EdgeLeague

Paper Analysis

Giuseppe Prisco - 1895709

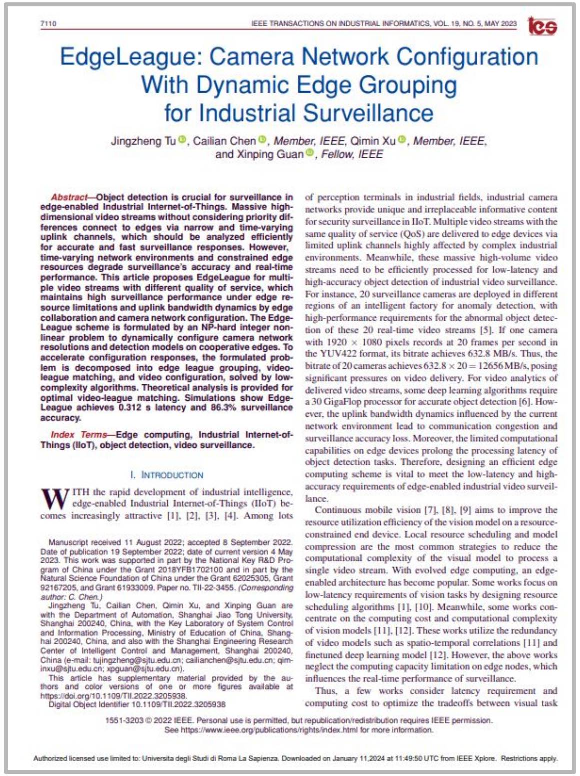
“**Engineering in Computer Science**” Master’s Degree
Mobile Applications and Cloud Computing (2023/2024)

Analyzed Paper

EdgeLeague: Camera Network Configuration With Dynamic Edge Grouping for Industrial Surveillance

2023 - Jingzheng Tu, Cailian Chen, Qimin Xu, and Xinping Guan

https://ieeexplore.ieee.org/document/9894446



EdgeLeague: Camera Network Configuration With Dynamic Edge Grouping for Industrial Surveillance

Jingzheng Tu, Cailian Chen, *Member, IEEE*, Qimin Xu, *Member, IEEE*, and Xinping Guan, *Fellow, IEEE*

***Abstract*—Object detection is crucial for surveillance in edge-enabled Industrial Internet-of-Things. Massive high-dimensional video streams without considering priority differences connect to edges via narrow and time-varying uplink channels, which should be analyzed efficiently for accurate and fast surveillance responses. However, time-varying network environments and constrained edge resources degrade surveillance's accuracy and real-time performance. This article proposes EdgeLeague for multiple video streams with different quality of service, which maintains high surveillance performance under edge resource limitations and uplink bandwidth dynamics by edge collaboration and camera network configuration. The EdgeLeague scheme is formulated by an NP-hard integer nonlinear problem to dynamically configure camera network resolutions and detection models on cooperative edges. To accelerate configuration responses, the formulated problem is decomposed into edge league grouping, video-league matching, and video configuration, solved by low-complexity algorithms. Theoretical analysis is provided for optimal video-league matching. Simulations show EdgeLeague achieves 0.312 s latency and 86.3% surveillance accuracy.**

***Index Terms*—Edge computing, Industrial Internet-of-Things (IIoT), object detection, video surveillance.**

I. INTRODUCTION

WITH the rapid development of industrial intelligence, edge-enabled Industrial Internet-of-Things (IIoT) becomes increasingly attractive [1], [2], [3], [4]. Among lots of perception terminals in industrial fields, industrial camera networks provide unique and irreplaceable informative content for security surveillance in IIoT. Multiple video streams with the same quality of service (QoS) are delivered to edge devices via limited uplink channels highly affected by complex industrial environments. Meanwhile, these massive high-volume video streams need to be efficiently processed for low-latency and high-accuracy object detection of industrial video surveillance. For instance, 20 surveillance cameras are deployed in different regions of an intelligent factory for anomaly detection, with high-performance requirements for the abnormal object detection of these 20 real-time video streams [5]. If one camera with 1920×1080 pixels records at 20 frames per second in the YUV422 format, its bitrate achieves 632.8 MB/s. Thus, the bitrate of 20 cameras achieves $632.8 \times 20 = 12656$ MB/s, posing significant pressures on video delivery. For video analytics of delivered video streams, some deep learning algorithms require a 30 GigaFlop processor for accurate object detection [6]. However, the uplink bandwidth dynamics influenced by the current network environment lead to communication congestion and surveillance accuracy loss. Moreover, the limited computational capabilities on edge devices prolong the processing latency of object detection tasks. Therefore, designing an efficient edge computing scheme is vital to meet the low-latency and high-accuracy requirements of edge-enabled industrial video surveillance.

Continuous mobile vision [7], [8], [9] aims to improve the resource utilization efficiency of the vision model on a resource-constrained end device. Local resource scheduling and model compression are the most common strategies to reduce the computational complexity of the visual model to process a single video stream. With evolved edge computing, an edge-enabled architecture has become popular. Some works focus on low-latency requirements of vision tasks by designing resource scheduling algorithms [1], [10]. Meanwhile, some works concentrate on the computing cost and computational complexity of vision models [11], [12]. These works utilize the redundancy of video models such as spatio-temporal correlations [11] and finetuned deep learning model [12]. However, the above works neglect the computing capacity limitation on edge nodes, which influences the real-time performance of surveillance.

Thus, a few works consider latency requirement and computing cost to optimize the tradeoffs between visual task

Manuscript received 11 August 2022; accepted 8 September 2022. Date of publication 19 September 2022; date of current version 4 May 2023. This work was supported in part by the National Key R&D Program of China under the Grant 2018YFB1702100 and in part by the Natural Science Foundation of China under the Grant 62025305, Grant 92167205, and Grant 61933009. Paper no. TII-22-3455. *(Corresponding author: C. Chen.)*

Jingzheng Tu, Cailian Chen, Qimin Xu, and Xinping Guan are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China, with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China, and also with the Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai 200240, China (e-mail: tujingzheng@sjtu.edu.cn; cailianchen@sjtu.edu.cn; qiminxu@sjtu.edu.cn; xpguan@sjtu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TII.2022.3205938.

Digital Object Identifier 10.1109/TII.2022.3205938

Index

Introduction

EdgeLeague

EdgeLeague is an edge-collaboration scheme employed for **Industrial Surveillance** with these characteristics:

1) It considers multiple video streams with different quality of service ➡ **Multi Edge** && **Multi Camera** scenario

2) It promotes algorithms that consider both **edge resource limitations** and **bandwidth dynamics** ➡ **High surveillance performance**

3) The main features of the proposed protocols include **edge collaboration** and **camera network configuration**

Setting

Real Scenario

Multiple video streams delivered to edge devices for low-latency and high-accuracy **object detection**

Problem

If 20 cameras record with 1920 × 1080 pixels resolution at 20 fps, total bitrate is 12656 MB/s ➡ video delivery pressures

Problems

- Bandwidth dynamics ➡ communication congestion and accuracy loss
- Limited computational capabilities on edge devices ➡ latency of detection

Solution

Design an efficient edge computing scheme

Other works proposed the following improvements:

- Focus on **requirements** of vision tasks ➡ Design **resource scheduling** algorithms

- Focus on **computing cost** and computational complexity of vision models ➡ utilize **spatio-temporal correlations**

However they neglect the computing capacity limitation on edge nodes

- Focus on **latency requirement** and computing cost ➡ optimize accuracy, video resolution, latency, and energy consumption tradeoffs

However it only investigates one-edge multi-camera architectures ➡ Not applicable for realistic factories with multiple edge nodes

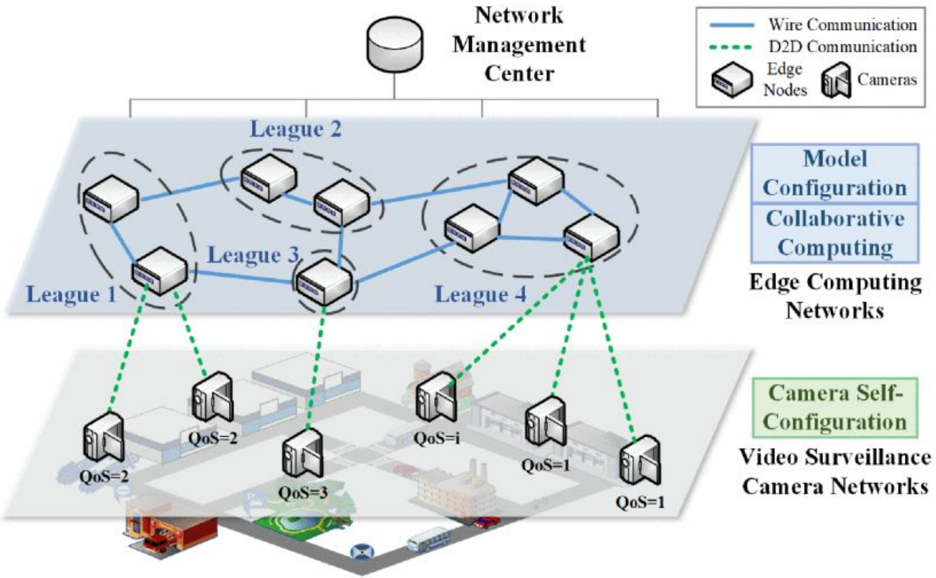
EdgeLeague Architecture

EdgeLeague Characteristics

- EdgeLeague considers a **multiedge-multicamera** scenario ➡ Edge nodes **collaborate** to process video streams
- EdgeLeague completely **runs on edge devices** ➡ No cloud offloading
- EdgeLeague can be applied to tasks other than **object detection** ➡ Widely deployable

EdgeLeague Schema

1) **Camera Node**: captures a real-time video stream and sends it to an edge node
2) **Edge Node**: they form collaborative edge leagues dynamically, the *access node* is the one edge node in each league that receives video streams
3) **Network Management Center**: is a high computation device that monitors the network



System Models

Video Surveillance

- **K cameras** or video streams $V = \{ v_1, v_2, \ldots, v_K \}$
- **QoS demands** for the K cameras $Q = \{ q_1, q_2, \ldots, q_K \}$
- transmission **bandwidth** of edge node ***i*** is b_i
- **computation capability** of edge node ***i*** is C_i

Edge Node Network

- the set of **edge nodes** is $N = \{ 1, 2, \ldots, N \}$
- on each node are deployed **M CNN models**, with $M = \{ 1, 2, \ldots, M \}$
- the set of **edge leagues** is $N^\circ = \{ N_1^\circ, N_2^\circ, \ldots, N_S^\circ \}$
- b_s° and C_s° are the **minimum bandwidth** and **computational capability** of each league s

Accuracy Model

- the input **resolution** of node ***i*** is r_i
- the **detection accuracy** on edge node ***i*** using CNN model ***j*** with input resolution r_i is denoted by the accuracy model $a_{ij}(r_i, x_{ij})$
- the **video bitrate** is $pk = \eta(\sum_{i=1}^{N} z_{ik} r_i)^2$

Problem Formulation

The objective of the problem is to minimize both the QoS **weighted latency** and maximize the **accuracy** of object detection:

Edge League Constraints

$$\sum_{s=1}^{S} |\mathcal{N}_s^{\circ}| = N$$

$$|\mathcal{N}_s^{\circ}| \neq 0, \forall \mathcal{N}_s^{\circ} \in \mathcal{N}^{\circ}$$

$$\mathcal{N}_u^{\circ} \cap \mathcal{N}_w^{\circ} = \varnothing, \forall u \neq w, u, w \in \{1, 2, \ldots, S\}$$

$$C_s^{\circ} = \sum\nolimits_{i \in \mathcal{N}_s^{\circ}} C_i$$

$$b_s^{\circ} = \min\{b_{ih} \mid h \in \mathcal{N}_s^{\circ}, i \text{ is the access node}\}.$$

Matching Constraints

$$\sum_{s=1}^{S} y_{sk} = 1, k \in \{1, 2, \ldots, K\}$$

$$\sum_{k=1}^{K} y_{sk} = 1, s \in \{1, 2, \ldots, S\}$$

Latency Constraints

$$l_{ij} = \frac{S}{b_i} + \frac{l_{ij}^{\text{CNN}}(r_i, x_{ij})}{|\mathcal{N}_s^{\circ}|} + \max_{h \in \mathcal{N}_s^{\circ}} \sum_{v \in \text{route}(i,h)} \frac{S}{b_{iv}|\mathcal{N}_s^{\circ}|}$$

$$\sum_{j=1}^{M} l_{ij} x_{ij} \leq \min_{k} L_k, i = 1, 2, 3 \ldots, N$$

Final Problem Formulation

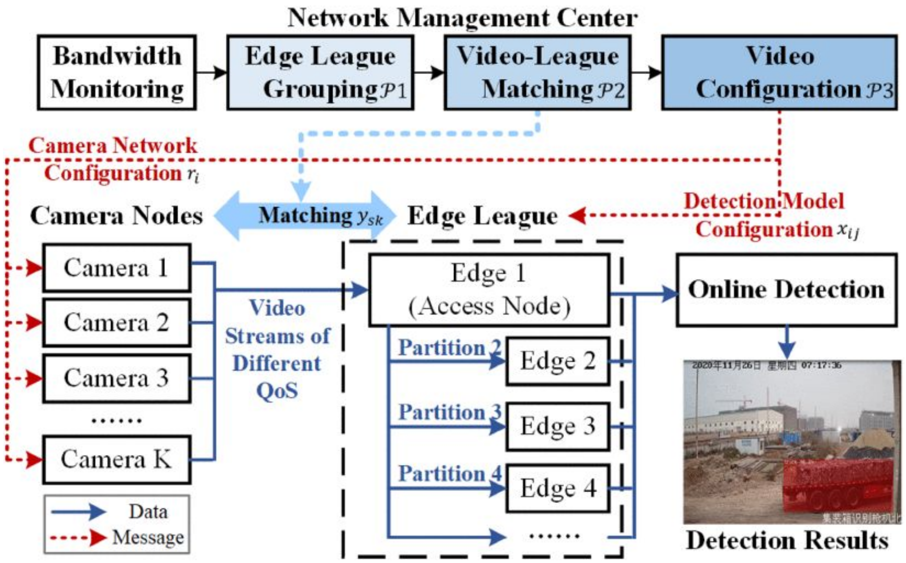
$$\mathcal{P}0: \min_{z_{ik}, x_{ij}, r_i} \sum_{k=1}^{K} \frac{1}{q_k} \sum_{i=1}^{N} z_{ik} \left(\sum_{j=1}^{M} x_{ij} l_{ij} - \omega \sum_{j=1}^{M} x_{ij} a_{ij} \right)$$

However the computational complexity of the problem is: $O(\frac{N!}{(N-K)!} K^{N-K} M^K K!)$

Problem Decomposition

A more **efficient** algorithm to solve the problem was designed following the proposed **workflow**:

1) Camera resolutions, CNN models, edge leagues and video-league matches are **randomly initialized**
2) If the bandwidth surpasses a threshold the Management Center computes the **Edge League grouping** and **video-league matching**
3) Each access node gets the video profile of each video stream connected to his league and **updates** the CNN models
4) The video profile is sent back to the connected cameras
5) Goto **2)** if there is no termination signal



Edge League Grouping (1)

This problem is transformed in a **winner determination problem**, that given a set of bids in an auction finds an allocation of items that **maximizes** the **bidder's utility**:

1) the **bidders** are the edge nodes $\mathcal{I} = \{ 1, 2, \ldots, S \}$

2) the **items** are the remaining (N - S) edge nodes J = { S + 1, . . . , N }

3) a **bubble** $\mathcal{B} \subseteq \mathcal{J}$ is a combination of items

Edge League Grouping (2)

The **brute-force algorithm** to solve the problem has $O(K^{N-K})$ computational complexity

The proposed **greedy algorithm** is $O(K(N-K))$

Algorithm 1: The Edge League Grouping Algorithm.

Require:
The set of bidders $\mathcal{I}$ and the set of items $\mathcal{J}$;
1: Initialize $\mathcal{B}_\alpha = \emptyset, \forall \alpha \in \mathcal{I}$ and $\mathcal{Q} = \mathcal{J}$;
2: **While** $\mathcal{Q} \neq \emptyset$ **Do**:
3: $\alpha^*, \beta^* = \arg\max\{u_\alpha(\mathcal{B}_\alpha \cup \{\beta\}) \mid \alpha \in \mathcal{I}, \beta \in \mathcal{Q}\}$;
4: $\mathcal{B}_\alpha = \mathcal{B}_\alpha \cup \{\beta^*\}, \mathcal{Q} = \mathcal{Q} \setminus \{\beta^*\}$;
5: **End While**
6: **return** $\{\mathcal{B}_\alpha | \alpha \in \mathcal{I}\}$;

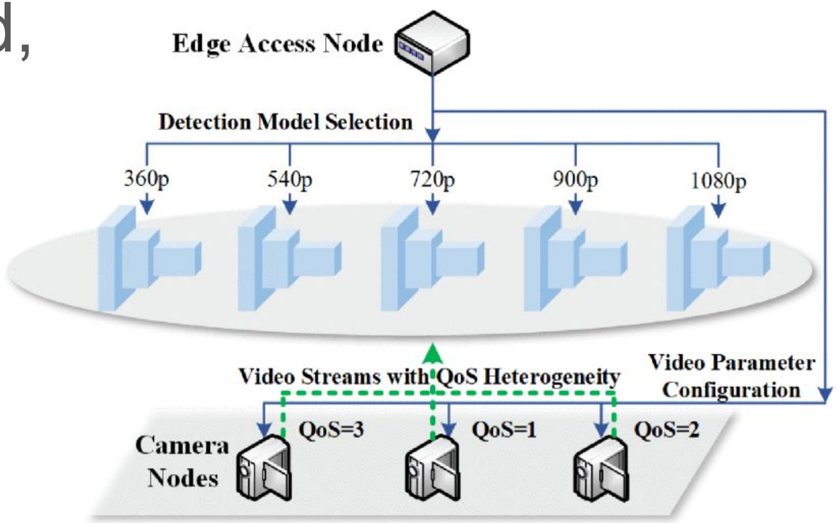
Video-League Matching

The objective of this problem is to find the **optimal matching sequence** of K Edge Leagues in N° in which the K video streams are processed with the **minimum QoS weighted latency**

The optimal matching sequence is the one in which the matching is based on the **descending order** of $\frac{1}{|\mathcal{N}_s^\circ|}\left(\frac{1}{C_s^\circ}+\frac{1}{b_s^\circ}\right)$

Video Configuration (1)

When a threshold is surpassed, the **reconfiguration** is performed and **video profiles** for each video stream and the **CNN models** are sent back to cameras and edge nodes



Video Configuration (2)

The problem can be solved by the **video configuration algorithm**, which has O(r_{max}NK) computational complexity

Thus the original complexity of $O(\frac{N!}{(N-K)!}K^{N-K}M^{K}K!)$ is reduced to the final complexity of $O(K(N-K)+r_{\max}NK)$

Algorithm 2: The Video Configuration Algorithm.

```
Require:
  QoS demand q_k and Network latency L_k of the k-th
   video;
  Uplink bandwidth b_i for edge node i;
  Latency model l_ij and accuracy model a_ij;
Initialize r_i, x_ij; u_max = 0
For j = 1 to M:
      u = sum_{k=1}^{K} (1/q_k) sum_{i=1}^{N} z_ik ( sum_{j=1}^{M} x_ij l_ij − ω sum_{j=1}^{M} x_ij a_ij );
      r_i = arg min_{x_ij, r_i} u;
   If Constraints (9), (10 b), (10 c) are satisfied:
         u_max ← u, r_i^* ← r_i, x_ij^* ← x_ij;
   End If
End For
returnResolution r_i^*; CNN model x_ij^*;
```

$$u = \sum_{k=1}^{K} \frac{1}{q_k} \sum_{i=1}^{N} z_{ik} \left(\sum_{j=1}^{M} x_{ij} l_{ij} - \omega \sum_{j=1}^{M} x_{ij} a_{ij} \right);$$

Performance Evaluation

Performance Compared to Other Methods (1)

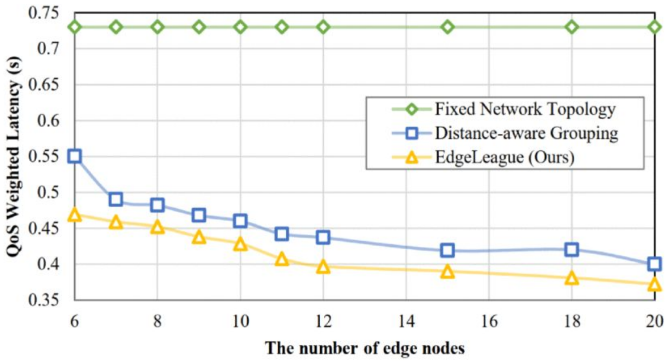
EdgeLeague **performance** is first compared with:

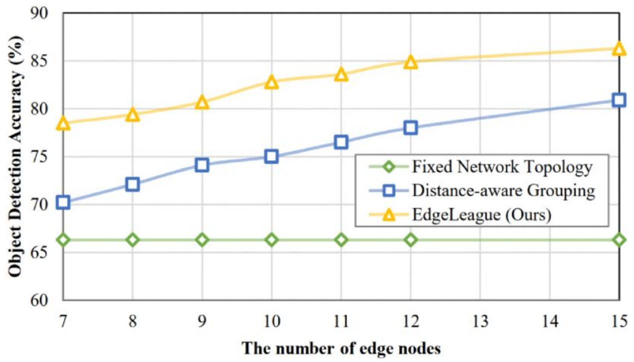
1) **Fixed Network Topology**, in which the network topology remains fixed and no cooperative edge leagues are formed

2) **Distance-Aware Grouping**, where adjacent nodes form edge leagues and video streams are sent to the nearest league

Performance Compared to Other Methods (2)

While for both the distance-aware grouping and EdgeLeague the **QoS weighted latency** decreases and **object detection accuracy** increases, for the fixed network topology they remain constant





Performance Compared to Other Methods (3)

Finally, as shown in the following table, EdgeLeague architecture achieves the highest accuracy and the lowest latency among the evaluated SOTAs

PERFORMANCE EVALUATION WITH THE STATE-OF-THE-ART

Methods	Architeture	Accuracy (%)	Latency (s)
[2]	One-camera one-edge	60.8	2.9841
DeepDecision [13]	One-camera one-edge	69.5	1.5773
JCAB [15]	Multi-camera one-edge	70.4	1.7910
[17]	One-camera Multiedge	74.2	1.2420
EdgeLeague (Ours)	Multi-camera Multiedge	75.7	0.6105

Edge League Grouping Performance (1)

The **greedy algorithm** for Edge League grouping has been compared with:

1. an heuristic algorithm based on **simulated annealing**

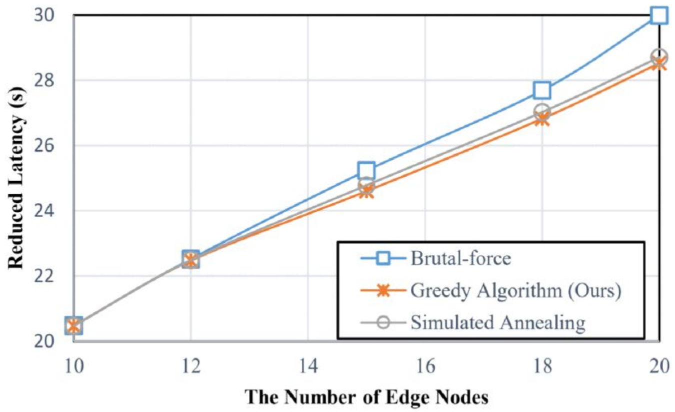
2. a **brute-force** algorithm

Edge League Grouping Performance (2)

The following table shows the comparison for the **running times**, with N as the number of edge nodes:

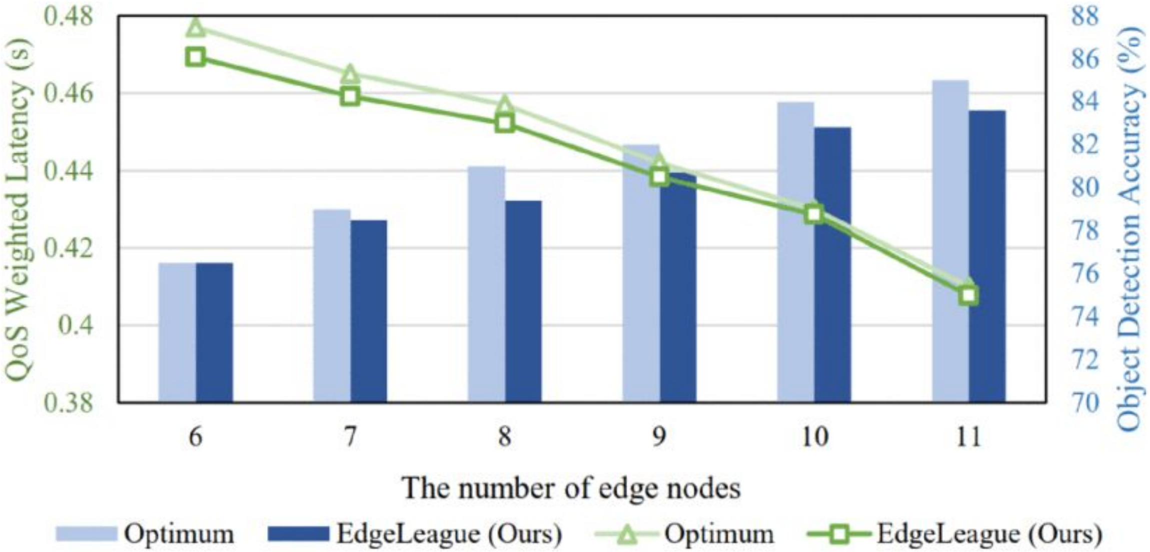
Methods	N=10	N=12	N=15	N=18	N=20
Brutal-force	0.481	1.500	7.431	15.782	29.590
Simulated annealing	0.360	0.458	0.511	0.573	0.649
Greedy algorithm (Ours)	0.343	0.376	0.389	0.396	0.444

However, since the **brute-force** algorithm always finds an **optimal solution**, it obtains a greater reduced latency compared to the **sub-optimal solutions** obtained by the **simulated annealing** and **greedy algorithms**



Performance Comparison with Optimum

This final graphs shows the performance obtained by EdgeLeague compared with the optimum:



Thanks for the Attention