Multi-Vehicle Task Offloading for Cooperative Perception in Vehicular Edge Computing

Amr M. Zaki[1], Sara A. Elsayed[1], Khalid Elgazzar[2], Hossam S. Hassanein[1]
[1]School of Computing, Queen's University, Kingston, ON, Canada
[2]Ontario Tech University, Oshawa, ON, Canada
amr.zaki@queensu.ca, selsayed@cs.queensu.ca, khalid.elgazzar@ontariotechu.ca, hossam@cs.queensu.ca

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10279594

***Abstract*—Autonomous vehicles heavily rely on sensor data to make pivotal driving and traffic management decisions. However, the reliability of such data can be profoundly impacted by many impairments, such as the adverse environmental and weather conditions, the presence of obstacles, and the vehicle's limited view of road and traffic conditions of larger areas. Collaboration between vehicles can help improve the perception of vehicles beyond their line-of-sight, and increase accurate detection of objects. Vehicular Edge Computing (VEC) has emerged as a propitious computing paradigm that can foster the realization of autonomous vehicles. However, maximizing the cooperative perception of vehicles has been mostly overlooked. In this paper, we propose the Cooperative Perception-based Task Offloading (CPTO) scheme. CPTO enables task offloading in VEC with the goal of maximizing the cooperative perception of vehicles and minimizing the latency of perception aggregation, while abiding by a certain deadline. Towards that end, we formulate the task offloading problem as a multi-objective 0-1 integer linear program (0-1 ILP). We also propose a greedy heuristic, called the CPTO-Heuristic (CPTO-H) scheme, to solve the optimization problem. Extensive simulations show that CPTO significantly outperforms the baseline task offloading scheme in terms of perception intensity, service capacity, and satisfaction ratio. Furthermore, CPTO-H closely approaches the optimal solution, with a small gap of up to 3.7% and 2.4% in terms of perception intensity and satisfaction ratio, respectively.**

***Index Terms*—Autonomous Vehicles, Vehicular Edge Computing, Cooperative perception, Task Offloading.**

I. Introduction

With the advent of Autonomous Vehicles (AVs), the global market of self-driving cars is expected to reach 400 billion by 2025 [1], and the number of AVs is expected to reach 62.4 million by 2030 [2]. In AVs, crucial driving and traffic management decisions are delegated to onboard computing units [3]. Fostering the self-driving process in AVs necessitates the reliance on multiple onboard sensors, including LiDAR, radar, and GPS, to sense the surrounding physical and atmospheric conditions and send the collected data to the onboard computing unit [3]. However, the reliability of such data can be hindered by several factors, such as severe weather, the vehicle's limited view of its surroundings, and the presence of obstacles [4]. Reliability issues can also be attributed to the inherent limitations of the on-board sensors, resulting from the trade-off between range and resolution [5]. This can profoundly impact the vehicle's traffic perception, thus diminishing road safety and traffic efficiency [5].

The vehicle's traffic perception can be improved by aggregating the sensed data from other vehicles and enabling cooperative perception [6]. This can be achieved by sharing sensor data among vehicles and exploiting extraneous data from other vehicles to ameliorate the detection capabilities of a single vehicle and enhance traffic situational awareness [3]. To transmit such extraneous data, researchers have focused on the cloud computing paradigm [7]. However, due to the massive number of onboard sensors, an AV is expected to trigger approximately 4 terabytes of data every two hours [6]. The transmission of such an excessive amount of data to remote data centers can lead to prolonged latency and excessive traffic influx at backhaul links, rendering real-time object detection via cloud computing impractical [7].

Vehicular Edge Computing (VEC) is considered a promising paradigm that can overcome the limitations mentioned above [8] [9]. This is because VEC moves the computing service within the proximity of the end-user, which can drastically curtail the delay [3], and significantly reduce the number of ineffective transmissions [9]. Edge nodes can either be static or mobile according to the availability of resources [10]. In VEC, vehicles can be used as edge nodes by exploiting their increasingly powerful onboard computing units [11]. Offloading cooperative perception tasks to vehicles to improve situational awareness has recently been proposed in the literature [7], [8], [12], [13].

Existing task offloading schemes that enable cooperative perception in VEC focus on optimizing specific metrics, such as latency and energy consumption [7], [10], [12], [13]. However, maximizing the level of cooperation between vehicles has been mostly overlooked. Intuitively, increasing the level of cooperation between vehicles can improve situational awareness [14], [15]. However, as the number of vehicles collaborating with the same vehicle (i.e., worker/edge node) increases, this can impact the worker's ability to reduce the delay of the perception aggregation process.

In this paper, we propose the Cooperative Perception-based Task Offloading (CPTO) scheme. CPTO aims to maximize the cooperative perception of vehicles and minimize the latency of the perception aggregation process while abiding by a deadline requirement. CPTO formulates the task offloading process as a multi-objective 0-1 Integer Linear Program (0-1 ILP). To the best of our knowledge, CPTO is the first task offloading scheme in VEC that maximizes the cooperative perception between vehicles while reducing aggregation latency. Considering that the formulated problem in CPTO is NP-hard, we also propose a greedy heuristic, called the CPTO-Heuristic (CPTO-H) scheme, to approximate the optimal solution and enable its practical use.

We evaluate the performance of CPTO and CPTO-H compared to a representative of the state-of-the-art schemes that focus on latency while overlooking the cooperative perception of vehicles. Simulation results show that CPTO and CPTO-H significantly outperform the baseline scheme regarding of perception intensity, service capacity, and satisfaction ratio. The performance of CPTO-H is also compared to the optimal solution rendered by CPTO. Simulation results show that CPTO-H closely approaches CPTO, with a performance gap of up to 3.7% and 2.3% regarding perception intensity and satisfaction ratio. This is while rendering a significant improvement, of up to 97.5%, regarding the time taken to solve the optimization problem compared to CPTO.

The remainder of the paper is organized as follows. Section II highlights some of the related work. Section III provides a detailed description of CPTO and CPTO-H. Section IV discusses the performance evaluation and simulation results. Finally, section V presents our conclusions and future directions.

II. Related Work

Extending situational awareness through cooperative perception by offloading computational tasks to moving vehicles that act as mobile edge nodes has recently been studied in several works [7] [12] [13] [10]. In [10], the authors study task allocation of video applications to vehicles or Road Side Units (RSUs) for object detection. They propose a multi-objective scheme that focuses on minimizing latency and quality loss. In [12], authors propose a task offloading scheme that improves the cooperative perception by maximizing the data transmission rate of vehicles to maximize the number of tasks that can run on vehicles or RSUs. In [13], a task offloading scheme is proposed that considers the task's deadline requirement, and the energy restrictions of vehicles when acting as edge nodes. In [7], the authors propose a cooperative group formation scheme for task offloading in VEC. They exploit the use of vehicles for distributed learning, where the quality of learning increases as the size of the formed group increases. The scheme focuses on forming the largest group possible while ensuring that the formed group is capable of completing the task. However, this scheme cannot be easily adopted for cooperative perception tasks. The system is designed for federated learning, which requires local models to be built in each car. More recent works [16] [17] have developed a reinforcement learning (RL) scheme using Deep Q-network (DQN) to determine when to offload, discard, and opt for local processing for cooperative perception to stop the system from using unnecessary offloading.

One aspect which has been overlooked in the literature is optimizing the amount of collaboration between vehicles while ensuring the stringent requirements of the real-time perception aggregation process. In this paper, we improve cooperative perception and situational awareness by proposing a multi-objective task offloading scheme that maximizes the number of collaborating vehicles while minimizing the latency of the perception aggregation process.

III. Cooperative Perception-based Task Offloading (CPTO)

In this section, we provide a detailed description of the the system model, the ILP optimization problem, the problem analysis, and the heuristic scheme CPTO-H.

A. System Model and Overview

Consider a set of vehicles U={u_1, u_2,..., u_n} moving in a certain area. Vehicles can act as users or workers (i.e., edge nodes). Users offload their perception tasks to workers, which can be moving vehicles or static RSUs. Let W ={w_1, w_2, ...w_m} be the set of workers that are willing to offer their computational resources to execute the offloaded tasks in exchange for some incentives.

Each user $u_i \in U$ offloads a perception task, which has a computation workload or intensity q_i (in CPU cycles/bit) and involves a certain bits of data λ_i, constituting the size of the perception frame. Each worker $w_j \in W$ has a maximum CPU clock speed or CPU frequency, denoted C_j (in CPU cycles/sec). The CPU frequency of worker w_j is divided equally among all the perception tasks offloaded to it. The number of offloaded tasks to worker w_j is the number of vehicles currently using this worker (i.e., cooperating together), and is denoted s_j. The distance between user u_i and worker w_j is denoted d_{ij}, and the propagation speed is denoted v. The data rate of the transmission link is denoted R_{ij}.

We assume that the perception frames are sent from users to workers at a fixed rate of one perception frame per second per user. Consider a certain perception range, denoted ξ. If a user moves outside this range, it must collaborate with a closer worker. The perception range is set in accordance with the European Telecommunications Standards Institute (ETSI) solution for short and medium distances, which are critical for the safety of connected vehicles [18], [19].

CPTO runs periodically every 1 sec, in which it considers offloading the perception tasks of a subset of users who have exceeded their perception range. The task offloading problem is formulated as a multi-objective optimization problem that aims to jointly maximize the cooperative perception of vehicles and minimize the latency of perception aggregation. The total latency of running the perception aggregation task of user u_i on worker w_j is denoted t_{ij}, and is given by Eq. 1, where α_{ij} is the computation latency, β_{ij} is the propagation latency, and γ_{ij} is the transmission latency.

$$t_{ij} = \alpha_{ij} + \beta_{ij} + \gamma_{ij} \tag{1}$$

As given by Eq. 2, the computation latency α_{ij} is the time it takes worker w_j to execute the perception aggregation task of user u_i. Note that the computation latency is affected by the number s_j of users currently using the worker. As the number of users increases, the worker's computational capability decreases, and thus the computation latency increases.

$$\alpha_{ij} = \frac{q_i \lambda_i}{C_j / s_j} \tag{2}$$

The propagation latency β_{ij} , given by Eq.3, is the time it takes the perception task to propagate from user u_i to worker w_j.

$$\beta_{ij} = d_{ij}/v \tag{3}$$

The transmission latency γ_{ij}, given by Eq. 4, is the time it takes to push the entire perception frame on the transmission link between user u_i to worker w_j.

$$\gamma_{ij} = \lambda_i / R_{ij} \tag{4}$$

CPTO minimizes the latency t_{ij} by maximizing the latency difference g_{ij}, which acts as the utility gain from changing the worker used for offloading, from the current worker w_a to a different one w_b, as given by Eq. 5.

$$g_{ij} = t_{ia} - t_{ib} \tag{5}$$

B. Problem Formulation

CPTO is formulated as a multi-objective 0-1 integer linear program (0-1 ILP) with quadratic constraints, where the decision variable x_{ij} is set to 1 if user u_i offloads the task to worker w_j, and 0 otherwise. The objectives are maximizing the latency difference g_{ij} and maximizing the cooperation of vehicles. CPTO maximizes the cooperative perception of vehicles by maximizing the summation of the placement decision variable x_{ij}. The optimization problem is given by Eq. 6.

$$\max_{x} \quad \sum_{i=0}^{n}\sum_{j=0}^{m} x_{ij} g_{ij} \tag{6a}$$

$$\max_{x} \quad \sum_{i=0}^{n}\sum_{j=0}^{m} x_{ij} \tag{6b}$$

$$\text{s.t.} \quad x_{ij}\left((\sum_{i=0}^{n} x_{ij}) + \kappa_j \right) \geq x_{ij}\mathbb{C}_i \qquad \forall j \in W \;\; \forall i \in U \tag{6c}$$

$$\sum_{i=0}^{n} x_{ij} d_{ij} \leq \xi_j \qquad \forall j \in W \tag{6d}$$

$$\sum_{j=0}^{M} x_{ij} \leq 1 \qquad \forall i \in U \tag{6e}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in U \;\; \forall j \in W \tag{6f}$$

The first optimization objective (6a) aims to maximize the latency difference g_{ij}. The second objective (6b) aims to maximize the cooperative perception by maximizing the number of vehicles cooperating at each worker.

Constraint (6c) controls the number of collaborating vehicles. This is because in case of a low congestion (multiple available workers), the delay objective over-dominates and CPTO tends to offload to workers with low number of collaborating vehicles to decrease the overall average latency. However, this could negatively affect the cooperation between vehicles, since the chosen workers would have a small number of cooperating vehicles. To help mitigate this case, constraint

Algorithm 1 Multi-Objective Heuristic (CPTO-Heuristic)

Input: Users (U), Workers (W), $percetion_{intensity}$,Delay
Output: SelectedWorkers

1: **for** $u \in Users$ **do**
2: $nearbyWorkersDelayPerception$ = [ ]
3: $SortedList$ = [ ]
4: $SelectedWorkers$ = [ ]
5: **for** $w \in Workers$ **do**
6: **if** $dist(u, w) <= perception_{range}$ & $delay! = 0$ **then**
7: $delay \leftarrow round(delay)$
8: $nearbyWorkers \leftarrow (w, delay, percetion_{intensity})$
9: **end if**
10: **end for**
11: $SortedList \leftarrow sort_{Delay}(nearbyWorkers)$
12: $SortedList \leftarrow sort_{Perception}(nearbyWorkers)$
13: $SelectedWorkers \leftarrow SortedList[0]$
14: $assign(u, SortedList[0])$
15: **end for**
16: **return** $SelectedWorkers$

(6c) is added to act as a lower bound on the cooperation between vehicles, where $\mathbb{C}_i$ denotes the lower bound of cooperation of each vehicle. Since the optimization problem only considers a subset of users who have exceeded their perception range ξ_j, κ_j is added to consider users who occupy the worker but are not involved in the optimization problem (i.e, users who haven't exceeded their perception range ξ_j).

Constraint (6d) ensures that the selected worker is located within the required perception range. Constraint (6e) ensures that the user is served by at most one worker. Constraint (6f) ensures that each element x_{ij} in the binary placement matrix X is set to either 0 or 1.

C. Problem Relaxation

The aforementioned problem can be considered as a 0/1 multi-objective multi-dimension Knapsack Problem (KP), which has been proven to be NP-hard [20].

The optimization problem in Eq. 6 contains a quadratic constraint, due to the multiplication of optimization variables in constraint (6c). To convert the problem to one with only linear constraints, constraint (6c) is relaxed to (7a). The term $((1 - x_{ij}) \times 2\mathbb{C}_i)$ is added to ensure that the cooperation constraint only works when a suitable placement is found by the optimization problem (i.e., $x_{ij} = 1$).

$$\left((\sum_{i=0}^{n} x_{ij}) + \kappa_j \right) + ((1 - x_{ij}) \times 2\mathbb{C}_i) \geq \mathbb{C}_i \;\; \forall j \in W \;\; \forall i \in U \tag{7a}$$

D. CPTO-H

The time complexity of solving the proposed NP problem grows exponentially with the scale of the problem. Therefore, solving using exact dynamic programming and branch-and-bound methods can only stay in the theoretical stage [20]. Thus, to solve the proposed optimization problem, CPTO-H is proposed, which is detailed in Algorithm 1.

CPTO-H first loops on the users who have exceeded their perception range (line 1). Then, for each user $u_i \in U$, a search is done to ensure that the correct placement is done on an eligible worker. In particular, this search is done to ensure that the selected worker is within the required perception range (line 6). Current workers are not included in the search, since they are considered to have exceeded the user's perception range. Thus, workers with a delay difference of zero are not included in the search procedure (line 6).

The delay difference is approximated to the nearest tenth (line 7), to simplify the sorting steps later in the procedure. Next, nearby eligible workers are added to a list with their respective delay differences and perception intensities (line 8). After the list is built, CPTO-H applies multiple sorting steps. First, the list is sorted in descending order according to the delay difference (line 11), then it is sorted according to perception intensities (line 12). The worker with the highest perception intensity and delay difference is found in the first index of the list. This worker is then added to a list of selected workers (line 13). Finally, user u_i is offloaded to the selected worker (line 14). This last step is essential since it ensures that workers' perception intensity is updated after each user's placement.

IV. Performance Evaluation

In this section, we evaluate the performance of CPTO and CPTO-H. To study the effect of optimizing the co-operative perception and aggregation latency, we compare CPTO and CPTO-H to a representative of the state-of-the-art optimization-based task allocation schemes that focus on aggregation latency and quality loss while overlooking cooperative perception [10]. To demonstrate the effect of considering cooperative perception in CPTO and CPTO-H, we slightly modify the scheme in [10] to optimize the delay rather than the delay and quality loss. We refer to the modified scheme as a Single Objective-CPTO (SO-CPTO).

We use the following performance metrics: 1) the average latency experienced starting from the time a perception frame request is sent until a response is received, 2) the average perception intensity, which is calculated as the average number of collaborating vehicles associated with workers, 3) the service capacity, which is the ratio of the number of successfully offloaded tasks to the total number of tasks, 4) the satisfaction ratio, which is the percentage of requests that have satisfied the deadline requirement, and 5) the average offloading decision latency, which is the average time taken by the different schemes to reach the offloading decision.

A. Simulation Setup

We have implemented CPTO, CPTO-H, and SO-CPTO using Python and integrated the simulator with IBM CPLEX optimization solver [21] to solve the optimization problems. Realistic mobility traces have been employed by using Luxembourg SUMO traffic dataset (LuST) [22]. This dataset includes the mobility patterns of buses in Luxembourg city, with an average speed of 22.3 kmph in routes of 26.44 min on average. The number of vehicles (users) is set to 90. We set the number of static workers to 30, while varying the number of moving workers from 90 to 210 to test the effect of the density of workers on the system.

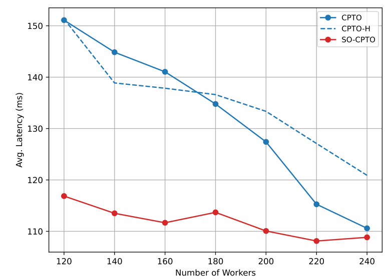

Fig. 1: Average latency of CPTO, CPTO-H, and SO-CPTO over a varying number of workers

The computation frequency of workers ranges from 22 to 23 GHz. The uplink data rate of the vehicles that act as users ranges from 23 to 25 Mbps. DSRC, which supports data rates of 4.5 to 27 Mbps [23], is the communication technology adopted. Following with the range specified by ETSI for short and medium distances [19], the perception range is set to 200 m. The perception lower bound $\mathbb{C}_i$ ranges from 1 to 2. The perception task studied throughout the simulations is similar to the one used in [12], which is the computation of image data. This computing task can include feature detection and perspective transformation, essential tasks for traffic perception. The computation intensity is set to $1e^9$ cycles/sec, and the perception frame size is 20 KB. The delay deadline is set to 0.6 sec. The simulation period is set to 5 minutes, and the optimization problem is solved periodically each 1 second. The number of users involved in the optimization problem for offloading is variable for each run of the decision-making schemes. Only the users who have exceeded the perception range are considered in the offloading schemes.

B. Simulation Results and Analysis

In our experiments, we assess the performance of CPTO, CPTO-H, and SO-CPTO over a varying number of moving workers. Simulation results are presented at a confidence level of 95%.

Figure 1 depicts the performance of the different schemes regarding the average latency. As the number of moving workers increases, the average delay decreases in all schemes because the system becomes less congested, as the number of workers available for each user increases. SO-CPTO yields the lowest average latency compared to CPTO (-14.6%) and CPTO-H (-17%) across the different number of moving workers. This is because SO-CPTO focuses only on optimizing the average latency, whereas both CPTO and CPTO-H jointly optimize the latency and the cooperative perception of vehicles. As depicted in Figure 1, CPTO-H closely approaches the optimal solution CPTO, with a performance gap of up to 4.6%.

We conduct the same comparison regarding the average perception intensity. As depicted in Figure 2, as the number of workers increases, the perception quality decreases (i.e., the number of vehicles offloading their cooperative perception

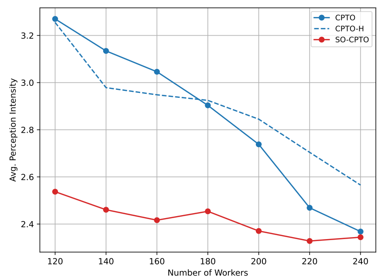

Fig. 2: Average perception intensity of CPTO, CPTO-H, and SO-CPTO over a varying number of workers

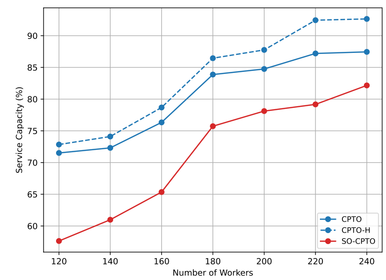

Fig. 3: Service capacity of CPTO, CPTO-H, and SO-CPTO over a varying number of workers

to a worker decreases). This is because the availability of placement options increases as the number of moving workers increases. Thus, the schemes choose the workers associated with the low number of users, so as to optimize the average delay. Consequently, the perception intensity decreases. CPTO and CPTO-H, account for both minimizing the average latency and maximizing the cooperation, render higher perception intensity across different numbers of workers compared to SO-CPTO, which only aims to optimize the average latency. CPTO shows an increase of 14.3%, while CPTO-H shows an increase of 16.1% when respectively compared to SO-CPTO. CPTO-H closely approaches the perception intensity of CPTO, with a performance gap of 4.4%. Figure 1 and Figure 2 adopt the same pattern as direct result of the perception intensity's effect on latency. The computation latency is the most significant factor in affecting the average latency of the system, where the computation latency decreases proportionally as the perception intensity decreases. On the other hand, both the propagation and the transmission delays have a minor effect on the system. This can be attributed to the fact that small distances between workers and users, result in the propagation delay having a minimal effect on the system. Additionally, the almost similar uplink data rate of users results in the transmission delay having a minor effect on the average latency of the system. This ultimately leaves the computation delay, affected by the number of vehicles cooperating at a worker, to have the biggest effect.

We assess the different schemes regarding the service capacity. As depicted by Figure 3, as the number of workers increases, the service capacity increases in all schemes. This can be attributed to the fact that as the number of workers increases congestion decreases, since the number of eligible workers available to a specific user increases, which increases the number of perception tasks that can be served. Since SO-CPTO focuses on optimizing the average latency only, it tends to choose workers that minimize the latency, without considering the need to assign all requests. Therefore, SO-CPTO sacrifices placing all the vehicles' requests to optimize the average latency. Accordingly, the service capacity of SO-CPTO is always lower than that of CPTO and CPTO-H. In contrast, CPTO and CPTO-H strive to tend to all vehicles' requests to maximize the cooperative perception. Thus, CPTO

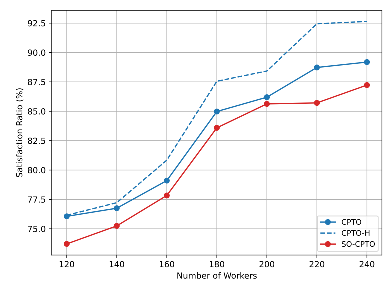

Fig. 4: Satisfaction ratio of CPTO, CPTO-H, and SO-CPTO over a varying number of workers

and CPTO-H render a higher service capacity of 11.7% and 15% increase when respectively compared to SO-CPTO. CPTO-H yields a higher service capacity than CPTO, with an increase of up to 3.7%. This is due to the fact that the perception constraint (6c) prevents CPTO from offloading to workers that have a perception intensity lower than the required lower bound of cooperation $\mathbb{C}_i$.

Figure 4 shows the satisfaction ratio of CPTO, CPTO-H, and SO-CPTO over varying number of workers. As shown in Figure 4, as the number of workers increases, the satisfaction ratio in all schemes increases. This is due to the resulting decrease in congestion, which in turn decreases the number of users sharing the same worker. This increases the available CPU speed dedicated to offloaded tasks, which decreases the computation delay, thus increasing the percentage of requests satisfied within the specified deadline. Note that SO-CPTO yields the lowest satisfaction ratio among all schemes. This is since SO-CPTO renders the lowest service capacity, which directly impacts the satisfaction ratio. In contrast, CPTO and CPTO-H optimize both the latency and cooperative perception, thus yielding a higher service capacity than SO-CPTO, which in turn leads to a higher satisfaction ratio. CPTO and CPTO-H shows an increase of 2% and 4.3% when respectively compared with SO-CPTO. CPTO-H achieved a higher satisfaction ratio of an 2.4% increase compared to CPTO due to the fact that it didn't consider the perception lower bound constraint (6c).

Figure 5 depicts the performance of CPTO and CPTO-H regarding the average offloading decision latency over a varying number of workers. As the number of workers

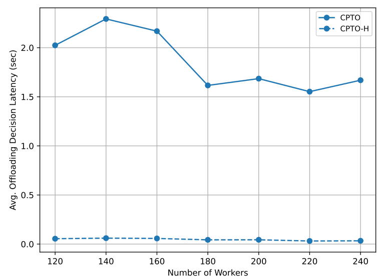

Fig. 5: Average offloading decision latency of CPTO and CPTO-H over varying number of workers

increases, a fewer number of users becomes associated with each worker, which reduces congestion, and thus the offloading task becomes easier. Thus, the average decision time for the offloading task decreases. The heuristic offloading scheme CPTO-H yields a significant reduction, of up to 97.5%, in the time taken to solve the offloading problem compared to the conventional optimization technique adopted by CPTO.

V. Conclusion and Future Work

In this paper, we have proposed the Cooperative Perception-based Task Offloading (CPTO) scheme. CPTO strives to foster reliable autonomous vehicles by maximizing the level of cooperative perception between vehicles. CPTO exploits task offloading in Vehicular Edge Computing (VEC) to jointly maximize cooperative perception and minimize perception aggregation latency. Towards that end, CPTO formulates the task offloading problem as a multi-objective 0-1 integer linear program (0-1 ILP). A greedy heuristic scheme, CPTO-H, has also been proposed to enable a practical approximation of the optimal solution rendered by CPTO. Extensive evaluations show that CPTO-H closely approaches CPTO, with a performance gap of up to 4.6%, 4.4%, 3.7%, and 2.4% regarding aggregation latency, perception intensity, service capacity, and satisfaction ratio, respectively. This is while sustaining a significant improvement regarding offloading decision-making latency, reaching up to 97.5%. In addition, simulations have shown that CPTO and CPTO-H outperform a baseline scheme that only focuses on latency, regarding perception intensity, service capacity, and satisfaction ratio. In the future, we plan further improvements in perception intensity by using predictive techniques that take environmental conditions into consideration.

Acknowledgment

This research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number ALLRP 549919-20.

References

[1] Statista, "Projected autonomous vehicle market size worldwide between 2021 and 2025," https://www.statista.com/statistics/1224515/av-market-size-worldwide-forecast/ , 2022.

[2] Business-Wire, "Self-driving cars market global forecast to 2030: Increasing demand for level 1 and level 2 cars and rising investment in development of level 4 and level 5 cars will propel market," https://www.businesswire.com/news/home/20220215005655/en , 2022.

[3] Tang, Gu *et al.*, "Vehicular edge computing for multi-vehicle perception," in *International Conference on Connected and Autonomous Driving (MetroCAD)*, 2021, pp. 9–16.

[4] Thandavarayan, Sepulcre, and Gozalvez, "Cooperative perception for connected and automated vehicles: Evaluation and impact of congestion control," *IEEE Access*, vol. 8, pp. 197 665–197 683, 2020.

[5] D. D. Yoon and Ali, "Performance of decentralized cooperative perception in v2v connected traffic," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[6] Business-Wire, "Intel editorial: For self-driving cars, there's big meaning behind one big number: 4 terabytes," https://www.businesswire.com/news/home/20170414005225/en/Intel-Editorial-For-Self-Driving-Cars-There.

[7] Liu and Blough, "Cooperative task-oriented group formation for vehicular networks," in *IEEE Consumer Communications Networking Conference (CCNC)*, 2022, pp. 584–592.

[8] Tang, Chen *et al.*, "Vecframe: A vehicular edge computing framework for connected autonomous vehicles," in *IEEE International Conference on Edge Computing (EDGE)*, 2021, pp. 68–77.

[9] Yu, Yang, and Zhang, "Edge-assisted collaborative perception in autonomous driving: A reflection on communication design," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, 2021, pp. 371–375.

[10] Zhu, Pastor *et al.*, "Fog following me: Latency and quality balanced task allocation in vehicular fog computing," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.

[11] Corporation, "Nvidia drive agx developer kit. [online]. available:," https://developer.nvidia.com/drive/drive-agx, 2020.

[12] Krijestorac, Memedi *et al.*, "Hybrid vehicular and cloud distributed computing: A case for cooperative perception," in *IEEE Global Communications Conference*, 2020, pp. 1–6.

[13] Kazmi, Otoum *et al.*, "A novel deep reinforcement learning-based approach for task-offloading in vehicular networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[14] Monteil, Billot *et al.*, "Cooperative highway traffic: Multiagent modeling and robustness assessment of local perturbations." *Transportation Research Record*, pp. 1–10, 01 2013.

[15] Guériau, Billot *et al.*, "How to assess the benefits of connected vehicles? a simulation framework for the design of cooperative traffic management strategies," *Transportation Research Part C Emerging Technologies*, vol. 67, 04 2016.

[16] Higuchi, Ucar *et al.*, "Vehicular edge offloading based on anticipated value of computational tasks," in *IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 1–5.

[17] Aoki, Higuchi, and Altintas, "Cooperative perception with deep reinforcement learning for connected vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 328–334.

[18] Thandavarayan, Sepulcre, and Gozalvez, "Redundancy mitigation in cooperative perception for connected and automated vehicles," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[19] ETSI-ITS, "Intelligent transport system (its); vehicular communications; basic set of applications; analysis of the collective - perception service (cps)," June 2019.

[20] He, Sheng *et al.*, "Multi-objective deep reinforcement learning based time-frequency resource allocation for multi-beam satellite communications," *China Communications*, vol. 19, no. 1, pp. 77–91, 2022.

[21] Cplex, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[22] Codecá, Frank *et al.*, "Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017.

[23] Ali, Ayalew *et al.*, "Feedbackless relaying for enhancing reliability of connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4621–4634, 2020.