Migration-Aware Optimized Resource Allocation in B5G Edge Networks

Tadeus Prastowo, Ayub Shah, Luigi Palopoli, Roberto Passerone
Department of Information Engineering and Computer Science
University of Trento, Trento, Italy 38123
Email: first.last@unitn.it

Giuseppe Piro
DEI
Politecnico di Bari, Bari, Italy
Email: giuseppe.piro@poliba.it

***Abstract*—The fifth-generation and beyond (B5G) communication systems are evolving for computation-intensive and communication-sensitive applications with diverse Quality-of-Service (QoS) requirements on processing, bandwidth, latency, and reliability. This work focuses on an ultra-dense edge network with Multi-access Edge Computing (MEC) facilities, serving agents that execute their tasks by touring the cells. Specifically, we propose a novel methodology for optimally and flexibly managing task offloading in the context of heterogeneous computing and communication services required by real-time robotic applications. Differing from many related work, the proposed approach takes the number of admitted service migrations and the QoS upper and lower bounds as binding constraints. We model the QoS evolution based on the agent positions, the MEC servers serving the agents, the QoS requirements, the communication capabilities in the edge network, and the computing capabilities of the servers. The model is formalized as a mixed-integer linear program (MILP) to obtain an optimal schedule for the service migrations and communication and computation bandwidth allocation. Experimental results show that the approach outperforms baseline approaches and can scale to large deployments.**

***Index Terms*—B5G task offloading, migration cost, MILP, computer simulation.**

I. Introduction

The research on the next generation of mobile networks is chasing the ambitious objective to jointly support, within a flexible and powerful communication and computing infrastructure, a very large number of heterogeneous services [1], [2]. In this context, an effective management of task offloading is crucial to deliver high-quality services [3], hence the need for the optimal management of computing and communication resources at the network edge.

This important topic has been investigated extensively. Many papers [4]–[18] focus on static scenarios, where user mobility is not explicitly taken into account. Other papers, e.g., [19]–[31], explicitly account for user mobility. They propose optimization algorithms or iterative procedures to optimize the allocation of servers to the tasks while respecting energy, latency, and communication delay constraints. Task offloading in mobile scenarios is certainly a complex problem. While moving across network attachment points, mobile users should be served by different servers at the edge, whose position and capabilities can satisfy the expected service level. The overall management is extremely dynamic. In case the offered service is implemented through dedicated VMs (virtual machines) or containers, such a dynamic scenario requires frequent migration operations [20]–[22], [24], [25], [27].

Unfortunately, state-of-the-art solutions have two main problems. First, many B5G use-cases (e.g., indoor robotic applications) will be enabled through ultra-dense cell deployments, where a limited area is covered by numerous base stations. The frequent handovers triggered by users' mobility would produce an erratic management of task offloading, resulting in excessive migrations of VMs or containers deployed at the network edge. Unfortunately, to our knowledge no solution in the literature considers the number of migrations as a system variable to be optimized (e.g., minimized). Second, while in real-time systems the QoS of a service can be related to its computation bandwidth [32], in edge computing the degrees of freedom that affect the QoS are greater and include at least the computation and communication bandwidth and the physical displacement of the VM delivering the service. Hence, the choice of these parameters can make for an adaptive QoS level between some minimum and maximum. Despite the evident benefits of an adaptive QoS, the flexible provisioning of advanced services in dynamic network conditions is quite ignored in the literature.

To close the gap, we propose a novel methodology to manage B5G task offloading optimally and flexibly in the context of real-time applications, which are represented well in the robotic domain: AGVs (automated guided vehicles) touring a large logistic facility using the edge facilities for computation-intensive tasks. Network attachment points offer wireless connectivity to agents (e.g., AGVs) that require heterogeneous services. The attachment points are connected to an edge network with computing capabilities provided by MEC servers. The agents then connect to one of the available MEC servers through edge links with fixed communication capabilities. In the robotic domain, it is reasonable to assume that agent mobility is predictable. Hence, the QoS dynamics is modeled in terms of the agent and VM positions, the service requirements, the link communication capabilities, the MEC server computing capabilities, and other parameters. The model is then translated into an MILP to get optimal VM positions and their optimal communication and computation bandwidth. Unlike the current state of the art, the QoS is a function of computing and communication requirements, end-to-end communication latency, and migration cost. Further-

more, while the computing and communication capabilities are constraints in the state of the art, the proposed novel approach is characterized by taking as constraints the number of admitted VM migrations and the QoS lower and upper bounds expected by the agents.

Therefore, we make three main contributions in this paper: 1) Section II presents the model of the QoS dynamic, 2) Section III translates the model into an MILP, and 3) Section IV evaluates the MILP effectiveness and scalability in various scenarios. Section V outlines our conclusions and future work.

II. System Model

B5G network dynamics are viewed at discrete times t with the network edge being required by every mobile agent $\mathcal{A}_i \in \mathbb{A}$ to deliver a number of real-time services by running their VMs $\mathbb{M}_i = \{\mathcal{M}_{i,1}, \ldots, \mathcal{M}_{i,m_i}\}$. We denote the set of all VMs with $\mathbb{M} = \bigcup_{\mathcal{A}_i \in \mathbb{A}} \mathbb{M}_i$. While $\mathcal{A}_i$ uses a number of VMs, $\mathcal{M}_{i,j}$ serves exactly one $\mathcal{A}_i$ and communicates with no other VM in $\mathbb{M}$.

Every cell $\mathcal{C}_c$ in the network $\mathbb{G}$ hosts a MEC server that has a fixed computation capacity Φ_c. Similarly, the links that connect every cell with the edge network have a fixed communication capacity Ψ_c, as well. Thanks to the edge network, a VM can migrate from the MEC server in some cell $\mathcal{C}_c$ to the MEC server in another cell $\mathcal{C}_{c'}$ to maximize its QoS. We use $\mu_{i,j,c,t}$ (resp. $\rho_{i,c,t}$) to denote the location of $\mathcal{M}_{i,j}$ (resp. $\mathcal{A}_i$) in terms of the network cells $\mathbb{G}$ such that $\mu_{i,j,c,t} = 1$ (resp. $\rho_{i,c,t} = 1$) if the MEC server that hosts $\mathcal{M}_{i,j}$ is in cell $\mathcal{C}_c$ (resp. $\mathcal{A}_i$ connects to the network by attaching itself wirelessly to cell $\mathcal{C}_c$) and $\mu_{i,j,c,t} = 0$ (resp. $\rho_{i,c,t} = 0$) otherwise. While the position of the agents $\rho_{i,c,t}$ is assumed to be known a priori, the location of the VMs $\mu_{i,j,c,t}$ is to be decided optimally.

Initially at time $t = 0$, no migration occurs, every $\mathcal{M}_{i,j}$ runs on exactly one MEC server in some cell $\mathcal{C}_c$, and every $\mathcal{A}_i$ is hosted by exactly one cell $\mathcal{C}_{c'}$, possibly $c = c'$. At any later time $t' > 0$, while migration may occur, every $\mathcal{M}_{i,j}$ still runs on exactly one possibly-different MEC server and every $\mathcal{A}_i$ is still hosted by exactly one possibly-different cell. It follows that (1) holds for every $\mathcal{M}_{i,j}$ and every $\mathcal{A}_i$ at any time t.

$$\begin{aligned} \sum_{\mathcal{C}_c \in \mathbb{G}} \mu_{i,j,c,t} &= 1 \\ \sum_{\mathcal{C}_c \in \mathbb{G}} \rho_{i,c,t} &= 1 \end{aligned} \tag{1}$$

Migrating $\mathcal{M}_{i,j}$ is assumed to consume negligible bandwidth and take one time unit with cost $\mathcal{E}_{i,j}$ and with the total migrations that can take place at any time t' being limited by some constant M.[1] We use $h_{i,j,t'}$ to denote the migration of $\mathcal{M}_{i,j}$ at time t' such that $h_{i,j,t'} = 1$ if $\mathcal{M}_{i,j}$ was at time $t' - 1$ not hosted by the MEC server in $\mathcal{C}_c$ but at time t' is hosted by the server in $\mathcal{C}_c$, else $h_{i,j,t'} = 0$. Hence, (2) and (3) hold.

$$h_{i,j,t} = \begin{cases} 0 & \text{, if } t = 0 \\ 1 - \sum_{\mathcal{C}_c \in \mathbb{G}} \mu_{i,j,c,t-1} \cdot \mu_{i,j,c,t} & \text{, otherwise} \end{cases} \tag{2}$$

$$\sum_{\mathcal{M}_{i,j} \in \mathbb{M}} h_{i,j,t} \leq M \tag{3}$$

Beside $\mu_{i,j,c,t}$, the computation bandwidth, denoted $\alpha_{i,j,t}$, and the communication bandwidth, denoted $\beta_{i,j,t}$, given to $\mathcal{M}_{i,j}$ at any time t are to be decided optimally as well subject to (4) and (5) with $\alpha_{i,j}^{\min}$ and $\beta_{i,j}^{\min}$ being the least bandwidth below which the service cannot be provided and $\alpha_{i,j}^{\max}$ and $\beta_{i,j}^{\max}$ being the greatest bandwidth above which further allocations yield no benefits on the QoS.

$$\alpha_{i,j}^{\min} \leq \alpha_{i,j,t} \leq \alpha_{i,j}^{\max} \tag{4}$$

$$\beta_{i,j}^{\min} \leq \beta_{i,j,t} \leq \beta_{i,j}^{\max} \tag{5}$$

On the other hand, the total computation and communication bandwidth allocated to the VMs hosted by a MEC server cannot exceed the maximum computation and link capacities of the server. It follows that the computation bandwidth $\phi_{i,j,c,t}$ allocated to $\mathcal{M}_{i,j}$ by the MEC server in a cell $\mathcal{C}_c$ at time t has to satisfy (6) and (7). Note that if the MEC server in $\mathcal{C}_c$ executes the VM $\mathcal{M}_{i,j}$ at time t, then $\phi_{i,j,c,t} > 0$ but $\phi_{i,j,c',t} = 0$ for the MEC servers in all other cells $\mathcal{C}_{c'}$. Furthermore, the communication bandwidth $\psi_{i,j,c,t}$ of the link that at time t connects an agent $\mathcal{A}_i$ to its VM $\mathcal{M}_{i,j}$ running on the MEC server in $\mathcal{C}_c$ depends on whether $\mu_{i,j,c,t} = \rho_{i,c,t}$. If it is, only the $\mathcal{A}_i \leftrightarrow \mathcal{M}_{i,j}$ link in $\mathcal{C}_c$ allocates some communication bandwidth. Else, the communication bandwidth is also allocated by all other links involved in routing the $\mathcal{A}_i \leftrightarrow \mathcal{M}_{i,j}$ communication. With respect to the routing, we assume that at any time t there exists exactly one logical loop-free cycle-free bidirectional route $\mathcal{R}_{c_s,c_e,t} \in \wp(\mathbb{G})$ from $\mathcal{C}_{c_s}$ to $\mathcal{C}_{c_e}$ where $\wp(\mathbb{G})$ is the power set of $\mathbb{G}$. Clearly, $\{\mathcal{C}_{c_s}, \mathcal{C}_{c_e}\} \subseteq \mathcal{R}_{c_s,c_e,t}$ (i.e., the start and end cells are on the route), $\mathcal{R}_{c_s,c_e,t} = \mathcal{R}_{c_e,c_s,t}$ (i.e., the route is bidirectional), and $\mathcal{R}_{c,c,t}$ refers to the agent-connecting wireless link. The binary value $\eta_{c,c_s,i,t}$ is used to denote whether at time t, cell $\mathcal{C}_c$ is on the route $\mathcal{R}_{c_s,c_e,t}$ between some cell $\mathcal{C}_{c_s}$ and the cell $\mathcal{C}_{c_e}$ where the agent $\mathcal{A}_i$ is. Specifically, if $\rho_{i,c_e,t} = 1$ and $\mathcal{C}_c \in \mathcal{R}_{c_s,c_e,t}$, then $\eta_{c,c_s,i,t} = 1$ (i.e., $\mathcal{C}_c$ is in $\mathcal{R}_{c_s,c_e,t}$ to route any bidirectional communication between some VM $\mathcal{M}_{i,j}$ running on the MEC server in $\mathcal{C}_{c_s}$ and its agent $\mathcal{A}_i$ in $\mathcal{C}_{c_e}$), else $\eta_{c,c_s,i,t} = 0$. The communication bandwidth allocated to each VM $\psi_{i,j,c,t}$ then has to satisfy (8) and (9) (i.e., the total allocated bandwidth cannot exceed the available one). Clearly, (7)/(9) fails for some VM $\mathcal{M}_{i,j}$ whenever at time t the MEC server in some cell $\mathcal{C}_c$ hosts too many VMs such that the sum of $\alpha_{i,j}^{\min}$/$\beta_{i,j}^{\min}$ of all the VMs $\mathcal{M}_{i,j}$ hosted by the server in $\mathcal{C}_c$ exceeds the respective capacity Φ_c/Ψ_c of the server in order to satisfy (4)/(5). In case of failing (7), migrating $\mathcal{M}_{i,j}$ to another server in $\mathcal{C}_{c'}$ may respect both (7) and (4) at time t. However, in case of failing (9), the migration cannot respect both (9) and (5) in

[1] This is realistic by choosing a suitable time unit and by reserving some computation and communication bandwidth, if not dedicating processor cores and network links, to support M concurrent migrations.

$\mathcal{C}_{c'}$ as $\mathcal{M}_{i,j}$ still needs at least $\beta_{i,j}^{\min}$ of the link capacity at $\mathcal{C}_c$ to communicate with $\mathcal{A}_i$, and hence, it follows that $\sum_{\mathcal{A}_i \in \mathbb{A}} \rho_{i,c,t} \left(\sum_{\mathcal{M}_{i,j} \in \mathbb{M}_i} \beta_{i,j}^{\min} \right) \leq \Psi_c$ for every cell $\mathcal{C}_c$ that hosts some agent at time t.

$$\phi_{i,j,c,t} = \mu_{i,j,c,t} \cdot \alpha_{i,j,t} \tag{6}$$

$$\sum_{\mathcal{M}_{i,j} \in \mathbb{M}} \phi_{i,j,c,t} \leq \Phi_c \tag{7}$$

$$\psi_{i,j,c,t} = \sum_{\mathcal{C}_{c_s} \in \mathbb{G}} \eta_{c,c_s,i,t} \cdot \mu_{i,j,c_s,t} \cdot \beta_{i,j,t} \tag{8}$$

$$\sum_{\mathcal{M}_{i,j} \in \mathbb{M}} \psi_{i,j,c,t} \leq \Psi_c \tag{9}$$

Another important aspect of the model is the end-to-end communication latency. Each route $\mathcal{R}_{c_s,c_e,t}$ has some end-to-end latency $\Lambda_{c_s,c_e,t}$ such that $\Lambda_{c_s,c_e,t} = \Lambda_{c_e,c_s,t}$ due to bidirectionality. The end-to-end latency $\lambda_{i,j,t}$ between $\mathcal{M}_{i,j}$ and $\mathcal{A}_i$ at t is then given by (10) where $\tau_{c_s,i,t}$ is the $\Lambda_{c_s,c_e,t}$ of the $\mathcal{C}_{c_e}$ that satisfies $\rho_{i,c_e,t} = 1$.[2] The end-to-end latency $\lambda_{i,j,t}$ is then required by (11) to satisfy some upper-bound beyond which $\mathcal{A}_i$ would fail in executing its real-time tasks.

$$\lambda_{i,j,t} = \sum_{\mathcal{C}_{c_s} \in \mathbb{G}} \mu_{i,j,c_s,t} \cdot \tau_{c_s,i,t} \tag{10}$$

$$\lambda_{i,j,t} \leq \lambda_{i,j}^{\max} \tag{11}$$

We now use the framework just shown to define the total QoS function. Specifically, we define in (12) the total QoS experienced by $\mathcal{M}_{i,j}$ at t in terms of a quality function $\mathcal{Q}_{i,j}^{+}$ and a migration cost $\mathcal{Q}_{i,j,t}^{-}$. The $\mathcal{Q}_{i,j}^{+}$ quantifies the QoS of $\mathcal{M}_{i,j}$ as a function of the given bandwidth for computation $\alpha_{i,j,t}$ and communication $\beta_{i,j,t}$ and the experienced end-to-end latency $\lambda_{i,j,t}$ in a manner that is specific to $\mathcal{M}_{i,j}$. The $\mathcal{Q}_{i,j,t}^{-}$, however, follows from the previous definitions and is given in (13) where the negative sign is justified by the migration's adverse effect on the QoS.

$$\mathcal{Q}_{i,j,t} = \mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t}) + \mathcal{Q}_{i,j,t}^{-} \tag{12}$$

$$\mathcal{Q}_{i,j,t}^{-} = -\mathcal{E}_{i,j} \cdot h_{i,j,t} \tag{13}$$

III. Formulation of the Optimization Problem

The model is formulated as a mixed-integer program (MIP) optimization problem in terms of (12), which is the different QoS experienced by every VM. Specifically, the MIP formulation maximizes (14) subject to (1)–(11) over a time horizon $\mathbb{H} \in (\wp(\mathbb{N}^+ \cup \{0\}) \setminus \{\emptyset\})$, which is a finite subset of the naturals.

$$\sum_{t \in \mathbb{H}} \sum_{\mathcal{M}_{i,j} \in \mathbb{M}} \mathcal{Q}_{i,j,t} \tag{14}$$

Hence, due to the time horizon, the symbols t and t' in the MIP formulation refer to the members of $\mathbb{H}$ with $t \in \mathbb{H}$ and $t' \in (\mathbb{H} \setminus \{\min \mathbb{H}\})$, while the time $t = 0$ refers to the time $t_0 = \min \mathbb{H}$ in the MIP formulation. We now discuss how to turn our non-linear MIP into a mixed-integer linear program (MILP) by getting rid of its non-linear forms: the products of decision variables in (2), (6), and (8) and the expression $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t})$ in (12). To that end, we adapt established techniques [33] as shown next.

To turn (2), which involves the product of two binary variables, into a linear form, we introduce the binary decision variable $z_{i,j,c,t'} \in \{0,1\}$ with $t' \in \mathbb{N}^+$ (positive naturals) and require that (15)–(17) hold. Since it can be shown by means of a truth table that $z_{i,j,c,t'} = \mu_{i,j,c,t'-1} \cdot \mu_{i,j,c,t'}$ if and only if (15)–(17) hold, the MILP formulation replaces (2) with (15)–(18).

$$z_{i,j,c,t'} \leq \mu_{i,j,c,t'-1} \tag{15}$$

$$z_{i,j,c,t'} \leq \mu_{i,j,c,t'} \tag{16}$$

$$z_{i,j,c,t'} \geq \mu_{i,j,c,t'-1} + \mu_{i,j,c,t'} - 1 \tag{17}$$

To turn (6), which involves the product of a real and a binary variables, into a linear form, we define A to be $\max_{\mathcal{M}_{i,j} \in \mathbb{M}} \alpha_{i,j}^{\max}$ and require that (19)–(21) hold. Since it can be shown that $\phi_{i,j,c,t} = \mu_{i,j,c,t} \cdot \alpha_{i,j,t}$ if and only if (19)–(21) hold, the MILP formulation replaces (6) with (19)–(21). Similarly for (8), we define B to be $\max_{\mathcal{M}_{i,j} \in \mathbb{M}} \beta_{i,j}^{\max}$ and introduce the real decision variable $\omega_{i,j,c_s,t}$ while requiring that (19)–(21) hold when $\omega_{i,j,c_s,t}$, B, $\mu_{i,j,c_s,t}$, and $\beta_{i,j,t}$ replace $\phi_{i,j,c,t}$, A, $\mu_{i,j,c,t}$, and $\alpha_{i,j,t}$, respectively. The MILP formulation then replaces (8) with $\psi_{i,j,c,t} = \sum_{\mathcal{C}_{c_s} \in \mathbb{G}} \eta_{c,c_s,i,t} \cdot \omega_{i,j,c_s,t}$ and the additional constraints.

$$h_{i,j,t} = \begin{cases} 0 & \text{, if } t = 0 \\ 1 - \sum_{\mathcal{C}_c \in \mathbb{G}} z_{i,j,c,t} & \text{, otherwise} \end{cases} \tag{18}$$

$$0 \leq \phi_{i,j,c,t} \leq A \cdot \mu_{i,j,c,t} \tag{19}$$

$$\phi_{i,j,c,t} \leq \alpha_{i,j,t} \tag{20}$$

$$\phi_{i,j,c,t} \geq \alpha_{i,j,t} - A\,(1 - \mu_{i,j,c,t}) \tag{21}$$

To turn $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t})$ into a linear form, we assume that the function $\mathcal{Q}_{i,j}^{+}$ is additively separable so that $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t}) = \sum_{k=1}^{3} U_{i,j}^{(k)} f_{i,j}^{(k)} \left(v_{i,j,t}^{(k)} \right)$ with $v_{i,j,t}^{(1)} = \alpha_{i,j,t}$, $v_{i,j,t}^{(2)} = \beta_{i,j,t}$, and $v_{i,j,t}^{(3)} = \lambda_{i,j,t}$. This assumption can be broadened to include non-separable functions [34]. Each $f_{i,j}^{(k)}$ is then approximated by a piecewise linear continuous function $\tilde{f}_{i,j}^{(k)}$ defined in (22), which segments the domain of $f_{i,j}^{(k)}$ into $n_{i,j}^{(k)}$ possibly-unequal intervals and approximates $f_{i,j}^{(k)}$ in every interval by a linear function. By introducing $n_{i,j}^{(k)} \in \mathbb{N}^+$ pairs of real $\delta_{i,j,t}^{(k),l}$ and binary $b_{i,j,t}^{(k),l}$ decision variables, if (23)–(26) hold with $b_{i,j,t}^{(k),n_{i,j}^{(k)}+1} = 0$, then $\sum_{k=1}^{3} U_{i,j}^{(k)} \tilde{f}_{i,j}^{(k)} \left(v_{i,j,t}^{(k)} \right)$ has (27) as its linear form. The term $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t})$ in (12) is then replaced with (27)

[2] As $\mathcal{C}_{c_e}$ always exists at any t due to $\mathcal{A}_i$ being exactly in one cell at any t, it follows that $\Lambda_{c_s,c_e,t}$ is always defined at any t due to the presence of $\mathcal{R}_{c_s,c_e,t}$ at any t. Hence, if $\Lambda_{c_s,c_e,t} > 0$ for every cell pair $(\mathcal{C}_{c_s}, \mathcal{C}_{c_e})$ and for time point t, then $\tau_{c_s,i,t} > 0$ also for every cell $\mathcal{C}_{c_s}$ and time point t.

while adding (23)–(26) as constraints.

$$\tilde{f}_{i,j}^{(k)}\left(v_{i,j,t}^{(k)}\right)=\begin{cases} m_{i,j}^{(k),1}\ \left(v_{i,j,t}^{(k)}-L_{i,j}^{(k),0}\right)+C_{i,j}^{(k)} \\ \qquad\qquad \text{, if } v_{i,j,t}^{(k)}\in\left[L_{i,j}^{(k),0},L_{i,j}^{(k),1}\right] \\ \qquad\qquad\vdots \\ m_{i,j}^{(k),n_{i,j}^{(k)}}\left(v_{i,j,t}^{(k)}-L_{i,j}^{(k),n_{i,j}^{(k)}-1}\right) \\ \qquad\qquad +\tilde{f}_{i,j}^{(k)}\left(L_{i,j}^{(k),n_{i,j}^{(k)}-1}\right) \\ \quad \text{, if } v_{i,j,t}^{(k)}\in\left(L_{i,j}^{(k),n_{i,j}^{(k)}-1},L_{i,j}^{(k),n_{i,j}^{(k)}}\right] \end{cases} \tag{22}$$

$$v_{i,j,t}^{(k)}=L_{i,j}^{(k),0}+\sum_{l=1}^{n_{i,j}^{(k)}}\delta_{i,j,t}^{(k),l} \tag{23}$$

$$0\le\delta_{i,j,t}^{(k),l}\le b_{i,j,t}^{(k),l}\left(L_{i,j}^{(k),l}-L_{i,j}^{(k),l-1}\right) \tag{24}$$

$$\delta_{i,j,t}^{(k),l}\ge b_{i,j,t}^{(k),l+1}\left(L_{i,j}^{(k),l}-L_{i,j}^{(k),l-1}\right) \tag{25}$$

$$b_{i,j,t}^{(k),l}\ge b_{i,j,t}^{(k),l+1} \tag{26}$$

$$\sum_{k=1}^{3}U_{i,j}^{(k)}\left(\sum_{l=1}^{n_{i,j}^{(k)}}m_{i,j}^{(k),l}\delta_{i,j,t}^{(k),l}+C_{i,j}^{(k)}\right) \tag{27}$$

Finally, our formulation is summed up in terms of its parameters and variables in Table I.

TABLE I
THE MILP PARAMETERS AND DECISION VARIABLES.

Parameters	
$\mathbb{A}$, $\mathbb{M}$, $\mathbb{G}$	The sets of agents $\mathcal{A}_i$, VMs $\mathcal{M}_{i,j}$, and cells $\mathcal{C}_c$.
$\mathcal{R}_{c_s,c_e,t}$	A bidirectional route $\mathcal{C}_{c_s}\leftrightarrow\mathcal{C}_{c_e}$ (i.e., $\mathcal{R}_{c_s,c_e,t}=\mathcal{R}_{c_e,c_s,t}$).
$\Lambda_{c_s,c_e,t}$	$\mathcal{R}_{c_s,c_e,t}$ end-to-end latency ($\Lambda_{c_s,c_e,t}=\Lambda_{c_e,c_s,t}\in\mathbb{R}^{\ge 0}$).
Φ_c, Ψ_c	$\mathcal{C}_c$ computation & communication capacities in $\mathbb{R}^{\ge 0}$.
$\mathcal{E}_{i,j}$	$\mathcal{M}_{i,j}$ migration cost in $\mathbb{R}^{\ge 0}$.
M	The cap in $\mathbb{N}^+$ on concurrent migration count at any t.
$\alpha_{i,j}^{\min}$, $\alpha_{i,j}^{\max}$	$\mathcal{M}_{i,j}$ computation bandwidth lower & upper bounds in $\mathbb{R}^{\ge 0}$.
$\beta_{i,j}^{\min}$, $\beta_{i,j}^{\max}$	$\mathcal{M}_{i,j}$ communication bandwidth lower & upper bounds in $\mathbb{R}^{\ge 0}$.
$\lambda_{i,j}^{\max}$	The upper bound on $\mathcal{M}_{i,j}\leftrightarrow\mathcal{A}_i$ end-to-end latency in $\mathbb{R}^{\ge 0}$.
$\rho_{i,c,t}$	One (zero) if $\mathcal{A}_i$ is (not) in $\mathcal{C}_c$ at time t.
With $l\in\mathbb{N}^+$ and $\tilde{f}_{i,j}^{(1)}$, $\tilde{f}_{i,j}^{(2)}$, and $\tilde{f}_{i,j}^{(3)}$ being defined in (22) as the piece-wise linear functions that approximate the contributions of $\alpha_{i,j,t}$, $\beta_{i,j,t}$, and $\lambda_{i,j,t}$, respectively, in the quality function $\mathcal{Q}_{i,j}^{+}$:	
$U_{i,j}^{(1)}$	$\tilde{f}_{i,j}^{(1)}(\alpha_{i,j,t})$ weight to approximate $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.
$U_{i,j}^{(2)}$	$\tilde{f}_{i,j}^{(2)}(\beta_{i,j,t})$ weight to approximate $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.
$U_{i,j}^{(3)}$	$\tilde{f}_{i,j}^{(3)}(\lambda_{i,j,t})$ weight to approximate $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.
$C_{i,j}^{(1)}$	$\tilde{f}_{i,j}^{(1)}(\alpha_{i,j,t})$ offset to approximate $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.
$C_{i,j}^{(2)}$	$\tilde{f}_{i,j}^{(2)}(\beta_{i,j,t})$ offset to approximate $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.
$C_{i,j}^{(3)}$	$\tilde{f}_{i,j}^{(3)}(\lambda_{i,j,t})$ offset to approximate $\mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.
$L_{i,j}^{(1),0}$,$L_{i,j}^{(1),l}$	The piece-wise interval endpoints of $\tilde{f}_{i,j}^{(1)}$.
$L_{i,j}^{(2),0}$,$L_{i,j}^{(2),l}$	The piece-wise interval endpoints of $\tilde{f}_{i,j}^{(2)}$.
$L_{i,j}^{(3),0}$,$L_{i,j}^{(3),l}$	The piece-wise interval endpoints of $\tilde{f}_{i,j}^{(3)}$.
$m_{i,j}^{(1),l}$	The piece-wise gradient of $\tilde{f}_{i,j}^{(1)}$ in $\left(L_{i,j}^{(1),l-1},L_{i,j}^{(1),l}\right]$.
$m_{i,j}^{(2),l}$	The piece-wise gradient of $\tilde{f}_{i,j}^{(2)}$ in $\left(L_{i,j}^{(2),l-1},L_{i,j}^{(2),l}\right]$.
$m_{i,j}^{(3),l}$	The piece-wise gradient of $\tilde{f}_{i,j}^{(3)}$ in $\left(L_{i,j}^{(3),l-1},L_{i,j}^{(3),l}\right]$.
Variables	
$\mu_{i,j,c,t}$	One (zero) if $\mathcal{M}_{i,j}$ is (not) in $\mathcal{C}_c$ at time t.
$\alpha_{i,j,t}$	$\mathcal{M}_{i,j}$ computation bandwidth at time t.
$\beta_{i,j,t}$	$\mathcal{M}_{i,j}$ communication bandwidth at time t.

IV. EVALUATION

We evaluate the effectiveness and the scalability of our MILP formulation in a realistic ultra-dense network where many robots and services are deployed in an indoor environment (i.e., an industrial scenario). For clarity of illustration and without loss of generality, we use different edge networks, different MILP solving stategies, and different MILP parameters, while considering that *in an individual scenario* the agents and the cells have the same types, the cells are statically interconnected by the same type of links, the MEC servers are of the same type with the same software stack, every server and every link have the same capacities, respectively, every VM migration has the same cost, and every VM has the same bounds on the computation and communication bandwidth and latency and the same quality function. And, *in all scenarios* we assume that every end-to-end communication latency depends on two factors: 1) the intra-cell latency along the wireless link used by an agent to connect to the network, which for simplicity is assumed to be the same for every agent in every cell at any time, and 2) the inter-cell latency along the links used to connect a pair of cells, which for simplicity is assumed to be the same for every cell pair at any time.

Specifically, Section IV-A uses the small edge network shown in Figure 1(a) with a mesh topology, while Section IV-B and IV-C use the large edge network shown in Figure 1(b) with both star and mesh topologies. Furthermore, taking Figure 1(b) as an N-by-N grid of cells roamed by $4N$ agents, Section IV-C also has further scenarios to evaluate larger values of N. While the scenarios evaluated in Section IV-A are solved with a time limit of 1 hour, the scenarios evaluated in Section IV-B and IV-C are solved until their MILP solutions are within 10% of the optimum. And, while different scenarios use different sets of MILP parameters, the different sets are derived from the following common assumptions.

Every scenario assumes a certain network shown in Figure 1 to derive their respective $\mathbb{G}$. As shown in Figure 1, each row/ column has two agents that start at the opposite ends facing each other to move forward at the same speed to the opposite edges only to restart by turning around, and due to having the same speed, every agent enters the next cell at the next time point. These assumptions are used by every scenario to obtain their respective $\mathbb{A}$ and $\rho_{i,c,t}$. To obtain their respective $\mathcal{R}_{c_s,c_e,t}$, every scenario assumes that in the star topology, all cells are connected to one aggregation point so that every end-to-end latency assumes one of two distinct values, while in the mesh topology, every cell pair is connected in the

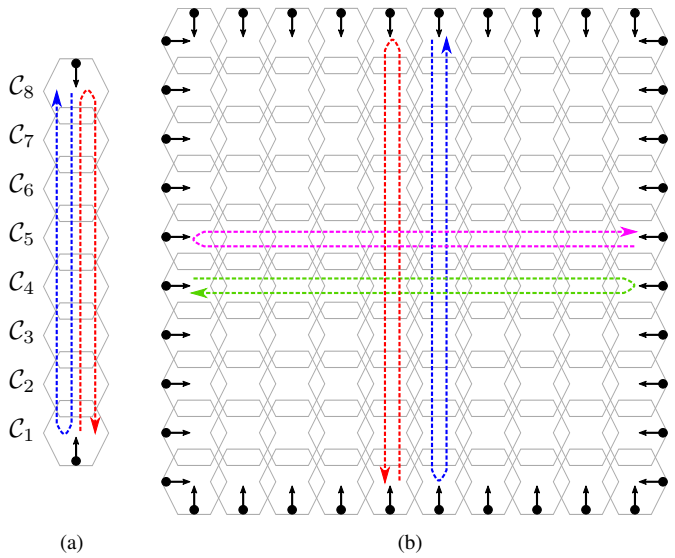

Fig. 1. Two different networks (a) and (b) are used in different subsections: (a) the 8-by-1 mesh network used in the scenarios evaluated in Section IV-A, showing the 2 agents as black circles, their initial headings at time t_0 with arrows, and their trajectories over the next 14 time points with dashed U-arrows, and (b) the 10-by-10 network used in the scenarios evaluated in Section IV-B and IV-C with both the star and mesh topologies, showing the 40 agents, their initial headings at time t_0, and the trajectories of, for clarity, only 4 agents over the next 14 time points.

Manhattan scheme[3] so that every end-to-end latency increases proportionally to the number of cells on the route. Additionally, every scenario assumes that in the edge network every communication bandwidth Ψ_c is at 1 Gbps (gigabits/s) and the intra-cell latency is at 2 ms while the latency along each inter-cell link is at 3 ms so that $\Lambda_{c_s,c_e,t} = 2 + 3(|\mathcal{R}_{c_s,c_e,t}| - 1)$ (e.g., every end-to-end latency in the star topology is 2 if not 5 ms). Every scenario then assumes that no limit exists on the number of concurrent migrations, and hence, they derive their respective M to be $|\mathbb{M}|$ (limiting M to different percentages of $|\mathbb{M}|$ is planned in our future work).

Finally, CPLEX CC8ATML 20.1.0 for GNU/Linux (ibm.com/analytics/cplex-optimizer) is used as the MILP solver on Ubuntu 16.04.7 on a Lenovo E40-80 laptop with 16 GiB RAM, no swap, and a 4-core Intel Core i3-5010U (2×64-bit 2.1-GHz cores, 2 threads/core). As its development environment (`oplide`) uses extra time and memory, the solver is run directly as `cplex -c "read i.lp" "$prm" mipopt "write o.sol"`. As `cplex` accepts a problem in the LP format, we first translate literally the formulation in Section III into a GMPL model[4] accepted by another MILP solver, GLPK (gnu.org/software/glpk). Then, for every scenario written in GMPL, we run GLPK as `glpsol --check --wlp i.lp -m model.glp -d data.glp` to translate the GMPL model (`model.glp`) and the scenario (`data.glp`) into an LP-format file (`i.lp`) without solving the MILP (`--check`).

[3]If $\mathcal{C}_{(x,y)}$ is $\mathcal{C}_c$ at $(x,y) \in (\mathbb{N}^+)^2$, $\mathcal{R}_{(x_1,y_1),(x_2,y_2),t} = \{\mathcal{C}_{(x,y_1)} \in \mathbb{G} | \min\{x_1,x_2\} \le x \le \max\{x_1,x_2\}\} \cup \{\mathcal{C}_{(x_2,y)} \in \mathbb{G} | y_1 \le y \le y_2\}$ for $y_1 \le y_2$.

[4]We make the model available at archive.org/details/model-202108.

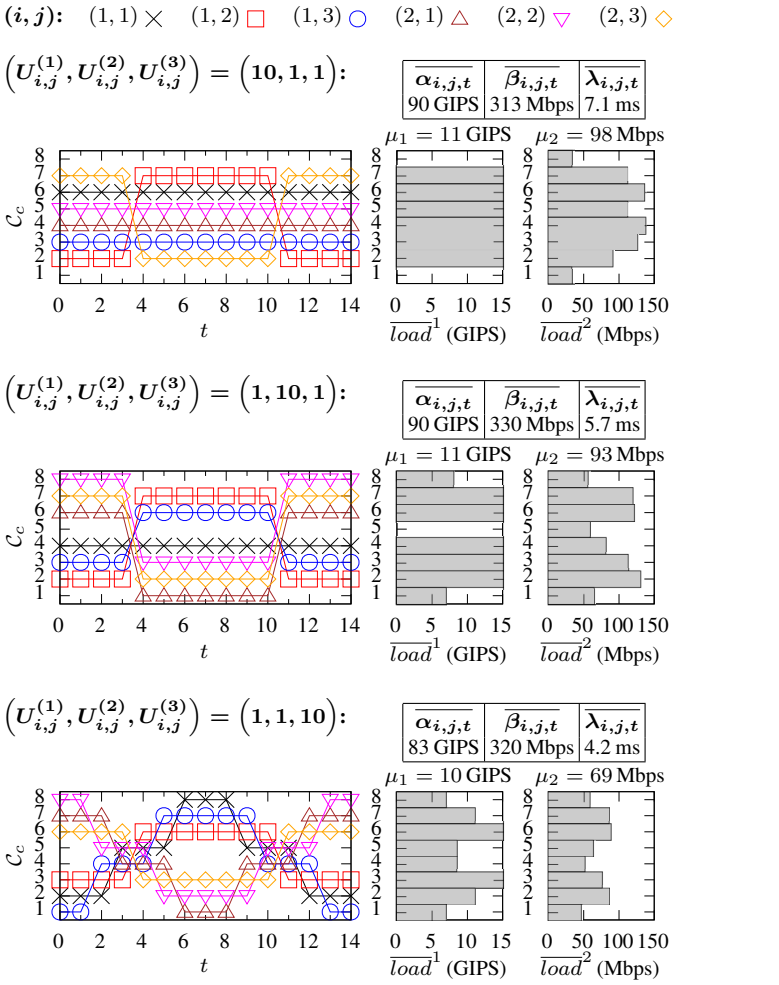

Fig. 2. The effectiveness of different quality functions as evaluated in Section IV-A on the network shown in Figure 1(a).

A. System Behavior in the Time Domain

Our formulation effectiveness is first shown by the effect of distinct quality functions on the small mesh network shown in Figure 1(a). The network is small to make the resulting plots easy to analyze. Furthermore, the mesh topology is used instead of the star topology because the mesh topology is more complex than the star topology in the way that the link of a cell may have to bear the traffic between a pair of other cells. This complexity then makes it easier to analyze the effect of quality functions that maximize the available link bandwidth.

The effect of distinct quality functions is shown using scenarios that make the solver favor some decision variables only by the quality functions. The quality functions obtained by distinct weights $\left(U_{i,j}^{(1)}, U_{i,j}^{(2)}, U_{i,j}^{(3)}\right)$ are used with the same 8 cells, 15 time points, 2 agents, 3 VMs/agent, $\Phi_c = 100$ GIPS (gigainstructions/s), $\alpha_{i,j}^{\min} = 15$ GIPS (any MEC server can host all VMs), $\alpha_{i,j}^{\max} = 90$ GIPS, $\beta_{i,j}^{\min} = 150$ Mbps (any link can route all communication channels), $\beta_{i,j}^{\min} = 900$ Mbps, $\lambda_{i,j}^{\max} = 23$ ms (migration is optional), $C_{i,j}^{(1)} = C_{i,j}^{(2)} = 0$, $C_{i,j}^{(3)} = 1$, $\left(L_{i,j}^{(1),0}, L_{i,j}^{(1),1}\right) = \left(\alpha_{i,j}^{\min}, \alpha_{i,j}^{\max}\right)$,

$\left(L_{i,j}^{(2),0}, L_{i,j}^{(2),1}\right) = \left(\beta_{i,j}^{\min}, \beta_{i,j}^{\max}\right)$, $\left(L_{i,j}^{(3),0}, L_{i,j}^{(3),1}\right) = \left(2, \lambda_{i,j}^{\max}\right)$, $m_{i,j}^{(1),1} = \frac{1}{\alpha_{i,j}^{\max} - \alpha_{i,j}^{\min}}$, $m_{i,j}^{(2),1} = \frac{1}{\beta_{i,j}^{\max} - \beta_{i,j}^{\min}}$, $m_{i,j}^{(3),1} = \frac{-1}{\lambda_{i,j}^{\max} - 2}$, and $\mathcal{E}_{i,j} = 80\% \max_{\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t}} \mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t})$. The scenarios are solved in an hour by setting `prm="set timelimit 3600"`.

Figure 2 shows the effect of distinct quality functions on the positions of the VMs over time on the left part, on the MEC server average processing $\overline{load}_c^1 = \frac{\sum_{\mathcal{M}_{i,j} \in \mathbb{M}, t \in \mathbb{H}} \phi_{i,j,c,t}}{|\mathbb{M}|+|\mathbb{H}|}$ on the middle part with the mean $\mu_1 = \frac{\sum_{\mathcal{C}_c \in \mathbb{C}} \overline{load}_c^1}{|\mathbb{C}|}$ shown at the top of each plot, on the mesh-network link average traffic $\overline{load}_c^2 = \frac{\sum_{\mathcal{M}_{i,j} \in \mathbb{M}, t \in \mathbb{H}} \psi_{i,j,c,t}}{|\mathbb{M}|+|\mathbb{H}|}$ on the right part with the mean $\mu_2 = \frac{\sum_{\mathcal{C}_c \in \mathbb{C}} \overline{load}_c^2}{|\mathbb{C}|}$ shown at the top of each plot, and on the average bandwidth of computation $\frac{\sum_{\mathcal{M}_{i,j} \in \mathbb{M}, t \in \mathbb{H}} \alpha_{i,j,t}}{|\mathbb{M}|+|\mathbb{H}|}$ and communication $\frac{\sum_{\mathcal{M}_{i,j} \in \mathbb{M}, t \in \mathbb{H}} \beta_{i,j,t}}{|\mathbb{M}|+|\mathbb{H}|}$ and latency $\frac{\sum_{\mathcal{M}_{i,j} \in \mathbb{M}, t \in \mathbb{H}} \lambda_{i,j,t}}{|\mathbb{M}|+|\mathbb{H}|}$ in the table at every row heading, which shows the distinct weights. The weight of 10 used in a row makes the solver favor $\alpha_{i,j,t}$/$\beta_{i,j,t}$/$\lambda_{i,j,t}$ if $U_{i,j}^{(1)}$/$U_{i,j}^{(2)}$/$U_{i,j}^{(3)}$ is 10. As placing VMs in the MEC server in the cell where their agent is results in the lowest latency and traffic born by intermediary links at the cost of higher migration frequency and lower computation bandwidth as VMs have to follow their agents and some servers have to host multiple VMs, Figure 2 shows that our formulation is effective at implementing distinct quality functions, e.g., migration frequency is highest for $U_{i,j}^{(3)} = 10$ to minimize $\lambda_{i,j,t}$ but lowest for $U_{i,j}^{(1)} = 10$ as $\alpha_{i,j,t}$ is highest when each server hosts just one VM.

B. System Key Performance Indicators (KPIs)

Our formulation effectiveness is then shown on the VM migration frequency, outage count, and average computation bandwidth and latency (system KPIs) for different VM migration costs $\mathcal{E}_{i,j}$ and MEC server computation capacities Φ_c by scenarios that use the network shown in Figure 1(b) with star/mesh topology and the same 100 cells, 19 time points, 40 agents, 3 VMs/agent, and the same QoS bounds and quality functions based on [35]–[38]: $(\alpha_{i,j}^{\min}, \alpha_{i,j}^{\max}) = (11, 13)$ GIPS, $(\beta_{i,j}^{\min}, \beta_{i,j}^{\max}) = (9, 11)$ Mbps, $\lambda_{i,j}^{\max} = 15$ ms, $U_{i,j}^{(1)} = U_{i,j}^{(2)} = U_{i,j}^{(3)} = 1$, $C_{i,j}^{(1)} = C_{i,j}^{(2)} = -1$, $C_{i,j}^{(3)} = 1$, $L_{i,j}^{(1),0} = L_{i,j}^{(2),0} = L_{i,j}^{(3),0} = 0$, $\left(L_{i,j}^{(1),1}, L_{i,j}^{(1),2}\right) = \left(\alpha_{i,j}^{\min}, \alpha_{i,j}^{\max}\right)$, $\left(L_{i,j}^{(2),1}, L_{i,j}^{(2),2}\right) = \left(\beta_{i,j}^{\min}, \beta_{i,j}^{\max}\right)$, $\left(L_{i,j}^{(3),1}, L_{i,j}^{(3),2}\right) = \left(5, \lambda_{i,j}^{\max}\right)$, $m_{i,j}^{(1),1} = \frac{1}{\alpha_{i,j}^{\min}}$, $m_{i,j}^{(2),1} = \frac{1}{\beta_{i,j}^{\min}}$, $m_{i,j}^{(3),1} = 0$, $m_{i,j}^{(1),2} = \frac{1}{\alpha_{i,j}^{\max} - \alpha_{i,j}^{\min}}$, $m_{i,j}^{(2),2} = \frac{1}{\beta_{i,j}^{\max} - \beta_{i,j}^{\min}}$, $m_{i,j}^{(3),2} = \frac{-1}{\lambda_{i,j}^{\max} - 5}$. Each scenario is solved to get a solution that is within 10% of the optimum by `prm="set mip tolerances mipgap 0.1"`. To highlight our formulation effectiveness, we use baseline scenarios that always migrate the VMs to the MEC server in the cell where their agent is with the same $(\alpha_{i,j}^{\min}, \alpha_{i,j}^{\max}) = (0, 11)$ GIPS and $(\beta_{i,j}^{\min}, \beta_{i,j}^{\max}) = (0, 9)$ Mbps so that their solutions give every VM the lowest latency but the highest migration frequency and possibly some outage times, each occurring at time t and at cell $\mathcal{C}_c$ if $\alpha_{i,j,t} < 11$ GIPS for the least possible number of VMs on the server in $\mathcal{C}_c$ when $|\{\mathcal{M}_{i,j} \in \mathbb{M} \,|\, \mu_{i,j,c,t} = 1\}| \times 11\,\text{GIPS} > \Phi_c$.

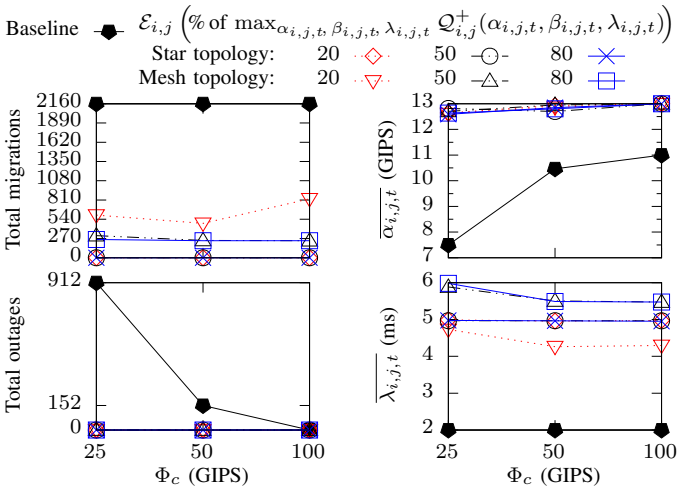

Fig. 3. The effectiveness of our MILP formulation as evaluated in Section IV-B on the network shown in Figure 1(b).

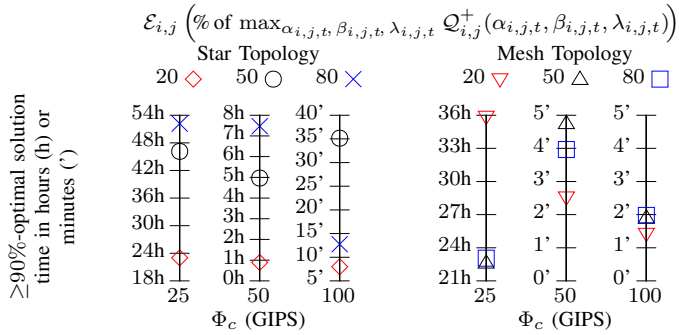

Fig. 4. The scalability of our MILP formulation as evaluated in Section IV-C on the network shown in Figure 1(b).

Figure 3 shows the system KPIs attainable by the optimal solutions for the baseline scenarios and by the ≥90%-optimal solutions for the other scenarios. Every ≥90%-optimal solution has no service outage by (4), (5), and (11) and sets $\beta_{i,j,t} = 11$ Mbps as $\Psi_c = 1$Gbps $> \frac{\Phi_c}{11\,\text{GIPS}} \times 11$Mbps, while the baseline solutions are plotted as one line in each KPI as they are equal despite the various topologies and migration costs. Figure 3 shows that our formulation is effective for the system KPIs as every ≥90%-optimal solution migrates much less often, especially in the star topology, has no outage, and gives much higher computation bandwidth even when the server is very constrained at 25 GIPS regardless of the topology, all of these with latency that is very acceptable in the robotic domain.

C. System Complexity

Our formulation scalability in handling complex systems is shown in Figure 4 by the time taken to get the ≥90%-optimal solutions plotted in Figure 3 and in Figure 5 by the time taken to get the ≥90%-optimal solutions for the mesh-network scenarios described in the previous section with $\Phi_c = 50$ GIPS and $\mathcal{E}_{i,j} = 80\% \max_{\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t}} \mathcal{Q}_{i,j}^{+}(\alpha_{i,j,t}, \beta_{i,j,t}, \lambda_{i,j,t})$

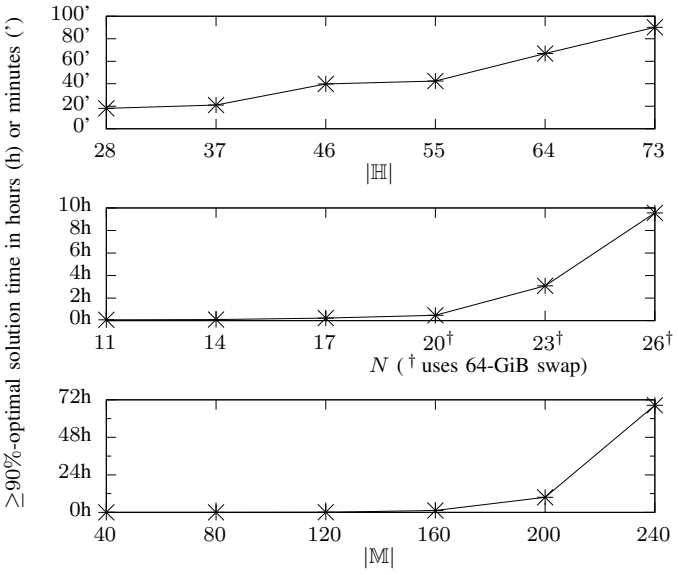

Fig. 5. The scalability of our MILP formulation as evaluated in Section IV-C on the mesh network shown in Figure 1(b) when $\Phi_c = 50$ GIPS and $\mathcal{E}_{i,j} = 80\% \max_{\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t}} \mathcal{Q}^+_{i,j}(\alpha_{i,j,t},\beta_{i,j,t},\lambda_{i,j,t})$.

when the scenarios use different time horizon lengths $|\mathbb{H}|$, grid sizes N, which mean different counts of cells $|\mathbb{G}|$ and agents $|\mathbb{A}|$, and VMs/agent counts, which mean different $|\mathbb{M}|$.

Figure 4 shows three important points about our formulation scalability: 1) the mesh topology takes less time than the star topology to solve, 2) the more constrained the MEC server is, the (possibly exponentially) longer the solution is obtained, and 3) compared to the previous point, migration cost has no significant effect on the solution time. Furthermore, Figure 5 shows two important points about our formulation scalability: 1) the fiercer the MEC server is contested, the (possibly exponentially) longer the solution is obtained (e.g., for 6 VMs/agent, the server-to-VM ratio is 100:240, but for $N = 26$, the ratio is 676:312, and hence, the servers are contested fiercer in the former than in the latter), and 2) memory becomes the main limitation as the number of cells, and hence the number of end-to-end routes, increases.

V. Conclusions and Future Work

In this paper, we have considered the problem of allocating resources at the edge of a B5G network to real-time services optimally by formulating an MILP whose decision variables are the amount of computation and communication resources and the MEC servers to execute the VMs providing the services at each time point. Using state-of-the art optimization tools allows us to treat problems of reasonable size in the number of cells and agents when the agent trajectories are known up-front and the optimization can be performed offline before starting the system operations. When the size of the problem grows or when the system is highly dynamic and requires online optimization, heuristic approaches are needed to produce high-quality sub-optimal solutions. This is one of the most promising research areas that we reserve for our future investigations, which include futuristic scenarios where the base stations are mobile (e.g., aerial or terrestrial vehicles) and need an optimal decision on their positions as well.

Acknowledgment

This work has received funding from the Italian Ministry of Education, University and Research (MIUR) through the PRIN project no. 2017NS9FEY entitled "Realtime Control of 5G Wireless Networks: Taming the Complexity of Future Transmission and Computation Challenges". The views and opinions expressed in this work are those of the authors and do not necessarily reflect those of the funding institution.

References

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2020.

[2] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, 2020.

[3] Q. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116 974–117 017, 2020.

[4] T. Dlamini, Á. F. Gambín, D. Munaretto, and M. Rossi, "Online resource management in energy harvesting BS sites through prediction and soft-scaling of computing resources," in *IEEE Symp. on Personal, Indoor, and Mobile Radio Comm.*, 2018, pp. 1820–1826.

[5] T. Subramanya, D. Harutyunyan, and R. Riggio, "Machine learning-driven service function chain placement and scaling in MEC-enabled 5G networks," *Computer Networks*, vol. 166, p. 106980, 2020.

[6] M. Berno, J. J. Alcaraz, and M. Rossi, "On the allocation of computing tasks under QoS constraints in hierarchical MEC architectures," in *4th Int. Conf. on Fog and Mobile Edge Comp. (FMEC)*, 2019, pp. 37–44.

[7] Z. Cheng, Q. Wang, Z. Li, and G. Rudolph, "Computation offloading and resource allocation for mobile edge computing," in *IEEE Symp. Series on Comp. Intelligence (SSCI)*, 2019, pp. 2735–2740.

[8] A. Bozorgchenani, F. Mashhadi, D. Tarchi, and S. S. Monroy, "Multi-objective computation sharing in energy and delay constrained mobile edge computing environments," *IEEE Trans. on Mob. Comp.*, 2020.

[9] B. Yang, X. Cao, J. Bassey, X. Li, T. Kroecker, and L. Qian, "Computation offloading in multi-access edge computing networks: A multi-task learning approach," in *IEEE Int. Conf. on Comm.*, 2019, pp. 1–6.

[10] D. Kirov, P. Nuzzo, R. Passerone, and A. L. Sangiovanni-Vincentelli, "Optimized selection of wireless network topologies and components via efficient pruning of feasible paths," in *Proc. of the* 55^{th} *Design Automation Conf.*, ser. DAC 2018, San Francisco, CA, June 24–28, 2018.

[11] P. Nuzzo, N. Bajaj, M. Masin, D. Kirov, R. Passerone, and A. L. Sangiovanni-Vincentelli, "Optimized selection of reliable and cost-effective safety-critical system architectures," *IEEE Trans. on Computer-Aided Design of ICs and Systems*, vol. 39, no. 10, pp. 2109–2123, 2020.

[12] H. Peng, Q. Ye, and X. S. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Comm.*, vol. 26, no. 4, pp. 156–162, 2019.

[13] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "LORM: Learning to optimize for resource management in wireless networks with few training samples," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 665–679, Jan 2020.

[14] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 158–11 168, 2019.

[15] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for mobile edge computing," in *Symp. on Mod. & Opt. in Mob., Ad Hoc, & Wi. Net. (WiOpt)*. IEEE, 2018, pp. 1–6.

[16] L. Ferdouse, A. Anpalagan, and S. Erkucuk, "Joint communication and computing resource allocation in 5G cloud radio access networks," *IEEE Trans. on Vehicular Tech.*, vol. 68, no. 9, pp. 9122–9135, 2019.

[17] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Transactions on emerging topics in computing*, 2019.

[18] G. Wang, F. Xu, and C. Zhao, "Multi-access edge computing based vehicular network: Joint task scheduling and resource allocation strategy," in *IEEE Int. Conf. on Comm. (ICC) Workshops*, 2020, pp. 1–6.

[19] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Trans. on Vehicular Tech.*, vol. 69, no. 3, pp. 3341–3356, 2020.

[20] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *IEEE Symp. on Personal, Indoor, and Mobile Radio Comm.*, 2016, pp. 1–6.

[21] S. Thananjeyan, C. A. Chan, E. Wong, and A. Nirmalathas, "Mobility-aware energy optimization in hosts selection for computation offloading in multi-access edge computing," *IEEE Open Journal of Com.Soc.*, 2020.

[22] C.-L. Wu, T.-C. Chiu, C.-Y. Wang, and A.-C. Pang, "Mobility-aware deep reinforcement learning with glimpse mobility prediction in edge computing," in *IEEE Int. Conf. on Comm. (ICC)*, 2020, pp. 1–7.

[23] W.-C. Chien, S.-Y. Huang, C.-F. Lai, H.-C. Chao, M. S. Hossain, and G. Muhammad, "Multiple contents offloading mechanism in ai-enabled opportunistic networks," *Computer Comm.*, vol. 155, pp. 93–103, 2020.

[24] A. Dalgkitsis, P.-V. Mekikis, A. Antonopoulos, and C. Verikoukis, "Data driven service orchestration for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–1, 2020.

[25] X. Yu, M. Guan, M. Liao, and X. Fan, "Pre-migration of vehicle to network services based on priority in mobile edge computing," *IEEE Access*, vol. 7, pp. 3722–3730, 2019.

[26] L. Chen, D. Yang, M. Nogueira, C. Wang, D. Zhang *et al.*, "Data-driven C-RAN optimization exploiting traffic and mobility dynamics of mobile users," *IEEE Transactions on Mobile Computing*, 2020.

[27] P. Roy, A. Tahsin, S. Sarker, T. Adhikary, M. A. Razzaque, and M. M. Hassan, "User mobility and quality-of-experience aware placement of virtual network functions in 5G," *Computer Communications*, vol. 150, pp. 367–377, 2020.

[28] Z. Hong, H. Huang, S. Guo, W. Chen, and Z. Zheng, "QoS-aware cooperative computation offloading for robot swarms in cloud robotics," *IEEE Trans. on Vehicular Tech.*, vol. 68, no. 4, pp. 4027–4041, 2019.

[29] F. Malandrino, C. F. Chiasserini, G. Einziger, and G. Scalosub, "Reducing service deployment cost through VNF sharing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2363–2376, 2019.

[30] B. Németh, N. Molner, J. J. Martín-Pérez, C. J. Bernardos, A. de la Oliva, and B. Sonkoly, "Delay and reliability-constrained VNF placement on mobile and volatile 5G infrastructure," *arXiv:2007.11870*, 2020.

[31] M. Afrin, J. Jin, A. Rahman, Y.-C. Tian, and A. Kulkarni, "Multi-objective resource allocation for edge cloud based robotic workflow in smart factory," *Future Gen. Computer Sys.*, vol. 97, pp. 119–130, 2019.

[32] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "Practical solutions for QoS-based resource allocation problems," in *Proceedings of the 19th IEEE Real-Time Systems Symposium*, 1998, pp. 296–306.

[33] S. P. Bradley, A. C. Hax, and T. L. Magnanti, *Applied Mathematical Programming*. Addison-Wesley, 1977, ch. 9.

[34] J. P. Vielma, S. Ahmed, and G. Nemhauser, "Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions," *Operations Research*, vol. 58, no. 2, pp. 303–315, 2010.

[35] G. Brown *et al.*, "Ultra-reliable low-latency 5G for industrial automation," *Technol. Rep. Qualcomm*, vol. 2, p. 52065394, 2018.

[36] "Fujitsu server: Fujitsu global," https://www.fujitsu.com/global/products/computing/servers/, May 2021, (Accessed on 05/26/2021).

[37] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Comm. Surveys & Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.

[38] L. Liu and Q. Fan, "Resource allocation optimization based on mixed integer linear programming in the multi-cloudlet environment," *IEEE Access*, vol. 6, pp. 24 533–24 542, 2018.