect a cloud AP. Finally, the edge plays the role of bringing UAV swarms and remote cloud together by providing computation resources with low latency for UAV swarms and reducing the communication cost for cloud. In the example of crowd sensing in smart cities, it is more efficient to preprocess stream data (e.g., video or photos) at the edge server for UAV swarms before uploading to the remote cloud so that the uploaded data size related to communication latency and energy consumption could be significantly reduced.

QoS Improvement

Latency Reduction: Our hybrid framework can improve the latency issue effectively for UAV swarms. First, by using the edge computing, the data size can be greatly reduced so that the total communication latency can be effectively

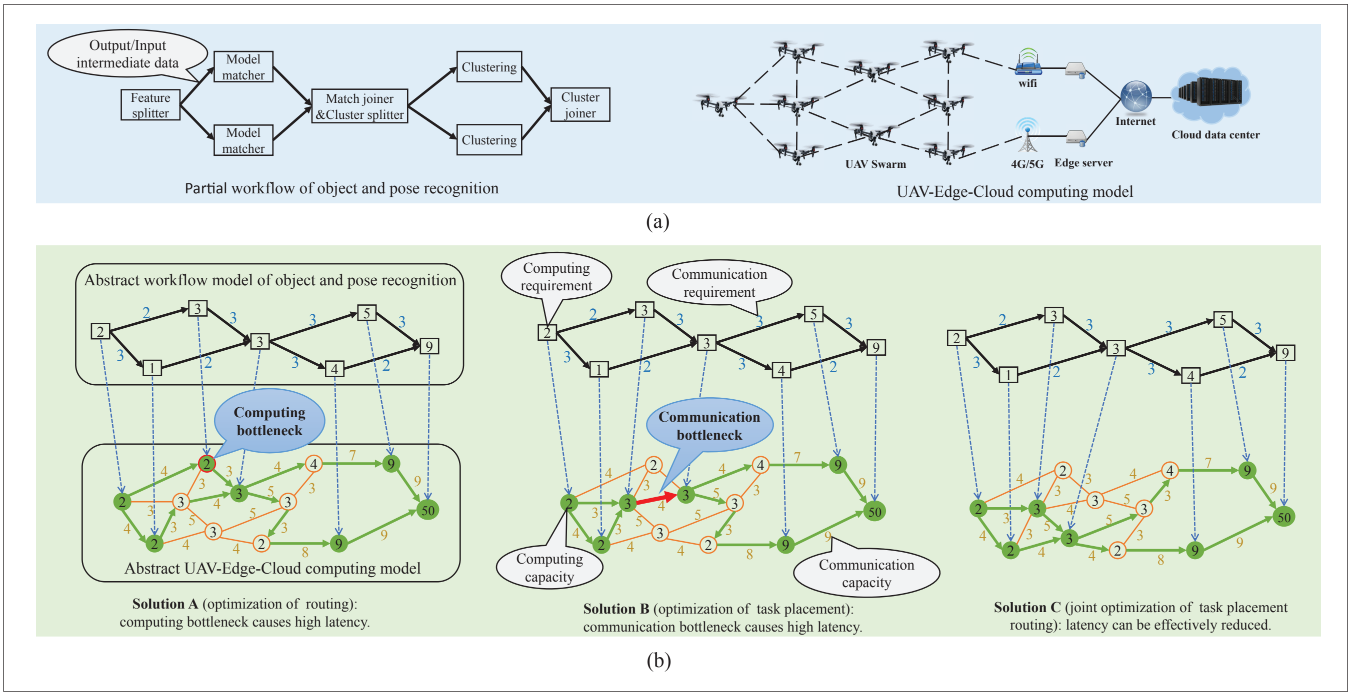

FIGURE 2. Illustration of allocating tasks in a work flow to the UAV-Edge-Cloud computing model: a) examples of real-world work flow and UAV-Edge-Cloud computing model; b) assigning a work flow to the UAV-Edge-Cloud computing model: a motivating example for joint optimization of task placement and routing.

improved. Second, with more communication constrained at the network edges by edge computing and less direct communication between UAV swarms and remote cloud, the communication latency can further reduced. Third, by offloading the tasks with high instruction complexity to edge and cloud, the computation workload at UAV swarms can be significantly reduced so that the computation latency can be greatly improved.

Energy Efficiency: Our hybrid framework can achieve energy efficiency for UAV swarms. First, by computation offloading, UAV swarms can migrate computation-intensive tasks to edge or remote cloud, which could effectively reduce the energy consumption. Second, by preprocessing large-scale data at the edge server, the uploaded data size could be reduced so that the energy consumption for data communication can be greatly saved. Furthermore, the stochastic characteristics of wireless networks (e.g., bandwidth fluctuations caused by building shields, and mobility issues) may consume excessive energy for communications, and UAV-AP-Edge cooperation can be used to reduce the energy consumption.

Operating Cost Cutting: In our proposed hybrid framework, the operating cost can be greatly reduced by minimizing the communication and computation costs. In detail, the communication cost can be greatly reduced because the data size uploading to remote cloud has been reduced by preprocessing at the edge servers near the UAV swarms. Furthermore, by computation offloading, the workload to remote cloud would be reduced because the edge servers in many cases are powerful enough for computation-intensive tasks. This could greatly reduce the computation cost at the cloud side.

Case Study: Joint Task Placement and Routing Planning for Latency-Critical UAV Swarms

In this section, we study a work flow allocation problem as a case study of the proposed UAV-Edge-Cloud computing architecture, which is shown in Fig. 1a.

Problem Statement

Overview: A work flow consists of several tasks. Streaming data is transferred between two neighboring tasks. A task may have more than one predecessor and successor tasks, which form several branches like the example work flow shown in Fig. 2a. A UAV swarm contains a group of UAVs that are connected with each other via wireless connections. When a work flow is assigned to a UAV swarm, each task should be allocated to a specific UAV. Then a group of routing paths in the UAV network should be found for the data transmission between two neighboring tasks. Edge servers and cloud servers are available for some computation-intensive tasks.

Task Placement: In a UAV swarm, UAVs offer different computing capabilities since they have different hardware configurations. Computation-intensive tasks that need large computing resources should be allocated to the powerful UAVs. However, due to the limited onboard resources, UAVs cannot support computation-intensive applications in general. That inspires us to place these computation-intensive tasks at edge servers or cloud servers that can offer more powerful computing capabilities.

Routing Planning: Routing plays an important role in computation offloading. Note that the links in UAV networks have different transmission delays. For example, the link delay between two UAVs may be very low because it is a LAN. In

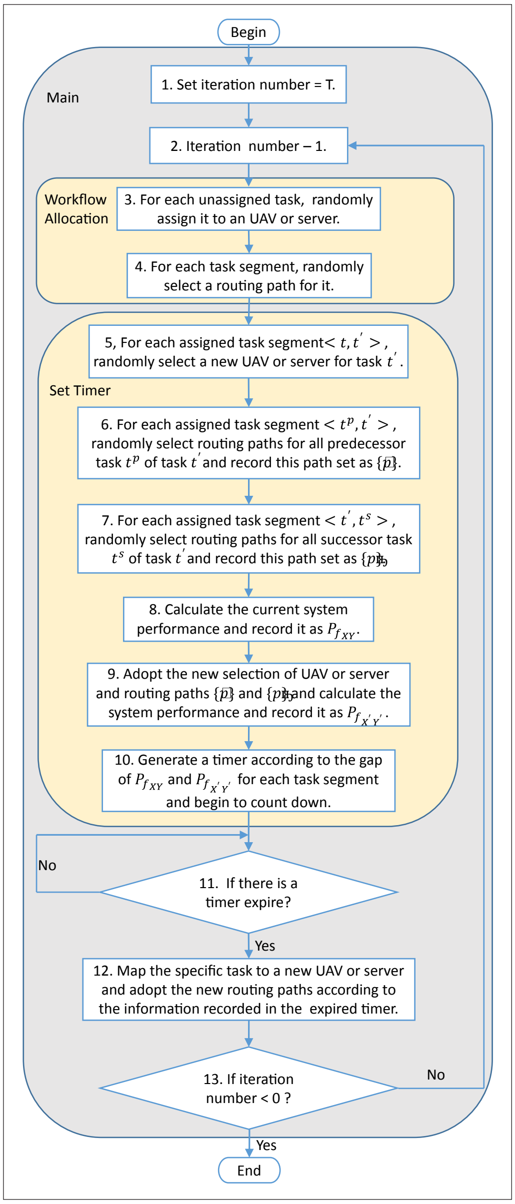

FIGURE 3. Illustration of MA algorithm to solve the Joint Task Placement and Routing (JTPR) problem.

contrast, the link delay between a UAV and an edge server or between an edge server and a cloud server can be much higher than that of a LAN link. Thus, for each pair of neighboring tasks, it is of great significance to select a routing path with low delay.

Joint Optimization of Task Placement and Routing Problem: Both task placement and routing are important for computation offloading. If we optimize the task placement problem solely, there is no doubt that allocating as many tasks as possible to the servers can achieve good performance in terms of gaining strong computing capabilities. However, it is adverse to latency-sensitive tasks. On the contrary, if we optimize the routing problem solely, allocating all tasks to the UAV swarm can achieve good performance in terms of low transmission delay, but is adverse to computation-intensive tasks. Compared to the single-goal optimization method, jointly optimizing the task placement and routing scheduling can achieve globally optimal performance that would be more useful in practice. Figure 2b shows a motivating example for joint optimization of task placement and routing. Solution A in Fig. 2b only considers the optimization of routing, in which a computing bottleneck may occur and cause high latency. Solution B only focuses on the optimization of task placement, in which communication bottleneck could cause high latency. Finally, solution C, which considers the joint optimization of task placement and routing, can avoid the computing and communication bottleneck and reduce the latency effectively.

Algorithm Design

The Markov-Approximation Algorithm: Referring to [11, 12], we then propose a Markov-approximation-based algorithm (MA), which is shown in Fig. 3, to solve the JTPR problem. Notice that Fig. 3 is the main algorithm that executes properly with the support of algorithms 1 and 2. We explain the basic idea of each algorithm as follows.

Work Flow Allocation (Alg. 1): In the beginning, all the work flows are unsatisfied because none of them has been placed at UAVs or servers yet. Algorithm 1 is in charge of allocating the work flow tasks to UAVs or servers. For each work flow, the program allocates its tasks to a UAV and finds feasible routing paths for task segments randomly. Because of the limitation of bandwidth, not all of them may be able to get an allocation. A part of the work flows may still remain unsatisfied after the initial allocation. In this manner, Algorithm 1 runs in each iteration in order to allocate work flows to the UAV swarm.

SetTimer (Algorithm 2): First, a new UAV or an edge server is randomly selected to execute a pair of consecutive tasks, which is also called a task segment. Then a routing path will be set for this task segment by randomly selecting a new routing path from a candidate path set. Because a task may have more than one predecessor and successor tasks, the system also randomly selects paths for all the predecessor and successor tasks. Next, the system performance under the current configuration is computed. Once the newly selected computing nodes and routing paths are adopted, the performance of the system under a new configuration is computed. Finally, an exponentially distributed random timer whose mean value is related to the performance gap that is generated independently with the current configuration. Then the new timer begins to count down.

Main (Algorithm 3): Algorithm 1 is used to allocate all the work flows to UAV swarm, while Algorithm 2 aims to create and maintain timers for all task segments. If the system detects an expiration of a timer, actions will be taken to jump to a new configuration by mapping the specific task to a new UAV or a server and adopting the new specific routing paths. Then the system updates the performance value and resets all the timers according to Algorithm 2. The main algorithm will not terminate until all the work flows have been allocated to the UAV swarm.

Performance Evaluation

We implemented a simulator in Python to evaluate the proposed algorithm. We first construct a computing architecture consisting of 21 UAVs,

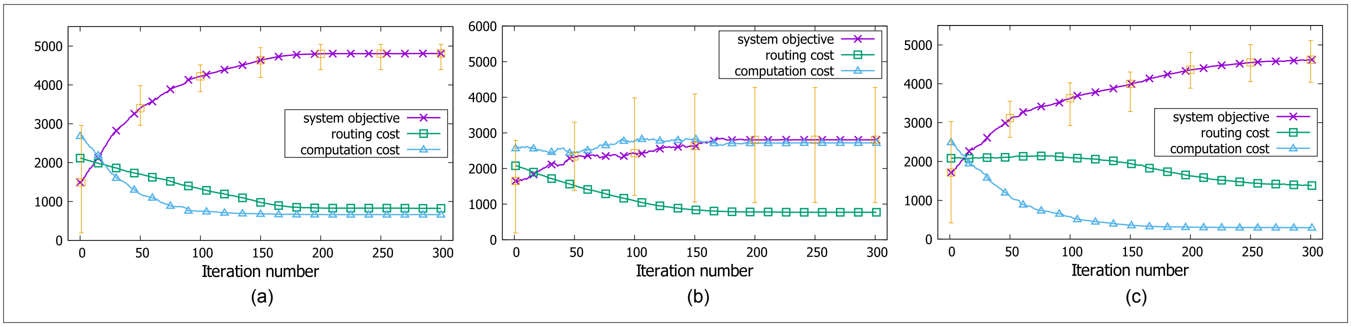

FIGURE 4. Results of simulation: a) performance under a = 1 and b = 1; b) performance under a = 0 and b = 1; c) performance under a = 1 and b = 0.

three edge servers, and one cloud server. The data processing capacity of each UAV is set randomly within a range from 1 to 100. The data processing capacities of the edge server and cloud server are set to 500 and 10,000, respectively. We then generate a parallel work flow set that contains 10 work flows and 90 tasks in total. The processing resources needed by each task is set from 50 to 1000 randomly, and the bandwidth needed by each task segment is set from 10 to 100 randomly. In each simulation, we run our algorithm 20 times and show the average value as the final result.

We first evaluate the feasibility of our approach. Let a and b denote the wight factor of computation cost and the routing cost, respectively. Then, under different settings of a and b, we can observe different running cases from Fig. 4a to Fig. 4c. At first, Fig. 4a shows that our algorithm quickly converges to a maximal performance after 200 iterations, which indicates that our algorithm can produce a feasible solution at this time slot. In contrast, the routing cost and computation cost exhibit decreasing trends. Then we explore an extreme case when weight factors a and b are set to 0 and 1, respectively. This setting means that we ignore the impact of task placement on system performance. Figure 4b shows that the routing cost decreases quickly, while the computation cost fluctuates several times and finally remains at a high level. Next, we conduct another simulation when weight factors a and b are set to 0 and 1, respectively. This time, it ignores the impact of routing planning on system performance. Figure 4c shows that the computation cost decreases quickly, while the routing cost remains high. From these two extreme cases, we can also see that the system objectives of the solely-routing-aware approach and the solely-task placement-aware approach converge more slowly than that of the joint study approach. Furthermore, the proposed joint control approach outperforms the other two single-objective approaches in terms of the final converged system objective.

In summary, Fig. 4 exhibits that the proposed MA algorithm could yield feasible solutions within a few iterations under different joint optimization control policies for latency-critical UAV swarms. In our implementation, the proposed algorithm can achieve very fast convergence within the unit of logical 0.001 s. Thus, according to the simulation results, the proposed algorithm converges within around 200 iterations, which indicates that the algorithm can converge within 200 ms if the algorithm is deployed in a real implementation.

Applications and Open Issues

This section discusses some applications and critical challenges of our hybrid framework for smart cities.

Benefits and Applications

Efficient Knowledge Sharing: As a human goes through childhood, having a big data collection process, UAV swarms could share learned knowledge and learn skills from other UAVs via a knowledge base using our proposed hybrid framework, as shown in Fig. 5a. One representative application for knowledge sharing is to determine the optimal strategy to grasp an object. With the support of our hybrid framework, UAVs with mechanical hands can send featured data obtained from onboard sensors to an edge-cloud hybrid computation framework. The edge-cloud processes the received data and quickly performs model matching using a knowledge base stored in the edge-cloud, and soon returns a set of candidate objects together with their possible grasping solutions back in low latency. The UAVs compare the received 3D CAD models from the edge-cloud with the detected point cloud, and then select the best grasping solution. Finally, model knowledge of new objects learned by UAVs is added to the knowledge base in the edge-cloud and becomes available for future grasping by other UAVs.

Efficient Collaboration: Our hybrid framework acts as a brain, providing not only powerful computation capacity to process stream data but also a large-scale knowledge base to support decision making. This allows UAV swarms to collaborate with each other with high QoS guarantees and accomplish sophisticated tasks more effectively, as shown in Fig. 5b. A representative application is the swarm navigation for crowd sensing in disaster sites. The surrounding environment after widespread damage in disaster sites is fairly unpredictable, and the task of swarm navigation is to determine the UAV swarm's own position and then plan a feasible path to effectively reach a predefined location. There have been many extensive research works, including both mapless approaches and map-based approaches [13]. However, these approaches usually suffer from QoS problems (e.g., high latency) due to the limited onboard storage and processing capacities in UAV swarms. Fortunately, our hybrid edge-cloud computing framework can provide a promising solution for edge-cloud-enabled swarm navigation that overcomes these weakness. It can pro-

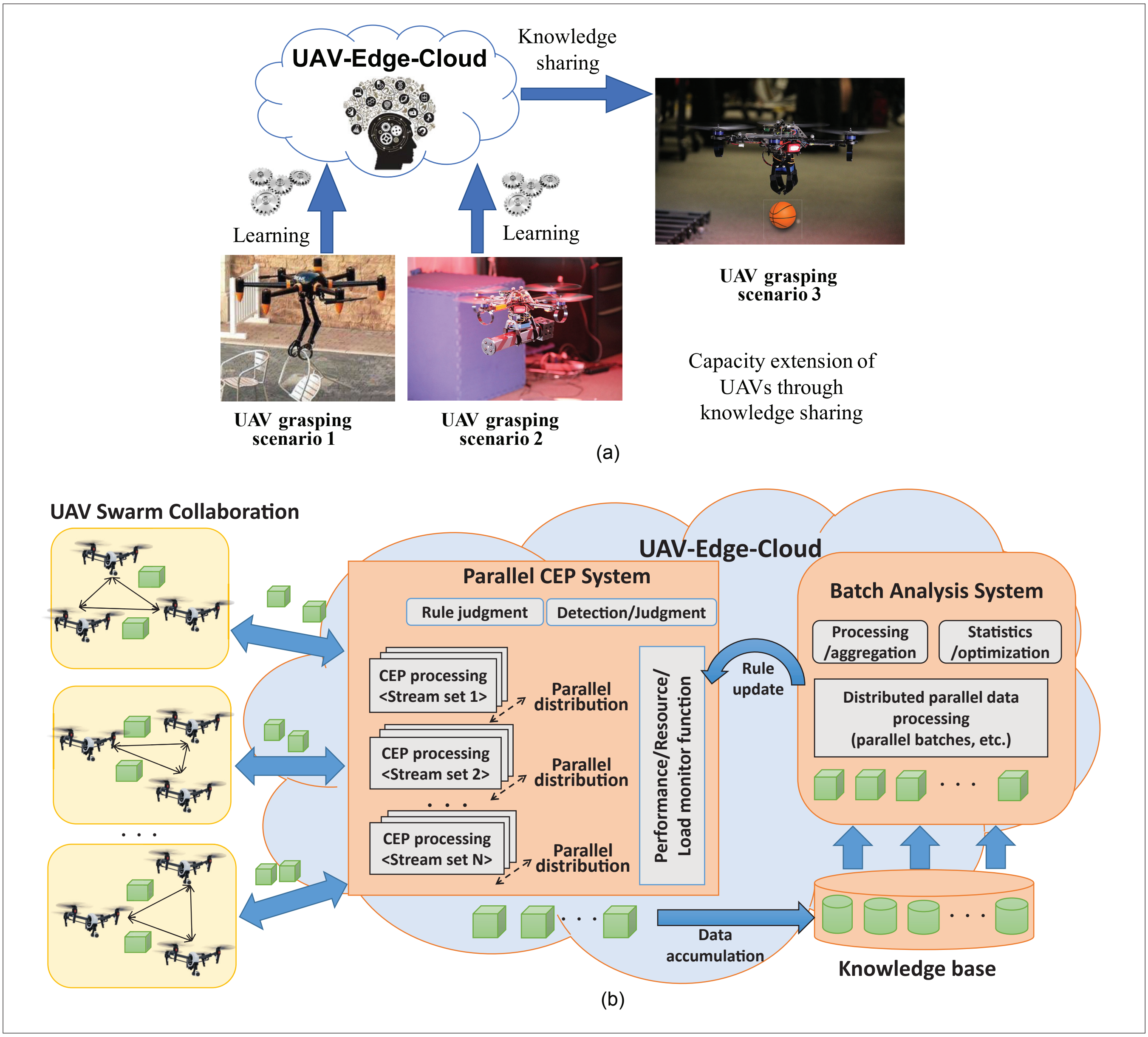

FIGURE 5. Data placement using golden division on a Zipf-like replica distribution: a) UAV swarm knowledge sharing: knowledge learned from scenarios 1 and 2 is applied to scenario 3; b) UAV swarm collaboration: UAV-Edge-Cloud computing acts as a brain.

vide not only a sufficient storage platform for the large-scale knowledge storage for mapping, but also powerful computing capacity to support the searching and building of the map for UAV swarm navigation with low latency and high energy efficiency.

Open Issues

Further QoS Improvement: Due to the strict latency and energy requirement, some mechanisms for QoS guarantees based on the UAV-Edge-Cloud model are further required. Application partitioning for efficiency computation offloading is a complex problem since identifying the resource-intensive components in workloads remains an issue. Besides, the stochastic characteristics of wireless networks could cause unpredictable latency and energy consumption [14], and QoS-aware communications are critical to fully realize the power of the UAV-Edge-Cloud model. Therefore, much effort, such as using fifth generation (5G) techniques [15], must be made to address the communication issues associated with real-time processing.

Resource Management: Resource management and scheduling is critical to QoS guarantees. First, much effort must be made to dynamically monitor the status of a UAV computing node (including location and speed, CPU utilization, network status, and remaining battery and storage space), and the status of edge and cloud servers (including available resources in edge and cloud, and the properties of the available resources). Second, cross-layer resource scheduling requires considerable research and development. High heterogeneity exists in cross-layer resources, even in the same layer, bringing the challenge of resource scheduling. Therefore, effective resource scheduling algorithms are significant to overcome the heterogeneity issues and ensure the QoS guarantees.

Conclusion

In order to allow UAV swarms to support resource-intensive applications such as crowd sensing for smart cities, in this article, we propose a novel UAV-Edge-Cloud computing model for resource enhancement and QoS guarantees to extend the capacity and intelligence of UAV swarms. Besides, based on our UAV-Edge-Cloud computing model, we study the joint task placement and routing problem for latency-critical UAV swarms using the Markov approximation algorithm for further QoS improvement. Finally, our evaluation shows the high efficiency of our proposed algorithm. In future work, experiments with and evaluation of real UAV swarm systems based on our UAV-Edge-Cloud computing model will be conducted.

High heterogeneity exists in cross-layer resources, even in the same layer, bringing the challenge of resource scheduling. Therefore, effective resource scheduling algorithms are significant to overcome the heterogeneity issues and ensure the QoS guarantees.

Acknowledgment

The work described in this article was supported by the National Key Research and Development Plan (2018YFB1003803), the National Natural Science Foundation of China (61722214, 61802450, 61872310), the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2016), the Natural Science Foundation of Guangdong (2018A030313005), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2017ZT07X355), and the Shenzhen Basic Research Funding Scheme (JCYJ20170818103849343).

References

[1] M. Dong *et al.*, "UAV-Assisted Data Gathering in Wireless Sensor Networks," *J. Supercomputing*, vol. 70, no. 3, 2014, pp. 1142–51.
[2] H. Menouar *et al.*, "Uav-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, Mar. 2017, pp. 22–28.
[3] L. Gu *et al.*, "A General Communication Cost Optimization Framework for Big Data Stream Processing in Geo-Distributed Data Centers," *IEEE Trans. Comp.*, vol. 65, no. 1, Jan. 2016, pp. 19–29.
[4] M. Chen *et al.*, "On the Computation Offloading at Ad Hoc Cloudlet: Architecture and Service Modes," *IEEE Commun. Mag.*, vol. 53, no. 6, June 2015, pp. 18–24.
[5] G. Mohanarajah *et al.*, "Rapyuta: A Cloud Robotics Platform," {*IEEE Trans. Automation Science and Engineering*, vol. 12, no. 2, 2015, pp. 481–93.
[6] Y. Li and W. Wang, "Can Mobile Cloudlets Support Mobile Applications?," *IEEE INFOCOM*, 2014, pp. 1060–68.
[7] B. Kehoe *et al.*, "A Survey of Research on Cloud Robotics and Automation," *IEEE Trans. Automation Science and Engineering*, vol. 12, no. 2, 2015, pp. 398–409.
[8] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, Jan./Feb. 2018, pp. 96–101.
[9] J. Kwak *et al.*, "Dream: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," *IEEE JSAC*, vol. 33, no. 12, 201, pp. 2510–235.
[10] W. Shi *et al.*, "Edge Computing: Vision and Challenges," *IEEE Internet of Things J.*, vol. 3, no. 5, 2016, pp. 637–46.
[11] M. Chen *et al.*, "Markov Approximation for Combinatorial Network Optimization," *IEEE Trans. Info. Theory*, vol. 59, no. 10, 2013, pp. 6301–27.
[12] H. Huang *et al.*, "Service Chaining for Hybrid Network Function," *IEEE Trans. Cloud Computing*, 2017. DOI:10.1109/TCC.2017.2721401.
[13] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated Robot Navigation Via Hierarchical Clustering," *IEEE Trans. Robotics*, vol. 32, no. 2, 2016, pp. 352–71.
[14] V. Sharma *et al.*, "Efficient Management and Fast Handovers in Software Defined Wireless Networks Using UAVs," *IEEE Network*, vol. 31, no. 6, Nov./Dec. 2017, pp. 78–85.
[15] Y. Hu, M. C. Gursoy, and A. Schmeink, "Relaying-Enabled Ultra-Reliable Low-Latency Communications in 5G," *IEEE Network*, vol. 32, no. 2, Mar./Apr. 2018, pp. 62–68.

Biographies

Wuhui Chen (chenwuh@mail.sysu.edu.cn) is an associate professor at Sun Yat-sen University, China. He received his Bachelor's degree from Northeast University, China, in 2008. He received his Master's and Ph.D. degrees from the University of Aizu, Japan, in 2011 and 2014, respectively. From 2014 to 2016, he was a JSPS research fellow in Japan. From 2016 to 2017, he was a researcher at the University of Aizu. His research interests include edge/cloud computing, cloud robotics, and blockchain.

Baichuan Liu (liubch6@mail2.sysu.edu.cn) received his B.S. degree from Northeastern University in 2018. He is currently pursuing an M.S. degree with the School of Data and Computer Science, Sun Yat-sen University. His research interests include edge computing, cloud computing, cloud robotics, and computation offloading.

Huawei Huang [M] (huawei.hwang@gmail.com) received his Ph.D. in computer science and engineering from the University of Aizu. His research interests mainly include software-defined networking (SDN), NFV, edge computing, and edge intelligence. He was a visiting scholar at the Hong Kong Polytechnic University (PolyU) from 2017 to 2018. He was a post-doctoral research fellow of JSPS between October 2016 and March 2018. He received the best paper award from IEEE TrustCom '16.

Song Guo [SM] (song.guo@polyu.edu.hk) received his Ph.D. in computer science from the University of Ottawa. He is currently a full professor in the Department of Computing, PolyU. Prior to joining PolyU, he was a full professor at the University of Aizu. His research interests are mainly in the areas of cloud and green computing, big data, wireless networks, and cyber-physical systems. He has published over 300 conference and journal papers in these areas and received multiple best paper awards from IEEE/ACM conferences. His research has been sponsored by JSPS, JST, MIC, NSF, NSFC, and industrial companies. He has served as an Editor of several journals, including *IEEE TPDS*, *IEEE TETC*, *IEEE TGCN*, *IEEE Communications Magazine*, and *Wireless Networks*. He has been actively participating in international conferences serving as General Chair and TPC Chair. He is a Senior Member of ACM and an IEEE Communications Society Distinguished Lecturer.

Zibin Zheng (zhzibin@mail.sysu.edu.cn) is a professor in the School of Data and Computer Science, Sun Yat-sen University. He received the Outstanding Ph.D. Thesis Award of the Chinese University of Hong Kong in 2012, the ACM SIGSOFT Distinguished Paper Award at ICSE 2010, the Best Student Paper Award at ICWS 2010, and an IBM Ph.D. Fellowship Award in 2010. He served as a PC member for IEEE CLOUD, ICWS, SCC, ICSOC, SOSE, and others. His research interests include service computing and cloud computing.