RESEARCH

A Risk-Sensitive Task Offloading Strategy for Edge Computing in Industrial Internet of Things

Xiaoyu Hao[1], Ruohai Zhao[1], Tao Yang[1,2*], Yulin Hu[3], Bo Hu[1,2*] and Yuhe Qiu[4]

*Correspondence:
taoyang@fudan.edu.cn;
bohu@fudan.edu.cn
[1]Research Center of Smart Networks and Systems, Department of Electronics Engineering, Fudan University, Shanghai, 200433, China
Full list of author information is available at the end of the article

Abstract

Edge computing has become one of the key enablers for ultra-reliable and low-latency communications in the industrial Internet of Things in the fifth generation communication systems, and is also a promising technology in the future sixth generation communication systems. In this work, we consider the application of edge computing to smart factories for mission-critical task offloading through wireless links. In such scenarios, although high end-to-end delays from the generation to completion of tasks happen with low probability, they may incur severe casualties and property loss, and should be seriously treated. Inspired by the risk management theory widely used in finance, we adopt the Conditional Value at Risk to capture the tail of the delay distribution. An upper bound of the Conditional Value at Risk is derived through analysis of the queues both at the devices and the edge computing servers. We aim to find out the optimal offloading policy taking into consideration both the average and the worst case delay performance of the system. Given that the formulated optimization problem is a non-convex mixed integer non-linear programming problem, a decomposition into sub-problems is performed and a two-stage heuristic algorithm is proposed. Simulation results validate our analysis and indicate that the proposed algorithm can reduce the risk in both the queueing and end-to-end delay.

Keywords: Ultra-reliable and low-latency communications, Edge computing, Industrial Internet of Things, Risk management theory, Conditional Value at Risk

1 Introduction

Intelligent factory automation is one of the typical applications envisioned in ultra-reliable and low-latency communications (URLLC) scenarios in the fifth generation (5G) and the coming sixth generation (6G) communications [1, 2]. In future smart factories, machines and sensors are seamlessly connected with each other through wireless links to conduct production tasks corporately. During the manufacturing process, a great number of operations of the machine and robots require complex control algorithms and intense data computation, such as travelling across zones to identify and pick up the objects and controlling the robotic arms to assemble components within a precise position alignment [3]. The limited built-in computing resources are not sufficient for the stringent latency requirements, so the tasks have to be offloaded to servers for processing [4]. Conventionally, the large volume of data generated at the local devices is uploaded to the cloud computing servers [5]. However, since the cloud computing servers are usually deployed remotely, the large roundtrip transmission latency as well as the possible network congestion make it hard to meet the stringent end-to-end delay requirements of the actuators and

Research Square

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

control units in the IIoT system. To overcome these difficulties, edge computing has emerged, where the servers are placed at the edge of the network to achieve a much lower transmission and processing latency [6].

There have been literatures focusing on improving the service efficiency of the edge computing systems [7, 8, 9, 10, 11]. The authors in [7] adopt the Markov decision process (MDP) to minimize the average delay of a mobile edge computing system by deciding whether to compute locally or offload the tasks to the edge server. In [8], the authors define a delay-based Lyapunov function instead of the queue length-based one, and minimize the average delay by optimizing resource scheduling under the Lyapunov optimization framework. Three-tier multi-server mobile computing networks are investigated in [9], where a cooperative task offloading strategy is proposed based on the Alternating Direction Method of Multipliers (ADMM). In [10], an adaptive learning-based task offloading algorithm is proposed to minimize the average delay for vehicle edge computing systems. Most relatively, [11] integrates the fog computing to the cloud-based industrial Internet of Things (IIoT), where the task offloading, transmission and computing resource allocation schemes are jointly optimized to reduce the service response latency in the unreliable communication environment.

The aforementioned works have only focused on the average delay performance, neglecting the worst case performance of the edge computing system. However, in the IIoT systems, the probability of a intense delay jitter usually matters much more than the average delay, since when the delay exceeds a certain threshold, severe accidents may incur such as the deadlock of the manufacture process, the damage to the machines and even casualties. Therefore, in such scenarios, not only the average delay performance, but also the potential hazard, i.e. the risk behind the tail distribution of the delay, should be carefully investigated. There have been some preliminary works to deal with the embedded risks in the edge computing systems [12, 13, 14]. In [12], the extreme value theory [15] is introduced to characterize the tail distribution of the task queue under a probability constraint imposed on the excess value, with which the offloading strategy is designed to minimize the energy consumption. The authors of [13] also apply the extreme value theory to the edge computing systems, in order to investigate the extreme event of queue length violation in the computation phase. Besides, in [14], the authors focus on a vehicular edge computing network where vehicles either fetch images from cameras or acquire synthesized images from an edge computing server. A risk-sensitive learning [16] based task fetching and offloading strategy is proposed to minimize the risk behind the end-to-end delay.

Different from the works mentioned above, we introduce the risk management theory [17], widely used in the field of finance, to the edge computing system in consideration of the uncertainty of the wireless channels. Value at risk (VaR) and Conditional Value at Risk (CVaR) are the two widely used tools to characterize risks. While VaR takes the Gaussian distribution as assumption and lacks convexity and sub-additivity, which makes it inapplicable in many cases, CVaR is a coherent risk measure of any type of probability distribution and much easier to handle in practice. Therefore, CVaR is employed in this work to model the risk of the task completion delay in the considered edge computing-assisted IIoT system. We aim to

minimize both the average delay and the CVaR by jointly designing the offloading and computing resource allocation strategy. The main contributions of this work is summarized as follows:

- We focus on the hazard incurred by the intense delay jitter in the edge computing-assisted IIoT scenario and introduce the risk management theory to the design of the offloading and computation resource allocation strategy.
- A cascade queueing model is constructed to describe the end-to-end delay property of the system. Due to the uncertainty of the wireless channel, the transmission time follows a general distribution, which makes the queueing model hard to analyze. By exploring the queueing theory and the risk management theory, we provide an upper bound for both the average end-to-end delay and the CVaR.
- A low-complexity risk-sensitive task offloading strategy is proposed, where both the average performance and the risk with respect to the end-to-end delay are optimized simultaneously. Simulations under the practical wireless environment in the automated factory validate the effectiveness of the proposed strategy in controlling the risk behind the intense delay jitter.

The remainder of this paper is organized as follows. We introduce the system model and analyze both the average delay and the CVaR in Section 2. In Section 3, we formulate the offloading and computation resource allocation problem and propose a low-complexity heuristic algorithm. In Section 4, numerical results are reported with discussions. Finally, Section 5 concludes the paper.

2 System model

2.1 Edge computing system

As shown in Fig. 1, we consider an edge computing-assisted IIoT system that consists of a set of $\mathcal{M} = \{1, 2, \cdots, M\}$ IIoT devices and a set of $\mathcal{N} = \{1, 2, \cdots, N\}$ edge computing servers (ECS). Each IIoT device $i \in \mathcal{M}$ randomly generate tasks of identical size of d_i bits, and we assume the task arrival process follows the Possion distribution with average arrival rate λ_i. We denote by ω_i the computation intensity of the task of device i [18], i.e. the number of CPU cycles required to process per bit data. Then, the total CPU cycles needed for a task of device i, denoted by c_i, can be calculated as $c_i = \omega_i d_i$.

Owing to the insufficiency of computation capability, the IIoT devices offload their tasks to the ECSs through wireless links. Each ECS $j \in \mathcal{N}$ is equipped with a CPU of N_j cores, which can work simultaneously and independently. We assume that the tasks of a device can only be offloaded to one ECS, while each core only processes the tasks from the same device, which means a ECS can receive tasks from multiple devices as long as the number of devices it serves doesn't exceed the number of the CPU cores [12]. Let $\mathbf{X}_{M\times N} = [x_{ij}]$ as the offloading matrix, where $x_{ij}, i \in \mathbf{M}, j \in \mathbf{N}$ is defined as follows:

$$x_{ij} = \begin{cases} 1\ , & \text{device } i \text{ offloads its tasks to ECS } j, \\ 0\ , & \text{otherwise.} \end{cases} \tag{1}$$

Under this definition, we can redescribe the offloading scheme in the considered system mathematically as $\sum_{j=1}^{N} x_{ij} = 1$ and $\sum_{i=1}^{M} x_{ij} \leq N_j$ for $\forall i \in \mathbf{M}, \forall j \in \mathbf{N}$.

Due to the massive deployment of devices in the complex industrial environment, it's impractical to obtain the instantaneous channel state information (CSI) accurately and timely. Therefore, in this work, we design a task offloading strategy based on statistics of the wireless links, i.e. the distribution of the channel gain. We assume blocking-fading channels such that the channel gain remains unchanged during the execution of one task and varies independently between two executions following an identical distribution, which is known a priori. Denote by g_{ij} the channel gain from device i to ECS j, the transmission rate can be expressed as follows:

$$R_{ij} = B \log_2 (1 + \frac{g_{ij} p_i}{N_0 \Phi_{ij}}), \tag{2}$$

where B is the bandwidth, N_0 is the noise power, p_i is the transmit power of device i and Φ_{ij} is the path loss from device i to ECS j. Without loss of generality, we assume that the noise power at each ECS is identical, and each IIoT device has an orthogonal channel with the same bandwidth B.

2.2 Queueing model

In the considered edge computing-assisted IIoT system, there are two kinds of queues: the queue at each device and the queue at each ECS, as depicted in Fig. 2. Without loss of generality, we assume that device i offloads its tasks to ECS j, and denote by Q_{ij}^D the queue formed at the device i. The arrival process of Q_{ij}^D follows the Poisson process, and the departure process is dependent on the transmission delay denoted by t_{ij}^D, which is given by

$$t_{ij}^D = \frac{d_i}{R_{ij}} = \frac{d_i}{B \log_2 (1 + \frac{g_{ij} p_i}{N_0 \Phi_{ij}})}. \tag{3}$$

Therefore, Q_i^D follows the M/G/1 model.

As for an ECS, tasks from each device connected to it form an independent queue. Denote by Q_{ij}^S the queue of tasks offloaded from device i to ECS j, then the arrival process of Q_{ij}^S is the same as the departure process of Q_{ij}^D. We denote by the f_{ij} the computation frequency allocated to device i by ECS j, then the computation delay denoted by t_{ij}^S, i.e. the time required to complete a task of device i, is calculated as $t_{ij}^S = c_i / f_{ij}$. As a result, the service time of a task follows a deterministic distribution and thus Q_{ij}^S follows the G/D/1 model.

Based on the queueing analysis, when device i offloads its tasks to ECS j, the total delay denoted by t_{ij} is given by

$$t_{ij} = W_{ij}^D + t_{ij}^D + W_{ij}^S + t_{ij}^S, \tag{4}$$

where W_{ij}^D and W_{ij}^S are the queueing delay at the device i and the ECS j, respectively. We 'll analyse both the average performance and the CVaR of the total delay in the following.

2.3 Average delay

According to (4), the average delay can be calculated as follows

$$E[t_{ij}] = E[W_{ij}^D] + E[t_{ij}^D] + E[W_{ij}^S] + E[t_{ij}^S]. \tag{5}$$

As for the queueing delay at device i, we denote by μ_{ij} the service rate of Q_{ij}^D, which can be calculated as the reciprocal of the average transmission time, i.e.

$$\mu_{ij} = \frac{1}{E[t_{ij}^D]} = \frac{d_i}{E[R_{ij}]}, \tag{6}$$

where the expectation is taken over the probability distribution of the channel gain. According to [19], the average queueing delay at device i can be expressed as follows

$$E[W_{ij}^D] = \frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})}, \tag{7}$$

where $\rho_{ij} = \lambda_i/\mu_{ij}$.

To analyze the queueing delay in the G/D/1 queueing model of Q_{ij}^S, we first give the following lemma [20].

Lemma 1 *In the G/G/1 queueing model, let λ, μ and W be the arrival rate, service rate and queueing delay, respectively, then an upper bound of the average queueing delay is given by*

$$E[W] \leq \frac{\lambda(\sigma_a^2 + \sigma_b^2)}{2(1-\rho)}, \tag{8}$$

where $\rho = \lambda/\mu$, σ_a^2 is the variance of the inter-arrival time, and σ_b^2 is the variance of the service time.

According to Lemma 1, we obtain the following theorem characterizing the upper bound of the average queueing delay at a ECS.

Theorem 1 *When device i offloads its tasks to ECS j, an upper bound of the average queueing delay at ECS j is given by*

$$E[W_{ij}^S] \leq \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)}, \tag{9}$$

where λ_{ij}^S is the arrival rate of Q_{ij}^S, $\rho_{ij}^D = \lambda_{ij}^S/\mu_{ij}$ is the traffic intensity and ${\sigma_{ij}^S}^2$ is the variance of the arrival interval of tasks offloaded from device i to ECS j.

Proof G/D/1 model can be seen as a special case of G/G/1 model with the service time following a deterministic distribution, the variance of which is zero. By substituting $\lambda = \lambda_{ij}^S$, $\sigma_a^2 = {\sigma_{ij}^S}^2$, $\sigma_b^2 = 0$ and $\rho = \rho_{ij}^S$ into (8), we get (9) and Theorem 1 is proved. □

Note that, due to the cascaded structure between Q_{ij}^D and Q_{ij}^S, the arrival rate λ_{ij}^S of Q_{ij}^S is equal to the departure rate of Q_{ij}^S, which can be evaluated from the analysis of the inter-departure time in [21]. Similarly, the variance of the inter-arrival time of Q_{ij}^S, i.e. ${\sigma_{ij}^S}^2$, can be derived from the variance of the inter-departure time of Q_{ij}^D as in [22].

Finally, combining (5), (6) and (9), the upper bound of the average total delay can be obtained in the following corollary.

Corollary 1 *When device i offloads its tasks to ECS j, an upper bound of the average total delay $E[t_{ij}]$ is given by*

$$E[t_{ij}] \leq \frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})} + \frac{1}{\mu_{ij}} + \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{c_i}{f_{ij}}. \tag{10}$$

Since each IIoT device offloads its tasks to only one ECS, for each device i the task completion time denoted by t_i can be expressed as $t_i = \sum_{j=1}^N x_{ij} t_{ij}$, and correspondingly an upper bound of the average total delay of device i based on (10) is given by

$$E[t_i] = \sum_{j=1}^N x_{ij} E[t_{ij}] \leq E[t_i]^*, \tag{11}$$

where

$$E[t_i]^* = \sum_{j=1}^N x_{ij} \left(\frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})} + \frac{1}{\mu_{ij}} + \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{c_i}{f_{ij}} \right). \tag{12}$$

2.4 Risk metric for delay

Risk in the considered edge computing-assisted system is mainly reflected in the high latency happened with low probability. Specifically, we introduce CVaR as a measure of risk to characterize the tail distribution of the delay. Before we formally define CVaR, we first give the definition of VaR [23].

Definition 1 *For a random variable X and a confidence level $\alpha \in (0,1)$, the α-VaR of X is the α-percentile of the distribution of X, which can be expressed mathematically as follows*

$$\mathrm{VaR}_\alpha(X) = \inf_\gamma \{\gamma : P(X > \gamma) \leq \alpha\}. \tag{13}$$

The CVaR measures the expected loss in the right tail given a particular threshold has been crossed, and can also be considered as the average of potential loss that exceed the VaR. The definition of CVaR is given as follows [24].

Definition 2 *For a random variable X and a confidence level $\alpha \in (0,1)$, the α-CVaR of X is given by*

$$\begin{aligned}\mathrm{CVaR}_\alpha(X) &= E[X|X > \mathrm{VaR}_\alpha(X)] \\ &= \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_\theta(X)d\theta. \end{aligned} \tag{14}$$

To characterize the CVaR of the total delay, we first analyze the CVaR of each part of the total delay in (4). Recall that the service process of the queue at the device follows general distribution, so it is quite difficult to directly characterize the probability distribution of the waiting time. However, in the considered IIoT scenario, it is reasonable to take the *heavy* traffic assumption. According to [25], the cumulative distribution function (CDF) of W_{ij}^D can be approximated as

$$F(W_{ij}^D) \approx 1 - \exp\left[-\frac{2(1-\rho_{ij})}{\lambda_i V_{ij}} W_{ij}^D\right], \tag{15}$$

where V_{ij} is the variance of the service time of Q_{ij}^D, i.e. the transmission time of a task. Based on (15), the CVaR of W_{ij}^D can be evaluated in the following theorem.

Theorem 2 *For a confidence level $\alpha \in (0,1)$, the α-CVaR of W_{ij}^D can be expressed as*

$$\mathrm{CVaR}_\alpha(W_{ij}^D) = \frac{\lambda_i V_{ij}}{2(1-\rho_{ij})}[1 - \ln(1-\alpha)]. \tag{16}$$

Proof According to the definition of VaR in Definition 1, we can obtain that

$$\begin{aligned}\mathrm{VaR}_\alpha(W_{ij}^D) &= \inf_\gamma \{\gamma : e^{-\frac{2(1-\rho_{ij})}{\lambda_i V_{ij}}\gamma} \le \alpha\} \\ &= \frac{-\lambda_i V_{ij}}{2(1-\rho_{ij})}\ln(1-\alpha). \end{aligned} \tag{17}$$

By substituting (17) to (14), the CVaR of W_{ij}^D can be calculated as follows.

$$\begin{aligned}\mathrm{CVaR}_\alpha(W_{ij}^D) &= \frac{1}{1-\alpha}\int_\alpha^1 \frac{-\lambda_i V_{ij}}{2(1-\rho_{ij})}\ln(1-\theta)d\theta \\ &= \frac{\lambda_i V_{ij}}{2(1-\rho_{ij})}[1 - \ln(1-\alpha)]. \end{aligned} \tag{18}$$

□

As for the transmission delay t_{ij}^D, we define a auxiliary function

$$\phi_\alpha(t_{ij}^D, \gamma) := \gamma + \frac{1}{1+\alpha}E[(t_{ij}^D - \gamma)^+], \tag{19}$$

where $(x)^+ = \max(0, x)$ and the expectation is taken over the distribution of the channel gain. According to [26], the CVaR of t_{ij}^D can finally be calculated as

$$\begin{aligned}\mathrm{CVaR}_\alpha(t_{ij}^D) &= \min_{\gamma\in\mathcal{R}} \phi_\alpha(t_{ij}^D, \gamma) \\ &= \min_{\gamma\in\mathcal{R}} \{\gamma + \frac{1}{1+\alpha} E[(t_{ij}^D - \gamma)^+]\}\end{aligned} \tag{20}$$

Now we turn to the queue at the ECS. Similar to (15), the CDF of W_{ij}^S can be approximated as

$$F(W_{ij}^D) \approx 1 - \exp\left[-\frac{2(1-\rho_{ij}^S)}{\lambda_{ij}^S {\sigma_{ij}^S}^2} W_{ij}^S\right], \tag{21}$$

and thus the CVaR of the queueing delay at the ECS can be evaluated in the following theorem.

Theorem 3 *For a confidence level $\alpha \in (0,1)$, the α-CVaR of W_{ij}^S can be expressed as*

$$\mathrm{CVaR}_\alpha(W_{ij}^S) = \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)}[1 - \ln(1-\alpha)]. \tag{22}$$

The proof of Theorem 3 is similar to Theorem 2 and is omitted here for brevity.

Since we assume constant computing capability at the ECS, the CVaR of the service time of Q_{ij}^S is at the same value as itself, i.e.

$$\mathrm{CVaR}_\alpha(t_{ij}^S) = \frac{c_i}{f_{ij}}. \tag{23}$$

With the CVaR of each part of the delay involved in the task offloading, we provide an upper bound of the CVaR of the total delay in the following theorem based on the convexity and subadditivity [27].

Theorem 4 *For a confidence level $\alpha \in (0,1)$, an upper bound of the α-CVaR of t_i is given by*

$$\mathrm{CVaR}_\alpha(t_i) \le \mathrm{CVaR}_\alpha(t_i)^*, \tag{24}$$

where

$$\mathrm{CVaR}_\alpha(t_i)^* = \sum_{j=1}^{N} x_{ij} \left[\mathrm{CVaR}_\alpha(W_{ij}^D) + \mathrm{CVaR}_\alpha(t_{ij}^D) + \mathrm{CVaR}_\alpha(W_{ij}^S) + \mathrm{CVaR}_\alpha(t_{ij}^S)\right]. \tag{25}$$

Proof According to the convexity, the α-CVaR of t_i satisfies the following Jensen inequality:

$$\mathrm{CVaR}_\alpha(t_i) = \mathrm{CVaR}_\alpha(\sum_{j=1}^{N} x_{ij} t_{ij}) \leq \sum_{j=1}^{N} x_{ij} \mathrm{CVaR}_\alpha(t_{ij}). \tag{26}$$

Furthermore, based on (4) and the subadditivity of the CVaR, we have the following inequality:

$$\mathrm{CVaR}_\alpha(t_{ij}) \leq \mathrm{CVaR}_\alpha(W_{ij}^D) + \mathrm{CVaR}_\alpha(t_{ij}^D) + \mathrm{CVaR}_\alpha(W_{ij}^S) + \mathrm{CVaR}_\alpha(t_{ij}^S). \tag{27}$$

By combining (26) and (27), Theorem 4 is proved. □

3 Problem formulation and solution

3.1 Problem formulation

In the design of the edge computing-assisted IIoT system, not only the average latency but also risk behind the intense delay jitter should be carefully considered. Taking into account both the average delay performance and the risk, we set the objective of the task offloading problem as the weighted sum of the average delay and the CVaR, i.e. the mean-risk sum. We have shown that obtaining an explicit expression of both the two terms is often cumbersome, especially for the complex wireless environment in the automated factories. Therefore, the two upper bounds of the average total delay and the corresponding CVaR derived in the previous section are adopted instead. Furthermore, in the considered mission-critical IIoT scenario, the performance of the whole system is usually determined by the device with the worst performance. As a result, we aim to minimize the maximum mean-risk sum among all the devices, which can be described as the following optimization problem:

$$\min_{\mathbf{X},\mathbf{f}} \max_{i} \quad E[t_i]^* + \beta \mathrm{CVaR}_\alpha(t_i)^* \tag{28}$$

$$\text{s.t.} \quad x_{ij} \in \{0,1\}, \forall i \in \mathcal{M}, \forall j \in \mathcal{N}, \tag{28a}$$

$$\sum_{j=1}^{N} x_{ij} = 1, \forall i \in \mathcal{M}, \tag{28b}$$

$$\sum_{i=1}^{M} x_{ij} \leq N_j, \forall j \in \mathcal{N}, \tag{28c}$$

$$\sum_{i=1}^{M} f_{ij} \leq F_j, \forall j \in \mathcal{N}, \tag{28d}$$

where $\mathbf{f} = [f_{ij}], i \in \mathcal{M}, j \in \mathcal{N}$ is the computation frequency allocation matrix, $\beta \in (0,1)$ is the weight of the CVaR, also called the risk-sensitive parameter, and F_j is the overall computation frequency of ECS j. Constraint (28b) is used to guarantee that the tasks generated by the device can only be offloaded to one ECS. Constraint (28c) and (28d) indicate that the number of devices served by a ECS should not exceed the number of its CPU cores, and the sum of the computation frequency allocated to these devices should not exceed its overall computation

frequency. Substituting (12), (18), (20), (22) and (23) to (28), we find that the optimization problem is a non-convex mixed integer non-linear problem (MINLP), which is NP-hard [28]. To reduce the computation overhead, we propose a heuristically algorithm, which will be described in details in the following subsection.

3.2 Problem solving

Recall that the CVaR of the queueing delay at the device in (20) is in the form of an minimization problem. Since the optimization variable γ in (20) is independent of the optimization variables $\mathbf{X}$ and $\mathbf{f}$ in (28), we can solve (20) first and substitute its optimal solution to (28) for the subsequent problem solving.

To solve (20), we introduce an auxiliary variable $z_ij = (t_{ij}^D - \gamma)^+$, and problem (20) can be transformed to the following problem

$$\min_{\gamma\in\mathcal{R},z_{ij}} \quad \gamma + \frac{1}{1+\alpha}E[z_{ij}] \tag{29}$$

$$\text{s.t.} \quad z_{ij} \geq t_{ij}^D - \gamma, \tag{29a}$$

$$z_{ij} \geq 0. \tag{29b}$$

Problem (29) is a stochastic optimization problem with the expectation taken over the channel gain g_{ij}. To approximate the expectation, we sample the probability distribution of g_{ij} [26], and a transformed problem is obtained as follows:

$$\min_{\gamma\in\mathcal{R},\mathbf{z_{ij}}} \quad \gamma + \frac{1}{1+\alpha}\frac{1}{K}\sum_{k=1}^{K} z_{ij}^k \tag{30}$$

$$\text{s.t.} \quad z_{ij}^k \geq t_{ij}^D - \gamma, \forall k \in \mathcal{K}, \tag{30a}$$

$$z_{ij}^k \geq 0, \forall k \in \mathcal{K}, \tag{30b}$$

where $z_{ij}^k, k \in \mathcal{K} = \{1, 2, \cdots, K\}$ are the samples of z_{ij}. Problem (30) is a linear optimization problem, the optimal solution of which, denoted by U_{ij}, can be obtained from the interior point (IPT) method [29].

With all the derived average and CVaR terms, problem (28) can be reformulated as the following optimization problem:

$$\min_{\mathbf{X},\mathbf{f}} \max_{i} \quad \sum_{j=1}^{N} x_{ij}\Bigg\{\Bigg[\frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})} + \frac{1}{\mu_{ij}} + \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{c_i}{f_{ij}}\Bigg] + \beta\Bigg[\Big[\frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{\lambda_i V_{ij}}{2(1-\rho_{ij})}\Big][1-\ln(1-\alpha)] + U_{ij} + \frac{c_i}{f_{ij}}\Bigg]\Bigg\} \tag{31}$$

$$\text{s.t.} \quad (28a), (28b), (28c), (28d). \tag{31a}$$

It is obvious that the objective function of problem (31) contains the term x_{ij}/f_{ij}, and thus problem (31) is still a non-convex MINLP. In order to reduce the computational complexity, we decompose the original problem into two sub-problems.

First, we consider the following problem:

$$\min_{\mathbf{X}} \max_{i} \quad \sum_{j=1}^{N} x_{ij} \Bigg\{ \Bigg[\frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})} + \frac{1}{\mu_{ij}} \Bigg] + + \beta \Bigg[\frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} [1 - \ln(1-\alpha)] + U_{ij} \Bigg] \Bigg\} \tag{32}$$

$$\text{s.t.} \quad (28a), (28b), (28c). \tag{32a}$$

It is worthwhile to mention that Problem (32) is a convex MINLP, which can be solved efficiently, e.g., via an outer approximation algorithm or an extended cutting plane algorithm [30]. Substitute the optimal offloading matrix for problem (32), denoted by $X^* = [x_{ij}^*], i \in \mathcal{M}, j \in \mathcal{N}$, to (28), and the second sub-problem can be formulated as follows:

$$\min_{\mathbf{f}} \max_{i} \quad \sum_{j=1}^{N} x_{ij}^* \Bigg\{ \Bigg[\frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})} + \frac{1}{\mu_{ij}} + \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{c_i}{f_{ij}} \Bigg] + \beta \Bigg[[\frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{\lambda_i V_{ij}}{2(1-\rho_{ij})}] [1 - \ln(1-\alpha)] + U_{ij} + \frac{c_i}{f_{ij}} \Bigg] \Bigg\} \tag{33}$$

$$\text{s.t.} \quad (28d). \tag{33a}$$

Problem (33) is a non-convex optimization problem. To transform it to a convex problem, we introduce an auxiliary variable $\mathbf{G} = [1/f_{ij}], i \in \mathcal{M}, j \in \mathcal{N}$, and an optimization problem equivalent to (33) is given by

$$\min_{\mathbf{G}} \max_{i} \quad \sum_{j=1}^{N} x_{ij}^* \Bigg\{ \Bigg[\frac{\lambda_i}{2\mu_{ij}^2(1-\rho_{ij})} + \frac{1}{\mu_{ij}} + \frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + c_i G_{ij} \Bigg] + \beta \Bigg[[\frac{\lambda_{ij}^S {\sigma_{ij}^S}^2}{2(1-\rho_{ij}^S)} + \frac{\lambda_i V_{ij}}{2(1-\rho_{ij})}] [1 - \ln(1-\alpha)] + U_{ij} + c_i G_{ij} \Bigg] \Bigg\} \tag{34}$$

$$\text{s.t.} \quad \frac{1}{F_j} \leq G_{ij} \leq \frac{1}{x_{ij}^* \lambda_{ij}^S c_i}, \forall i \in \mathcal{M}, \forall j \in \mathcal{N}, \tag{34a}$$

The right side of inequality (34a) is used to maintain the stability of the queue at the ECS. Problem (34) is a convex optimization problem, which can be solved by various algorithms such as the IPT method. Till now, both the offloading matrix $\mathbf{X}$ and the frequency allocation matrix $\mathbf{f}$ have been solved.

4 Results and discussion

In this section, we evaluate the proposed strategy through numerical results. In the simulation, we consider the scenario where 8 IIoT devices offload their tasks to 2 ECSs. Without loss of generality, we set the data size to 0.5 Mbits and the task computation intensity to 15 cycles/bit for each task of each device. Each ECS is equiped with a four-core CPU. The task arrival rate of each device, i.e. the

parameter of the Poisson process, is set to be uniformly distributed in $(10, 30)$. The bandwidth of each wireless channel is 10 MHz, and the transmission power, noise power at the receiver and the path loss are all set to be identical for each device at 30 dBm, 10^{-9} W and 70 dB, respectively. To characterize the fading channel in the practical automated factory, we set the channel distribution as a mixture of Rayleigh and log-normal distribution, which has been confirmed by the measurements in the real industrial environment [31]. The parameter of the Rayleigh distribution is set to be uniformly distributed in $(0.5, 1)$ for each IIoT-ECS pair, and correspondingly, the two parameters of the log-normal distribution are set to be uniformly distributed in $(1, 2)$ and $(0, 4)$ respectively. Finally, we set the confidence level to $\alpha = 0.99$.

Our proposed strategy considers both the queueing effect and the risk behind the total delay, and thus is denoted by queueing-based and risk-sensitive (Q-R) strategy in the following simulations. To evaluate the performance of the Q-R, we compare it with the following three strategies: i) the queueing-based and non-risk-sensitive (Q-NR) strategy, which considers the queueing effect but only optimizes the average delay performance, i.e. the weight of the CVaR is set to $\beta = 0$; ii) the non-queueing-based and risk-sensitive (NQ-R) strategy, which takes into account both the average delay and the CVaR, but does not consider the queueing effect; iii) the non-queueing-based and non-risk-sensitive (NQ-NR), which considers neither the queueing effect nor the risk. In the following simulations, we set the weight of the CVaR to $\beta = 2$ for risk-sensitive strategies.

We first investigate the complementary cumulative distribution function (CCDF) of the total delay under the four offloading strategies, since the CVaR captures the tail information of the delay distribution. As presented in Fig. 3, for the probability of ultra-high delay, the curve of Q-R is lower than that of Q-NR, and the cure of NQ-R is also lower than that of NQ-NR. This implies that by adding the CVaR to the optimization objective, the risk of high delay can be greatly reduced. On the other hand, we can see that for any value of the total delay, the CCDF under NQ-R and NQ-NR is greater than that under Q-R and Q-NR, which means the non-queueing strategies are more likely to arise a higher delay. This is reasonable, since the non-queueing strategies neglect the queueing effect in the strategy design, which leads to a higher queueing delay.

In Fig. 4 and Fig. 5, we compare how the delay performance evolves with the computation frequency of the ECS under the four strategies. Specifically, Fig.4 investigates relationship between the average delay and the computation frequency, and Fig. 5 focus on the 99th percentile of the total delay. It can be seen that with the increasing computation frequency, the average total delay and the 99th percentile decreases for all the four strategies. The reason is that the higher the computation frequency, the more computation resources allocated to the IIoT devices and thus the lower the computation delay. Furthermore, note that the delay doesn't descend much when computation frequency is relatively high. This is because for high computation frequency, both the computation delay and the queueing delay at the ECS is relatively low, and thus the total delay is mainly dependent on the queueing delay at the devices. We can also see that the queueing strategies outperforms the corresponding non-queueing strategies for both the average performance and the 99th percentile, which verifies the significance of the queueing analysis again. More

importantly, the two figures jointly indicate that the risk-sensitive strategies achieve nearly the same average total delay as the non-risk-sensitive ones, but greatly reduce the 99th percentile of the total delay by incorporating the risk to the design of offloading strategy. In other words, the intense delay jitter can be effectively controlled under the risk-sensitive strategies only at the price of very little degradation on the average performance.

Finally, we investigate the effect of the task size on both the average delay and the 99th percentile under the Q-R and Q-NR strategies. As shown in Fig. 6, the 99th percentile is higher than the average total delay for both strategies, since the former characterize the worst case delay. With the increase of the task size, both the average delay and the 99th percentile increase under both strategies. This is due to the fact that a larger task size lead to the higher transmission and computation delay. Furthermore, the 99th percentile of Q-R is always lower than that of Q-NR, while the two curves of the average delay almost coincide with each other. Note that the average delay under the Q-NR strategy is the lower bound of that under the Q-R. This implies that Q-R achieves nearly the same average performance as the Q-NR while simultaneously improving the worst case performance with respect to the total delay.

5 Conclusions

In this work, we introduce the risk management theory to design of the edge computing-assisted IIoT system. We explore the queueing theory and the properties of the CVaR to capture the tail distribution of the end-to-end delay, and provide two upper bounds of the average total delay and the CVaR. A joint task offloading and computation resource allocation problem is formulated to simultaneously minimize the average total delay and the risk. Since the problem is a non-convex MINLP, we decompose it into two sub-problems and design a two-stage heuristic algorithm. Finally, simulations are performed under the practical channel model in the automated factories, and the results verify that the proposed strategy can effectively control the risk of intense delay jitter while guaranteeing the average delay performance.

Acknowledgements
Not applicable

Funding
This work was supported by the Shanghai Municipal Natural Science Foundation (No. 19ZR1404700).

Abbreviations
URLLC: Ultra-reliable low-latency communications; 5G: fifth generation; 6G: sixth generation; MDP: Markov decision process; ADMM: Alternating Direction Method of Multipliers; IIoT: Industrial Internet of Things; VaR: Value at Risk; CVaR: Conditional Value at Risk; ECS: Edge computing server; CDF: Cumulative distribution function; MINLP: Mixed integer non-linear problem; IPT: Interior point; Q-R: Queueing-based and risk-sensitive; Q-NR: Queueing-based and non-risk-sensitive; NQ-R: Non-queueing-based and risk-sensitive; NQ-NR: Non-queueing-based and non-risk-sensitive; CCDF: Complementary cumulative distribution function

Availability of data and materials
Text for this section...

Ethics approval and consent to participate
Not applicable

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable

Authors' contributions
All the authors contributed to the system design, queueing and CVaR analysis, strategy design, simulations and the writing of this paper. All authors read and approved the final manuscript.

Authors' information
Not applicable

Author details
[1]Research Center of Smart Networks and Systems, Department of Electronics Engineering, Fudan University, Shanghai, 200433, China. [2]Key Laboratory of Electromagnetic Wave Information (MoE), Fudan University, Shanghai, 200433, China. [3]ISEK Research Group, RWTH Aachen University, Aachen, Germany. [4]China Mobile (Chengdu), Information & Telecommunication Technology Co., Ltd., Chengdu, China.

References

1. Gu, Z., She, C., Hardjawana, W., Lumb, S., McKechnie, D., Essery, T., Vucetic, B.: Knowledge-assisted deep reinforcement learning in 5g scheduler design: From theoretical framework to implementation. arXiv preprint arXiv:2009.08346 (2020)
2. She, C., Sun, C., Gu, Z., Li, Y., Yang, C., Poor, H.V., Vucetic, B.: A tutorial of ultra-reliable and low-latency communications in 6g: Integrating theoretical knowledge into deep learning. arXiv preprint arXiv:2009.06010 (2020)
3. Wan, J., Yang, J., Wang, Z., Hua, Q.: Artificial intelligence for cloud-assisted smart factory. IEEE Access **6**, 55419–55430 (2018)
4. Elbamby, M.S., Perfecto, C., Liu, C., Park, J., Samarakoon, S., Chen, X., Bennis, M.: Wireless edge computing with latency and reliability guarantees. Proceedings of the IEEE **107**(8), 1717–1737 (2019)
5. Velte, T., Velte, A., Elsenpeter, R.: Cloud Computing, a Practical Approach. McGraw-Hill, Inc., USA (2009)
6. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: Vision and challenges. IEEE Internet of Things Journal **3**(5), 637–646 (2016). doi:10.1109/JIOT.2016.2579198
7. Liu, J., Mao, Y., Zhang, J., Letaief, K.B.: Delay-optimal computation task scheduling for mobile-edge computing systems. In: 2016 IEEE International Symposium on Information Theory (ISIT), pp. 1451–1455 (2016)
8. Zhang, Y., Du, P., Wang, J., Ba, T., Ding, R., Xin, N.: Resource scheduling for delay minimization in multi-server cellular edge computing systems. IEEE Access **7**, 86265–86273 (2019)
9. Wang, Y., Tao, X., Zhang, X., Zhang, P., Hou, Y.T.: Cooperative task offloading in three-tier mobile computing networks: An admm framework. IEEE Transactions on Vehicular Technology **68**(3), 2763–2776 (2019). doi:10.1109/TVT.2019.2892176
10. Sun, Y., Guo, X., Song, J., Zhou, S., Jiang, Z., Liu, X., Niu, Z.: Adaptive learning-based task offloading for vehicular edge computing systems. IEEE Transactions on Vehicular Technology **68**(4), 3061–3074 (2019)
11. Shi, C., Ren, Z., Yang, K., Chen, C., Zhang, H., Xiao, Y., Hou, X.: Ultra-low latency cloud-fog computing for industrial internet of things. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2018)
12. Liu, C., Bennis, M., Debbah, M., Poor, H.V.: Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Transactions on Communications **67**(6), 4132–4150 (2019)
13. Zhu, Y., Hu, Y., Yang, T., Schmeink, A.: Reliability-optimal offloading in multi-server edge computing networks with transmissions carried by finite blocklength codes. In: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6 (2019). doi:10.1109/ICCW.2019.8757175
14. Batewela, S., Liu, C., Bennis, M., Suraweera, H.A., Hong, C.S.: Risk-sensitive task fetching and offloading for vehicular edge computing. IEEE Communications Letters **24**(3), 617–621 (2020). doi:10.1109/LCOMM.2019.2960777
15. De Haan, L., Ferreira, A.: Extreme Value Theory: an Introduction. Springer, New York (2007)
16. Mihatsch, O., Neuneier, R.: Risk-sensitive reinforcement learning. Machine learning **49**(2-3), 267–290 (2002)
17. Hessami, A.G. (ed.): Perspectives on Risk, Assessment and Management Paradigms. Books, vol. 5615. IntechOpen, London (2019). doi:10.5772/intechopen.77127. https://ideas.repec.org/b/ito/pbooks/5615.html
18. You, C., Huang, K., Chae, H., Kim, B.: Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Transactions on Wireless Communications **16**(3), 1397–1411 (2017). doi:10.1109/TWC.2016.2633522
19. Bertsekas, D.P., Gallager, R.G., Humblet, P.: Data Networks vol. 2. Prentice-Hall International, New Jersey (1992)
20. Kleinrock, L.: Queueing Systems, Volume 2: Computer Applications vol. 66. wiley, New York (1976)
21. Bertsimas, D.J., Nakazato, D.: The departure process from a gi/g/1 queue and its applications to the analysis of tandem queues (1990)
22. Yeh, P.-C., Chang, J.-F.: Characterizing the departure process of a single server queue from the embedded markov renewal process at departures. Queueing systems **35**(1-4), 381–395 (2000)
23. Jorion, P.: Value at risk (2000)
24. Acerbi, C., Tasche, D.: On the coherence of expected shortfall. Journal of Banking & Finance **26**(7), 1487–1503 (2002)
25. Kingman, J.F.: On queues in heavy traffic. Journal of the Royal Statistical Society: Series B (Methodological) **24**(2), 383–392 (1962)
26. Rockafellar, R.T., Uryasev, S., *et al.*: Optimization of conditional value-at-risk. Journal of risk **2**, 21–42 (2000)
27. Pflug, G.C.: Some remarks on the value-at-risk and the conditional value-at-risk. In: Probabilistic Constrained Optimization, pp. 272–281. Springer, Boston, MA (2000)

28. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization vol. 6. Athena Scientific, Belmont, MA (1997)
29. Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge university press, UK (2004)
30. Kronqvist, J., Bernal, D.E., Lundell, A., Grossmann, I.E.: A review and comparison of solvers for convex minlp. Optimization and Engineering **20**(2), 397–455 (2019)
31. Olofsson, T., Ahlen, A., Gidlund, M.: Modeling of the fading statistics of wireless sensor network channels in industrial environments. IEEE Transactions on Signal Processing **64**(12), 3021–3034 (2016)

Figures

Figure 1 System model

Figure 2 Queueing model

Figure 3 CCDF of the total delay

Figure 4 Average total delay versus the computation frequency where the task size is 0.5 Mbits

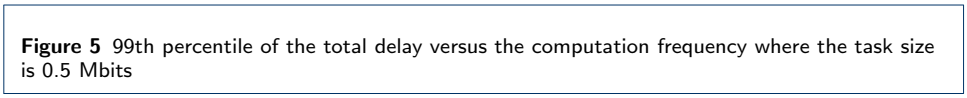

Figure 5 99th percentile of the total delay versus the computation frequency where the task size is 0.5 Mbits

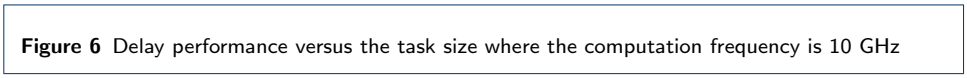

Figure 6 Delay performance versus the task size where the computation frequency is 10 GHz