Stochastic Coded Offloading Scheme for Unmanned Aerial Vehicle-Assisted Edge Computing

Wei Chong Ng, Wei Yang Bryan Lim, Zehui Xiong, Dusit Niyato, *IEEE Fellow*, Chunyan Miao,

Zhu Han, *IEEE Fellow*, and Dong In Kim, *IEEE Fellow*

Abstract—**Unmanned aerial vehicles (UAVs) have gained wide research interests due to their technological advancement and high mobility. The UAVs are equipped with increasingly advanced capabilities to run computationally intensive applications enabled by machine learning techniques. However, because of both energy and computation constraints, the UAVs face issues hovering in the sky while performing computation due to weather uncertainty. To overcome the computation constraints, the UAVs can partially or fully offload their computation tasks to the edge servers. In ordinary computation offloading operations, the UAVs can retrieve the result from the returned output. Nevertheless, if the UAVs are unable to retrieve the entire result from the edge servers, i.e., straggling edge servers, this operation will fail. In this paper, we propose a coded distributed computing approach for computation offloading to mitigate straggling edge servers. The UAVs can retrieve the returned result when the number of returned copies is greater than or equal to the recovery threshold. There is a shortfall if the returned copies are less than the recovery threshold. To minimize the cost of the network, energy consumption by the UAVs, and prevent over and under subscription of the resources, we devise a two-phase Stochastic Coded Offloading Scheme (SCOS). In the first phase, the appropriate UAVs are allocated to the charging stations amid weather uncertainty. In the second phase, we use the z-stage Stochastic Integer Programming (SIP) to optimize the number of computation subtasks offloaded and computed locally, while taking into account the computation shortfall and demand uncertainty. By using a real dataset, the simulation results show that our proposed scheme is fully dynamic, and minimizes the cost of the network and UAV energy consumption amid stochastic uncertainties.**

Index Terms—**Unmanned Aerial Vehicles, Coded Distributed Computing, Stochastic Integer Programming, Task Allocation, Internet-of-Things**

I. INTRODUCTION

Due to the rapid advancement of Internet of Things (IoT) enabled technologies, the number of wirelessly connected devices is increasing exponentially [1] and generating huge

WC. Ng and WYB. Lim are with Alibaba Group and Alibaba-NTU Joint Research Institute, Nanyang Technological University, Singapore

ZH. Xiong is with Singapore University of Technology and Design, Singapore

D. Niyato is with School of Computer Science and Engineering, Nanyang Technological University, Singapore.

C. Miao is with Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY) and School of Computer Science and Engineering, Nanyang Technological University, Singapore

Z. Han is with the Department of Electrical and Computer Engineering in the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701.

D. I. Kim is with the Department of Electrical and Computer Engineering, Sungkyunkwan University (SKKU), Suwon 16419, South Korea

amounts of data [2]. There are many new real-time applications enabled by wirelessly connected devices, such as augmented/virtual reality [3] and smart cities [4] that are delay-sensitive. For example, it is important to know the real-time traffic [5]/parking [6] information to regulate traffic flow. The increase in wirelessly connected devices exerts a tremendous burden on the wireless communication infrastructure. For example, in many urban areas that are covered by dense skyscrapers or when the end-users are in congested regions or at high-speed vehicular network [7], the content in the static roadside units (RSUs)/base stations (BSs) cannot be successfully delivered to the end-users.

One solution is to deploy Unmanned Aerial Vehicles (UAVs), also known as drones, to act as an airborne BS to collect and process data from the terrestrial nodes [8, 9]. UAVs are in different shapes and sizes, such as fixed wings or multirotors, and they can maintain a line-of-sight communication with the end-users to provide a better quality of service (QoS). Furthermore, UAVs can be flexibly deployed to inaccessible terrains or disaster relief operations, e.g., due to their size and mobility. Moreover, wireless connections can be established without a fixed infrastructure to extend communication coverage. However, apart from all those benefits, UAVs are faced with energy constraints [10], and thus, they cannot complete their computation tasks if the energy utilization is not scheduled correctly.

In this paper, we consider a network contains various UAVs, mobile charging stations, and edge servers that are attached to the BSs to run applications such as traffic monitoring [11, 12]. The UAVs are required to perform computation, e.g., distributed matrix multiplication, as it is central to many modern computing applications, including machine learning and scientific computing [13, 14] in applications such as post-disaster relief assistance [15] and crowd detection [16]. To alleviate some of the battery constraints of the UAVs, the matrix multiplication can be offloaded to ground-based edge servers for processing. The matrix multiplication in the UAVs can be accelerated by scaling the multiplication out across many distributed computing nodes in BSs or edge servers [17] known as the workers. However, there is a significant performance bottleneck that is the latency in waiting for the slowest workers, or "stragglers" to finish their tasks [17]. Coded distributed computing (CDC) is introduced to deal with stragglers in distributed high-dimensional matrix multiplication. In CDC, the computation strategy for each worker is carefully designed so that the UAV only needs to wait for the fastest subset of workers before recovering the

output [17]. The minimum number of workers that the UAV has to wait for to recover their results is known as the recovery threshold.

Apart from using the CDC technique to mitigate stragglers, there are three challenges in this network. The first challenge is the weather uncertainty. If the UAV is not properly allocated, it may not withstand the strong wind if its engines are not sufficiently powerful and the battery capacity is small. The second challenge is demand uncertainty. Typically, the edge servers in the BSs require the users to pay a subscription fee in advance, e.g., monthly subscription, so that the users, i.e., UAVs, can use the offloading service. For instance, in matrix multiplication, the size of the matrices, which is the demand, is not always the same. If the actual matrix size is very small, it will be cheaper to perform the local computation within the UAV. Therefore, an uncertainty of actual demand can result in an over-and under-subscription problem. The third challenge is the shortfall uncertainty. Once after the UAVs are allocated, they can perform full local computation, full offload or partial offloading. If the UAV decides to offload the computation to the edge servers in BSs, there is shortfall uncertainty that the copies cannot be returned by any edge servers on time to the UAV due to delays and link failure [18]. It means that the total copies that the UAV has is less than the recovery threshold, where each copy is a sub-portion of matrices involved in the matrix multiplication operation. Therefore, the UAV has to pay a correction cost to re-compute the number of shortfalls locally or re-offload them to match the recovery threshold. This correction cost also involves a hovering cost as the UAVs have to hover in the sky throughout the re-computation.

To overcome the three challenges mentioned above, we introduce the Stochastic Coded Offloading Scheme (SCOS). SCOS is a two-phase optimization scheme that adopts a CDC technique to reduce the total cost of the network:

- **Phase one (UAV type allocation):** The application owner will first allocate the appropriate UAV to each mobile charging station by considering the weather condition in each time slot. This weather uncertainty is modeled by a two-stage Stochastic Integer Programming (SIP) [19].
- **Phase two (task allocation):** There are a different number of time-frames/periods within the same time slot. For example, when the morning is the first time slot, each hour is treated as one period, and task allocation occurs in each period. Demand and shortfall are the two uncertainties in task allocation. Instead of performing local computation as the correction action to correct the shortfalls, the same decision options are provided to the UAVs until the z stage. Therefore, z-stage SIP is used to model the demand and shortfall uncertainty in various stages.

Extensive simulations are performed to evaluate the effectiveness of SCOS. The results show that SCOS can minimize the total cost and the UAVs' energy consumption, especially compared with the traditional deterministic baseline scheme.

The contributions of this paper are summarized as follows.

- The combination/integration yields fully dynamic on-demand computing solutions for emerging applications such as road traffic prediction for autonomous vehicles in which traditional approaches are ineffective due to their rigid and fixed deployment.
- Our SCOS is able to provide strategic scenario-based decision that adapts well with the weather condition in which the current solutions for UAVs are limited.
- The proposed SCOS can minimize the UAVs' overall costs by optimizing the task allocation. At the same time, it can also minimize all the UAVs' energy consumption. The optimal solution is achieved by considering both the demand and shortfall uncertainty.
- From the performance evaluation, we use the real data to validate that SCOS is the optimal scheme when the performance is compared with the Expected-Value Formulation (EVF) and random scheme.

The remainder of the paper is organized as follows: In Section II, we review the related works. In Section III, we present the system model. In Sections IV and V we formulate the problem. We discuss and analyze the simulation result in Section VI. Section VII concludes the paper.

II. RELATED WORK

A. UAV-enabled Mobile Edge Computing

Mobile edge computing (MEC) is regarded as a promising solution to break through the computation limitation [20]. Due to the flexibility of the UAVs, the UAV is an ideal mobile edge computing (MEC) platform for performing computing-intensive tasks for ground users. Furthermore, the UAV-enabled MEC platform can be quickly deployed in emergency response scenarios such as major traffic accidents [21]. There have been several works investigating the performance of UAV-enabled MEC. In [22], the authors studied the UAV-enabled MEC wireless powered system by considering both partial and binary computation offloading modes. Instead of using only the UAVs to act as the BSs, the authors in [23] installed the MEC servers on both UAVs and stationary BSs and presented a novel game-theoretic framework to serve their users more efficiently. In [24], the authors consider both computation bits and energy consumption to optimize the computation efficiency in a multi-UAV MEC system. The authors in [25] maximize the computation efficiency in partial computation offloading mode.

However, different from the work mentioned above, in this paper, we reduce energy consumption by adopting a CDC technique to mitigate stragglers in the network. The UAVs can recover the computed task if the returned tasks are greater than or equal to the recovery threshold.

B. Stochastic Integer Programming

Stochastic integer programming is one of the important tools to incorporate uncertainty in optimization problems [26]. SIP can be applied to various fields to solve the optimization problem, e.g., production planning [27]. SIP assumes uncertain data as random variables with known probability distributions, and uses sampled values from this distribution to build a scenario tree and optimize over the expectation [28]. SIP models can

correct the decisions using the concept of recourse. In this idea, some decisions have to be made before realizing uncertain parameters and some decisions after their realization [19]. SIP models can be formulated as the two-stage and multi-stage problems. For the two-stage SIP, stage one decisions are made 'here and now' at the beginning of the period without the uncertainty realization. Stage two decisions are taken 'wait and see' as the recourse action at the end of the period [29]. For example, in [30], the authors applied the two-stage SIP to optimize the resource provisioning cost in cloud computing. In the courier delivery serves, the authors in [31] uses the two-stage SIP to plan an optimal vehicle delivery route. A multi-stage SIP is a generalization of the two-stage SIP to the sequential realization of uncertainties. For example, the authors in [32] use a multi-stage SIP to optimize electricity generation, storage, and transmission investments over a long planning horizon. The recourse is the key concept behind SIP. In this problem, weather, demand, and shortfall uncertainties are constantly changing. Therefore, it is not possible to obtain one decision that is suitable for all scenarios. With the idea of recourse, corrective action can be made after a random event has taken place. To the best of our knowledge, the application of stochastic programming to coded distributed computing has been less studied.

C. Coded Distributed Computing

Distributed computing has been widely adopted to perform various computation tasks in different computing systems [33, 34]. Nevertheless, there are many design problems, i.e., computing frameworks are vulnerable to uncertain disturbances, such as node failures, communication congestion, and straggler nodes [18]. Only in recent years, coding techniques gained great success in improving the resilience of communication, storage, and cache systems to uncertain system noises [35]. The authors have [36] first presented the used of CDC to speed up matrix multiplication and data shuffling. As a result, a lot of the focus has been shifted to CDC. Followed by this study, CDC has been explored in many different computation problems, such as the gradients [37], large matrix-matrix multiplication [38], and multivariate polynomials [39].

There have been many other works to reduce the communication load [40, 41] that are capable of improving the overall communication time. The authors in [40] introduced a Coded MapReduce framework to reduce the inter-server communication load by a multiplicative factor that grows linearly with the number of servers in the system. The authors in [41] presented a technique known as Short-Dot to reduces the cost of computation, storage, and communication. Besides reducing the communication load, Short-Dot also tackles the straggler issue. It completes the computation successfully by ignoring the stragglers. More relevant to our study, the authors in [13] proposed PolyDot codes, which is a unified view of Matdot [13] and Polynomial codes [17] and leads to a trade-off between recovery threshold and communication costs.

However, the works mentioned above mainly focus on the designing of different CDC schemes. Therefore, in this paper, we adopt PolyDot codes in the UAV network to alleviate the straggler problem and improve network reliability.
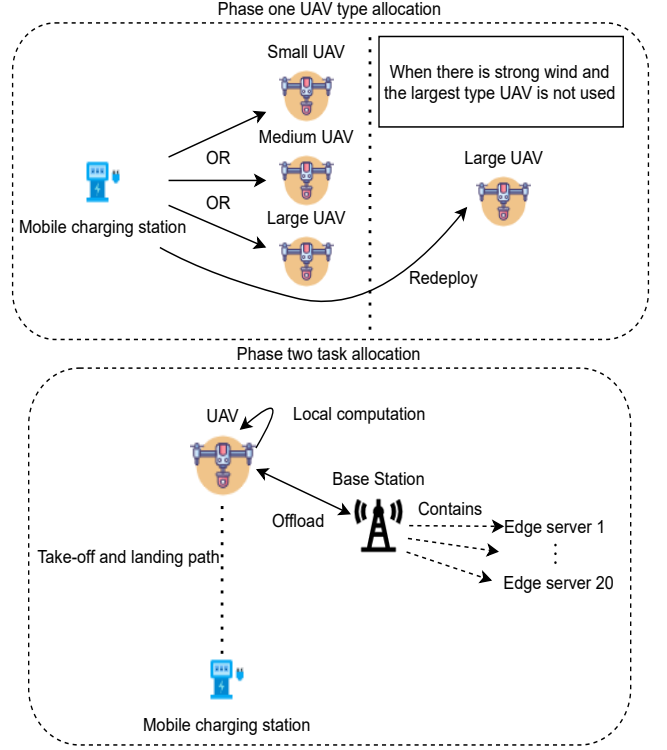

Fig. 1. An illustrative example of the network with $\mathcal{X} = \{1 : small, 2 : medium, 3 : large\}$, 1 mobile charging station $Y = 1$, 20 edge servers $q_1 = 20$ attached to 1 BS $F = 1$.

III. SYSTEM MODEL

The overall system model is shown in Fig. 1. We model the phase one (UAV type allocation) and phase two (task allocation) to complete applications defined by an application owner, e.g., road traffic monitoring [11] while considering various uncertainties. Since each edge server has limited computation capability, by deploy many edge servers at the BS, we can use constraints (53) and (54) from Appendix A to ensure that there will be enough computation resources to support the computation required by each UAV. The following sets are used to denote time slots, UAV types, mobile charging stations, and BSs.

- $\mathcal{T} = \{1, \ldots, \bar{t}, \ldots, \bar{T}\}$ represents the different time slot.
- $\bar{\mathcal{P}}^{\bar{t}} = \{1, \ldots, \bar{p}^{\bar{t}}, \ldots, \bar{P}^{\bar{t}}\}$ represents the period in time slot $\bar{t}$.
- The available UAVs are clustered into $|\mathcal{X}|$ types denoted by set $\mathcal{X}$, where $\mathcal{X} = \{1, \ldots, x, \ldots, X\}$. Specifically, the type refers to the battery capacity of the UAV in ascending order. For example, X is the largest type UAV that has the most battery and therefore leads to a longer flight time. The UAVs are owned by service provider $\bar{A}_1$. We use $x^{\bar{t}}$ to denote when type x UAV is used in time slot $\bar{t}$.
- $\mathcal{Y} = \{1, \ldots, y, \ldots, Y\}$ represents the UAV mobile charging stations, owned by service provider $\bar{A}_1$. All the mobile charging stations are deployed at pre-specified locations defined by application owner $\bar{A}_3$.
- Each of BS f is attached with q_f number of edge servers. $\mathcal{F} = \{1, \ldots, f, \ldots, F\}$ represents BSs with the height

of H_f. Edge servers are owned by service provider $\bar{A}_2$.

In phase one, the application owner first considers the weather uncertainty to pre-allocate the UAV types to each mobile charging station, also known as a UAV depot. Once the phase one optimization is done, all the UAVs will take off from their respective mobile charging stations which are located at (a_y, b_y). (a_y, b_y) are the x-y coordinates of mobile charging station y. At time slot $\bar{t}$, type x UAV will take off vertically to the height of $H_{y,x^{\bar{t}}}$ and hover in the sky for purposes such as traffic monitoring. $(a_y, b_y, H_{y,x^{\bar{t}}})$ and (a_f, b_f, H_f) are the three-dimensional coordinates of the type x UAV associated with mobile charging station y and edge servers in BS f, respectively, where $H_{y,x^{\bar{t}}} > H_f$ to maintain a line-of-sight (LoS) communication link between type x UAV and edge servers in BS f. For simplicity, we assume that UAV maintain a LoS link with the edge servers in the RSUs. Due to the hovering capability, we consider only the rotary-wing UAVs [42].

After the UAVs reach their respective heights, they can receive and process computation tasks. In this paper, we consider the task that the type x UAV computes is the matrix-matrix product **AB** involving the two matrices **A** and **B**. However, the UAV has limited computing and storage capability [43]. Therefore, the UAV can choose to offload a portion or the whole matrix multiplication to the edge servers [43]. In phase two, it derives the offloading decision to minimize the overall operation cost by considering the demand and shortfall uncertainties. Note that the key notations used in the paper are listed in Table I. In the following, we discuss the coded distributed computing model and UAV energy consumption model.

A. Coded Distributed Computing

Massive parallelization can speed up matrix multiplication. However, it has a computational bottleneck due to stragglers or faults. Coded computation is introduced to make matrix multiplications resilient to faults and delays, i.e., PolyDot codes [13]. In PolyDot codes, the system model typically consists of the followings [13]:

- Master node receives computation inputs, encodes and distributes them to the worker nodes.
- Worker nodes perform pre-determined computations on their respective inputs in parallel.
- Fusion node receives outputs from successful worker nodes and decodes them to recover the final output.

We consider that the type x UAV is our proposed network's master and fusion node. Each edge server in BS f is the worker and has the computation capability of $\tau_{f,q}$, where $\tau_{f,q}$ denotes the CPU computation capability of the edge server q in BS f (in CPU cycles per second).

The definitions of copy, successful workers, recovery threshold, shortfall, and demand are given as follows.

Definition 1. [Copy] a fraction of matrices **A** and **B** [13].

Definition 2. [Successful workers] Workers that finish their computation task and the task is received successfully by the UAV.

Definition 3. [Recovery threshold] The recovery threshold is the worst-case minimum number of successful workers required by the UAV to complete the computation [13].

Definition 4. [Shortfall] There exists a shortfall when the total returned copies from the local computation and from the workers are less than the recovery threshold.

Definition 5. [Demand] The demand is size of the matrix input N_y. It is always different as the input of the matrix multiplication is not always the same.

Following [13], two $N_y \times N_y$ square matrices $\mathbf{A}_y$ and $\mathbf{B}_y$ are considered. Note that our model can be applied to other matrices, e.g., non square matrices. Each of matrices $\mathbf{A}_y$ and $\mathbf{B}_y$ is sliced both horizontally and vertically. For example, $\mathbf{A}_y$ is sliced into $\frac{N_y}{t} \times \frac{N_y}{s}$ matrices and $\mathbf{B}_y$ is sliced into $\frac{N_y}{s} \times \frac{N_y}{t}$. We choose s and t such that they satisfy $st = m$ [13] and a copy is the m-th fractions of matrices **A** and **B**. Each edge server has a storage constraint that limits the edge server to store only m fractions of matrices **A** and **B** [13]. The recovery threshold k is defined [13] as:

$$k = t^2(2s - 1). \tag{1}$$

The processing by the workers may take a longer time when it is currently occupied with some other tasks. Therefore, the processing in the offloaded tasks is perceived to have failed if the duration exceeds the threshold time limit [44]. To recover the computed task, the sum of returned offloaded copies and locally computed copies must be greater than or equal to recovery threshold k.

The decision scenario of phase one and phase two are explained using recovery threshold k. In phase one, $|\mathcal{X}|$ scenarios may occur. Mobile charging station y chooses the UAV type to be used. In phase two, three scenarios may occur.

- The UAV can compute all copies locally, $M_{y,x^{\bar{t}}}^{(L)} \geq k$, where $M_{y,x^{\bar{t}}}^{(L)}$ indicates the number of copies that type x UAV from mobile charging station y computes locally at time slot $\bar{t}$ and k is defined in (1).
- The UAV can offload all copies to BS f, $M_{y,x^{\bar{t}},f}^{(O)} \geq k$, where $M_{y,x^{\bar{t}},f}^{(O)}$ denotes the number of copies that are offloaded to the edge servers in BS f at time slot t by the type x UAV from mobile charging station y.
- The UAV can compute some copies locally and offload some copies to the edge servers in BS f, $M_{y,x^{\bar{t}}}^{(L)} + M_{y,x^{\bar{t}},f}^{(O)} \geq k$

The final output can be decoded from all the return copies $M_{y,x^{\bar{t}}}^{(L)} + M_{y,x^{\bar{t}},f}^{(O)}$.

Similar to [13], the type x UAV associated with mobile charging station y uses $d_{enc} = N_y^2(M_{y,x^{\bar{t}}}^{(L)} + M_{y,x^{\bar{t}},f}^{(O)})$ symbols for encoding of matrices and $d_{dec} = N_y^2 t^2 (2s - 1) \log^2 t^2(2s - 1)$ to decode the returned matrices. Each copy contains m-th fractions of matrices **A** and **B**. UAV will transmit $d_{comm}^{to} = \frac{N_y^2}{m}$ symbols to each of the edge servers. Each copy requires $d_{cmp} = \frac{N_y^3}{mt}$ symbols for computation. After computation is completed, the edge server will send $d_{comm}^{fr} = \frac{N_y^2}{t^2}$ symbols back to the UAV.

TABLE I
LIST OF KEY NOTATION

Symbol	Definition
$\mathcal{T}$	Set of time slots while $\bar{t} \in \mathcal{T}$ denotes the time slot index
$\bar{\mathcal{P}}^t$	Set of periods in $\bar{t}$ while $\bar{p}^t \in \bar{\mathcal{P}}^t$ denotes the period index
$\mathcal{X}$	Set of UAV types while $x \in \mathcal{X}$ denotes the UAV type index
$\mathcal{Y}$	Set of mobile charging stations while $y \in \mathcal{Y}$ denotes the mobile charging station index
$\mathcal{F}$	Set of BSs while $f \in \mathcal{F}$ denotes the BS index
k	Recovery threshold
z	The number of stages in multi-stage SIP
$\gamma^{\bar{t}}$	Set of weather condition scenarios in $\bar{t}$ while $\mu_i^{\bar{t}} \in \gamma^{\bar{t}}$ denotes the weather condition scenario index
$\Theta^{\bar{t}}$	Set of demand scenarios in $\bar{t}$ while $\lambda_i^{\bar{t}} \in \Theta^{\bar{t}}$ denotes the demand scenario index
$\Omega^{(\bar{z},\bar{t})}$	Set of shortfall scenarios in stage $\bar{z} - 1$, where $2 < \bar{z} \leq z$, and $\bar{t}$ while $\omega_i^{(\bar{z},t)} \in \Omega^{(\bar{z},\bar{t})}$ denotes the shortfall scenario index
$T_y^{x\bar{t}}$	Binary variable at time slot $\bar{t}$ for mobile charging station y indicates whether type x UAV is used.
$T_y^{(\bar{t},X)}(\mu_i^{\bar{t}})$	Binary variable at time slot $\bar{t}$ for mobile charging station y indicates whether a correction on-demand type-X UAV is used in scenario $\mu_i^{\bar{t}}$, and X represents the largest UAV type.
$M_f^{(s)}$	Binary variable to indicate whether the edge servers in BS f will be used or not
$M_{y,x\bar{t}}^{(L,2)}(\lambda_i^{\bar{t}})$	Decision variable represents the number of copies computed locally by the type x UAV that is associated with mobile charging station y in stage 2, time slot $\bar{t}$ and scenario $\lambda_i^{\bar{t}}$
$M_{y,x\bar{t},f}^{(O,2)}(\lambda_i^{\bar{t}})$	Decision variable represents the number of copies offloaded to the edge servers in BS f by the type x UAV that is associated with mobile charging station y in stage 2, time slot $\bar{t}$ and scenario $\lambda_i^{\bar{t}}$
$\vdots$	$\vdots$
$M_{y,x\bar{t}}^{(L,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$	Decision variable represents the number of copies computed locally by the type x UAV that is associated with mobile charging station y in time slot $\bar{t}$, scenario $\lambda_i^{\bar{t}}$, ..., scenario $\omega_i^{(\bar{z},\bar{t})}$ and stage $\hat{z}$, where $1 < \hat{z} \leq z$
$M_{y,x\bar{t},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$	Decision variable represents the number of copies offloaded to the edge servers in BS f by the type x UAV that is associated with mobile charging station y in time slot $\bar{t}$, scenario $\lambda_i^{\bar{t}}$, ..., scenario $\omega_i^{(\bar{z},\bar{t})}$ and stage $\hat{z}$

B. UAV Hovering Energy

The propulsion energy consumption is needed to provide the UAV with sufficient thrust to support its movement. Note that we drop the time notation for ease of presentation. The propulsion power of a rotary-wing UAV with speed V can be modeled as follows [42]:

$$P_x(V) = P_{x,0}\left(1 + \frac{3V^2}{U_{tip}^2}\right) +$$
$$P_{x,1}\left(\sqrt{1 + \frac{V^4}{4v_0^4} - \frac{V^2}{2v_0^2}}\right)^{\frac{1}{2}} + \frac{1}{2}d_0\rho\bar{t}\mathbb{A}V^3, \quad (2)$$

where

$$P_{x,0} = \frac{\delta}{8}\rho \mathbf{s}\mathbb{A}\triangle_x^3 R^3, \quad (3)$$

$$P_{x,1} = (1+r)\frac{W_x^{3/2}}{\sqrt{2\mathbb{A}\rho}}. \quad (4)$$

$P_{x,0}$ and $P_{x,1}$ are two constants related to UAV's weight, rotor radius, air density, etc. U_{tip} denotes the tip speed of the rotor blade, v_0 is known as the mean rotor induced velocity in hover, d_0 and s are the fuselage drag ratio and rotor solidity, respectively. ρ and $\mathbb{A}$ are the air density and rotor disc area, respectively. r is the incremental correction factor to induced power. W_x is the type x UAV weight, δ is the profile drag coefficient, and $\triangle_x$ denotes blade angular velocity of the type x UAV. By substituting $V = 0$ into (2) [42], we obtain the power consumption for hovering status as follows:

$$P_{x,h} = P_{x,0} + P_{x,1}. \quad (5)$$

C. Local Computing Model

When one copy is processed locally, the local computation execution time of the type x UAV is expressed as [45]:

$$t_{y,x}^{local}(N_y) = \frac{\mathcal{C}_x d(\frac{N_y^3}{mt})}{\tau_x}, \quad (6)$$

where $\mathcal{C}_x$ is the number CPU cycles needed to process a bit, τ_x denotes the total CPU computing capability of the type x UAV, and $d(\cdot)$ is a function to translate the number of symbols to the number of bits for computation, i.e., if the 16 Quadrature Amplitude Modulation (QAM) is used, each symbol carries 4 bits [46]. The type x UAV takes $t_{y,x}^{enc}$ seconds to encode one copy of the matrices, and it is expressed as follows:

$$t_{y,x}^{enc}(N_y) = \frac{\mathcal{C}_x d(N_y^2)}{\tau_x}. \quad (7)$$

After the type x UAV obtains at least k copies, it will take $t_{y,x}^{dec}$ seconds to decode. $t_{y,x}^{dec}$ is defined as follows:

$$t_{y,x}^{dec}(N_y) = \frac{\mathcal{C}_x d(N_y^2 t^2(2s-1)\log^2 t^2(2s-1))}{\tau_x}. \quad (8)$$

D. UAV Communication Model

We assume that each UAV is allocated with an orthogonal spectrum resource block to avoid the co-interference among the UAVs [47]. The transmission rate from the type x UAV which is associated with mobile charging station y to the edge servers in BS f can be represented as [48]:

$$r_{y,x,f} = B_x \log_2(1 + P_x^C h_{y,x,f}/N_o), \quad (9)$$

where the wireless transmission power of the type x UAV at time slot $\bar{t}$ is expressed as P_x^C and B_x is the bandwidth. $h_{y,x,f}$

is the channel gains, and N_0 is the variance of complex white Gaussian noise. The UAV to edge server communication is most likely to be dominated by LoS links. Therefore, the air-to-ground channel power gain from the type x UAV to the edge servers in BS f can be modeled as follows [49]:

$$h_{y,x,f} = \frac{\beta_0}{\mathbb{D}^2_{y,x,f}}, \quad (10)$$

where

$$\mathbb{D}^2_{y,x,f} = (a_y - a_f)^2 + (b_y - b_f)^2 + (H_{y,x} - H_f)^2. \quad (11)$$

$\mathbb{D}_{y,x,f}$ denotes the distance between the type x UAV that is associated with mobile charging station y and the edge servers in BS f, and β_0 represents the reference channel gain at distance $d_0 = 1$m in an urban area [49]. We assume that for all the edge servers in the same BS f, they will have the same $\mathbb{D}^2_{y,x,f}$. The transmission time to offload one copy of matrix from the type x UAV to a edge server in BS f can be given as follows:

$$t^{to}_{y,x,f}(N_y) = \frac{d(\frac{N_y^2}{m})}{r_{y,x,f}}. \quad (12)$$

The energy $e_{y,x}$ required by the type x UAV to receive data from the edge server in BS f is defined as follows [50]:

$$e_{y,x}(N_y) = P^{re}_x \frac{d(\frac{N_y^2}{t^2})}{r_{f,y,x}}, \quad (13)$$

where P^{re}_x is the receiving power of type x UAV. $r_{f,y,x}$ is the transmission rate from edge servers in BS f to type x UAV which is associated with mobile charging station y. It is define similar to (9).

E. Problem Formulation

As an illustration, Fig. 2 depicts the decision process of the system across all the time slots. UAV type allocation is performed in each time slot $\bar{t}$. Throughout $\bar{t}$, the mobile charging stations will use the same UAV type to perform the task allocation in each period. Fig. 3 shows a detailed diagram that zooms into one-period in one-time slot, and it is explained in details in both Sections IV and V. In Section IV, the application owner $\bar{A}_3$ pays a reservation cost to make an advance booking for a different time slot for the use of the UAVs. The application owner can observe the weather condition via weather forecast as it may affect the status of the UAV. If the wind is too strong and the UAV used is not large type, the UAV may crash as it has insufficient energy to hover against the wind [51]. For example, a strong wind has high kinetic energy, kinetic energy leads to a higher density of the air, and it increases the UAV hovering power consumption. Low wind speed is referred to as wind speed that is less than $11m/s$ and turbulence level < 5 [52]. As a result, $\bar{A}_3$ has to request an on-demand X type UAV to perform the job. In order for SCOS to model the weather uncertainty, we formulate the two-stage SIP to optimize the UAV type allocation.

To achieve cost minimization, phase two (task allocation) in Section V has to consider two sources of uncertainty, i.e., the demand uncertainty and shortfall uncertainty. Demand uncertainty refers to the task required by the applications, such as traffic monitoring can be of different sizes, i.e., the task's size depends on the image resolution. Shortfall uncertainty refers to if the UAV offloads the copies to the edge servers, the computed copies may not return, or the number of copies returned is less than the recovery threshold due to delays and link failure. Therefore, we use multi-stage SIP to model the demand uncertainty to optimize the number of copies to compute locally $M^{(L)}_{y,x^{\bar{t}}}$ and offload $M^{(O)}_{y,x^{\bar{t}},f}$. For example, when the recovery threshold is $k = 4$ and the UAV decide to offload two copies of the task for the edge servers in BS f to compute, i.e., $M^{(O)}_{y,x^{\bar{t}},f} = 2$. Therefore, the UAV has to compute at least two more copies locally to match the recovery threshold $M^{(L)}_{y,x^{\bar{t}}} = 2$. In time slot $\bar{t}$, type x UAV will hover in the sky for a threshold time limit $t^{thresh}_{y,x^{\bar{t}}}$ to wait for the offloaded copies to return. Without loss of generality, $t^{thresh}_{y,x^{\bar{t}}}$ is set as the worst-case scenario, i.e., the time required to compute all copies locally by the UAV, i.e., $M^L_{y,x^{\bar{t}}} = k$. However, there is a probability that the edge servers in BSs may fail, i.e., the computed task is not returned to the UAV before $t^{thresh}_{y,x^{\bar{t}}}$. As a result, the UAV cannot complete the full task if the total returned copies are less than 4. When the UAV fails to receive sufficient number of copies, there are shortfalls, and hence, the UAV has to re-compute the shortfalls locally or re-offload to the edge servers. Since the UAV has limited computation capabilities, it can choose to re-compute the shortfall locally or re-offload to the edge servers until z stages, where z is the number of times of re-computations. In the meantime, the UAV has to continue hover in the sky when performing the re-computation. In order to model the shortfall uncertainty, we formulate z-stage SIP to optimize the numbers of copies to compute locally and to offload, and we can also optimize the number of stages required. Hence, this scheme will minimize the overall network cost, and the system model of this network is formulated as follows:

$$\min_{T^{x\bar{t}}_y, T^{x\bar{t}}_y(\mu^{\bar{t}}_i), M^{(s)}_f, \ldots, M^{(O,\hat{z})}_{y,x^{\bar{t}},f}(\lambda^{\bar{t}}_i, \ldots, \omega^{(\bar{z},\bar{t})}_i)} :$$

$$\sum_{\bar{t} \in \mathcal{T}} \mathcal{O}^{UAV}_{allocation}(\bar{t}) + \sum_{\bar{t} \in \mathcal{T}} \sum_{\bar{p}^{\bar{t}} \in \mathcal{P}^{\bar{t}}} \mathcal{O}^{Task}_{allocation}(\bar{p}^{\bar{t}}), \quad (14)$$

subject to: (19)-(22), (44)-(56)
where $\mathcal{O}^{UAV}_{allocation}(\bar{t})$ is the UAV type allocation cost in time slot $\bar{t}$ and it is defined in (17) in Section IV. $\mathcal{O}^{Task}_{allocation}(\bar{p}^{\bar{t}})$ is the task allocation cost within period $\bar{p}$ and period $\bar{p}$ is in time slot $\bar{t}$. The task allocation is defined in (35) in Section V-B.

IV. PHASE ONE: UAV TYPE ALLOCATION

This section introduces the SIP to optimize phase one (UAV type allocation) in SCOS by minimizing the total allocation. As described in Section III, the application owner $\bar{A}_3$ needs to make a reservation in advance to secure certain types of UAVs, which are own by $\bar{A}_1$. However, the weather condition is unknown and may vary at a different time slot $\bar{t}$. If the wind is too strong, the UAV is required to use more energy to hover at a fixed location [51]. As a result, the UAV will crash with insufficient energy, and the application owner has to make an
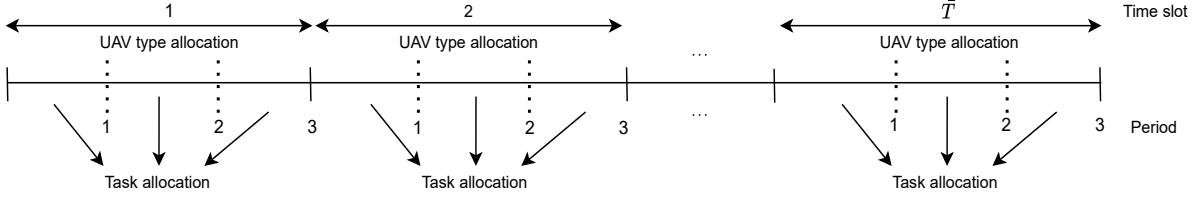
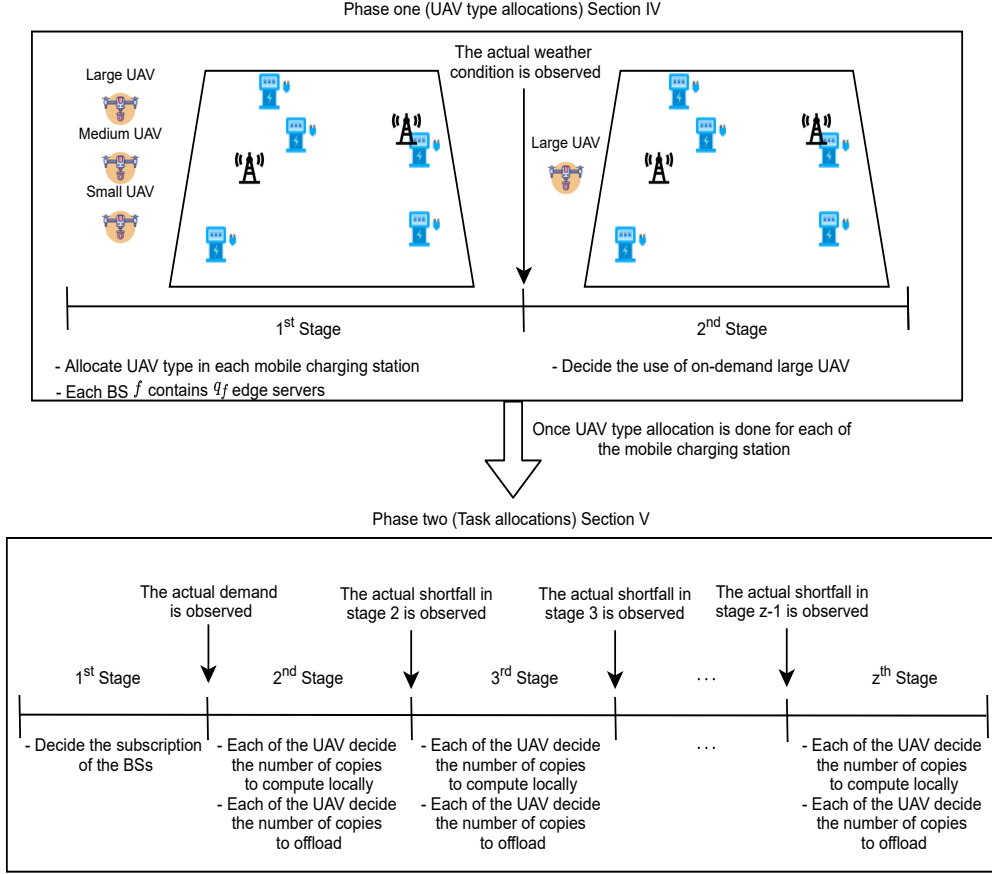Fig. 2. The decision process of the system across all the time slots $\bar{T}$.

Fig. 3. Decision making process of the system in one time slot with the using of three different types of UAV, $\mathcal{X} = \{1 : small, 2 : medium, 3 : large\}$.

on-demand request with a X type UAV. Fig. 3 illustrates the decision-making process of the system with the use of three UAV types, 1, 2 and 3, which represents small, medium and large, respectively.

Hence, we formulate this scheme as the two-stage SIP model.

- **First stage:** The application owner makes a reservation on the types of UAVs to be used. The decision will be made based on the available cost information and the probability distribution of the weather condition.
- **Second stage:** After knowing the exact weather condition, the application owner decides the correction action, which is the on-demand request to use the largest type X UAV.

Let $\mu_i^{\bar{t}} = \{G_1^{\bar{t}}(\mu_i^{\bar{t}}), \ldots, G_y^{\bar{t}}(\mu_i^{\bar{t}})\}$ denote weather condition scenarios i of all mobile charging stations at time slot $\bar{t}$. The

set of all weather scenarios is denoted by $\gamma^{\bar{t}}$, i.e., $\mu_i^{\bar{t}} \in \gamma^{\bar{t}}$ [53]. $G_y^{\bar{t}}(\mu_i^{\bar{t}})$ represents a binary parameter of the weather condition at time slot $\bar{t}$. For tractability, we only consider that each mobile charging station experiences only two types of weather condition. As shown in Table II, $G_y^{\bar{t}}(\mu_i^{\bar{t}}) = 1$ means that at time slot $\bar{t}$, the wind is strong in mobile charging station y and the UAV has crashed, and $G_y^{\bar{t}}(\mu_i^{\bar{t}}) = 0$ means otherwise. $\mathcal{P}(\mu^{\bar{t}})$ denotes the probability if scenario $\mu^{\bar{t}} \in \gamma^{\bar{t}}$ is realized. All of the scenarios can be obtained from historical records [31] or weather forecast. The cost function is proportional to the resources used. In total, there are $|\mathcal{X}| + 2$ types of payments. Note that we drop the time notation.

- C_r^x is the reservation cost for the type x UAV. It is defined as follows:

$$C_r^x = \alpha_1 B_x, \tag{15}$$

TABLE II
WEATHER UNCERTAINTY

Symbol	Definition
$G_y^{\bar{t}}(\mu_i^{\bar{t}}) = 0$	At time slot $\bar{t}$, the wind is weak or there is no wind at mobile charging station y. Low wind speed is referring to wind speed that is less than 11 m/s and turbulence level < 5 [52]
$G_y^{\bar{t}}(\mu_i^{\bar{t}}) = 1$	At time slot $\bar{t}$, the wind is strong at mobile charging station y. High wind speed is referring to wind speed that is greater than 11 m/s and turbulence level > 5 [52]

where B_x is the battery capacity of the type x UAV and α_1 is the cost coefficient.

- C_o^X is the on-demand cost for the type X UAV, which represents the largest UAV type. It is defined as follows:

$$C_o^X = \alpha_2 B_X, \qquad (16)$$

where α_2 is the cost coefficient with a similar role to α_1 and $\alpha_2 > \alpha_1$. B_X is the battery capacity of the type X UAV.

- C_p is the penalty cost. This penalty cost is the repair cost for the crashed UAV.

We formulate the UAV type allocation as a two-stage SIP model. There are $|\mathcal{T}|(|\mathcal{X}|+1)$ decision variables in this model.

- $T_y^{x\bar{t}}$ is a binary variable at time slot $\bar{t}$ for mobile charging station y indicates whether type x UAV is used. When $T_y^{x\bar{t}} = 1$, at time slot $\bar{t}$, mobile charging station y uses type x UAV and $T_y^{x\bar{t}} = 0$ means otherwise.
- $T_y^{(\bar{t},X)}(\mu_i^{\bar{t}})$ is a binary variable at time slot $\bar{t}$ for mobile charging station y indicates whether a correction on-demand type X UAV is used in scenario $\mu_i^{\bar{t}}$, and X represents the largest UAV type. When $T_y^{(\bar{t},X)}(\mu_i^{\bar{t}}) = 1$, at time slot $\bar{t}$, mobile charging station y performs a correction action by using the largest type-X UAV in scenario $\mu_i^{\bar{t}}$ and $T_y^{(\bar{t},X)}(\mu_i^{\bar{t}}) = 0$ means otherwise.

The objective function given in (17) and (18) is to minimize the cost of the UAV type allocation. The expressions in (17) and (18) represent the first- and second-stage SIP, respectively. The SIP formulation can be expressed as follows:

$$\min_{T_y^{x\bar{t}}, T_y^{(\bar{t},X)}(\mu_i^{\bar{t}})} :$$

$$\sum_{\bar{t}\in\mathcal{T}} \mathcal{O}_{allocation}^{UAV}(\bar{t}) = \sum_{\bar{t}\in\mathcal{T}}\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} T_y^{x\bar{t}} C_r^x + \mathbb{E}\left[\mathcal{Q}(T_y^{x\bar{t}}(\mu_i^{\bar{t}}))\right],$$
$$(17)$$

where

$$\mathcal{Q}(T_y^{x\bar{t}}(\mu_i^{\bar{t}})) = \sum_{\bar{t}\in\mathcal{T}}\sum_{\mu_i^{\bar{t}}\in\gamma^{\bar{t}}} \mathcal{P}(\mu_i^{\bar{t}})\sum_{y\in\mathcal{Y}} T_y^{(\bar{t},X)}(\mu_i^{\bar{t}})(C_o^X + C_p),$$
$$(18)$$

subject to:

$$\sum_{x\in\mathcal{X}} T_y^{x\bar{t}} = 1, \qquad \forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \quad (19)$$
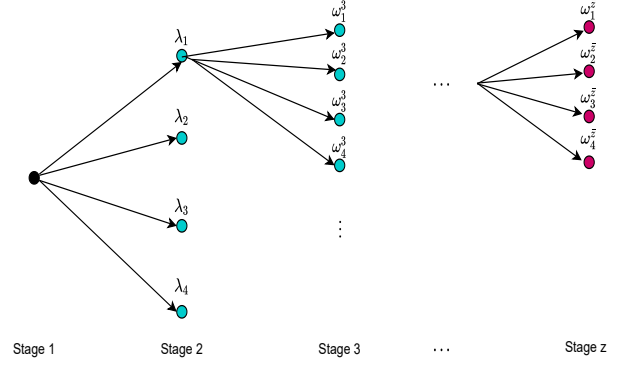

Fig. 4. A scenario tree structure for z-stage SIP in task allocation.

$$\sum_{x\in\mathcal{X}\setminus\{X\}} T_y^{x\bar{t}}(1 - G_y(\mu_i^{\bar{t}})) + T_y^{X\bar{t}} + T_y^{(\bar{t},X)}(\mu_i^{\bar{t}}) = 1,$$

$$\forall \bar{t}\in\mathcal{T}, \forall\mu_i^{\bar{t}}\in\gamma^{\bar{t}}, \forall y\in\mathcal{Y}, \quad (20)$$

$$T_y^{x\bar{t}} \in \{0,1\}, \qquad \forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall x\in\mathcal{X}, \quad (21)$$
$$T_y^{(\bar{t},X)}(\mu_i^{\bar{t}}) \in \{0,1\}, \quad \forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall\mu_i^{\bar{t}}\in\gamma^{\bar{t}}. \quad (22)$$

The constraint in (19) ensures that the application owner makes a reservation on the types of UAV. On the other hand, (20) ensures that the UAV crashes because of strong wind if the application owner previously reserves a UAV that is not largest type X. Then, the application owner has to perform a correction action by using the largest type X on-demand UAV. (21) and (22) are boundary constraints for the decision variables.

V. PHASE TWO: TASK ALLOCATION

Once the types of the UAVs are optimized from phase one in SCOS, we introduce the Deterministic Integer Programming (DIP) and SIP to optimize phase two (the number of copies to compute locally and to offload) by minimizing the UAV network cost. Note that for simplicity, we drop notation $\bar{p}^{\bar{t}}$ from phase two task allocation.

A. Deterministic Integer Programming System Model

In an ideal case, when the actual demand, which is the actual matrix size and the number of shortfalls, are precisely known ex-ante, the UAVs can choose the exact number of copies to compute locally or offload. Therefore, the correction for the shortfall is not needed, and the correction cost is zero. Similar to [54], the cost function is proportional to the UAVs offloaded data and to their demand for consuming computation resources. Choosing different sizes of UAVs will affect the payment value. In total, five types of payments are considered in DIP.

- $\bar{C}_f$ is the subscription cost for the edge servers in BS f.
- $\bar{C}_{y,x}$ denotes the UAV local computation cost and encoding cost for computing of one copy, i.e.,

$$\bar{C}_{y,x}(\mathcal{D}) = \alpha_3(t_{y,x}^{local}(\mathcal{D}) + t_{y,x}^{enc}(\mathcal{D})), \qquad (23)$$

where α_3 is the cost coefficient associated to the energy consumption and $\mathcal{D}$ is the actual demand.

- $C_{y,x^{\bar{t}},f}$ denotes the offloading cost and it consists of three parts. The first part is related to the transmission $t^{to}_{y,x^{\bar{t}},f}$ and encoding delay $t^{enc}_{y,x}$. The second part is the type x UAV energy consumption cost and the last part C_f is the service cost for edge servers in BS f. It is modeled as follows:

$$C_{y,x^{\bar{t}},f}(\mathcal{D}) = \alpha_3(t^{to}_{y,x^{\bar{t}},f}(\mathcal{D}) + t^{enc}_{y,x}(\mathcal{D})) \\ + \alpha_4 e_{y,x^{\bar{t}}}(\mathcal{D}) + C_f, \quad (24)$$

where α_4 is the cost coefficient with a similar role to α_3.

- $\underline{C}^{thresh}_{y,x^{\bar{t}}}$ denotes the hovering cost for $t^{thresh}_{y,x^{\bar{t}}}$ seconds. They are defined as follows:

$$\underline{C}^{thresh}_{y,x^{\bar{t}}} = t^{thresh}_{y,x^{\bar{t}}} k\alpha_5 P_{x^{\bar{t}},h}, \quad (25)$$

where α_5 is the cost coefficient with a similar role to α_3.

- $\hat{C}_{y,x}$ denotes the type x UAV decoding cost for the returned matrices as follows:

$$\hat{C}_{y,x}(\mathcal{D}) = \alpha_3 t^{dec}_{y,x}(\mathcal{D}). \quad (26)$$

A DIP can be formulated and minimize the total cost of the UAVs as follows:

$$\min_{M^{(s)}_f,\ldots,M^{(O)}_{y,x^{\bar{t}},f}} :$$

$$\sum_{\bar{t}\in\mathcal{T}} \mathcal{O}^{Task}_{allocation}(\bar{t}) = \sum_{\bar{t}\in\mathcal{T}}\sum_{f\in\mathcal{F}} M^{(s)}_f \bar{C}_f +$$

$$\sum_{\bar{t}\in\mathcal{T}}\sum_{y\in\mathcal{Y}}\sum_{f\in\mathcal{F}}\left(M^{(L)}_{y,x^{\bar{t}}}\bar{C}_{y,x}(\mathcal{D}) + M^{(TH)}_{y,x^{\bar{t}},f}\underline{C}^{thresh}_{y,x} + \right.$$

$$\left. M^{(O)}_{y,x^{\bar{t}},f}C_{y,x,f}(\mathcal{D}) + \hat{C}_{y,x}(\mathcal{D}) \right), \quad (27)$$

subject to:

$$\sum_{y\in\mathcal{Y}} M^{(O)}_{y,x^{\bar{t}},f} \le \sigma M^{(s)}_f, \qquad \forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \quad (28)$$

$$\sum_{y\in\mathcal{Y}} M^{(O)}_{y,x^{\bar{t}},f} \le q_f, \qquad \forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \quad (29)$$

$$M^{(O)}_{y,x^{\bar{t}},f} \le \sigma M^{(TH)}_{y,x^{\bar{t}},f}, \qquad \forall y\in\mathcal{Y}, \forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \quad (30)$$

$$\sum_{f\in\mathcal{F}} M^{(O)}_{y,x^{\bar{t}},f} \ge S^{\bar{t}}_y - M^{(L)}_{y,x^{\bar{t}}}, \qquad \forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \quad (31)$$

$$M^{(L)}_{y,x^{\bar{t}}} + \sum_{f\in\mathcal{F}} M^{(O)}_{y,x^{\bar{t}},f} - (S^{\bar{t}}_y - M^{(L)}_{y,x^{\bar{t}}}) \ge k, \\ \forall y\in\mathcal{Y}, \forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \quad (32)$$

$$M^{(s)}_f, M^{(TH)}_{y,x^{\bar{t}},f} \in \{0,1\}, \qquad \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \quad (33)$$

$$M^{(L)}_{y,x^{\bar{t}}}, M^{(O)}_{y,x^{\bar{t}},f} \in \{0,1,\ldots\}, \quad \forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}. \quad (34)$$

$M^{(s)}_f$ is a binary variable to indicate whether the edge servers in BS f will be used or not. $M^{(TH)}_{y,x^{\bar{t}},f}$ is a binary variable to indicate in time slot $\bar{t}$ whether the type x UAV which is associated with mobile charging station y will choose to offload or not. When $M^{(TH)}_{y,x^{\bar{t}},f} = 1$, in time slot $\bar{t}$ the UAV associated with mobile charging station y choose to offload some of the copies to BS f and $M^{(TH)}_{y,x^{\bar{t}},f} = 0$ means otherwise. The objective function in (27) is to minimize UAVs' total cost involving the UAVs' local computation cost and the UAVs' offloading cost. The constraint in (28) ensures that the subscription cost of the edge servers in the BS will be paid if they are used in any of the stages, where σ is a sufficiently large number. (29) ensures that the total number of copies offloaded to the edge servers must not exceed the total number of edge servers in BSs. (30) ensures that the threshold cost will be paid if the UAV perform offloading action. (31) ensures that the shortfalls should only exist if the number of copies offloaded is more than or equal to the shortfalls. (32) ensures that the number of copies computed locally and offloaded have to be at least equal to or larger than recovery threshold k. (33) indicates $M^{(s)}_f$ and $M^{(TH)}_{y,x^{\bar{t}},f}$ are binary variables. (34) indicates that $M^{(L)}_{y,x^{\bar{t}}}$ and $M^{(O)}_{y,x^{\bar{t}},f}$ are positive decision variables.

B. Stochastic Integer Programming System Model

This section introduces the SIP to minimize the total cost of the network by optimizing the number of copies to compute locally and to offload to the edge servers in BSs. The first stage consists of all decisions that have to be selected before the demand and shortfall are realized and observed. In the second stage and onwards, decisions are allowed to adapt to this information. In each stage, decisions are limited by constraints that may depend on previous decisions and observations.

As described in Section III, there is a subscription cost when the service provider $\bar{A}_3$ wants to use the edge servers in BSs for computation. Then, without knowing the demand, the type x UAV can decides the number of copies to compute locally $M^{(L)}_{y,x^{\bar{t}}}$ and the number of copies to offload $M^{(O)}_{y,x^{\bar{t}},f}$.

The computation process in the edge servers are not very reliable, as the edge servers might be processing some other task or congested. As a result, the computation time is much longer than the threshold limit $t^{thresh}_{y,x^{\bar{t}}}$. Therefore, if a copy is offloaded, there is a probability that the computation might fail, and it will require the type x UAV to re-offload again or compute it locally.

Hence, we formulate this framework as a z-stage SIP model.

- **First stage:** The application owner $\bar{A}_3$ decides to use the edge servers in BS f or not. The decision will be made based on the available cost information, the probability distribution of the demand, and the shortfall.
- **Second stage:** After knowing the exact demand, the application owner $\bar{A}_3$ decides the number of copies that are computed locally and the number of copies to be offloaded to the edge servers in BS f.
- **Third stage:** After knowing the exact shortfall in the previous stage, the $\bar{A}_3$ performs a correction action to re-decide the number of copies that is computed locally

and the number of copies to be offloaded to the edge servers in BS f.

$$\vdots$$

- z **stage:** After knowing the exact shortfall in the z-1 stage, $\bar{A}_3$ performs a correction action to re-decide the number of copies that is computed locally and the number of copies to be offloaded to the edge servers in BS f. To promote the UAV to complete the task, a huge penalty will occur if there is still a shortfall in stage z.

Let $\lambda_i^{\bar{t}} = \{D_1^{\bar{t}}, \ldots, D_y^{\bar{t}}\}$ denote the UAV demand scenario i across all mobile charging station y in time slot $\bar{t}$ and the set of demand scenarios is denoted by $\Theta^{\bar{t}}$, i.e., $\lambda_i^{\bar{t}} \in \Theta^{\bar{t}}$ [53]. $D_y^{\bar{t}}$ contains a discrete value from a finite set $\mathcal{W} = \{1, \ldots, W\}$, it represents the size of the task in UAV that is associated with mobile charging station y. Specifically, $D_y^{\bar{t}} = 1000$ means that in time slot $\bar{t}$ the matrix that UAV receives is in the size of 1000×1000. Let $\omega_i^{(\bar{z},\bar{t})} = \{\mathbb{F}_1^{(\bar{z},\bar{t})}, \ldots, \mathbb{F}_y^{(\bar{z},\bar{t})}\}$ denote the i-th shortfall scenario of the UAV in time slot $\bar{t}$ that is associated with its individual mobile charging station in stage $\bar{z}-1$, where $2 < \bar{z} \leq z$. The set of shortfall scenarios is denoted by $\Omega^{(\bar{z},\bar{t})}$, i.e., $\omega_i^{(\bar{z},\bar{t})} \in \Omega^{(\bar{z},\bar{t})}$. $\mathbb{F}_y^{(\bar{z},\bar{t})}$ represents a binary parameter of the shortfall in time slot $\bar{t}$ from the type x associated with mobile charging station y in stage $\bar{z} - 1$. For example, $\mathbb{F}_y^{(\bar{z},\bar{t})} = 1$ means that, in time slot $\bar{t}$ from the copies that the UAV has offloaded, at least $A_y^{(\bar{z},\bar{t})}$ copy did not return. As a result, the total number of copies that the UAV currently has is less than k, and $\mathbb{F}_y^{(\bar{z},\bar{t})} = 0$ means otherwise. In stage z, $\mathbb{F}_y^{(\bar{z},\bar{t})} = 0$ when $\mathbb{F}_y^{(\bar{z}-1,\bar{t})} = 0$. When there is no shortfall in the previous stage then, there will not be any shortfall in the next stage. Fig. 4 illustrates the stages with four scenarios at each stage. All of the scenarios can be obtained from the historical records.

The cost function used in SIP is similar to DIP with an additional penalty cost $\tilde{C}$. $\tilde{C}$ occurs when the UAV still has to perform a corrective action. In total, six types of payments are considered in z-stage SIP.

We formulate the task allocation as the z-stage SIP model. There are $|\bar{\mathcal{T}}|(z(f+2) - 1)$ decision variables in this model.

- $M_f^{(s)}$ is a binary variable to indicate whether the edge servers in BS f will be used or not. When $M_f^{(s)} = 1$, edge servers in BS f will be used and $M_f^{(s)} = 0$ means otherwise.
- $M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}})$ indicates in time slot $\bar{t}$ the number of copies to be offloaded to the edge servers in BS f by type x UAV which is associated with mobile charging station y in stage 2.
- $M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}})$ indicates in time slot $\bar{t}$ the number of copies computed locally by the type x UAV which is associated with mobile charging station y in stage 2.
- $M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}})$ is a binary variable to indicate in time slot $\bar{t}$ whether the type x UAV which is associated with mobile charging station y choose to offload or not in stage 3. When $M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}}) = 1$, the type x UAV associated with mobile charging station y chooses to offload some of the copies to the edge servers in BS f and $M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}}) = 0$ means otherwise.

$$\vdots$$

- $M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$ indicates in time slot $\bar{t}$ the number of copies to be offloaded by type x UAV which is associated with mobile charging station y to the edge servers in BS f in stage $\hat{z}$, where $1 < \hat{z} \leq z$.
- $M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$ indicates in time slot $\bar{t}$ the number of copies that the type x UAV which is associated with mobile charging station y is computing locally in stage $\hat{z}$.
- $M_{y,x^{\bar{t}},f}^{(TH,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$ indicates is a binary variable to indicate in time slot $\bar{t}$ whether the type x UAV which is associated with mobile charging station y choose to offload or not in stage $\hat{z}$.

The objective function given in (35) - (38) is to minimize the total cost of the network. The expressions in (35), (36), (37) and (38) represent the first, second, third and up till z stage objectives, respectively. $\mathcal{P}(\lambda_i^{\bar{t}})$ and $\mathcal{P}(\omega_i^{(\bar{z},\bar{t})})$ denote the probabilities if scenarios $\lambda_i^{\bar{t}} \in \Theta^{\bar{t}}$ and $\omega_i^{(\bar{z},\bar{t})} \in \Omega^{(\bar{z},\bar{t})}$ are realized, respectively. The SIP formulation can be expressed as follows:

$$\min_{M_f^{(s)}, \ldots, M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})} :$$

$$\sum_{\bar{t} \in \mathcal{T}} \mathcal{O}_{allocation}^{Task}(\bar{t}) = \sum_{\bar{t} \in \mathcal{T}} \sum_{f \in \mathcal{F}} M_f^{(s)} \bar{C}_f + \mathbb{E}\left[\mathcal{Q}(M_f^{(s)}(\lambda_i^{\bar{t}}))\right], \tag{35}$$

where

$$\mathcal{Q}(M_f^{(s)}(\lambda_i^{\bar{t}})) = \sum_{\bar{t} \in \mathcal{T}} \sum_{\lambda_i^{\bar{t}} \in \Theta^{\bar{t}}} \mathcal{P}(\lambda_i^{\bar{t}}) \sum_{y \in \mathcal{Y}} \sum_{f \in \mathcal{F}} \left(M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}}) \right.$$
$$\bar{C}_{y,x}(D_y(\lambda_i^{\bar{t}})) + M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}) C_{y,x,f}(D_y(\lambda_i^{\bar{t}})) +$$
$$\underline{C}_{y,x}^{thresh} + \hat{C}_{y,x}(D_y(\lambda_i^{\bar{t}})) +$$
$$\left. \mathbb{E}\left[\mathcal{Q}(M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})}))\right]\right), \tag{36}$$

$$\mathcal{Q}(M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})})) = \sum_{\bar{t} \in \mathcal{T}} \sum_{\omega_i^{(3,\bar{t})} \in \Omega^{(3,\bar{t})}} \mathcal{P}(\omega_i^{(3,\bar{t})}) \sum_{y \in \mathcal{Y}} \sum_{f \in \mathcal{F}}$$
$$\left(M_{y,x^{\bar{t}}}^{(L,3)}(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})}) \bar{C}_{y,x}(D_y(\lambda_i^{\bar{t}})) + M_{y,x^{\bar{t}},f}^{(O,3)}(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})}) \right.$$
$$C_{y,x,f}(D_y(\lambda_i^{\bar{t}}) + M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})}) \underline{C}_{y,x}^{thresh}$$
$$\left. + \mathbb{E}\left[\mathcal{Q}(M_{y,x^{\bar{t}},f}^{(O,3)})(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})}, \omega_i^{(4,\bar{t})})\right]\right), \tag{37}$$

$$\vdots$$

$$\mathcal{Q}(M_{y,x^{\bar{t}},f}^{(O,\hat{z})})(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})}, \ldots, \omega_i^{(\bar{z},\bar{t})}) = \sum_{\bar{t} \in \mathcal{T}} \sum_{\omega_i^{(\bar{z},\bar{t})} \in \Omega^{(\bar{z},\bar{t})}}$$

$$\mathcal{P}(\omega_i^{(\bar{z},\bar{t})}) \sum_{y \in \mathcal{Y}} \sum_{f \in \mathcal{F}} \left(M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})}) \bar{C}_{y,x}(D_y(\lambda_i^{\bar{t}})) + \right.$$
$$M_{y,x^{\bar{t}},f}^{(TH,\bar{z})}(\lambda_i^{\bar{t}}) \underline{C}_{y,x}^{thresh} + M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$$
$$C_{y,x,f}(D_y(\lambda_i^{\bar{t}})) + \left. \mathbb{F}_y^{\bar{z}}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})}) \tilde{C}\right), \tag{38}$$

subject to: Please see Appendix A

The constraints in (44) and (45) are the same as the constraint (28). (46) and (47) are the same as (30). (48) and (49) ensure that if the previous stage does not have a shortfall, and then the shortfall for the next stage should be zero. The shortfalls should only exist if the number of copies offloaded in the previous stage is more than or equal to the shortfall. They also ensure that the number of shortfalls can be reduced when the UAV performs local computations. (50) ensures that the number of copies computed locally and offloaded should be at least k. (51) and (52) ensure that if the UAV has a shortfall in the previous stage, the UAV has to compute the shortfall locally or re-offload to the edge servers in the BS to match k, i.e., the recovery threshold for task completion. (53) and (54) ensure that the total number of copies offloaded to the edge servers in each stage must not exceed the total number of edge servers in BSs. (55) indicates $M_f^{(s)}$, $M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}}, \omega_i^{(3,\bar{t})})$, $\ldots, M_{y,x^{\bar{t}},f}^{(TH,\bar{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$ are binary variables. (56) indicates $M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}})$, $M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}}), \ldots, M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$, $M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}}, \ldots, \omega_i^{(\bar{z},\bar{t})})$ are positive decision variables.

To solve SIP, we assume that the probability distribution of all scenarios in set $\gamma^{\bar{t}}$, $\Theta^{\bar{t}}$, $\Omega^{(3,\bar{t})}, \ldots, \Omega^{(\bar{z},\bar{t})}$ are known [55], then, the complexity of the problem increases exponentially when the total number of scenarios across all the stages increases [55, 56]. For phase one in Section IV, the total number of decision variables can be calculated from (39), where t', y', and x' denote the total number of time slots, mobile charging stations, and UAV types in the set, respectively. μ' is the total number of weather scenarios. The total number of constraints can be calculated from (40). For example, with six time slots, six mobile charging stations, three UAV types, and ten weather scenarios, phase one will have 468 decision variables and 864 constraints.

$$decisionVariables1 = t'y'x' + \mu't'y' \quad (39)$$

$$constraints1 = t'y' + 2\mu't'y' + t'y'x' \quad (40)$$

For phase two in Section V-B, the total number of decision variables can be calculated from (41), where f' denotes the total number of BSs in the set. λ', $\omega^{3'}$, $\ldots$, $\omega^{\bar{z}'}$ are the total number of demand, shortfall in stage 2, $\ldots$, shortfall in stage $\bar{z}-1$ scenarios. The total number of constraints can be calculated from (42).

$$decisionVariables2 = t'f' + t'\lambda'y'f' + t'\omega^{3'}y'f' + \ldots$$
$$+ t'\omega^{\bar{z}'}y'f' \quad (41)$$

$$constraints2 = 2(t'f'\lambda'\omega^{3'} + \ldots + t'f'\lambda'\omega^{3'} \ldots \omega^{\bar{z}'}) \quad (42)$$
$$t'f'\lambda') + 2(t'y'f'\lambda'\omega^{3'} + \ldots + t'y'f'\lambda'\omega^{3'} \ldots \omega^{\bar{z}'})$$
$$+ t'y'\lambda'\omega^{3'} + \ldots + t'y'\lambda'\omega^{3'} \ldots \omega^{\bar{z}'}$$
$$+ t'y'f'\lambda' + 2(t'y'f'\lambda'\omega^{3'} \ldots \omega^{\bar{z}'})$$

C. Stochastic Coded Offloading Scheme Flowchart

In Fig. 5, the flowchart of SCOS algorithm is shown. The algorithm for solving SCOS is presented in four steps (i.e., Step-1 to Step-4) as follows.

In Step-1, we obtain the weather probability for the first time slot, and it can be obtained from the historical records [31] or weather forecast. The weather uncertainty is then modeled using SIP from (17).

In Step-2, we save the solution from Step-1.

In Step-3, we obtain the demand probability from the road traffic data set released by Land Transport Authority Singapore [57].

In Step-4, we obtain the shortfall probability from the historical records. Using the allocated UAV type from Step-2, we modeled the demand probability from Step-3 and the shortfall probability using SIP from (35). After solving this problem, if the next period has task allocation, the solution is saved, and the algorithm will proceed to Step-3. Otherwise, the algorithm proceeds to the next decision box. If the timeslot is the last time slot for the next decision box, the algorithm will end and output the SIP decision variables solution. Otherwise, the algorithm will proceed to Step-1.
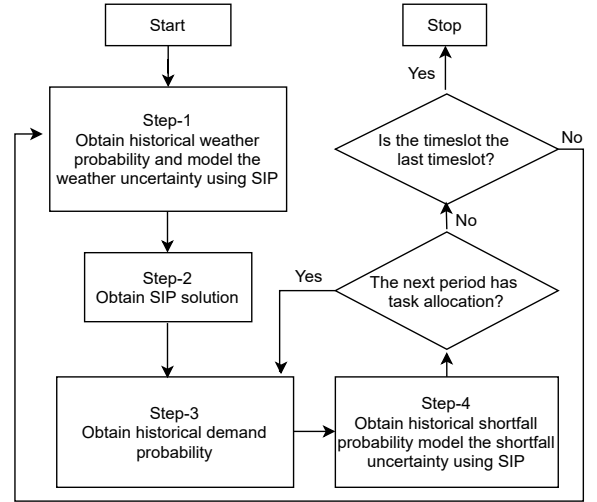
Fig. 5. Flowchart of SCOS algorithm.

VI. SIMULATION RESULT AND ANALYSIS

In this simulation, we consider the system model with one time slot $|\mathcal{T}| = 1$, six mobile charging stations, two BSs, sixty edge servers and thirty UAVs equally split into three UAV types. $\mathcal{X} = \{1 : small, 2 : medium, 3 : large\}$. The battery capacities B_1, B_2 and B_3 are $2375mAh$, $3500mAh$ and $5200mAh$, respectively [58]. A conceptual illustration of the system model is shown in Fig. 6. Fig. 6 is an x-y plane that shows the locations of the UAVs and the BSs. All the UAVs associated with their respective mobile charging stations are hovering at 100m, and the height of all edge servers in the BSs is 20m. They are randomly allocated in the area of $1000 \times 1000 \ m^2$. Each small grid is $25 \times 25 \ m^2$. We consider

Fig. 6. x-y coordinates of all the UAVs and BSs.

Fig. 7. Singapore traffic camera location.

Fig. 8. The probability distribution of the demand size.

TABLE III
EXPERIMENT PARAMETERS

Parameter	Values
UAV weight in kg, W_x	[8-12]
Air density in kg/m^3, ρ	1.225
Rotor radius in meter, R	0.5
Rotor disc area in m^2, $\mathbb{A}$	0.79
Blade angular velocity in radians/second, $\triangle_x$	[380-420]
Tip speed of the rotor blade, U_{tip}	200
Number of blades, b	4
Blade or aerofoil chord length, c	0.0196
Rotor solidity, defined as the ratio of the total blade area bcR to the disc area $\mathbb{A}$, $\mathbf{s}$	0.05
Fuselage drag ratio, d_0	0.3
Mean rotor induced velocity in hover, v_0	7.2
Profile drag coefficient, δ	0.012
Incremental correction factor to induced power, r	0.1
UAV hover height in meter, $H_{y,x}$	[80-100]
Height of edge servers in BS, H_f	20
UAV bandwidth in MHz, B_x	2
UAV transmit power in mW, P_x^C	32
UAV receiving power in mW, P_x^{re}	32
White Gaussian noise in dBm, N_0	-100
UAV computation power in GHz, τ_x	[0.6-1]
Number of CPU cycles needed to process a bit, $\mathcal{C}_x$	20
Channel gains, β_0	-60dB
C_f in \$	0.05
α_1	0.001
α_2	0.0015
α_3	0.5
α_4	0.5
α_5	0.0001

the case with $m = 2$ [13]. Therefore, we can substitute $s = \frac{2}{t}$ into (1), and we obtain the following:

$$k = 4t - t^2, \tag{43}$$

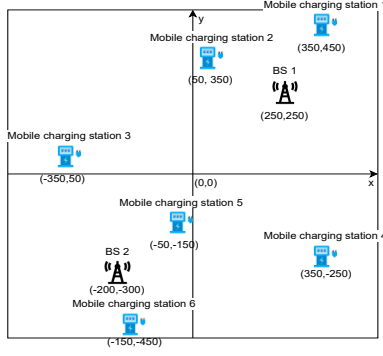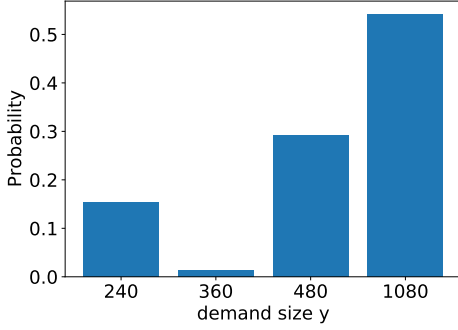by differentiating (43) with respect to the variable t and equating the result to zero. Then, we are able to obtain $t = 2$, $s = 1$, and $k = 4$. The simulation parameters are summarized in Table III and their values are from [48, 42]. For the presented experiments, we implement the SIP model using GAMS script [59, 60].

We obtain the finite set $\mathcal{W}$ from the road traffic data set released by Land Transport Authority Singapore [57]. As depicted in Fig. 7, there are 72 cameras in Singapore to monitor the traffic flow. We generate 1 set of the data in real-time to obtain $\mathcal{Y}$. The data are in the dimension of $row \times column$. Following [13] we are only considering square matrices, therefore, $W = row = column$. The probability distribution of the demand size is shown in Fig. 8 and $\mathcal{W} = \{240, 260, 480, 1080\}$. We use scenarios to represent the weather, demand, and shortfall uncertainties. In each scenario, each uncertainty will have a corresponding value. For example, when $G_y^{\bar{t}}(\mu_i^{\bar{t}}) = 1$. It means that the weather uncertainty in scenario i in mobile charging station y at time slot $\bar{t}$ is represented by the value of 1. For the road traffic data set, we obtain the dimension of the data, and we post-process the data to obtain $\mathcal{W}$ which is the demand size. We use the demand size to represent the actual demand uncertainty. Similar to weather

uncertainty, $\mathcal{W}$ is represented by its corresponding values 240, 260, 480, and 1080. As shown in Fig. 9, weather, demand, and shortfall uncertainties can be injected into the GAMS script using their corresponding values.

A. Allocation of UAV

We first evaluate the UAV type allocation for each mobile charging station. We consider a two-stage SIP, where the first stage is (17) and the second stage is (18). In this UAV type allocation, we consider two scenarios of weather condition, i.e., strong wind and weak wind, which are $|\gamma^1| = 2$ [30]. The two scenarios are that all locations have strong wind μ_1^1
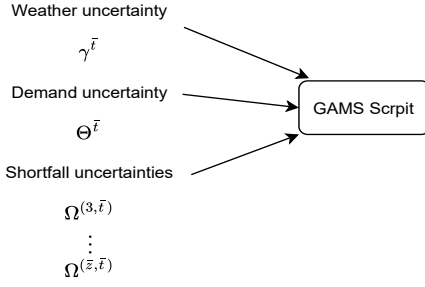
Fig. 9. The value of uncertainties are inserted into GAMS script.
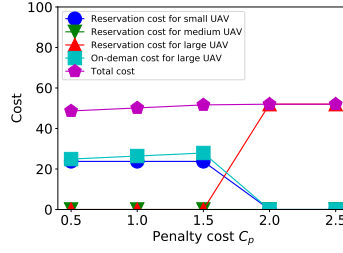
Fig. 10. The cost of UAV type allocation when varying the penalty cost.
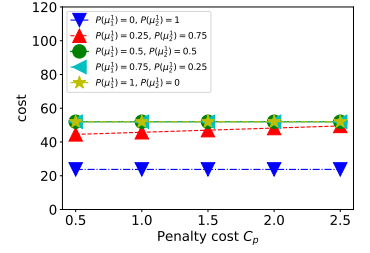
Fig. 11. The cost of the network by varying the penalty cost and the weather probability.

and no location has strong wind μ_2^1. We consider a stochastic system with $\mathcal{P}(\mu_1^1) = 0.3$ and $\mathcal{P}(\mu_2^1) = 0.7$. We perform the sensitivity analysis by varying the incurred cost and the weather condition probability.

1) Penalty cost C_p: We evaluate the UAV type allocation by varying the penalty cost C_p, and the result is presented in Fig. 10. When the penalty cost is low, the application owner still prefers to reserve the small UAV even though there is a crashing probability. However, the decision changed when the $C_p \geq 1.5$. The cost of the correction action, which is the on-demand cost of using a type 3 UAV, becomes much higher. $\bar{A}_3$ has to make a reservation on the type 3 UAV in the initial phase.

*2) Probability of the weather condition:* Next, we consider the same setting as that in Section VI-A1. We investigate the impact on the network by varying both the weather probability $P(\mu_1^1)$ and C_p. The result is shown in Fig. 11. When the weather is good $P(\mu_1^1) = 0$, all the charging stations will choose to use a type 1 UAV in stage 1 since there is no strong wind. When the probability of bad weather increases, e.g., $P(\mu_1^2) \geq 0.5$ all the charging stations tend to choose type 3 UAVs even when the penalty cost is low due to a high probability of UAV crashes.

B. Allocation of Task

We now perform the sensitivity analysis on task allocation using the UAV type allocation derived when $C_p = 2$. We perform the sensitivity analysis by varying the number of stages z, hovering cost $C_{y,x^{\bar{t}}}^{thresh}$, the probability of shortfalls, recovery threshold k as well as the types of UAV use, and we discuss the results in this subsection.

1) Cost structure: We first study the cost structure of the network. As an illustration, a primitive UAV network is considered with zero local computation $M_{y,x}^{L,2} = 0$, $M_{y,x}^{L,3} = 0$, and we consider a three-stage SIP, where the first stage is (35), the second stage is (36) and the third stage is (38). Furthermore, we first consider six mobile charging stations, one BS, one demand scenario $|\Theta^1| = 1$ and $|\Omega^{(3,1)}| = 1$. The demand is 1080. The shortfall scenario is that there are shortfalls from the copies that the UAV has offloaded. Since $k = 4$, the number of copies that are offloaded $M_{y,x,f}^{(O,2)}$ begins with 4. In Fig. 12, the costs in the first, second, third stages and the total cost under the different number of the

offloaded copies are presented. Since offloading is the only decision, the edge servers subscription cost in stage one should always paid. In the second stage, the cost is expected to increase as the number of copies that are offloaded increases. However, the cost in the third stage after knowing the actual shortfall decreases as the number of offloaded copies increases since the UAV needs to perform fewer re-computation to match the shortfall. Fig. 12 shows the optimal solution in this simple network. It can be identified that even in this simple network, the optimal solution is not trivial to obtain due to the uncertainty of shortfall. For example, the optimal cost is not the point where the cost in the second stage is the lowest of the cost in the third stage is the lowest. Therefore, SIP formulation is required to guarantee the minimum cost to the UAVs.

2) Number of stages z: We consider the case with four demand scenario $|\Theta^1| = 4$, one shortfall scenario $|\Omega^{(\bar{z},1)}| = 1$ for each SIP stage, and there is a shortfall in each stage. The four demand scenarios are made up of 4 demand size each, $\lambda_1^1 = 480$, $\lambda_2^1 = 240$, $\lambda_3^1 = 1080$ and $\lambda_4^1 = 360$. We vary the value of z from 3 to 6. The cost of the network is shown in Fig. 13 and Table IV indicates the value of the variables. The number of variables in the six-stage SIP is large. Therefore, for illustration, we only indicate the value for all the variables in mobile charging station 1 when $z = 3$. Then, We use the result from $z = 3$ to explain the findings. When the data size is 240, the offloading cost is more expensive than the local computation cost. Therefore, the UAV will perform local computation in stage 2 when the data size is 240. $M_{1,3^1}^{(L,2)}(\lambda_2^1) = 4$, $M_{1,3^1,f}^{(O,2)}(\lambda_2^1) = 0$, $M_{1,3^1,f}^{(L,3)}(\lambda_2^1) = 0$ and $M_{1,3^1,f}^{(O,3)}(\lambda_2^1) = 0$. However, for the rest of the data size, the offloading action is chosen. The UAV will offload more in stage 2 to overcome the shortfall, and this offloading cost is cheaper than the local computation cost. The total number of copies offloaded to the edge servers in BSs 1 and 2 is 24 each. The UAVs will always offload to the edge servers that is closer to it as the cost is lower. To support the offloading process, 24 edge servers have to be placed in both BSs 1 and 2. Since the shortfalls exist in every stage, as the number of stages increases, the UAV will perform more local computation in stage 2. When $z \geq 5$, the UAV will only consider local computation regardless of the demand, and the overall cost will remain the same. Since there is no offloading process,
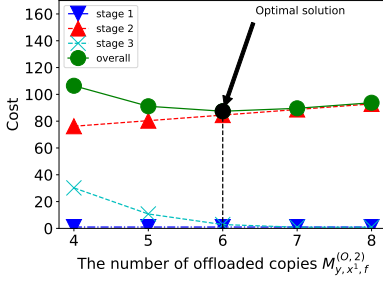
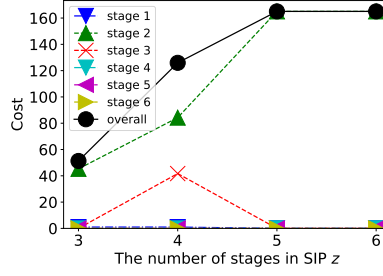Fig. 12. The optimal solution in a simple three-stage SIP UAV network.

Fig. 13. The cost of the network by varying the number of stages in SIP. Considered only UAV 1, BS 1 and BS 2.
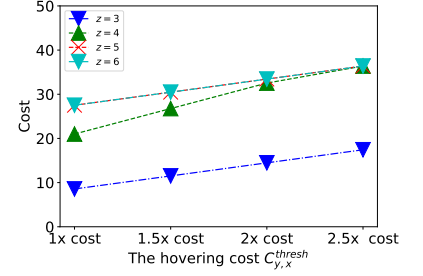

Fig. 14. The cost of the network by varying the hovering cost $C_{y,x^{\bar{t}}}^{thresh}$.
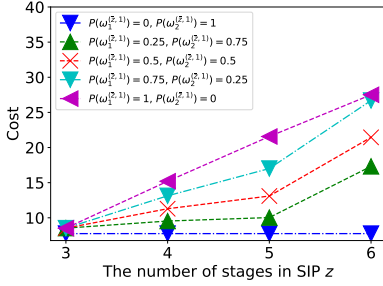
Fig. 15. The cost of the network by varying the probability of shortfall in each stage.
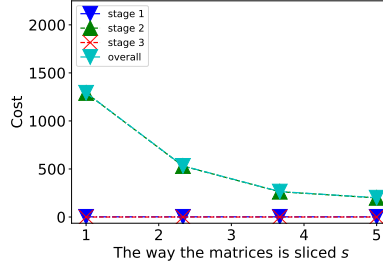
Fig. 16. The cost of the network by varying the variable s to change the recovery threshold.
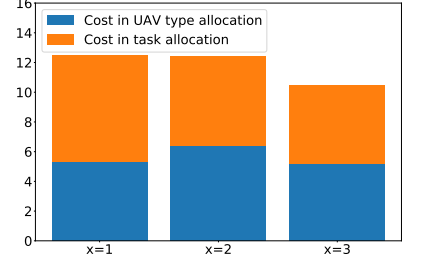
Fig. 17. The overall network cost when the UAV type x is varied.

edge servers' locations are no longer necessary as they are no longer needed.

3) Hovering cost $C_{y,x}^{thresh}$: We consider the case with mobile charging station 1 and BS 1. We vary both the hovering cost and the number of stages z to observe the impact on the network. We keep the rest of the parameters the same as Section VI-B2. Fig. 14 shows the cost of the network. It can be identified that when the hovering cost increases, it will also increase the re-offloading cost, therefore increasing the network cost. Since there is a shortfall scenario in every stage, the increase of hovering costs in every stage will affect the decision of the UAV. For example, when stage 4 has 2.5 times hovering cost, the UAV will perform full local computation regardless of the data size. Full local computation is also performed when $z \geq 5$.

4) Probability of the shortfalls: We consider the setting similar to Section VI-B3. There are two shortfall scenarios in each stage $|\Omega^{(\bar{z},1)}| = 2$. The two shortfall scenarios are i) all the UAVs have shortfalls $\omega_1^{(\bar{z},1)}$ in stage $\bar{z}$ and ii) all the UAVs do not have shortfall $\omega_2^{(\bar{z},1)}$. We observe the impact on the network by varying both the shortfall probabilities $P(\omega_1^{(\bar{z},1)})$, $P(\omega_2^{(\bar{z},1)})$ and the number of stages z. The result is shown in Fig. 15. Since there is no shortfall $P(\omega_1^{(\bar{z},1)}) = 0$, when z increases, the decision made by the UAV remains the same. The UAV can choose the cheapest decision, which is the offloading action. When $P(\omega_1^{(\bar{z},1)})$ increases, the number of copies that are computed locally and offloaded in the earlier stage also increases to reduce the penalty cost by the shortfalls. However, the number of scenarios increases exponentially with z [61], and this is the reason that it leads to a sharp increase

in cost from stages 5 to 6. For example, in a 6-stage SIP with two scenarios per stage, this results in a total of $5^6 = 15,625$ 64 scenarios [61]. When $z = 6$ and $P(\omega_1^{(\bar{z},1)}) = 1$, with the high re-offloading cost, full local computation is perform in the earlier stage.

5) Recovery threshold k: Similar to Section VI-B3, we consider three-stage SIP with four demand scenarios $|\Theta^1| = 4$, one shortfall scenario $|\omega^{(3,1)}| = 1$ and there is shortfalls in each offloading process. We monitor the impact of the network by varying the recovery threshold. We set $m = 4$ and we vary recovery threshold k by changing s from 1 to 5. Since k is controlled by (1), as s increases, k varies from 16 to 6. When $s = 1$, the recovery threshold is the highest $k = 16$. The result is shown in Fig. 16. The stage 1 cost remains constant as the BS is always subscribed. The cost of the network is also the highest as the UAV has to compute the most number of copies. When the data size is 240, 360, and 480, in stage 2, the UAV prefers to compute most of the copies locally as the local computation cost is much lower than the offloading cost plus the re-offloading cost in stage 3. The reason is that when k is high, the number of shortfalls is also high, which increases the re-offloading cost. This is why the stage 3 cost is zero. As s increases, the value of k decreases, the cost in local computation increases. Therefore, instead of local computation, some of the UAVs perform offloading process in stage 2. Throughout the simulations, the cost in stage 3 is zero. With the additional penalty and hovering cost, no UAVs prefer to compute the shortfalls in stage 3.

6) UAV types: Similar to Section VI-A1 and Section VI-B5, we consider 1 UAV and $C_p = 2$ and investigate the differences

TABLE IV
DECISION VARIABLE VALUE

Variable	$z=3$	$z=4$	$z=5$	$z=6$
$M_1^{(s)}$	1	1	0	0
$M_2^{(s)}$	0	0	0	0
$M_{1,3^1}^{(L,2)}(\lambda_1^1)$	0	4	4	4
$M_{1,3^1}^{(L,2)}(\lambda_2^1)$	4	4	4	4
$M_{1,3^1}^{(L,2)}(\lambda_3^1)$	0	0	4	4
$M_{1,3^1}^{(L,2)}(\lambda_4^1)$	0	4	4	4
$M_{1,3^1,1}^{(O,2)}(\lambda_1^1)$	8	0	0	0
$M_{1,3^1,1}^{(O,2)}(\lambda_2^1)$	0	0	0	0
$M_{1,3^1,1}^{(O,2)}(\lambda_3^1)$	8	4	0	0
$M_{1,3^1,1}^{(O,2)}(\lambda_4^1)$	8	0	0	0
$M_{1,3^1}^{(L,3)}(\lambda_1^1, \omega^{(3,1)})$	0	0	0	0
$M_{1,3^1}^{(L,3)}(\lambda_2^1, \omega^{(3,1)})$	0	0	0	0
$M_{1,3^1}^{(L,3)}(\lambda_3^1, \omega^{(3,1)})$	0	0	0	0
$M_{1,3^1}^{(L,3)}(\lambda_4^1, \omega^{(3,1)})$	0	0	0	0
$M_{1,3^1,1}^{(O,3)}(\lambda_1^1, \omega^{(3,1)})$	0	0	0	0
$M_{1,3^1,1}^{(O,3)}(\lambda_2^1, \omega^{(3,1)})$	0	0	0	0
$M_{1,3^1,1}^{(O,3)}(\lambda_3^1, \omega^{(3,1)})$	0	8	0	0
$M_{1,3^1,1}^{(O,3)}(\lambda_4^1, \omega^{(3,1)})$	0	0	0	0
Total number of edge server in BS 1	24	24	0	0
Total number of edge server in BS 2	24	24	0	0

Fig. 18. SIP comparing with EVF and random scheme.

comparison result. As shown in the result, EVF and random scheme cannot adapt to the change in cost. On the other hand, SIP can always achieve the best solution among the three to reduce the shortfall cost.

VII. CONCLUSION

In this paper, we have proposed the Stochastic Coded Offloading Scheme (SCOS) that employs a CDC technique to combines with computation offloading to minimize the energy consumption of the UAV network. We have first formulated the SCOS as the two-stage stochastic programming to optimize the UAV type allocation to account for the weather uncertainty. Then, we have formulated the z-stage stochastic programming to account for data size and task completion uncertainty. We have also conducted numerical experiments to verify that our proposed SCOS can optimize the total cost and the UAVs' energy consumption. Compared to the benchmark, SCOS based on SIP can achieve the best solution as it can adapt to changes in data size and task failure probability. We also know that a three-stage SIP is enough to formulate the problem through the simulations as it has the lowest cost.

in cost when we change the UAV type. The overall network cost is shown in Fig. 17. If the service provider initially uses type 1 and 2 UAVs, and when the wind is strong, the UAV cannot have sufficient energy to withstand the strong wind (weather uncertainty). As a result, the service provider has to deploy another UAV (type 3) to perform the task. However, if the service provider initially deploys a type 3 UAV, it will have sufficient energy to withstand the strong wind. There is no additional correction cost. Therefore, the UAV type allocation cost is cheaper if the service provider chooses to book type 3 UAV in advance. Furthermore, using type 3 UAV in task allocation makes the computation capability stronger than the other UAV types. As a result, it will have a shorter latency and will lead to a lower computation cost. Hence, the computation cost is also the lowest when a type 3 UAV is used. Therefore, the overall network cost is the lowest when a type 3 UAV is used.

C. Comparing between EVF, SIP and random scheme

We set $z = 3$ and compare the SIP with Expected-Value Formulation (EVF) [62] as well as the random scheme. Expected-value formulation uses the average values of shortfall as well as the demand and solves a DIP. For EVF, the total number of copies $(M_{y,x^{\bar{t}}}^{(L)} + M_{y,x^{\bar{t}},f}^{(O)})$ is fixed using the average value of shortfall and demand, an approximation scheme. In a random scheme, the values of the decision variables are randomly generated. We vary the price of the offloading action to compare the difference between EVF, SIP, and random schemes. This cost is combined with the UAV type allocation cost to observe the total cost of the network. Fig. 18 depicts the
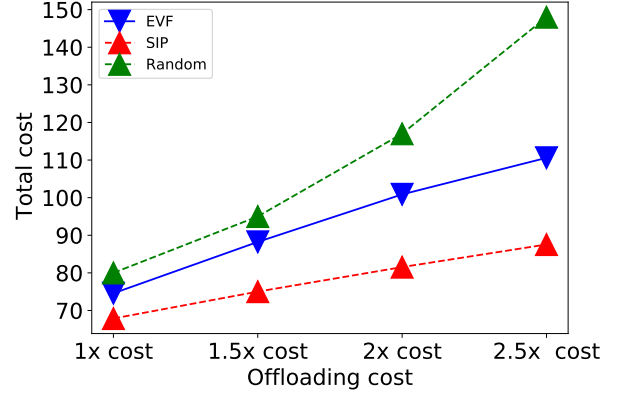
APPENDIX A
CONSTRAINTS OF z-STAGE SIP SYSTEM MODEL

$$\sum_{y\in\mathcal{Y}} M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}) \leq \sigma M_f^{(s)}, \quad \forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \tag{44}$$

$$\vdots$$

$$\sum_{y\in\mathcal{Y}} M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) \leq \sigma M_f^{(s)},$$
$$\forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots,$$
$$\forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}, \tag{45}$$

$$M_{y,x^{\bar{t}},f}^{(O,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) \leq \sigma M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}),$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})}, \tag{46}$$

$$\vdots$$

$$M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) \leq \sigma M_{y,x^{\bar{t}},f}^{(TH,\bar{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}),$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots,$$
$$\forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}, \tag{47}$$

$$\sum_{f\in\mathcal{F}} M_{y,x^{\bar{t}},f}^{(O,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) \geq \mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})$$
$$\left(\bar{A}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) - M_{y,x^{\bar{t}}}^{(L,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})\right),$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})}, \tag{48}$$

$$\vdots$$

$$\sum_{f\in\mathcal{F}} M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) \geq \left(\bar{A}_y^{(\bar{z},\bar{t})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) -\right.$$
$$\left. M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})})\right) \mathbb{F}_y^3(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})\ldots\mathbb{F}_y^{\bar{z}}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}),$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots,$$
$$\forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}, \tag{49}$$

$$M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}}) + M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}) \geq k,$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \tag{50}$$

$$M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}}) + \sum_{f\in\mathcal{F}} M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}) + \mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})$$

$$M_{y,x^{\bar{t}}}^{(L,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) + \sum_{f\in\mathcal{F}} \mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) M_{y,x^{\bar{t}},f}^{(O,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})$$

$$-\mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) \left(\bar{A}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) - M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})\right)$$
$$\geq k,$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})}, \tag{51}$$

$$\vdots$$

$$M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}}) + \sum_{f\in\mathcal{F}} M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}) + \ldots + \sum_{f\in\mathcal{F}} \mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})$$
$$\ldots\mathbb{F}_y^{(\bar{z},\bar{t})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})})+$$
$$\mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})\ldots\mathbb{F}_y^{\bar{z}}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})})$$
$$-\mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) \left(\bar{A}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}) - M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})\right)$$
$$-\ldots - \mathbb{F}_y^{(3,\bar{t})}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})})\ldots\mathbb{F}_y^{(\bar{z},\bar{t})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})})$$
$$\left(\bar{A}_y^{(\bar{z},\bar{t})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) - M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})})\right) \geq k,$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots,$$
$$\forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}, \tag{52}$$

$$\sum_{y\in\mathcal{Y}} M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}) \leq q_f, \forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \tag{53}$$

$$\vdots$$

$$\sum_{y\in\mathcal{Y}} M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) \leq q_f,$$
$$\forall \bar{t}\in\mathcal{T}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}}, \forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots,$$
$$\forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}, \tag{54}$$

$$M_f^{(s)}, M_{y,x^{\bar{t}},f}^{(TH,3)}(\lambda_i^{\bar{t}},\omega_i^{(3,\bar{t})}),\ldots, M_{y,x^{\bar{t}},f}^{(TH,\bar{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})})$$
$$\in\{0,1\},$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}},$$
$$\forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots, \forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}, \tag{55}$$

$$M_{y,x^{\bar{t}},f}^{(O,2)}(\lambda_i^{\bar{t}}), M_{y,x^{\bar{t}}}^{(L,2)}(\lambda_i^{\bar{t}}),\ldots, M_{y,x^{\bar{t}},f}^{(O,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}),$$
$$M_{y,x^{\bar{t}}}^{(L,\hat{z})}(\lambda_i^{\bar{t}},\ldots,\omega_i^{(\bar{z},\bar{t})}) \in\{0,1,\ldots\},$$
$$\forall \bar{t}\in\mathcal{T}, \forall y\in\mathcal{Y}, \forall f\in\mathcal{F}, \forall \lambda_i^{\bar{t}}\in\Theta^{\bar{t}},$$
$$\forall \omega_i^{(3,\bar{t})}\in\Omega^{(3,\bar{t})},\ldots, \forall \omega_i^{(\bar{z},\bar{t})}\in\Omega^{(\bar{z},\bar{t})}. \tag{56}$$

REFERENCES

[1] J. Du, E. Gelenbe, C. Jiang, H. Zhang, and Y. Ren, "Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2457–2467, 2017.

[2] K. Peng, J. Nie, N. Kumar, C. Cai, J. Kang, Z. Xiong, and Y. Zhang, "Joint optimization of service chain caching and task offloading in mobile edge computing," *Applied Soft Computing*, vol. 103, p. 107142, 2021.

[3] T. von Sawitzky, P. Wintersberger, A. Riener, and J. L. Gabbard, "Increasing trust in fully automated driving: route indication on an augmented reality head-up display," in *Proceedings of the 8th ACM International Symposium on Pervasive Displays*, Palermo Italy, June 2019.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, February 2014.

[5] S. Djahel, M. Salehie, I. Tal, and P. Jamshidi, "Adaptive traffic management for secure and efficient emergency services in smart cities," in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 2013, pp. 340–343.

[6] E. I. Vlahogianni, K. Kepaptsoglou, V. Tsetsos, and M. G. Karlaftis, "A real-time parking prediction system for smart cities," *Journal of Intelligent Transportation Systems*, vol. 20, no. 2, pp. 192–204, 2016.

[7] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, March 2017.

[8] H. Wang, J. Wang, J. Chen, Y. Gong, and G. Ding, "Network-connected uav communications: Potentials and challenges," *China Communications*, vol. 15, no. 12, pp. 111–121, December 2018.

[9] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 26–32, August 2018.

[10] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, "A green strategic activity scheduling for uav networks: A submodular game perspective," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 58–64, May 2016.

[11] K. Ro, J.-S. Oh, and L. Dong, "Lessons learned: Application of small uav for urban highway traffic monitoring," in *45th AIAA aerospace sciences meeting and exhibit*, Reno, June 2007.

[12] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, "Auction design and analysis for sdn-based traffic offloading in hybrid satellite-terrestrial networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2202–2217, 2018.

[13] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 278–301, July 2019.

[14] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with uav swarms," *IEEE Internet of Things Journal*, 2020.

[15] C. Tang, C. Zhu, X. Wei, H. Peng, and Y. Wang, "Integration of uav and fog-enabled vehicle: application in post-disaster relief," in *IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, Tianjin, China, January 2019, pp. 548–555.

[16] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, February 2017.

[17] Q. Yu, M. Maddah-Ali, and S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," *IEEE Transactions on Network Science and Engineering*, pp. 4403–4413, May 2017.

[18] B. Wang, J. Xie, K. Lu, Y. Wan, and S. Fu, "On batch-processing based coded computing for heterogeneous distributed computing systems," *arXiv preprint arXiv:1912.12559*, December 2019.

[19] J. R. Birge and F. Louveaux, "Introduction to stochastic programming," 2011.

[20] Z. Li, M. Xu, J. Nie, J. Kang, W. Chen, and S. Xie, "Noma-enabled cooperative computation offloading for blockchain-empowered internet of things: A learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2364–2378, 2020.

[21] Y. Zhou, C. Pan, P. L. Yeoh, K. Wang, M. Elkashlan, B. Vucetic, and Y. Li, "Secure communications for uav-enabled mobile edge computing systems," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 376–388, 2019.

[22] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[23] A. Asheralieva and D. Niyato, "Hierarchical game-theoretic and reinforcement learning framework for computational offloading in uav-enabled mobile edge computing networks with multiple service providers," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8753–8769, 2019.

[24] J. Zhang, L. Zhou, F. Zhou, B.-C. Seet, H. Zhang, Z. Cai, and J. Wei, "Computation-efficient offloading and trajectory scheduling for multi-uav assisted mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2114–2125, 2020.

[25] X. Zhang, Y. Zhong, P. Liu, F. Zhou, and Y. Wang, "Resource allocation for a uav-enabled mobile-edge computing system: Computation efficiency maximization,"

IEEE Access, vol. 7, pp. 113 345–113 354, 2019.

[26] C. M. Wang, "Stochastic integer programming: Decomposition methods and industrial applications," June 2014.

[27] W. H. Fleming, S. Sethi, and H. M. Soner, "An optimal stochastic production planning problem with randomly fluctuating demand," *SIAM Journal on Control and optimization*, vol. 25, no. 6, pp. 1494–1502, February 1987.

[28] C. L. Lara, J. D. Siirola, and I. E. Grossmann, "Electric power infrastructure planning under uncertainty: stochastic dual dynamic integer programming (sddip) and parallelization scheme," *Optimization and Engineering*, vol. 21, no. 4, pp. 1243–1281, October 2020.

[29] C. Li and I. E. Grossmann, "A review of stochastic programming methods for optimization of process systems under uncertainty," *Frontiers in Chemical Engineering*, vol. 2, p. 34, January 2021.

[30] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE transactions on services Computing*, vol. 5, no. 2, pp. 164–177, February 2011.

[31] S. Sawadsitang, R. Kaewpuang, S. Jiang, D. Niyato, and P. Wang, "Optimal stochastic delivery planning in full-truckload and less-than-truckload delivery," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, Sydney, NSW, Australia, June 2017.

[32] Y. Liu, R. Sioshansi, and A. J. Conejo, "Multistage stochastic investment planning with multiscale representation of uncertainties and decisions," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 781–791, April 2017.

[33] S. Kartik and C. S. R. Murthy, "Task allocation algorithms for maximizing reliability of distributed computing systems," *IEEE Transactions on computers*, vol. 46, no. 6, pp. 719–724, June 1997.

[34] K. Lu, J. Xie, Y. Wan, and S. Fu, "Toward uav-based airborne computing," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 172–179, August 2019.

[35] B. Wang, J. Xie, K. Lu, Y. Wan, and S. Fu, "Coding for heterogeneous uav-based networked airborne computing," in *IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, March 2019.

[36] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, August 2017.

[37] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*, Sydney, Australia, August 2017, pp. 3368–3376.

[38] K. Lee, C. Suh, and K. Ramchandran, "High-dimensional coded matrix multiplication," in *IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, August 2017, pp. 2418–2422.

[39] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *the 22nd International Conference on Artificial Intelligence and Statistics*, Naha, Okinawa, Japan, April 2019, pp. 1215–1225.

[40] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded mapreduce," in *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, April 2015, pp. 964–971.

[41] S. Dutta, V. Cadambe, and P. Grover, ""short-dot": Computing large linear transforms distributedly using coded short dot products," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6171–6193, July 2019.

[42] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, March 2019.

[43] 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, October 2018.

[44] S. Dutta, V. Cadambe, and P. Grover, "Coded convolution for parallel and distributed computing within a deadline," in *IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, August 2017, pp. 2403–2407.

[45] Q.-V. Pham, M. Zeng, R. Ruby, T. Huynh-The, and W.-J. Hwang, "Uav communications for sustainable federated learning," *arXiv preprint arXiv:2103.11073*, March 2021.

[46] E. Notes, "Qam formats: 8-qam, 16-qam, 32-qam, 64-qam, 128-qam, 256-qam." [Online]. Available: https://www.electronics-notes.com/articles/radio/modulation/quadrature-amplitude-modulation-types-8qam-16qam-32qam-64qam-128qam-256qam.php

[47] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus admm approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5087–5099, March 2019.

[48] J. Chen, S. Chen, S. Luo, Q. Wang, B. Cao, and X. Li, "An intelligent task offloading algorithm (itoa) for uav edge computing network," *Digital Communications and Networks*, vol. 6, no. 4, pp. 433–443, November 2020.

[49] M. Hua, Y. Wang, Q. Wu, H. Dai, Y. Huang, and L. Yang, "Energy-efficient cooperative secure transmission in multi-uav-enabled wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7761–7775, June 2019.

[50] J. S. Ng, W. Y. B. Lim, H.-N. Dai, Z. Xiong, J. Huang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Joint auction-coalition formation framework for communication-efficient federated learning in uav-enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, December 2020.

[51] N. Bezzo, K. Mohta, C. Nowzari, I. Lee, V. Kumar, and G. Pappas, "Online planning for energy-efficient and disturbance-aware uav operations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), December 2016, pp.

5027–5033.

[52] T. Chu, M. J. Starek, J. Berryhill, C. Quiroga, and M. Pashaei, "Simulation and characterization of wind impacts on suas flight performance for crash scene reconstruction," *Drones*, vol. 5, no. 3, p. 67, 2021.

[53] S. Sawadsitang, D. Niyato, P. Tan, and P. Wang, "Joint ground and aerial package delivery services: A stochastic optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2241–2254, October 2019.

[54] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, "Data offloading in uav-assisted multi-access edge computing systems: A resource-based pricing and user risk-awareness approach," *Sensors*, vol. 20, no. 8, p. 2434, April 2020.

[55] M. Dyer and L. Stougie, "Computational complexity of stochastic programming problems," *Mathematical Programming*, vol. 106, no. 3, pp. 423–432, December 2006.

[56] S. Rajendran and A. R. Ravindran, "Platelet ordering policies at hospitals using stochastic integer programming model and heuristic approaches to reduce wastage," *Computers & Industrial Engineering*, vol. 110, pp. 151–164, 2017.

[57] L. T. Authority, "Singapore traffic data," Accessed Aug. 1, 2021. [Online]. Available: https://data.gov.sg/dataset/traffic-images

[58] DJI, "Dji," Accessed Aug. 1, 2021. [Online]. Available: https://www.dji.com/sg

[59] D. Chattopadhyay, "Application of general algebraic modeling system to power system optimization," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 15–22, February 1999.

[60] G. D. Corporation, "General Algebraic Modeling System (GAMS) Release 24.2.1," Washington, DC, USA, 2013. [Online]. Available: http://www.gams.com/

[61] *GAMS - A User's Guide, GAMS Release 24.2.1*, GAMS Development Corporation, Washington, DC, USA, 2013. [Online]. Available: http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.p

[62] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *IEEE Asia-Pacific Services Computing Conference (APSCC)*, Singapore, January 2009, pp. 103–110.

Wei Yang Bryan Lim is currently an Alibaba Talent Programme PhD candidate with the Alibaba-NTU Joint Research Institute (JRI), Nanyang Technological University (NTU), Singapore. Prior to joining the JRI in 2019, he graduated with two First-Class Honors in Economics and Business Administration (Finance) from the National University of Singapore (NUS). During his PhD candidature, he has won 5 Best Paper Awards including in the IEEE Wireless Communications and Networking Conference (WCNC) and the IEEE ComSoc SPCC Best Paper Award. He regularly serves as a reviewer in leading journals and flagship conferences and is currently a review board member of the IEEE Transactions on Parallel and Distributed Systems and the assistant to the Editor-in-Chief of the IEEE Communications Surveys & Tutorials.
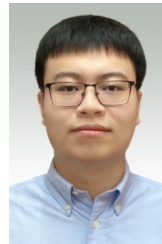
Zehui Xiong is currently an Assistant Professor in the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design. Prior to that, he was a researcher with Alibaba-NTU Joint Research Institute, Singapore. He received the PhD degree in Nanyang Technological University, Singapore. He was the visiting scholar at Princeton University and University of Waterloo. His research interests include wireless communications, network games and economics, blockchain, and edge intelligence. He has published more than 140 research papers in leading journals and flagship conferences and many of them are ESI Highly Cited Papers. He has won over 10 Best Paper Awards in international conferences and is listed in the World's Top 2% Scientists identified by Stanford University. He is now serving as the editor or guest editor for many leading journals including IEEE JSAC, TVT, IoTJ, TCCN, TNSE, ISJ. He is the recipient of IEEE TCSC Early Career Researcher Award for Excellence in Scalable Computing, IEEE CSIM Technical Committee Best Journal Paper Award, IEEE SPCC Technical Committee Best Paper Award, IEEE VTS Singapore Best Paper Award, Chinese Government Award for Outstanding Students Abroad, and NTU SCSE Best PhD Thesis Runner-Up Award. He is the Founding Vice Chair of Special Interest Group on Wireless Blockchain Networks in IEEE Cognitive Networks Technical Committee.

Dusit Niyato (M'09-SM'15-F'17) is currently a professor in the School of Computer Science and Engineering, at Nanyang Technological University, Singapore. He received B.Eng. from King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand in 1999 and Ph.D. in Electrical and Computer Engineering from the University of Manitoba, Canada in 2008. His research interests are in the areas of Internet of Things (IoT), machine learning, and incentive mechanism design.

Wei Chong Ng received B.Eng. degree in electrical and electronic engineering (Highest Distinction) from Nanyang Technological University, Singapore in 2020. He is currently pursuing the Ph.D. degree with Alibaba Group and Alibaba-NTU Joint Research Institute, Nanyang Technological University, Singapore. His research interests include the Metaverse, stochastic integer programming, and edge computing.

Chunyan Miao received the BS degree from Shandong University, Jinan, China, in 1988, and the MS and PhD degrees from Nanyang Technological University, Singapore, in 1998 and 2003, respectively. She is currently a professor in the School of Computer Science and Engineering, Nanyang Technological University (NTU), and the director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY). Her research focus on infusing intelligent agents into interactive new media (virtual, mixed, mobile, and pervasive media) to create novel experiences and dimensions in game design, interactive narrative, and other real world agent systems.

Zhu Han (S'01–M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019, and ACM distinguished Member since 2019. Dr. Han is a 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks."

Dong In Kim (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1990. He was a Tenured Professor with the School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada. Since 2007, he has been an SKKU-Fellowship Professor with the College of Information and Communication Engineering, Sungkyunkwan University (SKKU), Suwon, South Korea. He is a Fellow of the Korean Academy of Science and Technology and a Member of the National Academy of Engineering of Korea. He has been a first recipient of the NRF of Korea Engineering Research Center in Wireless Communications for RF Energy Harvesting since 2014. He has been listed as a 2020 Highly Cited Researcher by Clarivate Analytics. From 2001 to 2020, he served as an editor and an editor at large of Wireless Communications I for the IEEE Transactions on Communications. From 2002 to 2011, he also served as an editor and a Founding Area Editor of Cross-Layer Design and Optimization for the IEEE Transactions on Wireless Communications. From 2008 to 2011, he served as the Co-Editor-in-Chief for the IEEE/KICS Journal of Communications and Networks. He served as the Founding Editor-in-Chief for the IEEE Wireless Communications Letters, from 2012 to 2015. He was selected the 2019 recipient of the IEEE Communications Society Joseph LoCicero Award for Exemplary Service to Publications. He is the General Chair for IEEE ICC 2022 in Seoul.