

Article

Not peer-reviewed version

---

# Risk-aware Distributionally Robust Optimization for Mobile Edge Computation Task Offloading in Space–Air–Ground Integrated Networks

---

[Zhiyuan Li](#) <sup>\*</sup> and [Pinrun Chen](#)

Posted Date: 24 May 2023

doi: 10.20944/preprints202305.1689.v1

Keywords: Space–Air–Ground Integrated Network; Mobile Edge Task Offloading; Distributionally Robust Optimization; Conditional Value at Risk



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Article

# Risk-Aware Distributionally Robust Optimization for Mobile Edge Computation Task Offloading in Space–Air–Ground Integrated Networks

Zhiyuan Li <sup>1,2,3,\*</sup> and Pinrun Chen <sup>1</sup>

<sup>1</sup> School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China

<sup>2</sup> Jiangsu Provincial Key Laboratory of Industrial Network Security Technology, Zhenjiang, Jiangsu 212013

<sup>3</sup> Jiangsu Ubiquitous Data Intelligent Perception and Analysis Application Engineering Research Center, Zhenjiang, Jiangsu 212013

\* Correspondence: lizhiyuan@ujs.edu.cn; Tel.: +86-187-9600-6616

† Current Address: Jiangsu University, Zhenjiang, Jiangsu 212013, China.

**Abstract:** As an emerging network paradigm, Space-Air-Ground integrated networks (SAGIN) has attracted the attentions from academia and industry. That is because SAGIN can implement seamless global coverage and connections among electronic devices in space, air, and ground spaces. Additionally, the shortages of computing and storage resources in mobile devices greatly affect the quality of experiences for intelligent applications. Hence, we devise to integrate SAGIN as an abundant resource pool into mobile edge computing environments (MEC). To facilitate efficient processing, we need to solve the optimal task offloading decisions. Different from the existing MEC task offloading solutions, we have to face some new challenges, such as the fluctuation of processing capability for an edge computing node, the uncertainty of the transmission latency caused by the heterogeneous network protocols, the uncertain amount of uploaded tasks during a period, and so on. In this paper, we firstly describe a task offloading decision problem in new challenge environments. But, we cannot use the standard robust optimization and stochastic optimization methods to obtain the optimal result under the uncertain network environments. In this paper, we propose the condition value at risk-aware distributionally robust optimization algorithm, named as CVAR-DRO, to solve the task offloading decision problem. The proposed CVAR-DRO method combines the distributionally robust optimization and the condition value at risk model for solving the optimal result. And then, We have evaluated our approach in simulation SAGIN environments with the confidence interval, the number of mobile task-offloading and the various parameters. We compare our proposed CVAR-DRO algorithm with the state-of-the-art algorithms, such as the standard robust optimization algorithm, the stochastic optimization algorithm, the DRO algorithm, and the brute algorithm. The experimental results show that CVAR-DRO can get a sub-optimal mobile task-offloading decision. Overall, CVAR-DRO is more robust than others to the new challenges mentioned above in SAGIN.

**Keywords:** Space-Air-Ground Integrated Network; Mobile Edge Task Offloading; Distributionally Robust Optimization; Conditional Value at Risk

## 1. Introduction

With the booming development of the Internet of Things (IoT) ecosystem, users expect to obtain higher quality experiences (QoE) for various IoT applications. After the official deployment and operation of the fifth generation (5G) mobile system, the sixth generation (6G) mobile system is gradually coming into the limelight. The high-speed communication technologies can provide higher quality of service (QoS) for the IoT applications, such as intelligent transportation, intelligent agriculture, maritime surveillance, smart cities, natural disaster relief, and so on. However, the existing terrestrial communication networks cannot effectively provide the QoS and QoE guarantee for the

intelligent and low-latency IoT applications. In recent years, the Space-Air-Ground integrated network (SAGIN) architecture has attracted the attention from academia and industry. SAGIN is an integration of space layer, aerial layer, and ground layer. As a multidimensional network, SAGIN adopts different communication protocols in each segment or the integration of different segments to achieve high throughput and high-reliability data delivery [1].

However, we have to face some new challenges in SAGIN, such as the mobile edge computationally intensive tasks offloading and resources allocation in uncertain and heterogeneous network environments. The uncertain network parameters, such as the uncertain latency, the uncertain amount of arrival tasks, and the uncertain computation resources, may seriously impact the efficiency of edge task execution. Nowadays, the stochastic optimization (SO) [2] and the robust optimization (RO) [3–5] methods are proposed to solve the uncertainty problems. The SO method can use the probability distribution of measuring parameters to predict the potential uncertainty and obtain the mathematical expectation of objective function. The RO method can directly solve the target value under the worst-case conditions, and it doesn't need to obtain the probability distributions of measuring parameters. Distributionally robust optimization (DRO) can be viewed as a unifying framework for the SO and RO methods. The DRO method can replace the probability distribution of uncertain measuring parameters with a fuzzy set [6]. And then it can choose the worst case in the fuzzy set to get better robustness. Additionally, literature [7] proposed the conditional value at risk (CVaR) aware DRO method, which can reflect the potential risk and improve its stability.

There are two main technical challenges in solving the risk-aware two-stage DRO problem. Firstly, it is difficult to solve the DRO problem due to the fuzzy sets. Secondly, the first-stage offloading decision is the zero-one integer programming. And, the second-stage is a resources allocation procedure for latency insensitivity and computation-intensive tasks. But, in this paper, we have to solve the risk-aware two-stage DRO problem, which brings new challenge. If we can solve the risk-aware two-stage DRO problem, we can get the better revenue under the uncertainty and the quality of service(QoS) in SAGIN network architecture.

Our contributions are summarized as follows:

- Investigate the task offloading decision model in the SAGIN environment. The task offloading model consists of two stages: the first stage is task offloading decision, and the second stage is edge-cloud collaboration and cloud resource allocation.
- Building upon the work in (1), a fuzzy set of computational resources for edge computing nodes is constructed, and consideration for CVaR is introduced. Then, based on the theory of Lagrange duality, the model is transformed into a semidefinite programming form, and the Conditional Value at Risk (CVaR)-aware Distributionally Robust Optimization for Task Offloading (RaDROO) algorithm is proposed to solve the task offloading problem with distributional robustness under risk awareness.
- We conducted simulation experiments from two aspects. On the one hand, we adjust some parameters of the proposed model to get the optimal parameter values. On the other hand, we fix the parameters and compare with the the state of the art algorithms in different computation and network environments. The experimental results demonstrate that our proposed model and algorithm have better results than the state of the art methods in terms of usability, robustness and risk.

The remainder of this paper is organized as follows. Section II gives a brief introduction to SAGIN before reviewing the computation task offloading. Section III describes the network architecture and proposes a computation task offloading model under uncertain computation environments. Section IV gives the risk-aware DRO algorithm to obtain the optimal task offloading decision and resource allocation strategy. Section V shows the performance evaluation and analysis. Section VI concludes our work and presents the future work.

## 2. Related Work

In this section, we firstly introduce the SAGIN architecture. And then, we review the traditional computation task offloading, the computation task offloading under the uncertain network and computation environments, and the risk-aware computation offloading, respectively.

### 2.1. SAGIN Architecture

The SAGIN refers to the integration and synergy of systems from multiple domains-space, air, ground-to form a wide area coverage and high speed network for more efficient communication and data exchange. It is a proposed solution to address the growing need for enhanced communication and information sharing capabilities. SAGIN requires the use of various technical means, such as satellite communications, unmanned aerial vehicles and ground sensors, to enable collaborative operations and data sharing between different platforms. By sharing data and information in real time, situational awareness can be improved, decision making can be optimised, and mission effectiveness can be enhanced.

Over time, the SAGIN architecture has evolved rapidly and various projects like Global Information Grid (GIG) have been proposed and widely deployed [8]. Liu et al. [1] describe the communication network design and resource allocation algorithm of SAGIN in a high-dimensional network environment. Likewise, SAGIN can be invoked in MEC environments and has excellent performance in unique environments, *i.e.* desert, disaster scenarios. Yuet al. [9] considered the fine-grained offloading problem and caching problem and proposed the SAGIN framework which supports edge computing.

The impact of SAGIN on computing tasks offloading is multifaceted and can be summarized as follows:

- Cross-platform efficiency: Through the integration and collaborative operations achieved by SAGIN, different tasks can be offloaded and migrated between different platforms, thereby improving the efficiency of task execution.
- Increased Flexibility through Edge-Cloud Collaboration: In the SAGIN network, tasks can be dynamically allocated and scheduled, allowing them to be offloaded to the most suitable platforms based on their computational requirements. This improves resource utilization.

Overall, the impact of SAGIN on the field of task offloading is positive, as it can improve task execution efficiency, safety, flexibility, and coverage. An ECN built on the SAGIN framework will be able to more fully exploit its capabilities. Consolidate all available resources to provide computing resources for the computing tasks in this network environment.

### 2.2. Traditional Computation Offloading

First of all, a class of existing tasks offloading optimization algorithms in practice applications is on task dependency. Yuan et al. [10] paid an attention to a dependent task assignment problem over multiple mobile terminal devices (MTDs). In [11], an efficient partitioned search method was implemented for obtaining optimal solutions for task offloading policies and resource allocation under task-dependent models. Besides, without considering task dependency, whether to offload to edge computing nodes(ECN) for tasks and how much computing resources are allocated by ECNs according to the performance requirements for computing offloads is a hot topic of current researchers. The mainstream research directions under this scope are the following three. The first one is to explore task offloading decision schemes that minimize task execution delay[12,13]. Xiao et al. [14] designed a heat prediction method to analyze the dynamics of urban heat zones, aided by the design of a non-cooperative game-theoretic strategy selection based on regret matching to achieve minimum time delay. Dai, Y. et al. [15] proposed a JSCO algorithm to search for the solution to the optimization problem in a distributed manner with less overhead, using the integrated task processing delay as the performance metric. In [16], Farhangi, E.et al. proposed a novel offload approach OAMC, which

takes into account the dynamic changes of mobile applications and reduces the number of migrations and overall data movement while minimizing the turnaround time of mobile applications. Secondly, plenty of studies[17–20] focus on obtaining task offloading decisions in edge computing networks that minimize energy consumption, which is affected by the amount of offloading computations, MTD-MECs distance, channel conditions, application type, compression efficiency, and *etc.*. Hmimz et al. [19] jointly considered both the priority of certain MTDs and minimizing overall power consumption. Third, [21,22] assumes that the system can gain a certain amount of revenue by completing the task, and the goal is to find the offloading decision that maximizes the revenue. Samanta et al. [23] synthetically explores delay-sensitive and delay-tolerant edge services and guaranteed Quality of Experience(QoE) to be maximized over a long period using the Lyapunov method. Chen et al. [24] applied gaming offloading algorithms to maximize mobile network operator revenue.

In addition to the common is solved by traditional optimization methods, Some researchers[25] have also chosen to apply intelligent optimization algorithms to pick the optimal unloading decision. In [20], it developed a new adaptive inertia weight-based particle swarm optimization (NAIWPSO) algorithm to minimize the energy consumption of MTDs while considering the channel constraint condition during task offloading. Li et al. [26] proposed an algorithm, named EIPSO, based on an developed particle swarm optimization. Tout et al. [27] designed a multi-objective intelligent optimization algorithm based on genetic algorithm.

Machine learning and deep learning algorithms are now extraordinary prevalent in the research field in droves. Maleki, E. F. et al. [16] applied a Machine Learning(ML) algorithm called matrix complementation to design two fresh offloading methods, S-OAMC and G-OAMC, that enhance scalability and find low turnaround time offloading decisions. In [28], various machine learning techniques for the communication, network and security components of the future 6G environment in vehicular networks were profiled, and methods and directions for implementing artificial intelligence in future 6G vehicular networks are envisioned. However, machine learning algorithms typically require a significant amount of training data and cannot be directly computed. They rely on training on large datasets to learn patterns and make predictions. Additionally, machine learning algorithms are sensitive to changes in the environment and may require retraining or adaptation to maintain their performance.

**Table 1.** Content in the relevant literature.

Previous Work	Game	ML	RO	SO	DRO	mean risk-aware	SAGIN
[29]	✓						
[30], [3]			✓				
[16]		✓					
[7]					✓		
[2]				✓			
[31], [32]						✓	
[33]					✓		✓
Our Algorithm					✓	✓	✓

### 2.3. Uncertain-aware Computation Offloading

The above research scenarios whole assume that controllers has access to accurate computing resource and channel information either statically or dynamically. Nevertheless, in application environments, MECN are transient which can considerably impact resource utilization and user QoE. [30][3] investigates robust task offloading that is tolerant to failures in offloading scenarios and can effectively overcome this. In [4], robust tasks with tolerant server failures offloaded were contemplated to achieve higher application responsiveness. To counter the inaccuracy of channel measurements, [5] design a robust offloading strategy for channel information estimation errors. Facing the connection instability between MTDs and small clouds, [3] proposed a robust computation



offloading strategy with failure recovery. [34] introduced a mobility model and a trade-off fault tolerance mechanism for the offload system, taking into account the dependencies between services.

In contrast to the robust optimization described above, which sacrifices a lot of device performance to ensure its robustness during application, stochastic optimization is used. The investigations of Stochastic Optimization (SO) in MEC gives the probability distribution of channel resources, computational resources of ECNs varying randomly[2]. proposed a two-stage SO to tackle the challenge of uncertain dynamic environments.

#### 2.4. Risk-aware Computation Offloading

Unlike the uncertain or undefined parameter scenarios described above, risk considerations refer to unexpected conditions that occur during system operation. Bai et al. [29] solve risk-aware computation offloading (RCO) problem by considering the use of Bayesian Stackelberg Game to rationally disperse the nodes for task offloading when the server is attacked by outside miscreants. Apostolopoulos et al. [35] take account of the uncertainty in MEC server computational resources as well as storage space under heterogeneous networks. Schultz, R. et al. [32] explain the risk consideration in stochastic programming, i.e., mean-risk, and also give the expression formula for CVaR. With CVaR, Pan et al. [36] considered the availability of the MEC system in case of possible link failure. Mean-CVaR is commonly deployed in finance, where Zhang et al. [31] have studied portfolio selection under return fuzzy set conditions, considering a downside risk measure to reduce investment risk.

### 3. Network Architecture and System Model

In this section, we elaborate on the SAGIN architecture and channel models. Based on the compute, caching and communication (3C) requirements, We developed an optimization model for making computing offload decisions, aiming to minimize cost.

#### 3.1. SAGIN Architecture and Channel Models

SAGIN is a novel network structure integrating multiple dimensions. It adopts different protocols and occupies different frequency bands for establishing communications between the different dimensions[1]. We select the rational frequency band for communications by taking into account the distance between ECNs, the influence of current environmental factors on the link, the free-space path loss, tropospheric attenuation, and other factors.

In the study, tasks are uploaded to the nearest BSs from MTDs by default. As shown in Figure 1, the SAGIN architecture consists of five components: the MTD segment, ground segment, satellite segment, aerial segment, and cloud data centre (CDC) segment. Let  $\mathcal{B} = \{1, 2, \dots, B\}$  be used as the set of BSs. Furthermore, as a representative of the aerial segment, UAVs are despatched as the extension of the ground BS process. The set of the UAVs are denoted by  $\mathcal{U} = \{1, 2, \dots, U\}$ . The satellite segment consists of LEO satellites. Due to the utilization of higher communication frequency bands, satellites possess lower propagation delay and free space attenuation [37].

The SAGIN system offers three tasks offload strategies, depending on the task information uploaded by MTDs: firstly, selecting a BS that possess sufficient computing resources to calculate the task; secondly, applying for a UAV offload and compute the task; thirdly, selecting for a UAV as an intermediary and applying an LEO satellite to offload and compute the task. As a multi-dimensional 6G network, SAGIN integrates several network segments and uses communication protocols to achieve high reliability and high throughput data transmission [38].

The maximum achievable transmission rate  $K$ [bits/s] for the tasks in the channel can be expressed as follows.

$$\begin{cases} K = B \log_2 \left( 1 + \frac{P \times 10^{-[L_p/10]}}{\text{Noise}} \right), \\ L_p = 32.45 + 20 \lg d(\text{km}) + 20 \lg f(\text{MHz}). \end{cases} \quad (1)$$

Within the above equation,  $B$  (HZ),  $P$  and  $Noise$  represent channel bandwidth, average signal power and noise power;  $L_p$ ,  $d$ ,  $f$  represent propagation loss in free space, and the distance between ECNs and signal frequency, respectively.

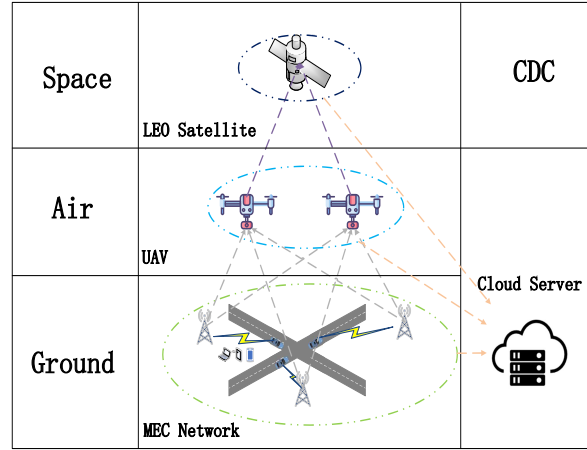


Figure 1. SAGIN model.

### 3.2. Computation Tasks Offloading Model

We suppose that there are  $n$  latency sensitivity and computation-intensive tasks offloaded to nearby BSs, where the set of tasks is denoted as  $\mathcal{N} = \{1, 2, \dots, N\}$ . And, each of tasks is characterized by any parameters,  $(n_i^{up}, n_i^{comp}, n_i^{down})$ ,  $i \in \mathcal{N}$ , in which  $n_i^{up}$  denotes the input and uploaded data size;  $n_i^{comp}$  denotes the number of CPU cycles required to process data;  $n_i^{down}$  denotes the processed and downloaded data size. In this research, the study's objective is 3C and the total cost of tasks, thus keeping the atomicity of the tasks, i.e. each task can only be served by one processing ECN.

In our proposed SAGIN system, there are four offload destinations: BSs, UAVs, LEO satellites, and CDC. We assume that  $x_{ib}^{BS}$ ,  $x_{iu}^{UAV}$ , and  $x_{is}^{LEO}$  are three binary parameters, where  $x_{ib}^{BS}$  denotes whether task  $i$  selects BS  $b$  as the offload target,  $x_{iu}^{UAV}$  denotes whether task  $i$  selects UAV  $u$  as the offload target and  $x_{is}^{LEO}$  denotes whether task  $i$  selects LEO satellite  $s$  as the offload target.

$$\sum_{b=1}^B x_{ib}^{BS} + \sum_{u=1}^U x_{iu}^{UAV} + \sum_{s=1}^S x_{is}^{LEO} = 1, \forall i \in \mathcal{N} \quad (2)$$

$$x_{ib}^{BS}, x_{iu}^{UAV}, x_{is}^{LEO} \in \{0, 1\}, b \in \mathcal{B}, u \in \mathcal{U}, s \in \mathcal{S} \quad (3)$$

The above equation indicates that each task would be offloaded to one ECN in the SAGIN architecture.  $x_{ib}^{BS} = 1$  if task  $n_i$  is offloaded to BS  $b$  and  $x_{ib}^{BS} = 0$  otherwise;  $x_{iu}^{UAV} = 1$  if task  $n_i$  is offloaded to UAV  $u$  and  $x_{iu}^{UAV} = 0$  otherwise;  $x_{is}^{LEO} = 1$  if task  $n_i$  is offloaded to LEO satellite  $s$  and  $x_{is}^{LEO} = 0$  otherwise. These three different computing units own different computing power and storage resources.  $f_{ib}^{BS}$ ,  $f_{iu}^{UAV}$ , and  $f_{is}^{SAT}$  denote the number of computing resources in per second requested by task  $i$  from BSs, UAVs, or LEO satellites, respectively.

We assume that tasks are uploaded to the nearest base station first and then propagate them from the base station to the target computing nodes. And since the propagation delay of tasks directly to edge computing nodes is much greater than uploading tasks to the nearest base station, we will ignore the upload delay of tasks.

The time taken by a task to be uploaded and downloaded between ECNs and the latency taken to calculate the task can be expressed as

$$T^{up} = \frac{n^{up}}{k^{up}} = \frac{n^{up}}{B \log_2(1 + \frac{P \times 10^{-[L_P/10]}}{N})} \quad (4)$$

$$T^{down} = \frac{n^{down}}{k^{down}} = \frac{n^{down}}{B \log_2(1 + \frac{P \times 10^{-[L_P/10]}}{N})} \quad (5)$$

$$T^{comp} = L^{time} - T^{up} - T^{down} \quad (6)$$

where  $T^{comp}$  is the time to execute the task,  $T^{up}$  and  $T^{down}$  represent the time of uploading and downloading the task message.

The overall latency of the tasks offloaded to the BSs spent by the task related to the transfer delay to the target BS, the computation delay of the BS, and the transfer delay of the result download.

$$T_{ib}^{BS} = T_{ib}^{up} + T_{ib}^{comp} + T_{ib}^{down}, i \in \mathcal{N} \quad (7)$$

It is one of the options to upload tasks to the UAVs when the computing resources on the ground are insufficient to cope with the large influx of tasks.

$$T_{iu}^{UAV} = T_{iu}^{up} + T_{iu}^{comp} + T_{iu}^{down}, i \in \mathcal{N} \quad (8)$$

The tasks process changes with the calculations of offloading tasks to LEO satellites. Tasks information must be uploaded to the nearest UAV, before spreading to satellites. After the task processing is completed, the data is returned in the original path. The total latency taken to offload the tasks to an LEO satellite can be expressed as

$$T_{is}^{LEO} = T_{iu}^{up} + T_{is}^{up} + T_s^{comp} + T_{iu}^{down} + T_{ib}^{down} \quad (9)$$

where  $T_{ibu}^{up}$ ,  $T_{ius}^{up}$ ,  $T_{isu}^{down}$ ,  $T_{iub}^{down}$  and  $T_{is}^{comp}$  represent uploading latency of task  $i$  from BSs to UAVs, and from UAVs to LEO satellites, downloading latency from LEO satellites to UAVs and from UAVs to BSs, and the computation delay of the task  $i$ , respectively.

For this paper, we have chosen to ensure that the task is completed within the time delay constraints. Therefore, in order to allocating of computing resources rationally, the node only provides the lowest possible computational resources,  $f_i = n_i^{comp} / L_i^T - T^{trans}$ , where  $T_i^{trans}$  represent the time taken for task  $i$  to be transmitted over the link.

There is an upper bound on the total amount of computing resources an ECN can have. As a result, the combined computational resources required for the tasks assigned to a node should not exceed the upper limit.

$$\sum_{i=1}^n f_i x_{ib}^{BS} \leq L_b^{Bcomp}, \forall b \in \mathcal{B}, \quad (10a)$$

$$\sum_{i=1}^n f_i x_{iu}^{UAV} \leq L_u^{Ucomp}, \forall u \in \mathcal{U}, \quad (10b)$$

$$\sum_{i=1}^n f_i x_{is}^{LEO} \leq L_s^{Lcomp}, \forall s \in \mathcal{S}, \quad (10c)$$



Consider the caching in 3C, where each ECN has its own storage limit.

$$\sum_{i=1}^N x_{ib}^{BS} n_i^{up} \leq L_b^{Bcache}, \forall b \in \mathcal{B} \quad (11a)$$

$$\sum_{i=1}^N x_{iu}^{UAV} n_i^{up} \leq L_u^{Ucache}, \forall u \in \mathcal{U} \quad (11b)$$

$$\sum_{i=1}^N x_{is}^{LEO} n_i^{up} \leq L_s^{Lcache}, \forall s \in \mathcal{S} \quad (11c)$$

$L_b^{Bcache}$ ,  $L_u^{Ucache}$ , and  $L_s^{Lcache}$  represent the cache upper bound of BS  $b$ , UAV  $u$ , and LEO satellite  $s$ , separately.

The cost for tasks offloading is closely related to the server type, the number of computing resources needed. In this paper, we focus on solving computational intensive and time delayed sensitive tasks. Therefore it is the key to minimize the offload cost by picking a reasonable offloading target. Hence

$$C_i = \sum_{b=1}^B \left( \delta_b^{BS} \times n_i^{comp} \times x_{ib}^{BS} \right) + \sum_{u=1}^U \left( \delta_u^{UAV} \times n_i^{comp} \times x_{iu}^{UAV} \right) + \sum_{s=1}^S \left( \delta_s^{LEO} \times n_i^{comp} \times x_{is}^{LEO} \right), i \in \mathcal{N} \quad (12)$$

$$C^{all} = \sum_{n=1}^N C_n \quad (13)$$

In the above equation,  $C_i$  denotes the final cost of task  $i$ ;  $C^{ALL}$  denotes the cost for accomplishing whole tasks;  $\delta_b^{BS}$ ,  $\delta_u^{UAV}$  and  $\delta_s^{LEO}$  represent the ratio between the fees and the data size that will be processed by BSs, UAVs, and LEO satellites, respectively.

$$\min_{x_{ib}^{BS}, x_{iu}^{UAV}, x_{is}^{LEO}} C^{all} \quad (14)$$

$$s.t. (2), (3), (10a), (10b), (10c), (11a), (11b), (11c)$$

### 3.3. Computing Resources Fussy Set Model

Realistically, the computational resources of each node can be greatly influenced by external factors, and the Qu et al. [30] then considered the resource uncertainty of MECN in a traditional environment. In this paper, for the computing resource uncertainty of ECNs, we propose a realistic scenario-based set of definite probability distribution to represent the possible range of uncertain resources.

Based on a fuzzy set of random parameters  $\xi$ , we will make the one-stage offloading decision and the two-stage cloud resource request.

Delage, E. et al. [39] considered a general bounded moments uncertainty set, based on realistic data, with exact moment uncertainty, i.e.,

$$\mathcal{D}(\xi, \sum, \mu_0, \gamma_1, \gamma_2) = \left\{ P \subseteq \mathcal{F} : \begin{aligned} &P\{\xi \in \mathcal{M}\} = 1 \\ &(\mathbb{E}[\xi] - \mu_0)^T \Sigma^{-1} (\mathbb{E}[\xi] - \mu_0) \leq \gamma_1 \\ &\mathbb{E}[(\xi - \mu_0)(\xi - \mu_0)^T] \leq \gamma_2 \Sigma \end{aligned} \right\} \quad (15)$$

where parameters  $\gamma_1$  and  $\gamma_2$ , based on historical data, are utilised to control the size of the ambiguity set and the conservatism of optimal solutions.  $\mathcal{M}$  is the closed convex set of the support of the currently known random  $\xi$ , and  $\Sigma$  and  $\mu_0$  are, respectively, its corresponding first-order and second-order moments. In contrast,  $\mathcal{F}$  is the set of all probability measures on the measurable space  $(R^m, \mathcal{B})$ , with  $\mathcal{B}$  the Borel  $\beta$ -algebra on  $R^m$ .

The first constraint indicates that the probability density sum of  $\xi$  is 1 over  $\mathcal{M}^\xi$ ; the second constraint assumes that the mean of  $\xi$  lies on an ellipsoid of size  $\gamma_1$  centred at  $\mu_0$ ; the third constraint forces that the covariance matrix lies in a positive semi-positive definite cone bounded by matrix inequalities.

#### 4. Risk-aware Distributionally Robust Optimization Tasks Offloading Algorithm Design

In this section, we consider fuzzy sets and CVaR to address the uncertainty of computational resources in edge computing nodes. We reconstruct the original problem with the uncertainty of the computational resources, due to the challenging nature of dealing directly with the DRO problem. We use Lagrangian duals to transform it into a semi-definite programming (SDP) problem to solve it.

In fact, we know that the offloading decision is an integer programming problem, and the additional computational resources requested from CDC are linear programming, which makes our problem (19) a MILP problem. Hence, we use branching implicit enumeration method to obtain the optimal offloading decision based on the uncertainty of computational resources and the minimum average cost.

##### 4.1. Network Architecture and System Model

In this section, We consider the transformation of the original problem into a two-stage DRO programming model with a mean-CVaR recourse function, namely a risk-aware DRO model.

Above all, we represent (14) in terms of matrix form variables and parameters. We consider the set  $\mathcal{D} = \mathcal{B} \cup \mathcal{U} \cup \mathcal{S}$  and  $M = B + U + S$ . Then, the formulated problem is written as follow:

$$\min_X \mathbb{N}^{compT} \times X \times \delta \quad (16)$$

$$s.t. \quad X \times O_m = O_n, \quad (16a)$$

$$X^T \times \mathbb{N}^{up} \leq L^{cache}, \quad (16b)$$

$$X^T \times F \leq L^{comp}, \quad (16c)$$

$$\begin{aligned} X &\in R^{N \times M}, \delta \in R^N, \mathbb{N}^{comp} \in R^M, \mathbb{N}^{up} \in R^N, \\ f &\in R^N, L^{cache} \in R^M, L^{comp} \in R^M \end{aligned} \quad (16d)$$

where the offloading strategy matrix variable  $X := \{x_{nm}\}_{n \in \mathcal{N}, m \in \mathcal{D}}$ , the upper bound of computing resource vector parameter  $L^{comp} := \{L_m^{comp}\}_{m \in \mathcal{D}}$ , the upper bound of cache ceiling vector parameter  $L^{cache} := \{L_m^{cache}\}_{m \in \mathcal{D}}$ , computing resource vector parameter  $F := \{f_i\}_{i \in \mathcal{N}}$ , the ratio between the cost and the data size vector parameter  $\delta := \{\delta_i\}_{i \in \mathcal{N}}$ , and the all ones vector parameter  $O$ .

Thus, given the uncertainty of the parameter  $L^{comp}$ , a two-stage offloading strategy model is created using the distribution set proposed above. We constructed a fuzzy set based on the historical data of the computing resources of the offload node. Expressed as:  $\xi = \tilde{L}^{comp}$ .

$$\min_X \mathbb{N}^{compT} \times X \times \delta + \mathbb{E}_{\xi} [\vartheta(X, \xi)], \quad (17)$$

$$s.t. ((16a) - (16d)),$$

$$\left\{ \begin{array}{l} P\{\xi \in \mathcal{M}^{\xi}\} = 1 \\ P \subseteq F^{\xi}: (\mathbb{E}[\xi] - \mu_0)^T \Sigma^{-1} (\mathbb{E}[\xi] - \mu_0) \leq \gamma_1 \\ \mathbb{E}[(\xi - \mu_0)(\xi - \mu_0)^T] \leq \gamma_2 \Sigma \end{array} \right\} \quad (17a)$$

$$with \vartheta(X, \xi) = \begin{cases} \min_Y \varphi^T \times Y, \\ (X \otimes F^{req})^T \times O_n \leq \xi + Y, \\ 0 \leq Y, \end{cases} \quad (17b)$$

$\vartheta(X, \xi)$  represent the additional cost of uploading tasks to the cloud when the real-time computing resources change in a phase of decision task offloading and are insufficient to meet demand. Vector  $\varphi$  indicates the ratio of additional costs to required computational power.

Instead, we further consider the mean-CVaR (Conditional Value at Risk) criterion in the model, which can be expressed in the following form:

$$VaR_{\alpha}(\zeta) = \inf \{v \in R : \mathbb{F}(v) \geq \alpha\} \quad (18a)$$

$$\begin{aligned} CVaR_{\alpha}(\zeta) &= \mathbb{E}\{\zeta | \zeta \geq VaR_{\alpha}(\zeta)\} \\ &= \min_{v \in R} \left\{ v + \frac{1}{1 - \alpha} \mathbb{E}[(\zeta - v)_+] \right\} \end{aligned} \quad (18b)$$

where  $\mathbb{F}$  is the cumulative distribution function of the function used to solve for the random variable  $\zeta$ , and  $\alpha$  is the given confidence level.  $(a)_+$  means  $\max(a, 0)$ .

This model gives the set of possible distributions  $\mathcal{M}^{\xi}$  for the unknown stochastic parameter  $\xi$  while ensuring its robustness using a mean-CVaR with two-stage minimization of fees. The following formulation will be made using min-max theory:

$$\min_X h(x) + \sup_{\xi} \{(1 - \lambda) \mathbb{E}[\vartheta(x, \xi)] + \lambda CVaR(\vartheta(x, \xi))\} \quad (19)$$

$$with \ h(x) = \mathbb{N}^{compT} \times X \times \delta,$$

$$s.t. \quad (16a), (16b), (16c), (16d), (17a), (17b),$$

which  $\lambda$  indicates the trade-off coefficient of conditional risk considered in the objective function and  $1 \geq \lambda \geq 0$ .

#### 4.2. Transform from DRO to SDP

Problem (19) has presented a DRO problem, below We further analyze the target formula and convert it to more explicit form.

$$\begin{aligned}
 & \sup_{\xi} \{ (1 - \lambda) \mathbb{E}[\vartheta(x, \xi)] + \lambda \text{CVaR}(\vartheta(x, \xi)) \} \\
 = & \sup_{\xi} \{ \min_{v \in R} \{ \lambda v + (1 - \lambda) \mathbb{E}[\vartheta(x, \xi)] + \frac{\lambda}{1 - \alpha} \mathbb{E}[(\vartheta(x, \xi) - v)_+] \} \} \\
 = & \min_{v \in R} \{ \lambda v + \sup_{\xi} \{ \mathbb{E}[(1 - \lambda) \vartheta(x, \xi) + \frac{\lambda}{1 - \alpha} (\vartheta(x, \xi) - v)_+] \} \}
 \end{aligned} \tag{20}$$

The ambiguous distribution set of the random parameter cannot be used directly for the solution, and we need to convert it into the common inequality form. Consider only the latter half of finding the optimal value in relation to the random parameter  $\xi$ . Instead of the original distribution set form, turn to the following semi-infinite conic linear problem form expressed as:

$$\sup_{\xi} \int_{\mathcal{M}} [(1 - \lambda) \vartheta(x, \xi) + \frac{\lambda}{1 - \alpha} (\vartheta(x, \xi) - v)_+] df(\xi) \tag{21}$$

$$s.t. \int_{\mathcal{M}} df(\xi) = 1, \tag{21a}$$

$$\int_{\mathcal{M}} (\xi - \mu_0)(\xi - \mu_0)^T df(\xi) \preceq \gamma_2 \Sigma \tag{21b}$$

$$\int_{\mathcal{M}} \begin{bmatrix} \Sigma & \xi - \mu_0 \\ (\xi - \mu_0)^T & \gamma_1 \end{bmatrix} df(\xi) \succeq 0, \tag{21c}$$

$$\xi \in \mathcal{M}^{\xi}, \tag{21d}$$

Secondly, we utilize the duality theory to convert it into a SDP problem. Consider the dual of problem (21):

(Dual) :

$$\inf_{r, H, \begin{bmatrix} Z & z \\ z^T & \hat{z} \end{bmatrix}} r + \langle H, \gamma_2 \Sigma - \mu_0 \mu_0^T \rangle + \langle Z, \Sigma \rangle + \hat{z} \gamma_1 - 2z^T \mu_0 \tag{22}$$

$$s.t. \quad u(x, \xi) - r - \xi H \xi^T + 2\xi^T H \mu_0 + 2z^T \xi \leq 0, \xi \in \mathcal{M}^{\xi}, \tag{22a}$$

$$r \in R, \tag{22b}$$

$$H \succeq 0, \tag{22c}$$

$$\begin{bmatrix} Z & z \\ z^T & \hat{z} \end{bmatrix} \succeq 0, \tag{22d}$$

$$\text{with } u(x, \xi) = (1 - \lambda) \vartheta(x, \xi) + \frac{\lambda}{1 - \alpha} (\vartheta(x, \xi) - v)_+$$

where  $r$ ,  $H$ , and  $\begin{bmatrix} Z & z \\ z^T & \hat{z} \end{bmatrix}$  are correspondingly the dual variables of constraints (21a), (21b), and (21c).  $\langle a, b \rangle$  represent  $\text{Trace}(AB)$ .

Due to the inclusion of  $(\cdot)_+$ , the inequality constraint (22a) has to be processed as follow:

$$\begin{aligned} & r + \xi H \xi^T - 2\xi^T H \mu_0 - 2z^T \xi \\ & \geq (1 - \lambda) \vartheta(x, \xi) + \frac{\lambda}{1 - \alpha} (\vartheta(x, \xi) - v)_+ \end{aligned}$$

When the random parameter  $\xi$  appears in the constraint, it cannot be directly substituted into constraint (22a). Therefore, it is necessary to consider the dual of problem  $\vartheta(x, \xi)$  such that the random parameters are transferred into the objective function. Another problem ensues, the coupling of variables the dual variable  $q$  and the variable  $Y$ . The constraint on the dual variable  $q$  confines it to a boxed region, as indicated by the constraint.

(Dual) :

$$\vartheta'(x, \xi) = \max_{q, p} q^T \left( (X \otimes F^{req})^T O_N - \xi \right), \quad (23)$$

$$\text{s.t. } \varphi - p \geq q \geq 0, \quad (23a)$$

$$p \geq 0, \quad (23a)$$

Responding to constraint (22a), taking its extreme value points, each point corresponds to two corresponding semi-definite matrix constraints. We transformed (22a) into a clearer and more concise form.

$$\begin{cases} r + \xi H \xi^T - 2\xi^T H \mu_0 - 2z^T \xi \\ \geq (1 - \lambda) \vartheta'(x, \xi) + \frac{\lambda}{1 - \alpha} (\vartheta'(x, \xi) - v) \\ = \left(1 + \frac{\lambda \alpha}{1 - \alpha}\right) \vartheta'(x, \xi) - \frac{\lambda v}{1 - \alpha} \\ r + \xi H \xi^T - 2\xi^T H \mu_0 - 2z^T \xi \geq (1 - \lambda) \vartheta'(x, \xi) \end{cases}$$

In the above constraints, we represent  $\left(1 + \frac{\lambda \alpha}{1 - \alpha}\right)$  as  $\phi$  and transform them into the following two Semi-definite matrix constraint.

$$\begin{cases} \begin{bmatrix} H & \frac{1}{2} \phi q^T - H \mu_0 - z \\ \left(\frac{1}{2} \phi q - H \mu_0 - z\right)^T & r - \phi q^T (X \otimes F^{req})^T O_N + \frac{\lambda v}{1 - \alpha} \end{bmatrix} \succcurlyeq 0 \\ \begin{bmatrix} H & \frac{1 - \lambda}{2} q - H \mu_0 - z \\ \left(\frac{1 - \lambda}{2} q - H \mu_0 - z\right)^T & r - (1 - \lambda) q^T (X \otimes F^{req})^T O_N \end{bmatrix} \succcurlyeq 0 \end{cases}$$



Then, dual problem (22) can be rewritten as following:

$$\min_{X,v,r,H,\begin{bmatrix} Z & z \\ z^T & \hat{z} \end{bmatrix}} h(x) + \lambda v + r + \left\langle H, \gamma_2 \sum -\mu_0 \mu_0^T \right\rangle \quad (24)$$

$$+ \langle Z, \sum \rangle + \hat{z} \gamma_1 - 2z^T \mu_0$$

$$s.t. \quad X \times O_M = O_N, \quad (24a)$$

$$X^T \times \mathbb{N}^{up} \leq L^{cache}, \quad (24b)$$

$$\begin{bmatrix} H & \frac{1}{2}\phi q^T - H\mu_0 - z \\ \left(\frac{1}{2}\phi q - H\mu_0 - z\right)^T & r - \phi q^T (X \otimes F^{req})^T O_N + \frac{\lambda v}{1-\alpha} \end{bmatrix} \succcurlyeq 0 \quad (24c)$$

$$\begin{bmatrix} H & \frac{1-\lambda}{2}q - H\mu_0 - z \\ \left(\frac{1-\lambda}{2}q - H\mu_0 - z\right)^T & r - (1-\lambda)q^T (X \otimes F^{req})^T O_N \end{bmatrix} \succcurlyeq 0 \quad (24d)$$

$$\begin{bmatrix} Z & z \\ z^T & \hat{z} \end{bmatrix} \succcurlyeq 0, H \succcurlyeq 0, r \in R, \quad (24e)$$

$$x_{ij} = \{0, 1\}, v \in R, \quad (24e)$$

#### 4.3. DRO-MILP Algorithm

In the above SDP problem, the X-matrix variables, 0-1 integer variables, and the Y-vector variables, continuous variables, are included. This makes the original problem a mixed-integer linear formulation, but he is not able to solve it directly. We relax the problem to a linear problem - simplifying the discrete variable X to a continuous variable from 0 to 1, i.e.,  $0 \leq x_{ij} \leq 1, \forall x_{ij} \in X$ .

In fact, the above SDP problem can be solved by using the SDPT3 solver through the CVX package in MATLAB. Then, the original problem MILP problem needs to be solved, and here we choose to use the branching implicit enumeration method to solve it, which a special branch and bound method for 0-1 integer problems, using the feature that the variables can only take two values of 0 or 1, to branch delimitation for the purpose of hidden enumeration. Where, for each task is divided into m sub-nodes, i.e., each task can only be offloaded and completely offloaded to one node.

Algorithm 1 present the progress of the calculation of the minimize cost of computation offloading. First of all, tasks should be sorted according to the number of CPU cycles they require, which determines the cost spent to a large degree. Then the solution of the relaxed LP problem is obtained, based on which the implicit enumeration method is performed to solve the solution of the branching subproblem. During the process of branch and bound, the two nodes with the highest values in the  $X_i$  vector, representing the most probable nodes for offloading the current task, are selected at each iteration. Constantly branching, updating the gradually increasing lower bound; constantly in delimitation, updating the gradually decreasing upper bound.

**Algorithm 1** Cost-based MILP algorithms for DRO task offloading problem.

---

**Input:** Convex Probability Distributional set  $\mathcal{M}^\sim$ , Stochastic Sample Space  $\mathcal{F}^\sim$ , Trade-off coefficient  $\lambda$ , Confidence Level  $\alpha$ , Ambiguous Set Parameters  $\gamma_1$  and  $\gamma_2$ ;  
**Output:** Offload Decision  $X$ , Optimal cost value to satisfy distribution robustness;  
 Solve the MILP formulation SDP problem with confidence level  $\alpha$ , to obtain optimal value of continuous variables  $X$  and optimal cost value;  
**for**  $i$  from 0 to  $N - 1$  **do**  
   Sort( $\mathcal{N}$ )  
    $[opt, X] = \max(X_i)$   
    $[x1_{ij}, x2_{ij}] = \max2(X)$   
   **if**  $SDP(X', x'_{i,j1} = 1) \leq SDP(X', x'_{i,j2} = 1)$  **then**  
      $x'_{i,j1} = 1$   
      $min\_cost[i] = SDP(X', X'_{i,j1} = 1)$   
   **else**  
      $x'_{i,j2} = 1$   
      $min\_cost[i] = SDP(X', X'_{i,j2} = 1)$   
   **end if**  
**end for**  
**if**  $min\_cost[N - 1] \neq NaN$  **then**  
   return  $X', min\_cost[N - 1]$   
**end if**

---

#### 4.4. Complexity Analysis

In Algorithm 1, a solver is used to solve the optimization problem. Here, the time complexity for solving the optimization problem is set to  $T$ , the number of unloading tasks is set to  $N$ , and the branch and bound method is used to traverse all unloading tasks. In the unloading target section, it is simplified to select only the optimal two node locations with a cycle number of. The time complexity of the proposed algorithm is  $O(TN)$ .

### 5. Performance Evaluation

In this section, we try the performance and effectiveness under different parameters; analyze the results of the above proposed algorithm under different risk confidence conditions; and compare it with the traditional algorithm to give its superiority.

#### 5.1. Simulation Setup

We accomplished the simulation on a laptop with an AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz, 16G RAM, and a Windows 10 OS. We envision a SAGIN-Cloud system where the coverage of the space layer contains the air layer and the coverage of the air layer contains the whole MECN. (The following parameters are all randomly within a certain range, given that the mean values are generated according to a normal distribution)

##### 1) Channel State Information (CSI)

For information transportation within ECNs, different frequency bands are borrowed between the different layers, resulting in different bandwidths and transmission speeds. Currently, the C-band is one of the most commonly used frequency bands for satellite operations, with the Ka-band being a latecomer. This paper assumes that data transmission between BSs and UAVs occupies the C-band, while the Ka-band is occupied between UAVs and LEO satellites. The bandwidth of C-band is 20MHz, while the available bandwidth of Ka-band can reach 3500MHz. As for signal frequency range, C-band and Ka-band are 3.4GHz-8GHz and 26.5-36GHz, respectively.

##### 2) ECNs Information

The horizontal and vertical coordinates of the BSs on the ground and the UAVs in the air are within  $(-1, 1)$  km, respectively, while vertically, they are all 0. In contrast, the UAVs are located at an altitude of  $(0.075, 0.15)$  km. LEO satellites are located in space and they are at a distance of  $(780, 800)$  km from the earth's surface. LEO satellites are located in space and they are  $(780, 800)$  km away from the surface. they are not too space constrained, so are deployed in a square area with four points as vertices,  $(-200, -200)$ ,  $(-200, 200)$ ,  $(200, -200)$ ,  $(200, 200)$ .

BSs, UAVs and LEO satellites possess CPU process speed of  $2 \times 10^9$  cycles/s,  $3 \times 10^8$  cycles/s, and  $5 \times 10^9$  cycles/s, respectively.

### 3) Comparison Algorithm

First, we propose to use the most basic comparison algorithm, namely the Brute-Force algorithm.

- In Brute-Force algorithm, It does not take into account any uncertainty and possible overflow of computation of task to the extent that it may eventually lead to an inability to obtain an optimal solution.

Secondly, we compare two traditional algorithms that take parameter uncertainty into account: robust optimization(RO) algorithm and stochastic optimization(SO) algorithm.

- In RO, only the uncertainty of computational resources of ECNs is considered and their possible worst-case scenarios are experimentally selected to ensure their robustness.
- In SO, the fuzzy set is constructed based on the historical data of the computational resources of the given ECNs, thus obtaining their means and variances.

Thirdly, there is another algorithm closest to our proposed one, which is the DRO algorithm for obtaining the optimal solution.

- In DRO [33], the mean value in the range of uncertainty set is obtained and its optimal robust result is obtained by minmax theory, which guarantees both robustness and practicality. However, there is also a drawback, namely that it is an uncertainty set constructed from historical data, which does not guarantee the stability of its uncertainty set and does not consider its risk.

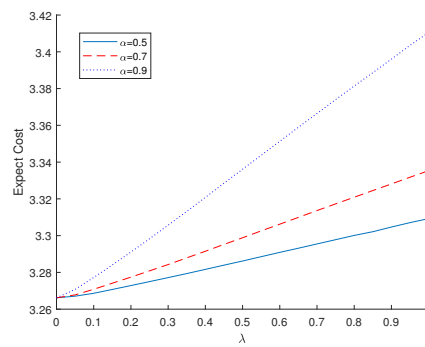
Lastly, we proposed CVaR-aware DRO algorithm on the basis of the traditional DRO algorithm.

- In CVaR-aware DRO, in addition to what the DRO considers, it also complements certain defects that it has. It considers CVaR, choosing only the  $\alpha - tail$  part as the benchmark, while using  $\lambda$  as a weight with the original part. That is to say, he selectively aggravates the proportion of the worse part of the results within the final result in order to guarantee its risk resistance.

## 5.2. Experiments

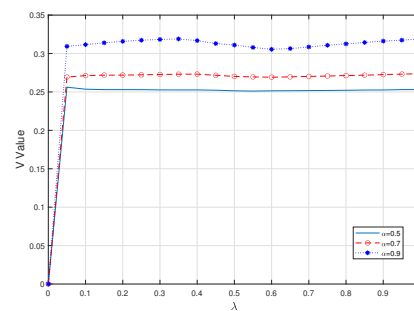
### 5.2.1. RaDROO algorithm

First observe the practical results of our proposed RaDROO algorithm under the practical problem of solving the optimization of offloading cost of edge network tasks. The optimal values of lambda in the range of  $(0, 1)$  are given as  $\alpha$  of 0.5, 0.7, and 0.9, respectively. As illustrated in Figure 2, when lambda is 0, it means that CVaR is not considered and weakened to a classical distribution robust problem, and the obtained results converge to a point. However as the  $\lambda$  increases, the consideration of risk is reinforced step by step, further constraining the range of values provided within the uncertainty set. This results in a progressive increase in the amount to be spent, but at the same time, a more superior and stable optimal choice is obtained. At the same time, the larger the lambda, i.e., the larger the weight considering the CVaR value, the worse the optimal value of the required spend.

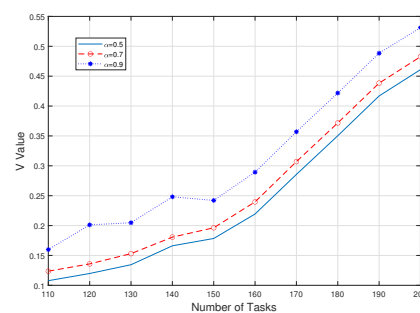


**Figure 2.** Expect cost under different  $\alpha$

Second, observing the value of  $V$  gives a more intuitive view of the degree of consideration of risk in CVaR. As shown in Figure 3 and Figure 4, it is obvious that the value of  $V$  is larger when a higher alpha value is taken, i.e., when the worse part of the fuzzy set is used as a criterion, and at the same time, this value is almost unaffected by the value of  $\lambda$  taken; however, with a fixed value of  $\alpha$ , as the simultaneous upload of amount of tasks increases, the value of  $V$  for each experiment also increases gradually and steadily, which means that the value of CVaR also increases, that is, the risk is further considered as a way to ensure the authenticity and practicality of the whole system.



**Figure 3.**  $V$  value under different  $\alpha$ .



**Figure 4.**  $V$  value between 110 and 200 tasks.

### 5.2.2. Comparing with other algorithms

For comparison with other algorithms, some of the parameters of RaDROO are  $\lambda = 0.5, \alpha = 0.5, \gamma_1 = 0, \gamma_2 = 1$ .

The comparison between algorithms takes the obtained target value as the evaluation criterion. Figure 5 shows the optimal cost results obtained by different algorithms at the same time as the amount of tasks received increases. Figure 6 and Figure 7 show the comparison of the results of the four algorithms at task volumes of 150 and 170. When the number of tasks is small, considering only known allocatable resources all algorithms give better results. As the total number of tasks that may be

uploaded simultaneously rises, the Brute-Force algorithm fails to obtain an optimal solution because it does not take into account the lack of computational resources. In contrast, RO, SO, DRO and RaDROO can obtain their optimal solutions and the optimal cost obtained increases with the increasing number of tasks.

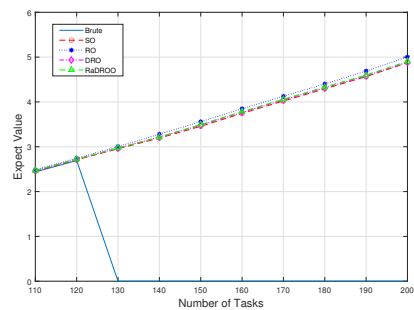


Figure 5. optimal cost of compared algorithms between 110 tasks to 200 tasks

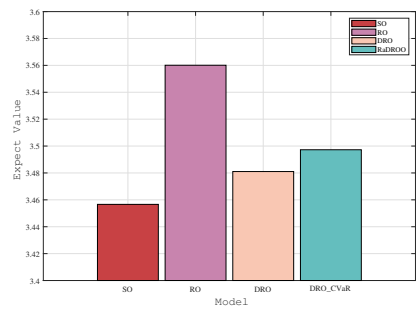


Figure 6. optimal cost under 150 tasks

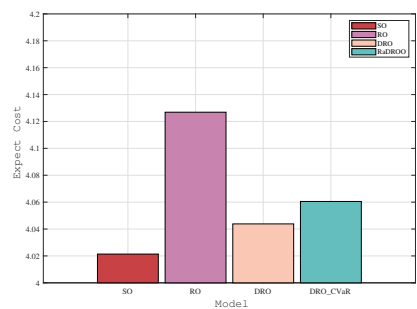


Figure 7. optimal cost under 170 tasks

In the two observations with the number of tasks 150 and 170, we can clearly get from Figure 6 and Figure 7 that the results obtained by the RaDROO are closer to those obtained by the traditional DRO algorithm. Therefore, it can be concluded that RaDROO ensures its robustness while considering CVaR to obtain higher risk resistance and better results under the uncertainty of computational resources.

The RaDROO algorithm obtains better results also pays a price in terms of computational time, as shown in Figure 8. the time taken to solve the RO, SO, Brute algorithm is close and lower. the DRO algorithm obtains the optimal two-stage mean based on fuzzy sets in the minmax theory of robust algorithms. The RaDROO algorithm considers CVaR on top of this, which further increases the complexity of the algorithm and takes more time to solve problem.

Overall, although RaDROO incurs a certain time cost, it strikes a balance between computational time and risk consideration by incorporating CVaR into the decision-making process. It achieves improved robustness compared to traditional methods while maintaining competitive performance.



By considering CVaR, RaDROO is able to effectively manage the risk associated with uncertain factors and make more informed task offloading decisions in the SAGIN environment.

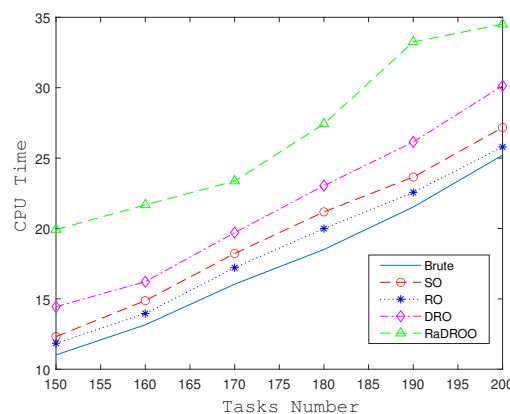


Figure 8. Execution time of each algorithm.

## 6. Conclusions

This paper focuses on the offloading decision problem of tasks in the SAGIN framework, considering the instability of computational resources of each node and the limited channel resources and storage resources of ECNs. Based on this problem, we propose the RaDROO algorithm, which considers while CVaR to improve its risk resistance on the basis of the traditional DRO algorithm. In addition, we conducted simulation experiments in a simulation environment, and the algorithm obtained better results than the traditional robust algorithm and close to the traditional DRO algorithm, and proved the effectiveness of the algorithm.

There are several directions in which we can extend this work in the future. First, the execution time of the RaDROO algorithm is a major problem that needs to be solved, which may be solved by optimizing the algorithm architecture and using a better model solving method; second, the task upload is transient, and we can consider the real-time task unloading decision problem within the SAGIN framework; third, the value of lambda is freely variable and it is a problem to choose a rational value; Finally, this experiment only considers the uncertainty of computational resources, in the actual environment, the storage space of ECNs and CSI also have uncertainty, and the impact of these uncertainties on the offloading decision is also worth considering.

**Acknowledgments:** This work was supported by Natural Science Foundation of Jiangsu Province under Grant BK20201415.

## References

1. J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.
2. W. Ma and L. Mashayekhy, "Quality-aware video offloading in mobile edge computing: A data-driven two-stage stochastic optimization," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*. IEEE, 2021, pp. 594–599.
3. M. Chen, S. Guo, K. Liu, X. Liao, and B. Xiao, "Robust computation offloading and resource scheduling in cloudlet-based mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2025–2040, 2020.
4. H. Wang, H. Xu, H. Huang, M. Chen, and S. Chen, "Robust task offloading in dynamic edge computing," *IEEE Transactions on Mobile Computing*, 2021.
5. Z. Wu, B. Li, Z. Fei, Z. Zheng, and Z. Han, "Energy-efficient robust computation offloading for fog-iot systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4417–4425, 2020.
6. H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, 2019.

7. A. Ling, J. Sun, N. Xiu, and X. Yang, "Robust two-stage stochastic linear optimization with risk aversion," *European Journal of Operational Research*, vol. 256, no. 1, pp. 215–229, 2017.
8. M. Albuquerque, A. Ayyagari, M. A. Dorsett, and M. S. Foster, "Global information grid (gig) edge network interface architecture," in *MILCOM 2007-IEEE Military Communications Conference*. IEEE, 2007, pp. 1–7.
9. S. Yu, X. Gong, Q. Shi, X. Wang, and X. Chen, "Ec-sagins: Edge-computing-enhanced space-air-ground-integrated networks for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5742–5754, 2021.
10. Y. Yuan, X. Xu, M. Sun, and P. Zhang, "Terminal cooperative interdependent computing task offloading for 6g," *IEEE Transactions on Network Science and Engineering*, 2022.
11. J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 235–250, 2019.
12. Y. Yang, X. Chang, Z. Jia, Z. Han, and Z. Han, "Towards 6g joint haps-mec-cloud 3c resource allocation for delay-aware computation offloading," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 2020, pp. 175–182.
13. Li, Zhiyuan and Peng, Ershuai, "Software-defined optimal computation task scheduling in vehicular edge networking," *Sensors*, vol. 21, no. 3, pp. 955, 2021.
14. Z. Xiao, X. Dai, H. Jiang, D. Wang, H. Chen, L. Yang, and F. Zeng, "Vehicular task offloading via heat-aware mec cooperation using game-theoretic method," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2019.
15. Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2018.
16. E. F. Maleki, L. Mashayekhy, and S. M. Nabavinejad, "Mobility-aware computation offloading in edge computing using machine learning," *IEEE Transactions on Mobile Computing*, 2021.
17. S. Chouhan, "Energy optimal partial computation offloading framework for mobile devices in multi-access edge computing," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2019, pp. 1–6.
18. F. Wang, J. Xu, and S. Cui, "Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2443–2459, 2020.
19. Y. Hmimz, T. Chanyour, M. El Ghmary, and M. O. C. Malik, "Energy efficient and devices priority aware computation offloading to a mobile edge computing server," in *2019 5th International Conference on Optimization and Applications (ICOA)*. IEEE, 2019, pp. 1–6.
20. Y. Wang, L. Wu, X. Yuan, X. Liu, and X. Li, "An energy-efficient and deadline-aware task offloading strategy based on channel constraint for mobile cloud workflows," *IEEE access*, vol. 7, pp. 69 858–69 872, 2019.
21. Z. Ma, S. Zhang, Z. Chen, T. Han, Z. Qian, M. Xiao, N. Chen, J. Wu, and S. Lu, "Towards revenue-driven multi-user online task offloading in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1185–1198, 2021.
22. D. Wang, X. Tian, H. Cui, and Z. Liu, "Reinforcement learning-based joint task offloading and migration schemes optimization in mobility-aware mec network," *China Communications*, vol. 17, no. 8, pp. 31–44, 2020.
23. A. Samanta and Z. Chang, "Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3864–3872, 2019.
24. N. Cheng, N. Lu, N. Zhang, X. Zhang, X. S. Shen, and J. W. Mark, "Opportunistic wifi offloading in vehicular environment: A game-theory approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1944–1955, 2016.
25. J. Bi, K. Zhang, H. Yuan, and Q. Hu, "Energy-aware task offloading with genetic particle swarm optimization in hybrid edge computing," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 3194–3199.
26. S. Li, H. Ge, X. Chen, L. Liu, H. Gong, and R. Tang, "Computation offloading strategy for improved particle swarm optimization in mobile edge computing," in *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*. IEEE, 2021, pp. 375–381.

27. H. Tout, A. Mourad, N. Kara, and C. Talhi, "Multi-persona mobility: joint cost-effective and resource-aware mobile-edge computation offloading," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1408–1421, 2021.
28. F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6g: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2019.
29. Y. Bai, L. Chen, L. Song, and J. Xu, "Risk-aware edge computation offloading using bayesian stackelberg game," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1000–1012, 2020.
30. Y. Qu, H. Dai, F. Wu, D. Lu, C. Dong, S. Tang, and G. Chen, "Robust offloading scheduling for mobile edge computing," *IEEE Transactions on Mobile Computing*, 2020.
31. X. Zhang and W. Sun, "Mean-cvar models for fuzzy portfolio selection," in *2010 International Conference on Intelligent System Design and Engineering Application*, vol. 1. IEEE, 2010, pp. 928–930.
32. R. Schultz and F. Neise, "Algorithms for mean-risk stochastic integer programs in energy," in *2006 IEEE Power Engineering Society General Meeting*. IEEE, 2006, pp. 8–pp.
33. Y. Chen, B. Ai, Y. Niu, H. Zhang, and Z. Han, "Energy-constrained computation offloading in space-air-ground integrated networks using distributionally robust optimization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 113–12 125, 2021.
34. S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE transactions on parallel and distributed systems*, vol. 26, no. 12, pp. 3317–3329, 2014.
35. P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1405–1418, 2020.
36. S. Pan, Z. Zhang, T. Xue, and G. Hu, "Enhancing availability for the mec service: Cvar-based computation offloading," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2020, pp. 342–347.
37. F. Vatalaro, G. E. Corazza, C. Caini, and C. Ferrarelli, "Analysis of leo, meo, and geo global mobile satellite systems in the presence of interference and fading," *IEEE Journal on selected areas in communications*, vol. 13, no. 2, pp. 291–300, 1995.
38. F. Qi, G. Mang, S. Zhang, and L. Liu, "A multi-layer architecture for space-air-ground network and iot services," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2021, pp. 1809–1813.
39. E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations research*, vol. 58, no. 3, pp. 595–612, 2010.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.