Mao, Wencan; Akgul, Ozgur Umut; Cho, Byungjin; Xiao, Yu; Yla-Jaaski, Antti

On-demand Vehicular Fog Computing for Beyond 5G Networks

Published in:
IEEE Transactions on Vehicular Technology

DOI:
10.1109/TVT.2023.3289862

Published: 01/12/2023

Document Version
Publisher's PDF, also known as Version of record

On-Demand Vehicular Fog Computing for Beyond 5G Networks

Wencan Mao, Ozgur Umut Akgul, Byungjin Cho, Yu Xiao, *Member, IEEE*, and Antti Ylä-Jääski

***Abstract*—Emerging compute-intensive and latency-sensitive vehicular applications are expected to be deployed at the edge instead of the cloud to shorten the network latency. Mobile fog nodes carried by moving vehicles, namely vehicular fog nodes (VFNs), have been proposed to complement the stationary fog nodes co-located with base stations to handle the spatio-temporal variations of demand in a cost-efficient way. Existing works on capacity planning for such vehicular fog computing (VFC) scenarios are built on the assumption of certain spatio-temporal patterns of vehicular traffic. They consider long-term capacity planning (e.g., updated every season) but leave the adaptation to temporary changes or unexpected variations out of scope. These solutions typically result in high computational costs and thus are not suitable for short-term capacity planning, which requires low-latency responses. To reduce time complexity, we propose an integer linear programming (ILP)-based framework called on-demand capacity planning (ODCP) to implement two-phase planning through optimizing the routing strategies of VFNs, with the aim of maximizing the profit and quality of service (QoS). More specifically, ODCP first predicts the traffic flow and resource demand using seasonal auto-regressive integrated moving average (SARIMA) and estimates the revenue using an economic model defined by service level agreement (SLA). With the estimated workload and revenue, the first phase (i.e., global planning) decides the ratio of tasks that can be served at the city scale and assigns VFNs to each region. The second phase (i.e., regional planning) assigns the VFNs to users within the same region and schedules the routes of VFNs based on the mobility of users. Experimental results show that the proposed solution achieves a higher performance in terms of profit and QoS than the existing single-phase capacity planning solutions. We also find that a large number of VFNs, a small region size, high penalty costs, and low travel and rental costs lead to high service rates, whereas a large region size and low travel, rental, and penalty costs lead to high profits.**

***Index Terms*—Capacity planning, vehicular fog computing (VFC), quality of service (QoS), integer linear programming (ILP), and techno-economic analysis.**

Manuscript received 19 May 2022; revised 10 September 2022 and 31 January 2023; accepted 21 June 2023. Date of publication 27 June 2023; date of current version 19 December 2023. This work was supported by the Academy of Finland under Grants 317432 and 318937. The review of this article was coordinated by Dr. Tomaso De Cola. *(Corresponding author: Byungjin Cho.)*

Wencan Mao, Ozgur Umut Akgul, Byungjin Cho, and Yu Xiao are with the Department of Information and Communications Engineering, Aalto University, 02150 Espoo, Finland (e-mail: wencan.mao@aalto.fi; ozgur.akgul@aalto.fi; byungjin.cho@aalto.fi; yu.xiao@aalto.fi).

Antti Ylä-Jääski is with the Department of Computer Science, Aalto University, 02150 Espoo, Finland (e-mail: antti.yla-jaaski@aalto.fi).

Digital Object Identifier 10.1109/TVT.2023.3289862

I. Introduction

THE emerging vehicular applications in beyond 5 G networks range from advanced driving (e.g., sensory applications and infotainment) to cooperative driving (e.g., collaborative intersection crossing and see-through passing). These applications involve intensive computing, which is expected to be offloaded to computing nodes at the edge instead of the cloud, due to the strict latency constraints [1]. The computing entities, which are called edge/fog nodes, can be deployed on network infrastructures (e.g., cellular base stations) and/or vehicles (e.g., buses and taxis). We denote the former as cellular fog nodes (CFNs) and the latter as vehicular fog nodes (VFNs). The concept of VFC [2] is motivated by the spatio-temporal variation in vehicular traffic, where the VFNs are envisioned to complement CFNs by utilizing their mobility to fulfill the temporary demand in a cost-efficient manner [2].

Capacity planning for VFC determines where to deploy the fog nodes and how much capacity should be deployed. Our previous work [3] proposed using commercial fleets (e.g., buses) to serve as VFNs and jointly decide the long-term capacity plans of VFNs and CFNs based on spatio-temporal traffic patterns learned from real-world traffic. Such a long-term capacity planning solution was not designed to handle the temporary changes in traffic patterns, due to occasional events, such as football matches and traffic accidents. Therefore, it was not suitable to be executed frequently (e.g., every hour or several minutes) due to the time complexity. To address this issue, this article focuses on a scenario where VFN trips are regularly scheduled to meet temporary needs. We call this scenario on-demand VFC (ODVFC).

Such an ODVFC scenario can be envisioned where we route the VFNs to places where new demand emerges. Unlike commercial fleets, the VFNs in ODVFC are installed on taxis or other vehicles with flexible schedules and adaptable routes. In this article, we target capacity planning in ODVFC, which is a challenging problem. Due to the Bellman called curse of dimensionality, the computational time of the capacity planning model increases with the number of tasks and VFNs [3]. Unlike long-term planning which allows sufficient time (i.e., days or weeks) to find the optimal solution, ODVFC is considered as short-term capacity planning that requires the solution to be found in minutes or even seconds.

To address the computation time limit, we propose an integer linear programming (ILP) based framework referred to as on-demand capacity planning (ODCP) to create the capacity

plan in two steps. During global planning, we determine the global routing strategy for VFNs. In this way, the service rate is maximized at the city scale with the overall capacity constraint. Meanwhile, the VFNs are routed to the corresponding regions with minimized traveling costs. During regional planning, we decide the task allocation and regional routing strategies for VFNs. More specifically, the VFNs are routed according to the locations of the users within the same region with minimized costs. Meanwhile, the VFNs are assigned to the users within the communication range with minimized service level agreement (SLA) violations. Such two-phase capacity planning enables parallel decision-making at the regional level, which greatly shortens the computational time.

The aim of ODCP is to maximize the profit of the service provider and the quality of service (QoS) received by the users. To anticipate the temporary changes in demand, the framework uses seasonal auto-regressive integrated moving average (SARIMA) to predict the upcoming traffic flow and estimate the computing workload frequently (e.g., hourly). We estimate the profit by using an economic model defined by the SLA. The economic model includes various parameters including the price for different service types as well as traveling, rental, capacity, and penalty costs. Then we formulate and solve the two-phase planning problem using ILP, with the constraint of estimated workload and the objective of maximizing profit. We simulate the ODVFC scenario with the real-world road network, cellular map, and application profiles as inputs. We evaluate the techno-economic performance of ODCP under different traffic scenarios where gradual or sudden changes in demand occur and compare it with other capacity planning solutions presented in the literature. We also assess the techno-economic implications of the impacting factors, including the number of VFNs, economic model, and region size.

The contributions of this work are listed as follows:

1) We propose the ODVFC scenario, where the VFNs are routed to the places where computing demand emerges. Unlike previous works on long-term capacity planning, ODVFC can quickly adapt to temporary changes in demand.
2) We propose an integer linear programming (ILP)-based framework called ODCP for capacity planning in ODVFC. By using prediction-based workload estimation, the anticipatory planning phase ensures timely and accurate vehicle routing solutions. By using two-phase capacity planning, it is sufficiently lightweight to be updated frequently.
3) ODCP improves the QoS and profit by increasing the accuracy of traffic prediction using SARIMA instead of the regression-based method. Furthermore, ODCP outperforms the existing vehicle routing method by balancing the QoS and costs in the objective function instead of regarding the demand as a hard constraint.

The rest of the article is organized as follows. Section II reviews the related works, and Section III demonstrates the system overview. We detail the demand prediction and resource provisioning in Sections IV and V, respectively. Sections VI and VII present the experimental setup and results, respectively. Finally, Section VIII presents the discussion, and Section IX concludes the article.

II. Related Work

This section compares our work with the state-of-the-art capacity planning, resource management, and vehicle routing methods, as listed in Table I.

A. Capacity Planning and Resource Management Methods

Three scenarios of edge/fog computing have been studied from the capacity planning perspective. The first scenario considers the capacity planning of stationary edge/fog nodes under certain constraints, such as latency constraints. For example, Stypsanelli et al. [4] formulated the capacity planning problem as an ILP problem that minimizes the costs and aimed to find the capacity planning strategy with the probability delay guarantee. Noreiki et al. [5] proposed a method to find the combinations of resource demand that can improve resource utilization in the edge computing environment. Noreiki et al. [6] further modeled the capacity planning problem using queuing theory, and aimed to minimize the number of homogeneous fog nodes while meeting the demands of latency-sensitive applications. In the second scenario where vehicles become users of computing services, the mobility of users is considered. For example, Hussain et al. [7] formulated an ILP problem aiming to minimize both latency and energy consumption in the vehicular environment, and including computing, network, cloud capacity, and link types as constraints. Premsankar et al. [8] utilized a mixed ILP model to find the location, number, and capacity of fog nodes in order to minimize the costs and fulfill the computational demand and network coverage requirements. Although these works have considered the mobility of users, their solutions do not support VFNs.

In the third scenario, the mobility of both users and fog nodes is considered. In our previous work [3], we considered both stationary CFNs and VFNs that follow regular routes and timetables, such as the fog nodes carried by buses, and proposed a long-term capacity planning solution based on ILP. In this work, we also target the third scenario where both users and fog nodes are mobile. Different from [3], we assume that the routes of VFNs can be scheduled frequently to fulfill temporary or unexpected demand with cost-efficiency. Such short-term capacity planning solutions are supposed to be lightweight compared with long-term capacity planning.

Resource management has also been studied for VFC. For example, Wu et al. [9] proposed a cooperative caching scheme for vehicles based on asynchronous federated and deep reinforcement learning to minimize the content transmission delay. They [10] further proposed a deep Q-network-based algorithm where the intelligent vehicular nodes are trained to learn the dynamic environment with maximized age fairness. Although ODVFC is also related to resource management with the aim of maximizing profit and QoS, the methods in [9], [10] cannot be used for ODVFC since they do not support dynamic VFN routing.

TABLE I
COMPARISON OF OUR WORK WITH THE EXISTING CAPACITY PLANNING METHODS [3], [4], [5], [6], [7], [8], RESOURCE MANAGEMENT METHODS [9], [10], AND VEHICLE ROUTING METHODS [11], [12], [13], [14], [15], WHERE UM REPRESENTS THE USER (E.G., USER VEHICLES AND CUSTOMERS/GOODS) MOBILITY AND RM REPRESENTS THE RESOURCE (E.G., FOG NODES AND TAXIS/DELIVERY VEHICLES) MOBILITY

Work	Method	Objective	Outputs	Constraints	UM	RM
[3]	Heuristic, integer linear programming	Minimize installation and operational costs	Location, number of CFNs, schedule of VFNs	Resource consumption, latency, resource capacity, availability	Yes	Yes
[4]	Mixed integer linear programming	Minimize installation and operational costs	Number and capacity of fog infrastructure	Probabilistic delay guarantees	No	No
[5]	Knapsack algorithm	Improve edge utilization	Combinations of resource demands	Latency, resource consumption	No	No
[6]	Queuing model	Minimize number of fog nodes	Number of fog nodes	Latency, resource consumption	No	No
[7]	Integer linear programming	Minimize latency and energy	Location, number, capacity of fog nodes	Node and network capacity, link type, cloud capacity	Yes	No
[8]	Mixed integer linear programming	Minimize deployment costs	Location, number, power of fog nodes	Network coverage, computational demand	Yes	No
[9]	Asynchronous federated and deep reinforcement learning	Minimize the content transmission delay	Location to cache the content.	link capacity, cache capacity, mobility, content popularity	No	Yes
[10]	Deep Q-network	Maximize age fairness utility	Size of minimum contention window	Age of information, transition probability, vehicle density	Yes	Yes
[11], [12]	Dynamic attention model, REINFORCE algorithm	Minimize distance	Routes of vehicles	Customer demand and location, vehicle capacity	No	Yes
[13]	Genetic algorithm, particle swarm optimization	Minimize number of vehicles and distance	Routes of vehicles	Customer demand and location, vehicle capacity, time windows	Yes	Yes
[14]	Multi-agent attention model, policy gradient	Minimize number of vehicles and distance	Routes of vehicles	Customer demand and location, vehicle capacity, time windows	Yes	Yes
[15]	Heuristic, integer linear programming	Minimize distance	Routes of UAVs	Resource consumption, resource capacity, cell range	Yes	No
ODCP	Heuristic, integer linear programming, SARIMA	Maximize profit and QoS	Routes of VFNs	Resource consumption, latency, resource capacity, cell range	Yes	Yes

Due to space limitations, we have not detailed the inputs.

B. Vehicle Routing Methods

The vehicle routing problem (VRP) is a well-studied problem in the literature, mainly used for scheduling taxis to pick up passengers or scheduling logistic cars to deliver goods. Similar to our problem, VRP aims to route the vehicles to the location of demand with minimized traveling costs. The works in [11] and [12] proposed scheduling the routes of vehicles to minimize the traveling costs using deep reinforcement learning. However, the demands in the works remain at the same level across the time horizon.

VRP with time windows (VRPTW) considers the dynamic demand which changes over time, similar to ODVFC. Different methods, such as the genetic algorithm [13], particle swarm optimization [13], multi-agent attention model and policy gradient [14], have been proposed to solve the VRPTW problem. These solutions do not fit well when applied to solve the capacity planning for ODVFC, due to the following reasons. Previous works on VRPTW assumed that the demand is known beforehand, which means they did not handle unexpected changes in the demand. Apart from this, the objective of the VRPTW is to route the vehicles to the exact places as the customers. However, in ODVFC, the VFNs are supposed to provide services for the vehicles within the one-hop communication range. Finally, previous works on VRPTW did not consider different levels of QoS. In other words, they always tried to satisfy all the demands; however, in our case, the ratio of users to be served varies with time and region. In most works related to VRP, the focus has been placed on scheduling the routes of service providers. There also exist works that try to move the users instead of service providers. For example, Bekkouche et al. [15] proposed an unmanned aerial vehicle (UAV) path planning solution based on ILP and a heuristic. In this case, UAVs are the users of edge computing services provided by stationary cellular fog nodes. Different from [15], our work aims to schedule the routes of VFNs in order to fulfill the demand generated by other vehicle users. We consider only ground vehicles in our case.

III. SYSTEM OVERVIEW

In this section, we demonstrate an exemplary scenario of ODVFC and present the system architecture in terms of network architecture and methodology.

A. Exemplary Scenario

Dedicated short-range communications (DSRC) and cellular V2X (C-V2X) are the most widely used radio access technologies for vehicular communication. For ODVFC, we assume that communication between vehicles is implemented with 5 G new radio (NR) V2X. In this work, we assume that the vehicles communicate with others connected to the same cell using 5 G V2N. Note that the application data exchanged between VFNs and the user vehicles can be implemented with 5 G NR V2V after the task assignment is completed. In addition, VFC, including ODVFC, is meant to support the offloading of compute-intensive and latency-sensitive tasks from vehicles to nearby CFNs or VFNs. In ODVFC, we focus on routing VFNs with flexible schedules to fulfill the demand which is not covered by CFNs or the VFNs with fixed routes. Fig. 1(a) illustrates an exemplary scenario of ODVFC that we envision. In this work, a city is divided into multiple connected but independent regions, each

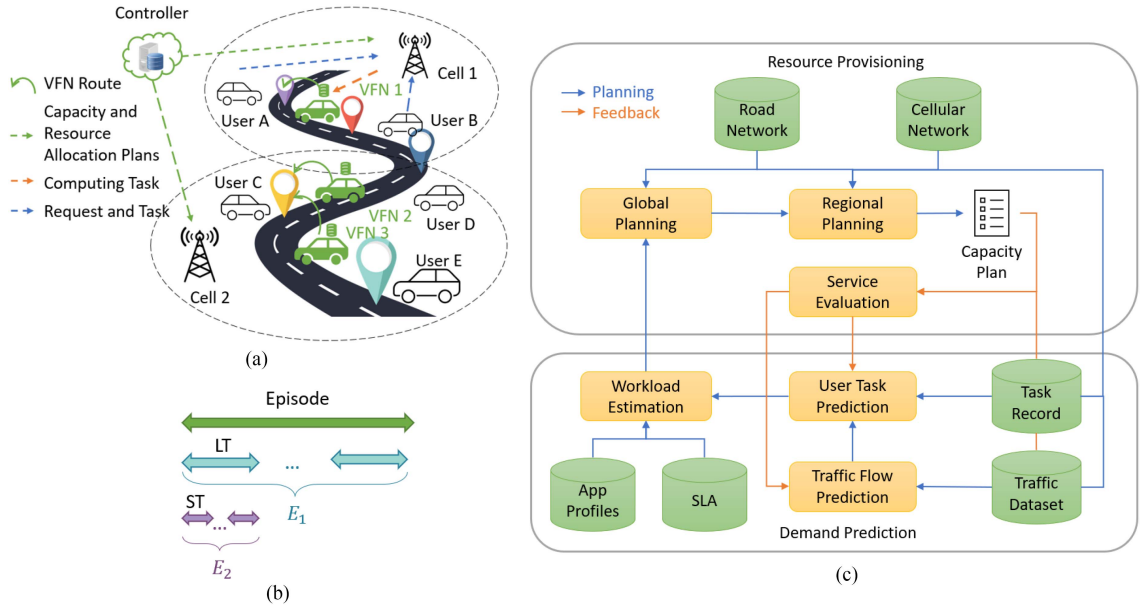

Fig. 1. Overview of ODVFC. (a) Exemplary scenario, (b) Time horizon, and (c) Flowchart of ODCP.

consisting of multiple nodes. The nodes are the start or end points of the road segments that compose the road. In the region shown in the figure, two cellular cells are deployed along a road that consists of at least five nodes. There are five users in the figure. Based on the workload estimation, three VFNs have been allocated to this region beforehand. At the upcoming time slot, *User A* and *User B* will arrive at the purple and red nodes, respectively, both are located within the communication range of *Cell 1*. Thus *VFN 1*, whose capacity is high enough to serve both *User A* and *User B*, is sent to the purple node. Similarly, we route *VFN 2* and *VFN 3* to the yellow node, so that they can serve the *Users C, D, and E*. To ensure the availability of a VFN in a given cell when a latency-critical service is received, we assume that VFNs are routed to the regions before the demand is generated. To fulfill this assumption, it is essential to be able to accurately predict the traffic flow and the corresponding demand for computational resources in the next moment or time window. In the case of ODVFC, the capacity plan needs to be updated frequently, which requires the capacity planning solution to be lightweight enough to calculate the routes within minutes or even seconds.

B. System Architecture

We assume that both the VFNs and user vehicles are equipped with 5 G onboard units (OBUs). When a user generates a computing task, it will send a request to the cellular base station that it connects to, including its information (e.g., vehicle ID, location, speed, and direction) and the task-related information (e.g., service type and latency requirement). The cellular base station receives the capacity and resource allocation plans from the central entities (i.e., the controller in Fig. 1(a)) and informs the user whether the task can be executed or not, and in the former case, the ID of the VFN is assigned to the user. The user and VFN update their locations periodically to the connected base stations. When they are within the communication range of the same cellular cell, the base station initiates the task offloading between them. We utilize lightweight containers for service provision [16], and the result is transmitted to the user while the task is executed on the VFN. When either the user or the VFN moves out of the cellular cell, the tasks would be migrated to the new cell with iterative live migration [16].

We follow a data-driven methodology to implement capacity planning in ODVFC. The flowchart of ODCP is presented in Fig. 1(c), where the demand and supply are modeled separately. To predict the demand, we need both the city-scale vehicular traffic distribution and the resource consumption profiles of different vehicular applications. The inputs for demand prediction, as described in Section IV-A, are the traffic dataset, task record, application profiles, and SLA. The traffic dataset and task record are used for predicting traffic flows and the arrivals of computing tasks, respectively. Together with the vehicular application profiles, which describe the patterns of the central processing unit (CPU) and graphics processing unit (GPU) consumption of each task, the workloads can be estimated. In other words, we can estimate how much CPU and GPU capacity is needed for satisfying the latency requirements of the services run on the vehicles in each region. From the SLA negotiated with the users, we know the price for each task and the penalty cost if the SLA is violated. Based on the price and penalty cost, we estimate the revenue w.r.t. the workload, which is fed to the resource provisioning module.

Apart from the estimated workload and revenue provided by the demand prediction module, the inputs for resource provisioning also include the road and cellular maps, as described in Section V-A. To balance the trade-off between the computational time and the techno-economic performance of the capacity planning solution, we divide the road network of the considered city into multiple regions and create the capacity plan in two steps, namely global planning and regional planning. Afterward, we

feed the capacity plan to the service evaluation module in order to get feedback for the next iteration.

The decision-makers of capacity planning are the global and regional agents, which are controllers (e.g., CFNs) running the optimization problems. The global agent is located in the central cloud. It collects information from the regional agents and evaluates the service performance. The regional agents are distributed at the centroids, which are the nodes that have the minimum traveling distance to all other nodes in each region, calculated by the Floyd-Warshall algorithm [17]. The regional agents monitor the state of each VFN and each user in their regions and forward them to the global agent.

During global planning, the global agent is responsible for deciding the QoS level (i.e., what percent of tasks are planned to be served in each region) and the global routing plan (i.e., how many VFNs will be routed to each region) based on the demand prediction. In this step, each episode (i.e., the duration of the capacity planning horizon) is discretized into $E1$ long time slots (LTs), cf. Fig. 1(b), based on which we update the global capacity plan. During regional planning, the regional agent takes charge of determining the regional routing plan (i.e., which VFN will be routed to which node) and the resource allocation plan (i.e., which VFN will serve which user) based on the upcoming location as well as CPU and GPU consumption of each user in the region. In this step, each LT is discretized into $E2$ short time slots (STs), cf. Fig. 1(b), based on which we update the regional capacity plan.

Finally, the regional capacity plan is fed to the service evaluation module to measure the achieved QoS and profit. At the end of each LT, the global agent evaluates the prediction accuracy of traffic flow in the current LT and uses it as feedback to improve the prediction accuracy in the next LT. At the end of each episode, the global agent evaluates the cost-efficiency of the sequential capacity plans on the episode as feedback for the capacity planning for the next episode.

TABLE II
NOTATIONS AND DEFINITIONS

Notation	Definition
$E1$	Number of LTs in an episode
$E2$	Number of STs in an LT
A_j, A_t	Accurate traffic flow in region j / in LT t
F_j, F_t	Forecast traffic flow in region j / in LT t
I	Set of VFNs in the city
J	Set of regions plus the depot
K	Set of QoS levels
L	Set of VFNs in a region
M	Set of nodes in a region
N	Set of tasks planned to be served in a region
P	Set of computing tasks in a region
Q	Set of required VFNs in a region
S	Symmetric mean absolute percentage error
r_{ser}	City-scale QoS (in terms of service rate)
C_{pro}	City-scale profit in an LT
C_{rev}	City-scale revenue in an LT
$c_{\text{tra}}, C_{\text{tra,cs}}$	Unit traveling cost, City-scale traveling cost in an LT
$c_{\text{rent}}, C_{\text{rent}}$	Unit rental cost, City-scale rental cost in an LT
$c_{\text{cap}}, C_{\text{cap}}$	Unit capacity cost, City-scale capacity cost in an LT
α_{pen}	Ratio of penalty cost and price of a task
$C_{\text{tra,rg}}$	Regional traveling cost in an ST
C_{pen}	Regional penalty cost in an ST
x_{pq}	Whether task p is assigned to VFN q
$c(p), g(p)$	CPU and GPU consumption of task p
$B_{\text{CPU}}, B_{\text{GPU}}$	CPU and GPU capacity of VFN
x_{ij}	Whether VFN i will travel to region j
d_{ij}	Traveling time for VFN i to region j
a_{ij}	Whether VFN i is active in region j
y_{jk}	Whether region j will select QoS level k
δ_{jt}	Number of served tasks in region j in ST t
η_{jt}	Number of total tasks in region j in ST t
γ_{jk}	Number of tasks in region j with QoS level k
w_{jk}	Estimated workload in region j with QoS level k
θ_{jk}	Revenue in region j with QoS level k
x_{lm}	Whether VFN l will travel to node m
d_{lm}	Traveling time from VFN l to node m
y_{mn}	Whether VFNs on node m will serve task n
h_{mn}	Whether VFNs on node m can connect to task n
ϵ_n	Whether the SLA is violated for task n
β_n	Penalty cost of task n according to the SLA
v	Speed of VFN when calculating traveling time

IV. DEMAND PREDICTION

The demand prediction model aims to estimate the workload that is generated in each region during each LT based on the traffic flow and task volume prediction. In this section, we detail it in terms of inputs, traffic flow prediction, task volume prediction, and problem formulation. The notations and definitions used in the model are listed in Table II.

A. Inputs for Demand Prediction

The inputs for demand prediction include the traffic dataset, task record, application profiles, and SLA.

1) Traffic Dataset: According to the traffic flow theory [18], traffic flow describes both macroscopic and microscopic behaviors of vehicle traffic. In this work, the macroscopic traffic dataset records the number of vehicles in each region during each LT, while the microscopic traffic dataset consists of the locations of each vehicle in each ST. The former is used for traffic flow prediction and workload estimation, while the latter is used for regional planning and service evaluation.

2) Task Record: We define each computing task involved in vehicular applications with two attributes: the service type and the latency requirement (note that there can be multiple levels of latency requirement for one service, cf. Table III). The information of the scheduled tasks is collected from all the users into the regional agents during each ST. The key information from each region is further forwarded to the global agent and stored in the task record in each LT. From the user's perspective, the computing task will either be served (i.e., delivered within the pre-defined latency constraint) or unserved.

3) Application Profiles: Apart from the number of computing tasks, the computing demand also depends on how much CPU and GPU resource is required for processing each task within its latency constraint [5]. We containerize exemplary services into Docker images [19] and follow the benchmark algorithm proposed in [3] to measure the CPU and GPU consumption corresponding to different combinations of service types and latency requirements.

4) Service Level Agreement: Following the common practice in cloud computing, we use SLA to describe commitments

TABLE III
APPLICATION PROFILES AND SLA, WHERE L1-L3 ARE DIFFERENT STANDARDS OF LATENCY REQUIREMENTS, C1-C3, G1-G3, AND P1-P3 ARE THE CPU CONSUMPTION, GPU CONSUMPTION, AND PRICES, RESPECTIVELY, CORRESPONDING TO EACH LATENCY REQUIREMENT

Service type	Latency (ms)			CPU usage (%)			GPU usage (%)			c_{price} (MU/task)		
	L1	L2	L3	C1	C2	C3	G1	G2	G3	P1	P2	P3
Object detection	10	25	100	12%	9%	4%	27%	20%	9%	10	8	6
Semantic segmentation	100	150	250	11%	8%	6%	19%	15%	12%	8	7	6
Lane detection	10	25	100	45%	31%	13%	0%	0%	0%	8	6	4
Video transcoding	100	150	250	24%	16%	10%	0%	0%	0%	6	5	4

between the service provider and customers selected by the global agent [20]. SLA defines negotiable parameters, such as desired QoS, prices, and penalties, which provide inputs for calculating the most profitable resource allocation plan that avoids or minimizes the violations of the agreement [20]. In this work, we define the economic model in SLA as follows: for a given task, the service provider will charge the price w.r.t. the service type and latency requirement (cf. Table III) if the task is anticipated to be served during the global planning phase. However, if the VFNs fail to serve the task during the regional planning phase, the penalty cost (directly proportional to the price) will be refunded to the user. The penalty cost is usually higher than the price of the task, which motivates the model to reduce the violation of SLA.

B. Traffic Flow and Task Volume Prediction

Auto-regressive integrated moving average (ARIMA) is a widely-used method for predicting traffic flow. As an extended algorithm of ARIMA, seasonal ARIMA (SARIMA) is particularly useful to model seasonal traffic behavior [21]. In this work, we use SARIMA to predict the macroscopic traffic flow (i.e., the number of vehicles in each region) with a minimum Akaike information criterion (AIC) [22]. During an LT, the actual traffic flow and the predicted traffic flow in the region j are denoted by A_j and F_j, respectively.

We assume that each user will keep one active computing task in each ST, which means that the number of tasks to be executed is equal to the number of vehicles. We define QoS levels based on the percentage of tasks that are planned to be served by the available VFNs. For example, if the set of QoS levels is K, the highest level $|K|$ ($|.|$ represents the cardinality of the set) indicates all the computation requests from vehicles are planned to be handled by the VFNs, while the lowest level 1 indicates only $1/|K|$ of tasks are planned to be served. The number of tasks to be executed in the region j under QoS level k, denoted by γ_{jk}, can be calculated as

$$\gamma_{jk} = F_j \times k/|K|. \tag{1}$$

We predict the number of tasks belonging to each service type and latency requirement based on sampled probability. More specifically, after aggregating the user tasks selected in each LT, we calculate the probability of selecting each service type and latency requirement based on the statistics. This probability is then multiplied by the predicted number of tasks to be executed in order to predict the number of each type of task that will be generated during the upcoming LT. Finally, we accumulate the prices of the tasks planned to be served in each region for every QoS level. We represent it as the revenue θ, which is an input for global planning.

C. Workload Estimation

In ODVFC, the computing tasks generated from the users are expected to be offloaded to the VFNs. Therefore, we formulated the problem of workload estimation (i.e., how much workload would be generated or needs to be handled) as a capacity planning problem of finding the minimum computing capacity requirement. For the sake of simplicity, we assume that the VFNs contain homogeneous configurations and provide identical services to the requests generated by the users.

In this problem, P represents the set of computing tasks, and Q represents the set of VFNs that are required for handling P. Assume that the resource consumption of each task does not exceed the capacity of any single VFN, thus in the worst case, the number of required VFNs is equal to the number of tasks. This is denoted as $|P| = |Q| = \gamma$. The CPU and GPU consumption of each computing task p is represented by $c(p)$ and $g(p)$, respectively. The values of $c(p)$ and $g(p)$ depend on the service type and latency requirement. The maximum capacity of the CPU and GPU, depending on the configurations of the VFNs, are given by B_{CPU} and B_{GPU}, respectively. The workload estimation problem can be expressed in (2a)–(2e). The objective function (2a) aims to minimize the workload, namely the number of VFNs required to serve all the computing tasks generated by the users. The binary decision variable x_{pq} indicates if the task p is assigned to the VFN q. The symbol $\lceil . \rceil$ represents the ceiling function, and the term $\left\lceil \frac{\sum_{p \in P} x_{pq}}{\gamma} \right\rceil$ indicates that if at least one task is assigned to the VFN, then the VFN will be required. Constraints (2b) and (2c) are the *CPU capacity constraint* and *GPU capacity constraint*, respectively, which ensure that the computing tasks assigned to each VFN do not exceed the corresponding CPU or GPU capacity defined in B_{CPU} and B_{GPU}, respectively. Finally, constraint (2d) is the *non-repetitive assignment constraint*, which guarantees that each computing task is only assigned to one VFN.

$$\min_{x_{pq}} \sum_{q \in Q} \left\lceil \frac{\sum_{p \in P} x_{pq}}{\gamma} \right\rceil \tag{2a}$$

$$\text{s.t.} \quad \sum_{p \in P} c(p) x_{pq} \leq B_{\text{CPU}}, \forall q \in Q \tag{2b}$$

$$\sum_{p \in P} g(p) x_{pq} \leq B_{\text{GPU}}, \forall q \in Q \tag{2c}$$

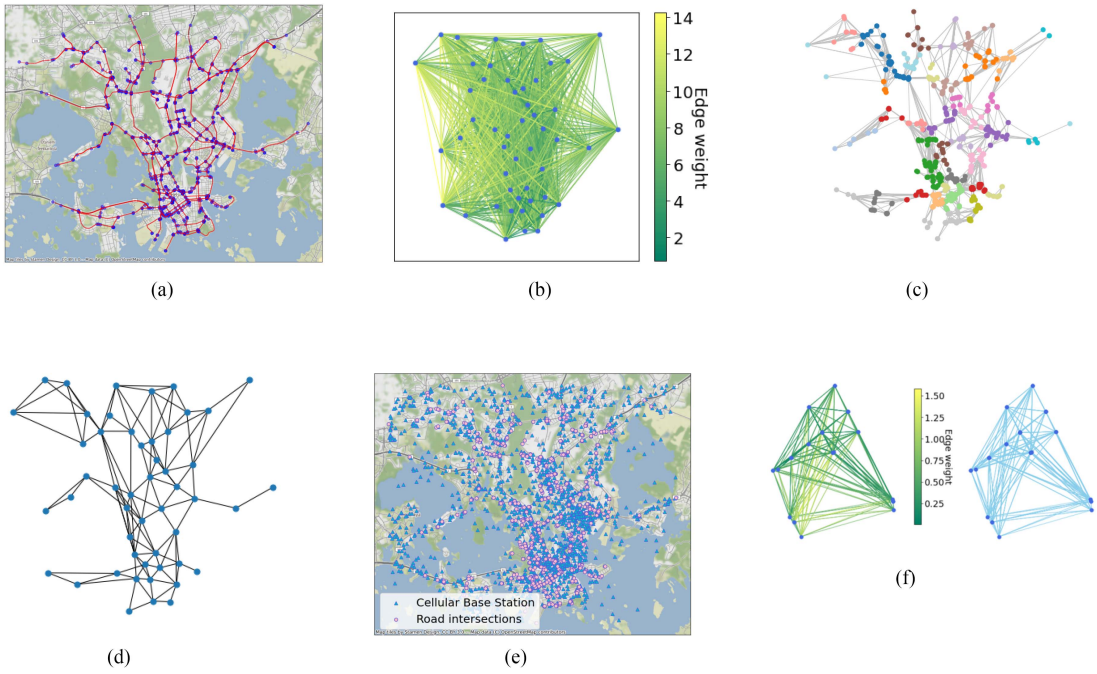

Fig. 2. Graphs generation using graph theory in Helsinki. (a) City-scale road network, where the red lines are road segments and the blue dots are nodes. (b) Traveling distance matrix of the road network, where the legend represents the distance in kilometers. (c) Exemplary road clustering using k-means, where each dot is a node and each color represents a region. (d) City-scale region network, where each dot is a region. (e) Map overlay of the cellular base station and road maps. (f) Left: Traveling distance matrix of the road network in a region, where the legend represents the distance in kilometers. Right: cellular coverage map in a region, where two adjacent nodes are within the coverage of the same cellular base station.

$$\sum_{q \in Q} x_{pq} = 1, \forall p \in P \tag{2d}$$

$$x_{pq} \in \{0, 1\}, \forall p \in P, \forall q \in Q. \tag{2e}$$

We use the *nextfit heuristic* [23] to solve the workload estimation problem as follows: we first choose a random VFN and assign tasks to it until all the CPU and GPU resources have been allocated. We repeat this process of choosing random VFNs and assigning tasks until we have assigned all tasks or all VFNs are fully booked. The heuristic method could offer two benefits: low complexity and bounded solutions. First, since the complexity of the *next-fit heuristic* is $O(n)$, the method can solve the problem in polynomial time [23]. Second, while the solution based on the heuristic suffers from the sub-optimal issue, the approximated solution could act as an upper bounding for the workload. This would avoid underestimating the workload, which is particularly problematic in the initial assumption in global planning that a VFN assigned to one region can serve all the users within the same region. The workload estimation problem is solved in each LT for each region at each QoS level. The value of the objective function is represented as the workload w, which is an input for global planning.

V. Resource Provisioning

During the resource provisioning phase, we need to decide the QoS level (i.e., the percentage of the tasks that are expected to be served), the routing plan (i.e., when and where to send each VFN), as well as the resource allocation plan (i.e., which user will be served by which VFN). When the numbers of users and VFNs increase, the joint scheduling of all the tasks and VFNs will be time-consuming. To speed up the decision-making, we propose to create the capacity plan in two phases, including global planning and regional planning phases. In this section, we explain the input for resource provisioning, the formulation of global and regional planning problems, and the metrics used for evaluating techno-economic performance.

A. Inputs for Resource Provisioning

Apart from microscopic traffic data, SLA, and estimated workload, the inputs for resource provisioning include the road and cellular maps, based on which we generate the road network and cellular coverage graphs, respectively, using graph theory [24]. The city-scale road network (cf. Fig. 2(a)) is a weighted directed graph extracted from HERE Traffic API [25], where each edge represents a road segment defined by HERE, and the nodes are either the starting or the ending point of a road segment. Two nodes are adjacent to each other if there is a road segment in between. The edge weights are set by the traveling distance through a road segment. Consequently, different driving directions have different edge weights. We calculate the traveling distance matrix of each pair of nodes following the Floyd-Warshall algorithm [17] (cf. Fig. 2(b)). We then calculate the traveling time based on the traveling distance, assuming the speed of VFN is a constant value v (cf. Table IV). and each color represents a region. (d) City-scale region network, where each dot is a region. (e) Map overlay of the cellular base station and road maps. (f) Left: Traveling distance matrix of the road network in a region, where the legend represents the

TABLE IV
SIMULATION CONFIGURATIONS FOR THE EXPERIMENTS

Parameter	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Unit
Traffic model	*STM, TM-SC, TM-OE*	*STM*	*STM*	*STM*	/
Scenario	*ODCP (with AT, TP-WF, TR-NF), VR, RG*	*ODCP with AT*	*ODCP with AT*	*ODCP with AT*	/
c_{tra}	1.0	1.0	0.5, 1.0, 1.5, 2.0	1.0	MU/minute
c_{rent}	5	5	5, 10, 15, 20	5	MU/device
α_{pen}	1.5	1.5	0.5, 0.75, 1.0, 1.5	1.5	/
c_{cap}	0	0, 10, 30, 50	0	0	MU/device
$\lvert I\rvert$	30	20, 30, 40, 50, 60, 70	30	30	/
$\lvert J \setminus \{0\}\rvert$	20	20	20	10, 12, 14, 16, 18, 20	/
$\lvert K\rvert$	10				/
$\lvert E1\rvert$	24				hour
$\lvert E2\rvert$	12				5 minutes
v	30				km/h

distance in kilometers. Right: cellular coverage map in a region, where two adjacent nodes are within the coverage of the same cellular base station.

We divide the city-scale road network into multiple connected but independent regions using k-means clustering (cf. Fig. 2(c)). More specifically, the algorithm aims to divide the nodes into a pre-defined number of regions to minimize the sum of distances between the nodes and their respective regional centroids. Various numbers of regions are used to evaluate the impacts of clustering (cf. Table IV). We assume that a VFN is dedicated to the users located in the same region and within the communication range of the VFN.

The city-scale region network (cf. Fig. 2(d)) is a weighted directed graph, where each node is a region. Two regions are adjacent to each other if any nodes within them are adjacent. The weight of the edge is the traveling distance between the centroids of the regions. We extract the road network for each region together with the traveling distance matrix of each pair of nodes (cf., Fig. 2(f) on the left), and calculate the traveling time based on the distance.

For each region, we also create a cellular coverage graph to describe the locations and coverage of cellular base stations. The graph is created using map overlay (cf. Fig. 2(e)). The first step is to plot the locations of nodes from the road map. The next step is to plot the locations of base stations from the cellular map. Then, we find the overlap between the nodes and the coverage of base stations and generate the cellular coverage graph. The cellular coverage graph (cf. Fig. 2(f) on the right) is an unweighted undirected graph, where each node represents a node. Two nodes are adjacent to each other if they are within the coverage of the same cellular base station.

B. Global Planning

The global planning problem aims to decide the QoS level for each region and the global routing plan based on workload estimation. The decision is made by the global agent in each LT. In this problem, I denotes the set of VFNs in the city, J denotes the set of regions plus the depot, and the depot is represented by 0. K denotes the set of QoS levels. The output of the workload estimation model, denoted as w_{jk}, represents the estimated workload in the region j when it chooses the QoS level k.

The city-scale revenue in an LT is defined as

$$C_{\text{rev}} = \sum_{j \in J} \sum_{k \in K} \theta_{jk} y_{jk} \tag{3}$$

where θ_{jk} represents the revenue gained from the region j when the QoS level k is selected, based on the prices defined in the SLA, and y_{jk} is the binary variable indicating whether the region j will select the QoS level k.

The city-scale traveling cost in an LT is defined as

$$C_{\text{tra,cs}} = c_{\text{tra}} \times \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \tag{4}$$

where c_{tra} denotes the unit traveling cost per VFN per minute, d_{ij} denotes the time it would take for *VFN* i to travel from the location at the end of the current LT to the region j, based on the traveling distance matrix of the city-scale region network, and x_{ij} is the binary variable indicating whether the VFN i will travel to the region j.

The city-scale rental cost in an LT is defined as

$$C_{\text{rent}} = c_{\text{rent}} \times \sum_{i \in I} \sum_{j \in J} a_{ij} x_{ij} \tag{5}$$

where c_{rent} represents the unit rental cost per VFN per LT, and a_{ij}, which indicates whether the VFN i is active or not in the region j, equals 0 when the VFN is in the depot ($j = 0$) and equals 1 otherwise.

The global planning problem can be formulated in (6a)–(6f). The objective function (6a) aims to maximize the profit, which is the difference between the revenue and the traveling and rental costs on a city scale. Constraint (6b) is the *QoS constraint*, which guarantees that we have routed enough VFNs to each region according to their QoS level selection. Constraint (6c) is the *one destination constraint*, which prevents any VFN from going to multiple regions during one LT. Finally, constraint (6d) is the *one QoS level constraint*, which ensures that a region can only select one QoS level during one LT.

$$\max_{x_{ij}, y_{jk}} \quad C_{\text{rev}} - C_{\text{tra,cs}} - C_{\text{rent}} \tag{6a}$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ij} \leq \sum_{k \in K} w_{jk} y_{jk}, \forall j \in J \tag{6b}$$

$$\sum_{j \in J} x_{ij} = 1, \forall i \in I \tag{6c}$$

$$\sum_{k\in K} y_{jk} = 1, \forall j \in J \tag{6d}$$

$$x_{ij} \in \{0,1\}, \forall i \in I, \forall j \in J \tag{6e}$$

$$y_{jk} \in \{0,1\}, \forall j \in J, \forall k \in K. \tag{6f}$$

C. Regional Planning

During the global planning phase, we assume that a VFN assigned to one region is allowed to serve all the users located inside the same region. However, in practice, a VFN can only serve the users within the communication range. In ODVFC, we assume that the VFNs located at the same node can share their resources to serve the surrounding users. The above two aspects are considered in the regional planning model. The regional planning problem aims to determine the regional routing plan and the resource allocation plan based on the upcoming locations of the users (e.g., collected from the vehicles' navigation system) and their task selections. The decision is made by the regional agent in each ST. In this problem, L, M, and N represent the sets of VFNs, nodes, and tasks in the region, respectively. The CPU and GPU consumption of the task n are denoted as $c(n)$ and $g(n)$, respectively.

Similar to the definitions presented in Section V-B, the regional traveling cost in an ST is defined as

$$C_{\text{tra,rg}} = c_{\text{tra}} \times \sum_{l\in L}\sum_{m\in M} d_{lm}x_{lm} \tag{7}$$

where d_{lm} is the traveling time from the *VFN* l to the node m, and x_{lm} is the binary variable representing whether the *VFN* l will travel to the node m.

The regional penalty cost in an ST is defined as

$$C_{\text{pen}} = \sum_{n\in N} \beta_n \epsilon_n \tag{8}$$

where β_n denotes the penalty cost of the task n, and ϵ_n is a binary variable that represents whether the SLA is violated for the task n.

The regional planning problem can be formulated in Equations (9a)–(9g). Our objective function (9a) aims to minimize regional traveling and penalty costs. We represent the SLA violation by ϵ_n defined in (9b). The binary variable y_{mn} denotes whether the VFNs on the node m will serve the task n. h_{mn} indicates whether the VFNs on the node m can connect to the user of the task n, based on the location of the user at the ST as well as the adjacency matrix of the cellular coverage graph. Constraint (9c) is the *one destination constraint*, which means each VFN can only go to one node during each ST. Constraints (9d) and (9e) are the *CPU capacity constraint* and *GPU capacity constraint*, respectively, which avoid the computing tasks assigned to the VFNs at each node from exceeding the maximum capacity of VFNs.

$$\min_{x_{lm},y_{mn}} \quad C_{\text{tra,rg}} + C_{\text{pen}} \tag{9a}$$

$$\text{s.t.} \quad \epsilon_n = 1 - \sum_{m\in M} h_{mn}y_{mn}, \forall n \in N \tag{9b}$$

$$\sum_{m\in M} x_{lm} = 1, \forall l \in L \tag{9c}$$

$$\sum_{n\in N} c(n)y_{mn} \le \sum_{l\in L} x_{lm}, \forall m \in M \tag{9d}$$

$$\sum_{n\in N} g(n)y_{mn} \le \sum_{l\in L} x_{lm}, \forall m \in M \tag{9e}$$

$$x_{lm} \in \{0,1\}, \forall l \in L, \forall m \in M \tag{9f}$$

$$y_{mn} \in \{0,1\}, \forall m \in M, \forall n \in N. \tag{9g}$$

The global and regional planning problems were developed in Python 3.7 and solved using the Gurobi [26] solver.

D. Performance Metrics

We evaluated the accuracy of traffic flow prediction using the symmetric mean absolute percentage error (SMAPE). Since the traffic flow could sometimes be 0, the mean absolute percentage error (MAPE) cannot be used here. The SMAPE during each LT, denoted by S, is calculated as follows:

$$S = \frac{1}{|J|-1}\sum_{j\in J\setminus\{0\}} \frac{|F_j - A_j|}{(A_j + F_j)/2} \tag{10}$$

where $J \setminus \{0\}$ is the set of regions except the depot, A_j and F_j represent the actual traffic flow and the predicted traffic flow using SARIMA for region j during the LT, respectively. We use service rate r_{ser} to evaluate the achieved QoS. It calculates the ratio of tasks served during each LT as follows:

$$r_{\text{ser}} = \frac{1}{|J|-1}\frac{1}{|E2|}\sum_{j\in J\setminus\{0\}}\sum_{t\in E2} \delta_{jt}/\eta_{jt} \tag{11}$$

where $E2$ is the set of STs in an LT, δ_{jt} and η_{jt} are the number of served tasks and the overall number of tasks in the region j in ST t, respectively. It is worth mentioning that the achieved QoS r_{ser} can be different from the QoS level selected for each region during global planning, due to the different assumptions mentioned in V-C and the potential errors in traffic flow and task volume prediction.

To evaluate the long-term economics, we assume there is a capacity cost c_{cap} for each deployed VFN. The city-scale capacity cost in an LT can be calculated as

$$C_{\text{cap}} = c_{\text{cap}} \times |I| = c_{\text{ins, man}} \times |I|/Ls \tag{12}$$

where $|I|$ is the number of VFNs in the city, $c_{\text{ins, man}}$ is the installation and maintenance cost per VFN, and Ls is the lifespan of the VFNs in LTs.

We use profit C_{pro} to evaluate the economic benefit of ODVFC during each LT, which is the difference between the city-scale revenue and the overall traveling, rental, capacity, and penalty costs. It is calculated as follows:

$$\begin{aligned} C_{\text{pro}} &= (6a) - \sum_{j\in J\setminus\{0\}}\sum_{t\in E2}(9a) - C_{\text{cap}} \\ &= C_{\text{rev}} - C_{\text{tra,cs}} - C_{\text{rent}} - C_{\text{cap}} \\ &\quad - \sum_{j\in J\setminus\{0\}}\sum_{t\in E2} C^{j,t}_{\text{tra,rg}} - \sum_{j\in J\setminus\{0\}}\sum_{t\in E2} C^{j,t}_{\text{pen}}. \end{aligned} \tag{13}$$

By maximizing Equation (6a) during global planning and minimizing Equation (9a) during regional planning, we have transformed the multi-objective of maximizing QoS (in terms of minimizing penalty cost C_{pen}) and maximizing revenue C_{rev} into a single objective, namely maximizing profit C_{pro} in Equation (13). Moreover, we assume that the penalty cost of a task c_{pen} is proportional to the price of the task c_{price}, with their ratio α_{pen} denoted as

$$\alpha_{\text{pen}} = c_{\text{pen}}/c_{\text{price}}. \tag{14}$$

When we adjust the value of α_{pen}, we are essentially adjusting the weights of the QoS and revenue.

At the end of each episode, the values of r_{ser} and C_{pro} are used for techno-economic evaluation. Between them, r_{ser} reflects the general performance while C_{pro} is more sensitive to extreme scenarios, for example, those with large spatio-temporal variation in the demand.

VI. Experimental Setup

This section describes the experimental setup including the input datasets, traffic models, capacity planning strategies, as well as simulation and network configurations.

A. Input Datasets

As discussed in Section IV-A, the inputs for demand prediction include traffic dataset, task record, application profile, and SLA. We also need road and cellular maps to generate the graphs discussed in Section V-A. Below we describe the datasets used in this study.

1) Road and Cellular Maps: The road map of Helsinki, more specifically, the area ranging from latitude 60.222306, longitude 24.858754 to latitude 60.142211, longitude 24.993980, is extracted from HERE Traffic API [25]. There exist 865 nodes in the target area, and the nodes are usually dense in the city downtown and sparse in the city suburb. We collect the locations and adjacency relationship of the nodes, as well as the traveling time between them to generate the road network. We set the VFN depot in Pasila, a major transportation hub in Helsinki, where the Pasila railway station, the second busiest station in Finland, and the terminal of many local buses are located. The cellular map is extracted from OpenCellID [27], which includes the latitudes and longitudes of all the base stations that support/will support 5 G in the area. We select the base stations provided by Elisa [28], the largest Finnish telecommunication operator, and set the communication range of each base station as 300 meters [29].

2) Traffic Dataset: Based on the road network, the traffic dataset is generated using Simulation of Urban MObility (SUMO) [30]. We generate the microscopic traffic flow (i.e., the location of each user in each ST) following the approach of *activity-based demand generation* [31], which creates the trips of the individual vehicles based on the description of the city population. To simulate different routes and schedules of school, employed, unemployed, and retired populations, we set the age distribution according to the data published by Statistics Finland in the year 2022 [32]. The working time of the inhabitants is distributed from 6:00 to 18:00. We set two peak hours during weekdays, one in the morning and the other in the afternoon (cf. Fig. 3(a)–(c)). We assume that the homes, schools, and working places are evenly distributed in the city. During each day, each inhabitant either commutes regularly between their school/working place and home (for school and employed travelers) or wanders in the city (for unemployed and retired travelers). The above-mentioned trips together form the microscopic traffic flow. Considering that there would be other random activities (e.g., entertainment after work), we set the *uniformRandomTraffic* parameter as 0.2, which indicates that 80% of the traffic flow is generated by regular commuting. After collecting the microscopic traffic data, we aggregate the macroscopic traffic flow in each region.

3) Task Record: During the experiment, each user generates a computing task per ST. The regional and global agents will aggregate this information and store them in the task record. We set uniform distribution for each service type as well as each latency requirement.

4) Application Profiles: We take four exemplary service types for a case study and set the input to be a 1280x720 driving footage video captured at 25 frames per second.

- *Object detection:* represents ultra-low latency and GPU-intensive service. It is implemented with YOLOv5s [33] trained on the COCO dataset [34]. It divides images into grids, where each cell in the grid is responsible for detecting objects within itself.
- *Semantic segmentation:* represents low latency and GPU-intensive service. It is implemented using Image Segmentation Keras [35] with the VGG-UNET model. The model is trained on the Cityscapes dataset [36]. It classifies each pixel in an image into one of the predefined classes using a fully convolutional network.
- *Lane detection:* represents ultra-low latency and CPU-intensive Service. It is implemented with OpenCV [37] in Python. It includes image processing techniques such as color selection, canny edge detection, region of interest selection, and Hough transform line detection.
- *Video transcoding:* represents low latency and CPU-moderate service. It is implemented using HandBrake video transcoder [38] with x265 video encoder and mp4 container.

We assume that each service type is associated with three latency requirements. The application profile is measured on a computer equipped with an Intel Core i7-7700 K CPU and an NVIDIA GeForce RTX 2080 Ti GPU.

5) Economic Model in SLA: In this work, the prices and costs are purely designed for comparison purposes and given in monetary units (MU). The price of a task c_{price} is listed in Table III, and the ratio between the penalty cost and price of a task α_{pen}, unit traveling cost c_{tra}, unit rental cost c_{rent}, and unit capacity cost c_{cap} are listed in Table IV. We set the prices following two rules: i) the price of the task will be higher when the CPU and GPU consumption is higher; ii) the price of the GPU consumption is higher than the CPU consumption.

B. Traffic Models

The duration of a capacity planning simulation is an episode. We assume that all the VFNs leave the depot at the beginning

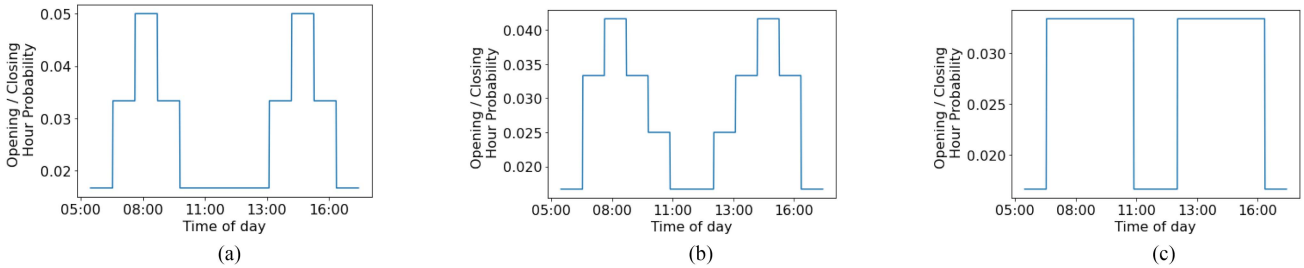

Fig. 3. Working hour distribution in different traffic models. (a) Working hour distribution in *STM*, *TM-OE*, and the first three days in *TM-SC*. (b) Working hour distribution in the middle three days in *TM-SC*. (c) Working hour distribution in the last four days days in *TM-SC*.

of each episode, and return to the same depot at the end of the episode [11], [12], [13], [14]; thus, the episode needs to be long enough to ensure VFNs would be able to travel back. During the experiment, we set the length of each episode as 1 d, each LT as 1 h, and each ST as 5 minutes. The lengths of LT and ST are set to be larger than the traveling time among the regions in the city and among nodes in a region, respectively, to ensure that the VFN can arrive at the destination before the next LT/ST starts. We generate the microscopic traffic data for ten working days with ST intervals. We consider three types of traffic models:

- *Stationary Traffic Model (STM):* The traffic pattern remains unchanged during the ten days.
- *Traffic Model with a Seasonal Change (TM-SC):* There is a gradual change in the distribution of the working hours (cf. Fig. 3(a)–(c)) due to a more flexible schedule, such as the hybrid working schedule during the pandemic period.
- *Traffic Model with an Occasional Event (TM-OE):* There is a sudden change in the demand magnitude (i.e., the total number of users doubled from 5000 to 10000) over the last four days due to an occasional event, such as an international football match.

Among them, *STM* is the baseline, while *TM-SC* and *TM-OE* represent the scenarios where temporary changes occur in the demand. We collect the macroscopic traffic flow in terms of the number of vehicles in each region. We use the first nine days to train the SARIMA model with LT intervals and use the last day for prediction.

C. Capacity Planning Strategies

Due to the uncertainty in traffic flow, the traffic flow prediction cannot always be accurate. The impacts of traffic flow prediction accuracy on capacity planning are analyzed through the following scenarios:

- *ODCP with Accurate Traffic (AT)*: Assume that the traffic flow prediction is 100% accurate and use the proposed model for capacity planning.
- *ODCP with Traffic Prediction With Feedback (TP-WF)*: Predict traffic flow using SARIMA, which takes the prediction errors in the previous time slot as feedback and uses the proposed model for capacity planning.
- *ODCP with Traffic Regression with No Feedback (TR-NF)*: Estimate the traffic flow based on Gaussian process regression [3] from the historical data and use the proposed model for capacity planning.

TP-WF is our proposal, *AT* is the baseline, and *TR-NF* is used for comparison.

We also compare ODCP with the following strategies presented in the literature:

- *Vehicle Routing method (VR):* Formulate the planning problem as VRPTW [13] and solve it using ILP.
- *Randomly Go and serve (RG):* The VFNs randomly travel among the regions and serve the demand that is within the same region (i.e., a naive approach).

D. Simulation Configurations

We design four sets of experiments. The simulation configurations for each experiment are detailed in Table IV. We repeat each experiment 10 times to get the mean values and statistical confidence intervals, and we use the same random seed for task generation in each comparison group. The first experiment aims to evaluate the techno-economic performance of different scenarios described in Section VI-C. We present the results in Sections VII-A and VII-B. Assuming the traffic flow prediction is correct, we want to see how different parameters affect the techno-economic performance of the capacity plan. The second experiment aims to find the impacts of the number of VFNs on performance. We change the number of VFNs from 20 to 70 and change the capacity cost from 0 to 50 while keeping other parameters fixed. The results are presented in Section VII-C. The third experiment aims to evaluate how the cost parameter values, including the unit traveling, rental, and penalty costs, will affect the performance of the capacity plan. We change them once per time while keeping other parameters unchanged. The results for this experiment are presented in Section VII-D. The last experiment aims to find the impacts of region size on performance. We change the regional number from 10 to 20, while keeping other parameters fixed. We present the results in Section VII-E.

E. Network Configurations

The latency experienced by the users consists of both computing and network latency. The experimental settings presented in the previous sections only consider computing latency since ODCP is focused on planning the computational capacity in

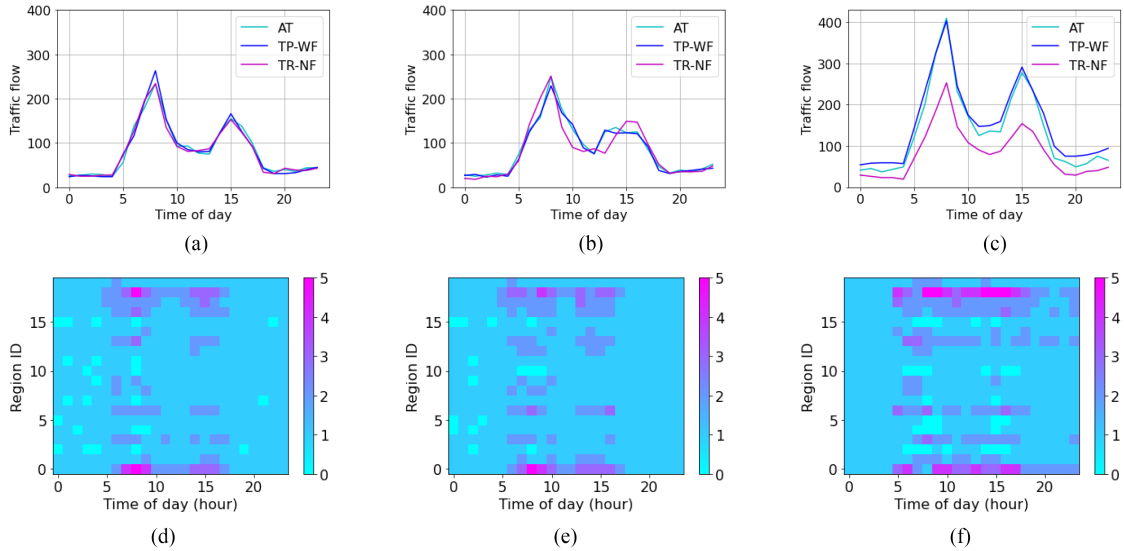

Fig. 4. (a)–(c): Traffic flow prediction results in terms of the number of the user vehicles in the city. (d)-(f): Global routing plans created by *ODCP with TP-WF* under different traffic scenarios, where the x-axis represents the 24 LTs, the y-axis represents the 20 regions, and the color bar represents the density of the active VFNs in each region. (a) F_t under *STM*. (b) F_t under *TM-SC*. (c) F_t under *TM-OE*. (d) Number of active VFNs in each region during each LT under *STM*, where mean=1.14, std=0.44, correlation with A_t=89%. (e) Number of active VFNs in each region during each LT under *TM-SC*, where mean=1.18, std=0.40, correlation with A_t=94%. (f) Number of active VFNs in each region during each LT under *TM-OE*, where mean=1.32, std=0.69, correlation with A_t=74%.

TABLE V
PHYSICAL AND NETWORK PARAMETERS IN THE NETWORK SIMULATION

Parameter	Value
Air interface	5G NR n78
Multiple access	OFDM/OFDMA
Duplex separation	TDD
Frequency band	3500 MHz
Channel bandwidth	100 MHz
Minimum required SINR	−10 dB
Number of base stations	12
Height of base stations	30 m
Transmission power	46 dB
Propagation model	Dominant path model
Simulation area	$1km \times 1km$
Simulation granularity	$1m \times 1m$
Simulation horizon	200 s
Simulation time slot	10 ms

the time granularity of ST. On the other hand, the network latency depends on the network and the applications using the bandwidth on the network. Therefore, we demonstrate an exemplary scenario using our capacity planning solution together with a real-time VFC simulator VFogSim [39] to measure the network latency experienced by the users in a region located in central Helsinki. We set the physical and network parameters according to Table V, and repeat each measurement 20 times to get the cumulative distribution function (CDF) of network latency distribution. We consider various traffic scenarios where the number of vehicles within a region ranges from 50 to 150 and compare the network latency experienced by the users (including queuing and migration latency) using the following strategies, assuming that the capacities of CFN and VFN are equal:

- *ODVFC:* The scenario created by ODCP.
- *CFN Only – Off-peak:* Deploying only CFNs according to the demand during off-peak hours.
- *CFN Only – Peak:* Deploying only CFNs according to the demand during peak hours.

The results are presented in Section VII-F.

VII. EXPERIMENTAL RESULT

This section presents the results of the experiments described in Sections VI-D and VI-E.

A. Traffic Flow Prediction

We compare the performance of our traffic flow prediction algorithm (i.e., *TP-WF*) with *TR-NF* in different traffic scenarios including *STM*, *TM-SC*, and *TM-OE*. As shown in Fig. 4(a)–(c), *TP-WF* outperforms *TR-NF* when compared with *AT* which represents the ground truth. *TP-NF* is not able to react as fast as *TP-WF* to the gradual change in case of *TM-SC*. In addition, *TP-NF* significantly underestimates the demand in the case of *TM-OE* where sudden change occurs. As shown in Table VI, the traffic flow prediction SMAPE using *TP-WF* is lower than the one using *TR-NF*, both in terms of mean values and confidence intervals. This is because *TP-WF* uses feedback from the previous LT to train the model and leads to higher prediction accuracy.

B. Capacity Planning

Fig. 4(d)–(f) present the global routing plans of *ODCP with TP-WF* in terms of the number of active VFNs routed to each region during each LT. Averaged over all the LTs, more active VFNs are needed for each region in the case of *TM-OE* than in the cases of *STM* and *TM-SC*, and the deviation of it is smaller in the case of *TM-SC* than in the case of *STM*. These results are consistent with the traffic patterns described in Section VI-B

TABLE VI
TECHNO-ECONOMIC PERFORMANCE METRICS WITH MEAN VALUES AND CONFIDENCE INTERVALS IN DIFFERENT SCENARIOS

Scenario	**Average SMAPE S - Mean value [95% confidence interval]**		
	STM	***TM-SC***	***TM-OE***
ODCP with AT	0%	0%	0%
ODCP with TP-WF	33.47% [27.86%, 39.09%]	30.19% [25.72%, 34.66%]	29.83% [26.14%, 33.52%]
ODCP with TR-NF	34.03% [27.44%, 40.62%]	41.74% [35.86%, 47.63%]	56.54% [52.29%, 60.79%]
VR with TP-WF	33.47% [27.86%, 39.09%]	30.19% [25.72%, 34.66%]	29.83% [26.14%, 33.52%]
RG	/	/	/
Scenario	**Average Service rate r_{ser} - Mean value [95% confidence interval]**		
	STM	***TM-SC***	***TM-OE***
ODCP with AT	93.65% [92.79%, 94.51%]	94.08% [93.35%, 94.81%]	88.51% [86.57%, 90.45%]
ODCP with TP-WF	92.22% [91.59%, 92.85%]	91.85% [91.21%, 92.49%]	87.95% [85.95%, 89.95%]
ODCP with TR-NF	92.07% [91.32%, 92.82%]	91.15% [90.33%, 91.97%]	79.23% [77.68%, 80.78%]
VR with TP-WF	91.40% [90.35%, 92.46%]	91.37% [90.45%, 92.29%]	85.84% [83.60%, 88.08%]
RG	71.82% [70.34%, 73.31%]	73.23% [71.63%, 74.83%]	70.57% [69.11%, 72.03%]
Scenario	**Overall Profit $\sum C_{pro}$ (MU/episode) - Mean value [95% confidence interval]**		
	STM	***TM-SC***	***TM-OE***
ODCP with AT	48665 [48298, 49032]	49968 [49646, 50291]	84079 [83312, 84847]
ODCP with TP-WF	47824 [47616, 48032]	48708 [48233, 49182]	82608 [82106, 83110]
ODCP with TR-NF	47517 [47299, 47735]	47687 [47588, 47787]	75754 [75376, 76132]
VR with TP-WF	36062 [35435, 36688]	41454 [41142, 41766]	16819 [15613, 18025]
RG	24981 [21828, 28133]	26874 [25655, 28093]	47916 [45253, 50578]

For example, 0% SMAPE represents that all the traffic flow predictions are accurate, and 100% service rate represents that all the tasks are served.

since the occasional event results in higher demand in *TM-OE*, and the gradual changing time schedule leads to more evenly distributed demand in *TM-SC*. Furthermore, there exist strong correlations between the traffic flow and the active VFNs utilized during each LT in all traffic scenarios. Among them, the correlation in *TM-OE* is slightly lower than in other cases because the limited number of VFNs cannot meet the peak demands in *TM-OE*.

Table VI shows the techno-economic performance of different scenarios described in Section VI-C, in terms of the average service rate and overall profit in an episode. First, we consider the *ODCP* scenario that we propose and evaluate the impacts of the traffic flow prediction accuracy on the techno-economic performance. *ODCP with AT* scenario assumes that the traffic is known beforehand; thus, it has the highest service rate and profit (i.e., the optimum values). *ODCP with TP-WF* scenario has a higher performance in terms of service rate and profit than the case of *ODCP with TR-NF* because a more accurate traffic flow prediction helps to adapt better to the temporary changes in demand.

Then, we compare the performance of *ODCP* with other capacity planning strategies. With the same traffic flow prediction method (i.e., *TP-WF*), the service rate of *VR* is similar to that in *ODCP*. However, the profit in *VR* is much lower than that in *ODCP*, especially under *TM-OE*. This is because *VR* plans to serve all the tasks, without considering the trade-off between the QoS and costs. When the demand exceeds the overall capacities of the VFNs, it will lead to higher penalty costs than *ODCP*, and the extreme amount of demand in *TM-OE* has intensified this situation. Finally, *RG* has an undesirable performance due to non-adaptation to the demand, with up to 20% lower service rate and 91% lower profit than *ODCP with TP-WF*. In case of 20 regions, *RG* uses 95% of VFNs throughout the day and stores the remaining 5% at the depot. Since not all VFNs are used during peak hours, the service rate is lower than other strategies; meanwhile, a high number of VFNs are utilized during off-peak hours, which leads to the waste of resources and costs. Consequently, *ODCP* outperforms others, especially when the demand shows high temporary changes.

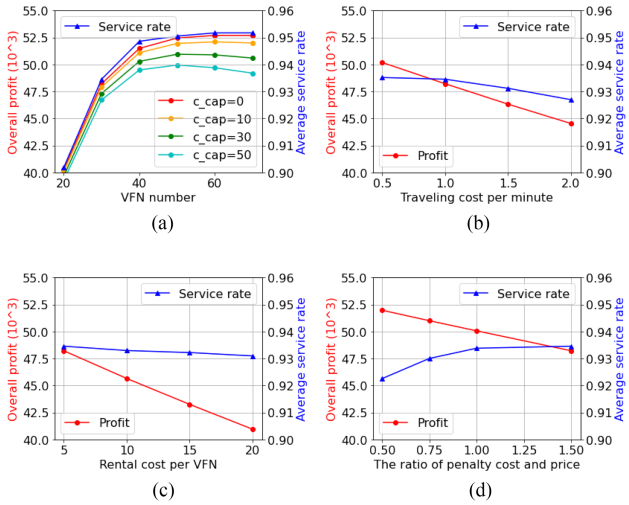

Fig. 5. Impact analysis of the deployed capacity and the economic model in use on the achieved profit and service rate. (a) Number of VFNs versus overall profit and service rate. (b) Unit traveling cost versus overall profit and service rate. (c) Unit rental cost versus overall profit and service rate. (d) Unit penalty cost versus overall profit and service rate.

C. Impacts of the Number of VFNs

Fig. 5(a) shows the techno-economic implications of the number of deployed VFNs. The increasing number of VFNs leads to service rate improvement and higher profit until they reach a saturation point. This is because the calculation of profit in Equation (13) considers not only the traveling and rental costs but also the revenue and penalty cost, which are tightly associated with the QoS received by the users. The objective function not only maximizes the economic gain but

also the QoS simultaneously. Meanwhile, the increase in QoS and profit gradually slows down, and they achieve the saturation point when the number of VFNs is large enough that there is sufficient capacity. For example, when there are 50 VFNs, all tasks are planned to be served, and a further increase in the number of VFNs lets more VFNs remain at the depot, which can no longer increase the service rate or profit. On the other hand, more VFNs also result in more installation and maintenance costs. When the number of VFNs becomes excessive, the profit will decrease, particularly when the capacity cost is high.

D. Impacts of the Economic Model

Fig. 5(b)–(d) show the techno-economic implications of the cost parameters values. The overall profit decreases linearly with the unit traveling, rental, and penalty costs, respectively, since these costs are independent of each other in (13). However, the cost parameter values have different effects on the average service rate. When the unit traveling cost becomes higher, the global and regional traveling costs will increase accordingly. As a result, the global and regional agents tend to move the VFNs as little as possible, which may lead to degraded QoS. When the traveling cost keeps increasing, the degradation in QoS becomes faster (cf. Fig. 5(b)). When the unit rental cost becomes higher, the rental cost increases, which forces the global agent to deploy as few VFNs as possible. Consequently, the service rate decreases steadily when the unit rental cost increases (cf. Fig. 5(c)). When the penalty cost per user increases, the regional planner is motivated to serve more tasks. However, the potential to increase the service rate in this way is limited by the number of VFNs available in the region, or by the total amount of tasks to be served. Therefore, the increase in service rate slows down when the unit penalty cost increases, and it will stop increasing when the unit penalty cost has reached a certain level (e.g., 1.5 times the value of the prices) (cf. Fig. 5(d)). Fig. 5(d) also shows the trade-off between the QoS and revenue, as mentioned in Section V-D. When the value of α_{pen} increases, the QoS has a higher weight than the revenue. Therefore, the service rate increases, and the profit decreases, and vice versa.

E. Impacts of Region Size

Fig. 6 reports how the selection of the region size affects the techno-economic performance of the capacity plan. We divide the same urban area into different numbers of regions. From Fig. 6(a), we can see when the region size becomes smaller, more VFNs will be used, which can result in a higher service rate but also higher costs. According to Fig. 6(b), during off-peak hours, using smaller regions can increase the service rate. In contrast, it is more profitable to use a larger region size during peak hours, as illustrated in Fig. 6(c). These are partially caused by the assumption that a VFN assigned to one region is only allowed to serve users within the same region. When the region size is large, a VFN in practice cannot serve all the users within the same region, due to the limited coverage of each cell. On the other hand, with a larger region size, there are fewer regional boundaries, and the VFN utilization rate can be higher. However, as shown in Fig. 6(d), the execution time of regional planning greatly increases with the region size during peak hours. When

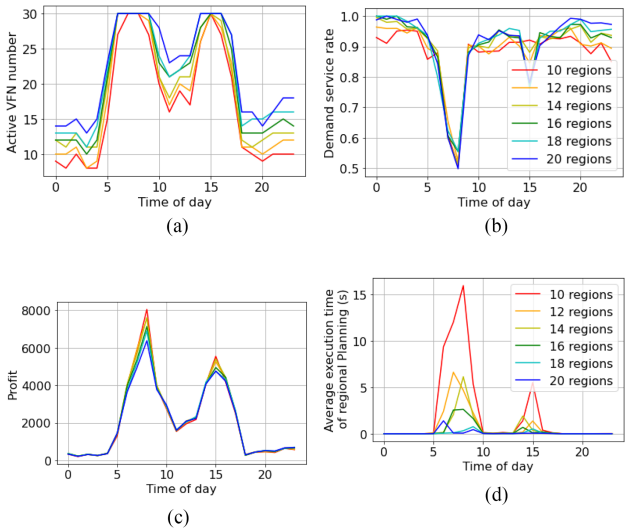

Fig. 6. Impact analysis of region size. (a) Impacts of region size on the VFN utilization. (b) Impacts of region size on the service rate. (c) Impacts of region size on the hourly profit. (d) Impacts of region size on the average execution time of regional capacity planning.

the region size becomes very large, for example, the whole city as an extreme case, the capacity plan would become the optimal solution in terms of profit, but there is no guarantee to get the optimal capacity plan within a feasible time.

In practice, we can set a threshold value for the service rate when selecting the region size. When the service rate reaches the threshold, there is no need to further decrease the region size. On the other hand, we can set a limit for the execution time of regional planning. When the execution time reaches the time limit, there is no need to further increase the region size.

F. Network Latency Experienced by the Users

As shown in Table VII, *ODVFC* has similar performance compared to *CFN only – Off-peak* when the network latency requirement is relaxed (e.g., 150 ms) or when the user density is low (e.g., 50 users). However, *ODVFC* has significantly better performance (i.e., up to 17%) compared to *CFN only – Off-peak* when the latency requirement is stringent (e.g., 50 ms) and when the users are densely distributed (e.g., 150 users). Therefore, ODVFC is suitable for serving latency-sensitive applications, especially in the urban environment. Compared to *CFN only – Peak*, *ODVFC* has similar network latency; by using a flexible number of VFNs, it can save 14 and 6 VFNs when there are 50 and 100 users, respectively; thus, it is a cost-efficient strategy.

VIII. Discussion

In this section, we analyze the time complexity and network overhead of ODCP. We also discuss the limitations and future directions.

A. Time Complexity

To evaluate the time complexity of ODCP, we measure the execution time of the model by running it on a core of an Intel

TABLE VII
CDF OF NETWORK LATENCY EXPERIENCED BY THE USERS USING DIFFERENT DEPLOYMENT STRATEGIES

Scenario	ODVFC				CFN Only - Off-peak				CFN Only - Peak			
	50ms	100ms	150ms	Capacity	50ms	100ms	150ms	Capacity	50ms	100ms	150ms	Capacity
50 users	82%	92%	98%	6 VFN + 12 CFN	80%	92%	98%	12 CFN	82%	92%	98%	32 CFN
100 users	78%	94%	96%	14 VFN + 12 CFN	73%	93%	96%	12 CFN	79%	94%	97%	32 CFN
150 users	77%	90%	95%	20 VFN + 12 CFN	60%	87%	94%	12 CFN	77%	90%	95%	32 CFN

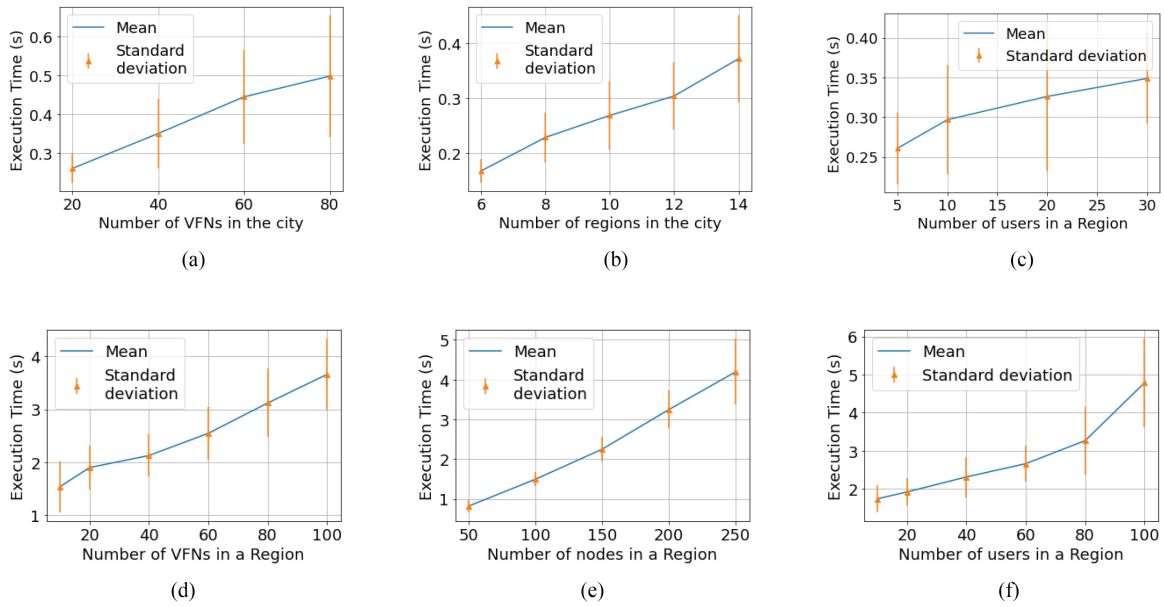

Fig. 7. Scalability analysis of capacity planning model. (a) Execution time of global planning versus the number of VFNs in the city $|I|$ ($|J \setminus \{0\}| = 10, |K| = 10$). (b) Execution time of global planning versus the number of regions in the city $|J \setminus \{0\}|$ ($|I| = 50, |K| = 10$). (c) Execution time of global planning versus the number of QoS levels $|K|$ ($|I| = 50, |J \setminus \{0\}| = 10$). (d) Execution time of regional planning versus the number of VFNs in a region $|L|$ ($|M| = 150, |N| = 50$). (e) Execution time of regional planning versus the number of nodes in a region $|M|$ ($|L| = 50, |N| = 50$). (f) Execution time of regional planning versus the number of users in a region $|N|$ ($|L| = 50, |M| = 150$).

i7-10510 U CPU. First, we evaluate the time complexity of the demand prediction model. If we use SARIMA for traffic flow prediction in a city with 20 regions, the average training and prediction time are both around 29 seconds (i.e., 58 seconds in total). With the same setting, the average modeling and prediction time using Gaussian process regression is about 44 seconds, which is more computationally efficient. However, as shown in Table VI, by using SARIMA (i.e., *TP-WF*), we can increase the profit by 9.05% and service rate by 8.72% compared to using Gaussian process regression (i.e., *TR-NF*) under *TM-OE*. Therefore, there is a trade-off between performance and computational time in the case of traffic flow prediction using training-based and regression-based methods.

Then, we evaluate the scalability of the capacity planning model. The *nextfit heuristic* can solve the workload estimation problem in polynomial time [23]. To measure the scalability of the global and regional planning models, we use random inputs with various sizes of data and compare the execution time. We repeat each measurement 20 times to get the mean values and standard derivations.

Fig. 7(a)–(c) show how the execution time of the global planning model changes with the number of VFNs in the city $|I|$, the number of regions $|J \setminus \{0\}|$, and the number of QoS levels $|K|$, respectively. It can be seen that the increasing rates are close to linear in the case of $|I|$ and $|J \setminus \{0\}|$, and slightly slower than linear in the case of $|K|$. When we set the QoS levels, we essentially divide the tasks in each region into $|K|$ groups and make batch decisions about serving them or not. In the optimal cases, when the value of $|K|$ is equal to or larger than the number of tasks, we will individually consider whether to serve each task or not, and the service rate will be the optimal value. However, this process will take a long time when the number of tasks is high. Therefore, similar to determining the regional size, we can set a threshold value for the service rate when selecting $|K|$. When the service rate reaches the threshold, there is no need to further increase the number of levels.

Fig. 7(d)–(f) show how the execution time of the regional planning model changes with the number of VFNs $|L|$, the number of nodes $|M|$, and the number of tasks $|N|$ in a region, respectively. It can be seen that the increasing rates are close to linear in the case of $|L|$ and $|M|$, and slightly faster than linear in the case of $|N|$. By using two-phase capacity planning, we greatly reduce the scale of $|L|$, $|M|$, and $|N|$. Therefore, the execution time is significantly shortened compared to single-phase capacity planning.

B. Network Overhead

We study the impact of the proposed method on network data overhead by measuring the data exchange among the users, regional agents, and the global agent with the network configuration described in Section VI-E. The information that the regional agents need to collect from the users includes the vehicle ID, vehicle dynamics per second (e.g., location, speed, and direction), service type, and latency requirement, which is collected every ST. We measure the data overhead in the regional agent for 10 minutes. With 150 users in the peak scenario, the data overhead is around 3500 kB. Meanwhile, the information that the global agent needs to collect from the regional agents includes the region ID and the vehicle ID, service type, and latency requirement of the users located in the region. Such data, which is collected every LT, is usually small (i.e., less than 1 kB for each user). The data overhead is around 400 kB for the city in this case.

C. Limitations and Future Directions

In this work, we assume that the VFNs have the same configuration and can serve all the requests. In the future, we will explore how the heterogeneity of the VFNs would affect the techno-economic performance of the capacity plan. In addition, we assume that the traveling time between two regions is fixed (i.e., depending purely on the distance). In our future work, we will consider dynamic traveling time (i.e., depending both on the distance and the real-time traffic condition), and we will investigate the impacts of travel delays on the achievable QoS. While the current ODCP assumes that the VFNs use 5 G NR V2N [40] to provide computing service to the users within the one-hop communication range, we will also consider V2V communication [41] and multi-hop collaboration in the future. Finally, this article addresses the capacity planning problem from season to ST. In the future, we will further shorten the time granularity and study the problem from LT to several seconds.

IX. Conclusion

In this work, we propose ODCP, a data-driven capacity planning framework for ODVFC. While the existing long-term capacity planning solutions cannot timely adapt to the temporary changes in demand, we formulate and solve the two-phase capacity planning problem using ILP, which shortens the execution time of the capacity planning model by enabling parallel capacity planning at the regional level. Through a city-scale simulation, we evaluate the service rate and profit of different capacity planning solutions under various traffic scenarios. We also compare the network latency experienced by the users among different fog node deployment solutions. The experimental results show that the proposed capacity planning solution achieves high performance in both service rate and profit. Furthermore, if a smaller region size is used, it will lead to a higher service rate during off-peak hours; if a larger region size is used, it will lead to higher profit during peak hours.

Overall, this article focuses on short-term capacity planning, but we can combine this article with our previous work [3] for both short-term and long-term capacity planning. More specifically, we can derive the regular demand based on the regression of daily traffic flow and fulfill it by CFN placement and bus scheduling following the approach proposed in [3]. Afterward, we can estimate the temporary demand based on the prediction of hourly traffic flow and fill the capacity gap by ODCP.

References

[1] C. Zhu, G. Pastor, Y. Xiao, and A. Ylä-Jääski, "Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 58–63, Oct. 2018.

[2] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2017, pp. 6–9.

[3] W. Mao, O. U. Akgul, A. Mehrabi, B. Cho, Y. Xiao, and A. Ylä-Jääski, "Data-driven capacity planning for vehicular fog computing," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13179–13194, Aug. 2022.

[4] I. Stypsanelli, O. Brun, S. Medjiah, and B. J. Prabhu, "Capacity planning of fog computing infrastructures under probabilistic delay guarantees," in *Proc. IEEE Int. Conf. Fog Comput.*, 2019, pp. 185–194.

[5] M. Noreikis, Y. Xiao, and A. Ylä-Jääski, "QoS-oriented capacity planning for edge computing," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[6] M. Noreikis, Y. Xiao, and Y. Jiang, "Edge capacity planning for real time compute-intensive applications," in *Proc. IEEE Int. Conf. Fog Comput.*, 2019, pp. 175–184.

[7] M. M. Hussain, M. Alam, and M. M. Beg, "Vehicular fog computing-planning and design," *Procedia Comput. Sci.*, vol. 167, pp. 2570–2580, 2020.

[8] G. Premsankar, B. Ghaddar, M. Di Francesco, and R. Verago, "Efficient placement of edge computing devices for vehicular applications in smart cities," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp.*, 2018, pp. 1–9.

[9] Q. Wu, Y. Zhao, Q. Fan, P. Fan, J. Wang, and C. Zhang, "Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 66–81, Jan. 2023.

[10] Q. Wu, S. Shi, Z. Wan, Q. Fan, P. Fan, and C. Zhang, "Towards V2I age-aware fairness access: A DQN based intelligent vehicular node training and test method," *Chin. J. Electron.*, vol. 32, pp. 1–15, 2022.

[11] B. Peng, J. Wang, and Z. Zhang, "A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems," in *Proc. Artif. Intell. Algorithms Appl.: 11th Int. Symp.* 2020, pp. 636–650.

[12] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takác, "Deep reinforcement learning for solving the vehicle routing problem," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9861–9871.

[13] K. Prag, M. Woolway, and B. A. Jacobs, "Optimising the vehicle routing problem with time windows under standardised metrics," in *Proc. IEEE 6th Int. Conf. Soft Comput. Mach. Intell.*, 2019, pp. 111–115.

[14] K. Zhang, F. He, Z. Zhang, and M. Li, "Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach," *Transp. Res. Part C: Emerg. Technol.*, vol. 121, 2020, Art. no. 102861.

[15] O. Bekkouche, T. Taleb, M. Bagaa, and K. Samdanis, "Edge cloud resource-aware flight planning for unmanned aerial vehicles," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2019, pp. 1–7.

[16] C. Zhu et al., "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4150–4161, Jun. 2019.

[17] Ramadiani, D. Bukhori, Azainil, and N. Dengen, "Floyd-Warshall algorithm to determine the shortest path based on Android," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 144, 2018, Art. no. 012019, doi: 10.1088/1755-1315/144/1/012019.

[18] L. Elefteriadou, *An Introduction to Traffic Flow Theory* (Springer Optimization and Its Applications Series), vol. 84. Berlin, Germany: Springer, Jan. 2014.

[19] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, p. 2, 2014.

[20] N. Ranaldo and E. Zimeo, "Capacity–driven utility model for service level agreement negotiation of cloud services," *Future Gener. Comput. Syst.*, vol. 55, pp. 186–199, 2015.

[21] S. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transport Res. Rev.*, vol. 7, no. 21, pp. 1–9, 2015.

[22] V. A. Profillidis and G. N. Botzoris, "Trend projection and time series methods," in *Modeling of Transport Demand*, V. A. Profillidis and G. N. Botzoris, Eds., Amsterdam, TheNetherlands: Elsevier, 2019, ch. 6, pp. 225–270.
[23] H. Fujiwara, R. Adachi, and H. Yamamoto, "Algorithm nextfit for the bin packing problem," *Formalized Math.*, vol. 29, no. 3, pp. 141–151, 2021, doi: 10.2478/forma-2021-0014.
[24] T. Kloks and M. Xiao, *A Guide to Graph Algorithms*. Singapore: Springer, Mar. 2022.
[25] HERE WeGo Contributors, "Here traffic API," 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://developer.here.com/documentation/traffic
[26] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2022, Accessed: Mar. 22, 2022. [Online]. Available: http://www.gurobi.com
[27] OpenCellid contributors, "Opencellid," 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://opencellid.org/
[28] Elisa Corporation, "Elisa," 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://elisa.fi/
[29] A. M. Niknejad and G. Hueber, *Millimeter-Wave Circuits for 5G and Radar* (The Cambridge RF and Microwave Engineering Series). Cambridge, U.K.: Cambridge Univ. Press, 2019.
[30] P. A. Lopez et al., "Microscopic traffic simulation using sumo," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2575–2582. [Online]. Available: https://elib.dlr.de/124092/
[31] L. Codeca, R. Frank, S. Faye, and T. Engel, "Luxembourg sumo traffic (LUST) scenario: Traffic demand evaluation," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 52–63, Summer 2017.
[32] Statistics Finland, "Population and society," 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://www.stat.fi/tup/suoluk/suoluk_vaesto_en.html
[33] G. Jocher et al., "ultralytics/yolov5: V3.1 - bug fixes and performance improvements," Oct. 2020, doi: 10.5281/zenodo.4154370.
[34] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312
[35] D. Gupta and R. J. wala, "Image segmentation Keras: Implementation of segnet, FCN, UNet, PSPNet and other models in Keras," Dec. 2020.
[36] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding,"*CoRR*, vol. abs/1604.01685, 2016. [Online]. Available: http://arxiv.org/abs/1604.01685
[37] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, 2000.
[38] HandBrake Contributors, "HandBrake," 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://handbrake.fr/
[39] O. U. Akgul, W. Mao, B. Cho, and Y. Xiao, "VFogSim: A data-driven platform for simulating vehicular fog computing environment," *IEEE Syst. J.*, vol. 17, no. 3, pp. 5002–5013, Sep. 2023.
[40] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications," *IEEE Veh. Technol. Mag.*, vol. 12, no. 4, pp. 30–39, Dec. 2017.
[41] B. Cho and Y. Xiao, "Learning-based decentralized offloading decision making in an adversarial environment," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11308–11323, Nov. 2021.

Wencan Mao received the B.E. degree in vehicle engineering from the Wuhan University of Technology, Wuhan, China, in 2017, and the M.S. degree in mechanical engineering in 2019 from Aalto University, Espoo, Finland, where she is currently working toward the Ph.D. degree with the Department of Information and Communications Engineering. Her research interests include edge computing, Internet of Things, vehicular networking, resource allocation, and capacity planning.

Ozgur Umut Akgul received the B.S. degree in electronics enginnering and electrical engineering and the M.S. degree in computer engineering from Istanbul Technical University, Istanbul, Turkey, in 2011 and 2014, respectively, and the Ph.D. degree in information technology from Politecnico di Milano, Milan, Italy, in 2019. He is currently a Postdoctoral Researcher with the Department of Communications and Networking, Aalto University, Espoo, Finland. His research interests include optimization models, mathematical programming, and machine learning, with the application of these techniques to wireless network problems, such as wireless resource allocation, edge computing, anticipatory network optimization, infrastructure and resource sharing, and network slicing.

Byungjin Cho received the Doctoral degree in communications engineering from Aalto University, Espoo, Finland, in 2016. He is currently a Postdoctoral Researcher with the Department of Communications and Networking, Aalto University. His research focuses on resource managements in networked systems using algorithmic decision theory.

Yu Xiao (Member, IEEE) received the Doctoral degree in computer science from Aalto University, Espoo, Finland, in 2012. She is currently an Associate Professor with the Department of Information and Communications Engineering, Aalto University. Her research interests include edge computing, wearable sensing, and extended reality.

Antti Ylä-Jääski received the Ph.D. degree from ETH Zürich, Zürich, Switzerland, in 1993. From 1994 to 2009, he was with Nokia, Espoo, Finland, in several research and research management positions, with a focus on future Internet, mobile networks, applications, services, and service architectures. Since 2004, he has been a Tenured Professor with the Department of Computer Science, Aalto University, Espoo. His research interests include mobile cloud computing, mobile multimedia systems, pervasive computing and communications, indoor positioning and navigation, energy efficient communications and computing, and Internet of Things.