Multiple Granularity Online Control of Cloudlet Networks for Edge Computing

Lei Jiao[1], Lingjun Pu[2,3], Lin Wang[4], Xiaojun Lin[5], Jun Li[1]
[1]University of Oregon, USA [2]Nankai University, China
[3]Guangdong Key Laboratory of Big Data Analysis and Processing, China
[4]Technische Universität Darmstadt, Germany [5]Purdue University, USA

***Abstract*—Operating distributed cloudlets at optimal cost is nontrivial when facing not only the dynamic and unpredictable resource prices and user requests, but also the low efficiency of today's immature cloudlet infrastructures. We propose to control cloudlet networks at multiple granularities—fine-grained control of servers inside cloudlets and coarse-grained control of cloudlets themselves. We model this problem as a mixed-integer nonlinear program with the switching cost over time. To solve this problem online, we firstly linearize, "regularize", and decouple it into a series of one-shot subproblems that we solve at each corresponding time slot, and afterwards we design an iterative, dependent rounding framework using our proposed randomized pairwise rounding algorithm to convert the fractional control decisions into the integral ones at each time slot. Via rigorous theoretical analysis, we exhibit our approach's performance guarantee in terms of the competitive ratio and the multiplicative integrality gap towards the offline optimal integral decisions. Extensive evaluations with real-world data confirm the empirical superiority of our approach over the single granularity server control and the state-of-the-art algorithms.**

I. Introduction

Provisioning services at the network edge in close proximity to end users with ultra low latency is becoming one of the most essential goals pursued by many service providers today. The key enablers towards this goal are *cloudlets*, i.e., small data centers, machine rooms, and server clusters at diverse locations such as WiFi neighborhoods, enterprise premises, and telecom central offices [1], [2]. A local cloudlet network can be shown as Figure 1, where users access the service through the WiFi networks, and cloudlets are co-located with the WiFi access points and are connected via wireline backhaul networks.

Similar to large data centers, in order to save the operational expense while serving the time-varying workload (e.g., user requests), it is often essential to dynamically switch on and off the servers [3], [4] in the cloudlets; however, only switching on/off the servers in a cloudlet is often insufficient, and it is necessary to switch on/off the entire cloudlet as well. This is because cloudlets are usually resource-inefficient and can incur a considerable amount of operational expense via the non-IT equipments. For example, the cooling equipment in small data centers is usually dedicated, atomic, and cannot be turned off partially to match the number of servers or the amount of heat [5]. Consider the Power Usage Effectiveness (PUE), the ratio of a data center's total energy consumption, such as IT, cooling, and lightning, over its IT energy consumption. It is recently reported that small (1$\sim$25 servers) and media (26$\sim$500 servers) data centers typically have PUEs of $1.5 \sim 2.1$ [6]; even the better modular and container-based micro data centers can have PUEs up to 1.4. This is in stark contrast to large data centers, such as Google's, with PUEs as low as 1.1 [7]. Therefore, controlling servers in cloudlets but leaving cloudlets themselves always on can consume significant non-IT energy.

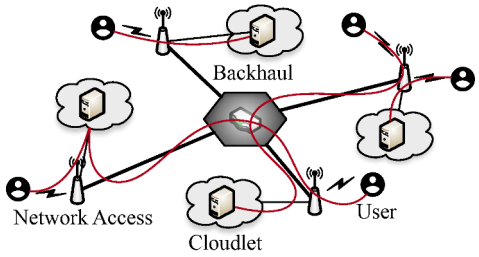

Fig. 1: An example cloudlet network structure

In this paper, we refer to this problem of jointly controlling the on/off status of the cloudlets and the servers as the "multiple granularity" cloudlet control problem. What complicates this problem is the "switching cost" incurred every time when turning on/off the cloudlets and the servers. For example, when booting additional servers to provide more service instances, the switching cost is embodied in the time needed for server initialization, the bandwidth needed for state migration, or any cost related to system oscillation, reliability risk, and hardware wear and tear [3], [4]. For cloudlets in continuous time slots, a control decision at one time slot not only incurs the operational cost at that time slot, based on the number of running cloudlets and servers at that time slot and the resource price at that time slot, but also affects the switching cost to be incurred between that time slot and the next time slot, depending on the number of the additional cloudlets and servers that will be turned on at the next time slot. In an "online" setting as is often the case, with no knowledge about the control decision at the next time slot, as it has not been made until the next time slot, it is nontrivial to make a good decision at the current time slot.

Another important differentiating factor of this multiple granularity cloudlet control problem lies in the intrinsically intertwined control decisions that also need to be made for the workload distribution across cloudlets with distance-dependent cost (e.g., delay). Unlike large data centers connected via wide area networks, local cloudlets are often highly distributed and connected by high-speed (e.g., optical) backhaul networks [1], making it flexible to distribute and serve workloads from

different locations. For example, workloads can be moved from one cloudlet to another to overcome the capacity limit or leverage the cheaper resources, while introducing moderate additional delay. Note this is different from the traditional data center load balancing or request distribution problem, as the workload distribution decision in this case affects not only the total delay and the total operational cost per time slot, but also the total switching cost of cloudlets and servers across time slots. Typical data center request distribution algorithms do not consider the switching cost [8], and are indeed not optimal in the cloudlet networks scenario.

Research to date has never investigated the cloudlet control problem from a multi-granularity perspective. Those that switch on/off servers [3], [4], [9], [10] do not shut down clouds and data centers, while those on resource allocation and job scheduling in edge clouds [2], [11]–[13] do not toggle servers. Existing online algorithmic techniques also fall insufficient for our multi-granularity control problem. They focus on either fractional control decisions only [3], [10] or fixed/bounded resource prices [4], [9], and cannot make multi-granularity decisions intertwined with distance-dependent cost simultaneously. It is unclear how to apply them to our problem while preserving or adapting their performance guarantees.

We model the multi-granularity control problem for cloudlet networks, controlling the servers inside cloudlets, the cloudlets themselves, and the workload distribution across cloudlets. We make no assumption on workload and resource price dynamics, so our models are general and can capture the workload variations due to user arrivals, departures, mobilities, and flash crowds [14], and the time-varying prices such as those of wholesale electricity [15] and spot virtual machines [16]. We model the switching costs of servers and cloudlets, accounting for increasing their numbers while capturing the fact that shutting them down is fast and incurs negligible cost [3]. With a simple yet general affine model for the delay, we minimize the total cost and enforce the constraints for workload distribution and processing, and also for cloudlet-server association.

Inspired by two separate techniques of regularization [17] and pipage rounding [18], [19], we propose a novel online algorithmic framework to solve our multi-granularity cloudlet control problem which turns out to be a mixed-integer nonlinear program. First, we make fractional control decisions online. We design an online algorithm that uses a carefully-designed logarithmic function to replace the nonlinear switching cost and decouples our problem into a series of one-shot subproblems solvable at each corresponding time slot by only taking the dynamic inputs at that time slot and the solution of the previous time slot. Next, we convert our fractional control decisions into integral ones. We design a randomized, pairwise rounding algorithm, where a pair of fractions are rounded together every time without violating any constraint of our problem, and plug this rounding algorithm into an iterative "rounding and re-solving" process to accommodate the multi-granularity control decisions at each time slot. Via rigorous formal proofs, we exhibit our approach's worst-case performance guarantee as a parameterized constant, which is also a product of the competitive ratio for our fractional online step and the multiplicative integrity gap for our rounding step, i.e., for arbitrary dynamic inputs, the total cost incurred over time by the integral control decisions produced by our approach on the fly without knowing future inputs is guaranteed not to exceed this constant times the total cost incurred over time by the offline optimal integral control decisions with the complete knowledge about all the future inputs in advance.

We conduct extensive evaluations using London's underground network of all its 268 stations [20] to simulate the cloudlet network and the real-world dynamic passenger numbers at each station [21] to simulate the workload, for a one-week period in November 2016, assuming all cloudlets are powered by an European wholesale electricity market [15]. We find the following results. For a typical cloudlet PUE of 1.4, our proposed multiple granularity online control algorithm achieves $15\% \sim 40\%$, $30\% \sim 50\%$, and $40\% \sim 60\%$ less total cost over time than a state-of-the-art online algorithm, a very advanced optimization solver, and the single granularity server control strategy, respectively. As the PUE grows to 2, our algorithm can achieve up to 65% less total cost than the single granularity control algorithm, and up to 25% less total cost than the next best algorithm in a pool of multiple different combinations of fractional online algorithms and rounding algorithms. Our algorithm saves the cloudlets usage significantly, and scales very well as the problem size increases.

II. Models and Problem Formulation

A. System Models

Cloudlets, Capacity, and Delay. We consider a network of multiple distributed cloudlets or small data centers, represented by the set $\mathcal{I}$, and multiple network access points, represented by the set $\mathcal{J}$. Cloudlets connect to one another via wireline backhaul networks, and every cloudlet is reachable from every network access point. A user connects to one of the access points, e.g., the closest one, to access the cloudlets. We have $\mathcal{I} \subseteq \mathcal{J}$ if all the cloudlets under consideration are co-located with the network access points. The cloudlet $i \in \mathcal{I}$ has its integral capacity C_i, referring to its number of servers or server clusters, depending on how the servers inside the cloudlets are managed. We use d_{ij}, $\forall i \in \mathcal{I}$, $\forall j \in \mathcal{J}$ to denote the delay between the cloudlet i and the network access point j.

Workload and Processing. We consider the entire system over $|\mathcal{T}|$ continuous time slots, where $\mathcal{T} \stackrel{\text{def}}{=} \{1, 2, ..., T\}$. The network access point j has the aggregated workload λ_{jt} at the time slot t. This may represent the number of user requests to be processed, or the number of jobs to be offloaded. λ_{jt} can be served by any one cloudlet or multiple cloudlets in $\mathcal{I}$, within each cloudlet's capacity limit. If served from a remote cloudlet, the corresponding access delay will be incurred. λ_{jt} often changes dynamically, i.e., users arrive, leave, move, or generate different amounts of workload over time. We make no assumption on how λ_{jt} varies across locations and over time. Workload is processed by servers. We allow all the cloudlets to be heterogenous, and use R_i, $\forall i$ as a coefficient to convert the

amount of workload to the number of servers or server clusters. Without loss of generality, we require $\frac{1}{R_i}$ to be a positive integer, i.e., the number of requests that can be handled by a single server or server cluster at the cloudlet i.

Unit Operational Cost. The unit operational cost (or the resource price), modeled for servers and cloudlets respectively, is the operational expense when running or using one server or one cloudlet per time slot. Such operational expense can include the electricity cost, the carbon footprint, the hardware/software maintenance/license fee, and the human labor cost. Note that, for cloudlet, we use the unit operational cost to refer to the cost for running its non-IT equipments, such as cooling, lightning, power distribution/conversion facilities. We denote the unit operation cost of servers and cloudlets as p_{it}^s and p_{it}^b, $\forall i$, $\forall t$, respectively. We make no assumption on how they vary across locations and over time.

Unit Switching Cost. As we propose the multi-granularity control, both the servers inside the cloudlets and the cloudlets themselves can be switched on and off dynamically. Toggling servers and cloudlets indeed incurs the "switching cost", which can capture the hardware wear and tear, system risk associated with the toggling operations, or the service performance degradation due to the lead or the initialization time of booting new software resources, loading profiles, migrating states, and so on. We denote the switching cost for switching on one server and one cloudlet as c_i^s and c_i^b, $\forall i$, respectively.

Control Variables. We have three types of control variables: $x_{ijt} \geq 0$, $\forall i$, $\forall j$, $\forall t$, referring to the amount of the workload distributed to the cloudlet i from the network access point j at the time slot t; $y_{it} \in \{0, 1, 2, 3, ...\}$, $\forall i$, $\forall t$, referring to the integral number of servers or server clusters activated at the cloudlet i to process the workload at the time slot t; $z_{it} \in \{0, 1\}$, $\forall i$, $\forall t$, denoting whether to activate the cloudlet i at the time slot t.

B. Problem Formulation

We formulate the multi-granularity control problem:

$$\begin{aligned}
\min \quad & \textstyle\sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& \textstyle + \sum_t \sum_i c_i^s \left(y_{it} - y_{it-1}\right)^+ + \sum_t \sum_i c_i^b \left(z_{it} - z_{it-1}\right)^+ \\
\text{s.t.} \quad & \textstyle\sum_i x_{ijt} \geq \lambda_{jt}, \qquad \forall j, \forall t, && \text{(1a)} \\
& \textstyle y_{it} \geq R_i \sum_j x_{ijt}, \qquad \forall i, \forall t, && \text{(1b)} \\
& C_i z_{it} \geq y_{it}, \qquad \forall i, \forall t, && \text{(1c)} \\
& x_{ijt} \geq 0, \qquad \forall j, \forall i, \forall t, && \text{(1d)} \\
& z_{it} \leq 1, \qquad \forall i, \forall t, && \text{(1e)} \\
& y_{it} \in \{0, 1, 2, 3, ...\}, z_{it} \in \{0, 1\}, \forall i, \forall t. && \text{(1f)}
\end{aligned}$$

The total cost consists of multiple components, as in the objective: the total delay of distributing or migrating the workload, the total operational cost and switching cost incurred by the cloudlets, and the total operational cost and switching cost incurred by the servers. We adopt a special function, $(x)^+ \stackrel{\text{def}}{=} \max\{x, 0\}$, to calculate the switching cost of increasing the resources. Constraint (1a) ensures that all of the workload are distributed and served. Constraint (1b) ensures that a sufficient number of servers are switched on to serve the corresponding workload. Constraint (1c) ensures that the server activation can only be performed within the capacity of the corresponding cloudlet, if the cloudlet itself is switched on. Constraint (1d), together with Constraints (1a)$\sim$(1c), ensures that all the control variables are non-negative. Constraint (1e) sets the upper limit for the cloudlet control variables. With Constraint (1f), the problem formulation does not actually need Constraint (1e); however, we keep it to facilitate our algorithm design and performance analysis. Also note there can exist weights associated to the five terms in the objective, but we remove them for the ease of presentation.

III. Algorithm Design

A. Challenges and Idea

Algorithmic Challenges. The major challenges for solving our optimization problem online stem from two aspects.

The first aspect is the *online uncertainty*. The switching cost couples the decision of the current time slot with that of the next time slot, i.e., any decision made for the current time slot will influence the switching cost between it and the possible decisions that can be made for the next time slot for the inputs not revealed yet. We would want an online algorithm that has "competitive" guarantees, i.e., the total cost over time of our online decisions made without future knowledge should not exceed a constant (i.e., the *competitive ratio*) times that of the offline optimal algorithm which knows all inputs in prior.

The second aspect is the *nonconvexity and intractability*. We highlight that our problem is a mixed-integer program, and we need to make integer decisions to dictate servers and cloudlets to switch on and off. A mixed-integer program is inherently nonconvex and NP-hard. It is already hard to find the optimal solution offline, not to mention we want it online. Instead of a heuristic that comes with no optimality guarantee, we would want an algorithm that provides integer solutions which incur the total cost no greater than a constant (i.e., the *multiplicative integrality gap*) times the total cost of the optimal solutions.

Algorithmic Idea. We remove the integral constraints to relax our problem to a linear program. For this linear relaxation, we design an online algorithm based on the *regularization* technique [17], i.e., by replacing the nonlinear switching cost in the objective with a carefully-designed logarithmic function, so that with zero knowledge about the future we can make control and workload distribution decisions with a competitive ratio r_1 by solving a series of the "regularized" one-shot problems at the corresponding time slots in an online fashion. Afterwards, based on the *pipage rounding* technique [18], [19], we design a randomized pairwise rounding algorithm running at each time slot to round the fractional decisions of servers and cloudlets into integers with a multiplicative integrality gap r_2, while maintaining feasible workload distributions. $r = r_1 r_2$ is the overall approximation ratio of our approach.

B. Proposed Algorithms

Notations and Formulations. To describe our algorithms formally, we introduce additional notations used throughout the rest of this paper: $\mathbf{P}$ is the relaxed problem of our original

problem over time; $\mathbf{P_t}$ is the one-shot problem at t for the relaxed problem $\mathbf{P}$; $\widetilde{\mathbf{P}}_\mathbf{t}$ is the regularized problem of $\mathbf{P_t}$. P, $\mathsf{P}_\mathbf{t}$, and $\widetilde{\mathsf{P}}_\mathbf{t}$ denote the objective functions of their corresponding problems. We use $\mathbf{D}$ to refer to the Lagrange dual problem of $\mathbf{P}$, and use D to refer to the objective function of this dual problem. $\mathbf{x_t}$, $\mathbf{y_t}$, $\mathbf{z_t}$ are shorthand for x_{ijt}, y_{it}, z_{it}, $\forall i$, $\forall j$, $\forall t$, respectively. We use different diacritical marks together with such variable symbols to represent the solutions obtained by solving the different problems, e.g., $\widetilde{\mathbf{z}}_\mathbf{t}$ denotes the (fractional) solution by solving the regularized problem and $\bar{\mathbf{z}}_\mathbf{t}$ denotes the (integral) solution after applying the rounding algorithm. We have more notations as such in Algorithm 1 and Figure 2. We also have some ad-hoc, auxiliary notations in Algorithm 2.

The linear relaxation of our problem is $\mathbf{P} = \sum_t \mathbf{P_t}$, with $\mathbf{P_t}$ as below. We remove (1f), and afterwards, we introduce the additional variables w_{it}, v_{it}, $\forall i$, $\forall t$ and the additional constraints (2b) $\sim$ (2d), without changing the optimal fractional solutions of $\mathbf{x_t}$, $\mathbf{y_t}$, and $\mathbf{z_t}$.

$$
\begin{aligned}
\min \quad & \mathsf{P_t} = \textstyle\sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_i p^b_{it} z_{it} \\
& + \textstyle\sum_i c^s_i w_{it} + \sum_i c^b_i v_{it} \\
\text{s.t.} \quad & w_{it} \geq y_{it} - y_{it-1}, \forall i, && \text{(2a)} \\
& v_{it} \geq z_{it} - z_{it-1}, \ \forall i, && \text{(2b)} \\
& w_{it} \geq 0, \qquad \forall i, && \text{(2c)} \\
& v_{it} \geq 0, \qquad \forall i, && \text{(2d)} \\
& \text{(1a)} \sim \text{(1e)}, \text{without ``}\forall t\text{''}.
\end{aligned}
$$

The regularized one-shot slice $\widetilde{\mathbf{P}}_\mathbf{t}$ is as below, which uses special logarithmic terms to replace the original switching cost. Note $\widetilde{\mathbf{P}}_\mathbf{t}$ uses the optimal solution of $\widetilde{\mathbf{P}}_\mathbf{t-1}$ as the input. Also, we have $\sigma_i = \ln\left(1 + \frac{C_i}{\varepsilon}\right)$, $\forall i$ and $\sigma' = \ln\left(1 + \frac{1}{\varepsilon}\right)$, where $\varepsilon > 0$ is a configurable parameter of our algorithm.

$$
\begin{aligned}
\min \quad & \widetilde{\mathsf{P}}_\mathsf{t} = \textstyle\sum_i \sum_j d_{ij} x_{ijt} \\
& + \textstyle\sum_i p^s_{it} y_{it} + \sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& + \textstyle\sum_i p^b_{it} z_{it} + \sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\
\text{s.t.} \quad & \text{(1a)} \sim \text{(1e)}, \text{without ``}\forall t\text{''}.
\end{aligned}
$$

Algorithms. We design our online algorithm as Algorithm 1, based on the definitions of $\mathbf{P_t}$ and $\widetilde{\mathbf{P}}_\mathbf{t}$. It iteratively invokes our *randomized pairwise dependent rounding* algorithm, i.e., Algorithm 2, to convert the fractional solutions to the integral solutions in an online manner. The name "dependent rounding" is due to the fact that in our case $\widetilde{\mathbf{y}}_\mathbf{t}$ and $\widetilde{\mathbf{z}}_\mathbf{t}$ cannot be rounded independently, as they are connected via the constraints, and they always need to be rounded altogether in order to keep the integral solutions still satisfying the constraints after rounding.

Algorithm 1 runs at t, takes the optimal fractional solution $(\widetilde{\mathbf{x}}_\mathbf{t-1}, \widetilde{\mathbf{y}}_\mathbf{t-1}, \widetilde{\mathbf{z}}_\mathbf{t-1})$ from $t-1$ as the input, and solves $\widetilde{\mathbf{P}}_\mathbf{t}$ to obtain the optimal fractional solution $(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t})$ for t. Afterwards, it invokes Algorithm 2 to round the fractional $\widetilde{\mathbf{z}}_\mathbf{t}$ to the integral $\bar{\mathbf{z}}_\mathbf{t}$. Next, it takes $\bar{\mathbf{z}}_\mathbf{t-1}$ as input, fixes $\bar{\mathbf{z}}_\mathbf{t}$, and solves $\mathbf{P_t}$ to obtain the solution $(\mathbf{x^*_t}, \mathbf{y^*_t}, \bar{\mathbf{z}}_\mathbf{t})$. It then invokes Algorithm 2 again but to round the fractional $\mathbf{y^*_t}$ into the integral $\bar{\mathbf{y}}_\mathbf{t}$. Finally, it takes $(\bar{\mathbf{y}}_\mathbf{t-1}, \bar{\mathbf{z}}_\mathbf{t-1})$ as input, fixes $(\bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$, and solves $\mathbf{P_t}$ to obtain the solution $(\mathbf{x^{**}_t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$. Figure 2 illustrates the online execution of our algorithms.

Algorithm 1: Online algorithm, $\forall t$

1. Solve $\widetilde{\mathbf{P}}_\mathbf{t}$ to obtain its solution $(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t})$;
2. Invoke Algorithm 2 to round $(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t})$ to $(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$;
3. Fix $(\bar{\mathbf{z}}_\mathbf{t})$, solve $\mathbf{P_t}$ to obtain its solution $(\mathbf{x^*_t}, \mathbf{y^*_t}, \bar{\mathbf{z}}_\mathbf{t})$;
4. Invoke Algorithm 2 to round $(\mathbf{x^*_t}, \mathbf{y^*_t}, \bar{\mathbf{z}}_\mathbf{t})$ to $(\mathbf{x^*_t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$;
5. Fix $(\bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$, solve $\mathbf{P_t}$ to obtain its solution $(\mathbf{x^{**}_t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$.

Algorithm 2: Randomized pairwise rounding, $\forall t$

1. To round $\widetilde{\mathbf{z}}_\mathbf{t}$, replace $\bar{u}_{it}$ by $\bar{z}_{it}$, $\widehat{u}_{it}$ by $\widetilde{z}_{it}$, and U_i by C_i, $\forall i$;
2. To round $\mathbf{y^*_t}$, replace $\bar{u}_{it}$ by $\bar{y}_{it}$, $\widehat{u}_{it}$ by y^*_{it}, and U_i by $\frac{1}{R_i}$, $\forall i$;
3. $\theta_{it} \stackrel{\text{def}}{=} \widehat{u}_{it} - \lfloor \widehat{u}_{it} \rfloor$, $\forall i$;
4. $\mathcal{I}'_t \stackrel{\text{def}}{=} \mathcal{I} \setminus \{i \,|\, \theta_{it} \in \{0, 1\}\}$;
5. **while** $|\mathcal{I}'_t| > 1$ **do**
6. Select i_1, $i_2 \in \mathcal{I}'$, where $i_1 \neq i_2$;
7. $\omega_1 \stackrel{\text{def}}{=} \min\{1 - \theta_{i_1 t}, \frac{U_{i_2}}{U_{i_1}} \theta_{i_2 t}\}$,
8. $\omega_2 \stackrel{\text{def}}{=} \min\{\theta_{i_1 t}, \frac{U_{i_2}}{U_{i_1}} (1 - \theta_{i_2 t})\}$;
9. With the probability $\frac{\omega_2}{\omega_1 + \omega_2}$, set
10. $\theta'_{i_1 t} = \theta_{i_1 t} + \omega_1$, $\theta'_{i_2 t} = \theta_{i_2 t} - \frac{U_{i_1}}{U_{i_2}} \omega_1$;
11. With the probability $\frac{\omega_1}{\omega_1 + \omega_2}$, set
12. $\theta'_{i_1 t} = \theta_{i_1 t} - \omega_2$, $\theta'_{i_2 t} = \theta_{i_2 t} + \frac{U_{i_1}}{U_{i_2}} \omega_2$;
13. Set $\bar{u}_{i_1 t} = \lfloor \widehat{u}_{i_1 t} \rfloor + \theta'_{i_1 t}, \mathcal{I}'_t = \mathcal{I}'_t \setminus \{i_1\}$, if $\theta'_{i_1 t} \in \{0, 1\}$;
14. Set $\bar{u}_{i_2 t} = \lfloor \widehat{u}_{i_2 t} \rfloor + \theta'_{i_2 t}, \mathcal{I}'_t = \mathcal{I}'_t \setminus \{i_2\}$, if $\theta'_{i_2 t} \in \{0, 1\}$;
15. **end**
16. **if** $|\mathcal{I}'_t| = 1$ **then**
17. Set $\bar{u}_{it} = \lceil \widehat{u}_{it} \rceil$ for the only $i \in \mathcal{I}'_t$;
18. **end**

Algorithm 2 is used to round $\widetilde{\mathbf{z}}_\mathbf{t}$ and $\mathbf{y^*_t}$ to $\bar{\mathbf{z}}_\mathbf{t}$ and $\bar{\mathbf{y}}_\mathbf{t}$, respectively, and thus needs to be invoked twice in Algorithm 1. Let us take $\widetilde{\mathbf{z}}_\mathbf{t}$ as an example to interpret this algorithm. The main loop is Line 5 through 15, where in each specific iteration either Line 10 or 12 is executed. This main loop ensures three things. Firstly, either $\widetilde{z}_{i_1 t}$, $\widetilde{z}_{i_2 t}$, or both are rounded into the integer(s) after every iteration of the loop. Secondly, we have $C_{i_1} \theta'_{i_1 t} + C_{i_2} \theta'_{i_2 t} = C_{i_1} \theta_{i_1 t} + C_{i_2} \theta_{i_2 t}$ after every iteration. This is the key for keeping the integral solutions after rounding still satisfying the constraints of $\mathbf{P_t}$. Thirdly, we have $\mathsf{E}(\bar{z}_{it}) = \widetilde{z}_{it}$, $\forall i \in \mathcal{I} \setminus \mathcal{I}'_t$ after the loop, where E denotes the expected value. This is useful for deriving the integrality gap, as shown next.

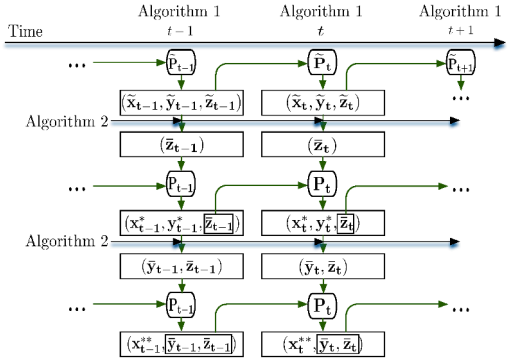

Fig. 2: Online execution of Algorithms 1 and 2

IV. Performance Analysis

To derive the performance bounds provided by our proposed algorithms, we establish a chain of inequalities, where r_1 and r_2 are constants:

$$\begin{aligned}
&\mathsf{E}(\mathsf{P}(\{\mathbf{x_t^{**}}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\})) && \text{(4a)}\\
&\leq r_2 \mathsf{P}(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\}) && \text{(4b)}\\
&\leq r_1 r_2 \mathsf{D}(\{\pi(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}), \forall t\}) && \text{(4c)}\\
&\leq r_1 r_2 \mathsf{P}^{opt}. && \text{(4d)}
\end{aligned}$$

Having introduced *randomized* rounding in our algorithms, we bound the *expectation* of the total cost over time of our online, integral decisions, i.e., $\mathsf{E}(\mathsf{P}(\{\mathbf{x_t^{**}}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\}))$ in (4a), by the total cost over time of the offline, optimal fractional decisions, i.e., P^{opt} in (4d). P^{opt} is an intrinsic lower bound of the optimum of our original problem with integral variables, because the problem $\mathbf{P}$ is a relaxation of our original problem. Thus, $r_1 r_2$ also applies to the optimum of our original problem.

We further divide the derivation chain into two parts. The competitiveness refers to (4b) $\leq$ (4d), i.e., we bound "fractional online" by r_1 times "fractional offline". The integrality gap refers to (4a) $\leq$ (4b), i.e., we bound "integral online" by r_2 times "fractional online". To prove (4b) $\leq$ (4d), we use the Lagrange dual problem $\mathbf{D}$ as a bridge [17], and we only need to prove that (4b) $\leq$ (4c), because (4c) $\leq$ (4d) holds naturally due to weak duality. π is a mapping that we need to construct in order to convert a primal solution to a feasible dual solution so that it can be evaluated in the objective function D.

A. Competitive Ratio

We establish $\mathsf{P}(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\}) \leq r_1 \mathsf{D}(\{\pi(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}), \forall t\})$ and derive r_1 in this section. We derive and write the Lagrange dual problem $\mathbf{D}$, the optimality conditions of $\widetilde{\mathbf{P}}_\mathbf{t}$, the mapping π, and based on all of these, we show Theorem 1 to derive r_1.

Deriving the Lagrange Dual Problem. We derive $\mathbf{P}$'s dual problem $\mathbf{D}$, where α_{jt}, β_{it}, ρ_{it}, γ_{ijt}, μ_{it}, ϕ_{it}, and τ_{it} are the corresponding dual variables:

$$\begin{aligned}
\max\quad & \mathsf{D} = \textstyle\sum_t \sum_j \lambda_{jt}\alpha_{jt} + \sum_t \sum_i (\sum_j \lambda_{jt} - \frac{C_i}{R_i})\mu_{it} && \\
\text{s.t.}\quad & d_{ij} - \alpha_{jt} + \beta_{it} - \gamma_{ijt} = 0, \forall j, \forall i, \forall t, && \text{(5a)}\\
& p^s_{it} - \tfrac{\beta_{it}}{R_i} + \tfrac{\rho_{it}}{R_i} + \phi_{it} - \phi_{it+1} = 0, \forall i, \forall t, && \text{(5b)}\\
& p^b_{it} - \tfrac{C_i\rho_{it}}{R_i} + \tfrac{C_i\mu_{it}}{R_i} - \tfrac{C_i}{R_i}\textstyle\sum_i \mu_{it} + \tau_{it} - \tau_{it+1} = 0, \\
& \qquad \forall i, \forall t, && \text{(5c)}\\
& c^s_i - \phi_{it} \geq 0, \forall i, \forall t, && \text{(5d)}\\
& c^b_i - \tau_{it} \geq 0, \forall i, \forall t, && \text{(5e)}\\
& \text{all dual variables} \geq 0. && \text{(5f)}
\end{aligned}$$

Characterizing the Regularized Solution. In the meantime, we note $\widetilde{\mathbf{P}}_\mathbf{t}$'s optimal (primal) solution $(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t})$ satisfies its Karush-Kuhn-Tucker (KKT) conditions. Here, we transform $\widetilde{\mathbf{P}}_\mathbf{t}$ to an equivalent form [22], as below:[1]

$$\begin{aligned}
\min\quad & \widetilde{\mathsf{P}}_\mathsf{t} && \\
\text{s.t.}\quad & \text{(1a)} \sim \text{(1d), without "}\forall t\text{"}, && \\
& \textstyle\sum_i \frac{C_i}{R_i} z_{it} - \frac{C_i}{R_i} z_{it} \geq \sum_j \lambda_{jt} - \frac{C_i}{R_i}, \forall i. && \text{(6a)}
\end{aligned}$$

With this new form, using α_j, β_i, ρ_i, γ_{ij}, and μ_i to denote the optimal dual solution, we have $\widetilde{\mathbf{P}}_\mathbf{t}$'s KKT conditions:

$$\begin{aligned}
& d_{ij} - \alpha_j + \beta_i - \gamma_{ij} = 0, \forall j, \forall i, && \text{(7a)}\\
& p^s_{it} - \tfrac{\beta_i}{R_i} + \tfrac{\rho_i}{R_i} + \tfrac{c^s_i}{\sigma_i} \ln \tfrac{\widetilde{y}_{it}+\varepsilon}{\widetilde{y}_{it-1}+\varepsilon} = 0, \forall i, && \text{(7b)}\\
& p^b_{it} - \tfrac{C_i\rho_i}{R_i} - \tfrac{C_i}{R_i}\textstyle\sum_i \mu_i + \tfrac{C_i}{R_i}\mu_i + \tfrac{c^b_i}{\sigma'} \ln \tfrac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} = 0, \forall i, && \text{(7c)}\\
& \alpha_j(\textstyle\sum_i \widetilde{x}_{ijt} - \lambda_j) = 0, \forall j, && \text{(7d)}\\
& \beta_i(\tfrac{\widetilde{y}_{it}}{R_i} - \textstyle\sum_j \widetilde{x}_{ijt}) = 0, \forall i, && \text{(7e)}\\
& \rho_i(\tfrac{C_i}{R_i}\widetilde{z}_{it} - \tfrac{\widetilde{y}_{it}}{R_i}) = 0, \forall i, && \text{(7f)}\\
& \gamma_{ij}\widetilde{x}_{ijt} = 0, \forall j, \forall i, && \text{(7g)}\\
& \mu_i(\textstyle\sum_i \frac{C_i}{R_i}\widetilde{z}_{it} - \frac{C_i}{R_i}\widetilde{z}_{it} - (\sum_j \lambda_{jt} - \frac{C_i}{R_i})) = 0, \forall i, && \text{(7h)}\\
& \text{primal and dual solutions} \geq 0. && \text{(7i)}
\end{aligned}$$

Construct the Mapping. We use a mapping π to jointly map $\widetilde{\mathbf{P}}_\mathbf{t}$'s optimal primal solution and optimal dual solution to a feasible solution of $\mathbf{D}$ at t, denoted by $\pi(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t})$. We do not write the optimal dual solution in the notation $\pi(\cdot)$ for the ease of presentation. We construct π as below, and it can be verified that the constructed solution satisfies (5a) $\sim$ (5f):

$$\begin{aligned}
&\alpha_{jt} = \alpha_j, \forall j\,; \beta_{it} = \beta_i, \forall i\,; \rho_{it} = \rho_i, \forall i\,; \gamma_{ijt} = \gamma_{ij}, \forall j, \forall i\,;\\
&\phi_{it} = \tfrac{c^s_i}{\sigma_i} \ln \tfrac{C_i+\varepsilon}{\widetilde{y}_{it-1}+\varepsilon}, \forall i\,; \tau_{it} = \tfrac{c^b_i}{\sigma'} \ln \tfrac{1+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon}, \forall i\,; \mu_{it} = \mu_i, \forall i.
\end{aligned}$$

Bounding. Using the KKT conditions (7a) $\sim$ (7i) and the mapping π, we bound the operational cost and the switching cost in $P(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\})$, respectively.

Theorem 1. *$\mathsf{P}(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\}) \leq r_1 \mathsf{D}(\{\pi(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}), \forall t\})$, where $r_1 = 1 + (1+\varepsilon)\ln(1+\frac{1}{\varepsilon})\sum_i \frac{C_i}{R_i} + \max_i\{(C_i+\varepsilon)\ln(1+\frac{C_i}{\varepsilon})\}\sum_i \frac{1}{R_i}$.*

Proof. **Step 1:** Bounding the operational cost

$$\begin{aligned}
& \textstyle\sum_t \sum_i \sum_j d_{ij}\widetilde{x}_{ijt} + \sum_t \sum_i p^s_{it}\widetilde{y}_{it} + \sum_t \sum_i p^b_{it}\widetilde{z}_{it} && \\
&\leq \textstyle\sum_t \sum_i \sum_j (\alpha_{jt} - \beta_{it} + \gamma_{ijt})\widetilde{x}_{ijt} && \\
&\quad + \textstyle\sum_t \sum_i (\frac{\beta_{it}}{R_i} - \frac{\rho_{it}}{R_i} - \frac{c^s_i}{\sigma_i} \ln \frac{\widetilde{y}_{it}+\varepsilon}{\widetilde{y}_{it-1}+\varepsilon})\widetilde{y}_{it} && \\
&\quad + \textstyle\sum_t \sum_i (\frac{C_i}{R_i}\rho_{it} + \frac{C_i}{R_i}\sum_i \mu_{it} - \frac{C_i}{R_i}\mu_{it} - \frac{c^b_i}{\sigma'} \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon})\widetilde{z}_{it} && \text{(9a)}\\
&\leq \textstyle\sum_t \sum_i \sum_j (\alpha_{jt} - \beta_{it} + \gamma_{ijt})\widetilde{x}_{ijt} && \\
&\quad + \textstyle\sum_t \sum_i (\frac{\beta_{it}}{R_i} - \frac{\rho_{it}}{R_i})\widetilde{y}_{it} && \\
&\quad + \textstyle\sum_t \sum_i (\frac{C_i}{R_i}\rho_{it} + \frac{C_i}{R_i}\sum_i \mu_{it} - \frac{C_i}{R_i}\mu_{it})\widetilde{z}_{it} && \text{(9b)}\\
&= \textstyle\sum_t \sum_i \sum_j \alpha_{jt}\widetilde{x}_{ijt} + \sum_t \sum_i (\frac{C_i}{R_i}\sum_i \mu_{it} - \frac{C_i}{R_i}\mu_{it})\widetilde{z}_{it} && \text{(9c)}\\
&= \textstyle\sum_t \sum_j \lambda_{jt}\alpha_{jt} + \sum_t \sum_i (\sum_j \lambda_{jt} - \frac{C_i}{R_i})\mu_{it} && \text{(9d)}\\
&= \mathsf{D} &&
\end{aligned}$$

(9a) is due to (7a)$\sim$(7c). (9c) is due to (7e)$\sim$(7g). (9d) follows from (7d) and (7h). (9b) follows from $\sum_t \widetilde{y}_{it} \ln \frac{\widetilde{y}_{it}+\varepsilon}{\widetilde{y}_{it-1}+\varepsilon} \geq 0$ and $\sum_t \widetilde{z}_{it} \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} \geq 0$. We show the latter, and the former can be shown analogously. We rewrite its left-hand side as $\sum_t \widetilde{z}_{it} \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} = \sum_t (\widetilde{z}_{it}+\varepsilon) \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} - \sum_t \varepsilon \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon}$ and then we have the following, $\forall i$:

$$\textstyle\sum_t (\widetilde{z}_{it}+\varepsilon) \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} - \sum_t \varepsilon \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon}$$

[1] In fact, this transformation introduces a number of constraints, all of which are analogous to (6a). Due to the redundancy, we do not write all of them. Our analytic technique here indeed applies, if one desires to work with all the constraints—new dual variables, KKT conditions, and corresponding terms in the derivations just need to be added. This transformation is required for our performance analysis, but not required for solving $\widetilde{\mathbf{P}}_\mathbf{t}$ in our algorithms.

$$\geq \left(\sum_t (\widetilde{z}_{it}+\varepsilon)\right) \ln \frac{\sum_t (\widetilde{z}_{it}+\varepsilon)}{\sum_t (\widetilde{z}_{it-1}+\varepsilon)} + (\widetilde{z}_{i0}+\varepsilon) \ln \frac{\widetilde{z}_{i0}+\varepsilon}{\widetilde{z}_{iT}+\varepsilon} \quad (10a)$$
$$\geq \sum_t (\widetilde{z}_{it}+\varepsilon) - \sum_t (\widetilde{z}_{it-1}+\varepsilon) + \widetilde{z}_{i0} - \widetilde{z}_{iT} \quad (10b)$$
$$= 0.$$

(10a) follows from (11a) below, and (10b) follows from (11b) below. We also leverage $\widetilde{z}_{i0}=0$, $\forall i$, by our definition. (11a) and (11b) are the two facts:

$$\left(\sum_n p_n\right) \ln \frac{\sum_n p_n}{\sum_n q_n} \leq \sum_n p_n \ln \frac{p_n}{q_n}, \forall p,q>0, \quad (11a)$$
$$p-q \leq p \ln \frac{p}{q}, \forall p,q>0. \quad (11b)$$

Step 2: Bounding the switching cost

Firstly, we have $\eta' = (1+\varepsilon)\sigma'$ and define $\mathcal{I}'_{t,z} \stackrel{\text{def}}{=} \{i \,|\, \widetilde{z}_{it} > \widetilde{z}_{it-1}\}$, $\mathcal{I}''_{t,z} \stackrel{\text{def}}{=} \mathcal{I}'_{t,z} \cap \{i \,|\, \rho_{it}>0\}$, $\mathcal{I}'''_{t,z} \stackrel{\text{def}}{=} \mathcal{I}''_{t,z} \cap \{i \,|\, \beta_{it}>0\}$. We bound the switching cost incurred by $\widetilde{\mathbf{z}}_\mathbf{t}$:

$$\sum_t \sum_i c_i^b \left(\widetilde{z}_{it} - \widetilde{z}_{it-1}\right)^+$$
$$= \sum_t \sum_{i\in\mathcal{I}'_{t,z}} c_i^b \left(\widetilde{z}_{it} - \widetilde{z}_{it-1}\right) \quad (12a)$$
$$\leq \sum_t \sum_{i\in\mathcal{I}'_{t,z}} c_i^b (\widetilde{z}_{it}+\varepsilon) \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} \quad (12b)$$
$$\leq \eta' \sum_t \sum_{i\in\mathcal{I}'_{t,z}} \frac{c_i^b}{\sigma'} \ln \frac{\widetilde{z}_{it}+\varepsilon}{\widetilde{z}_{it-1}+\varepsilon} \quad (12c)$$
$$\leq \eta' \sum_t \sum_{i\in\mathcal{I}'_{t,z}} \left(\frac{C_i}{R_i}\rho_{it} + \frac{C_i}{R_i}\sum_i \mu_{it}\right) \quad (12d)$$
$$\leq \eta' \sum_t \sum_{i\in\mathcal{I}''_{t,z}} \left(C_i\left(\frac{\beta_{it}}{R_i} - \frac{c_i^s}{\sigma_i} \ln \frac{\widetilde{y}_{it}+\varepsilon}{\widetilde{y}_{it-1}+\varepsilon}\right) + \frac{C_i}{R_i}\sum_i \mu_{it}\right) \quad (12e)$$
$$\leq \eta' \sum_t \sum_{i\in\mathcal{I}''_{t,z}} \left(\frac{C_i\beta_{it}}{R_i} + \frac{C_i}{R_i}\sum_i \mu_{it}\right) \quad (12f)$$
$$\leq \eta' \sum_t \sum_{i\in\mathcal{I}'''_{t,z}} \left(\frac{C_i}{R_i}(\alpha_{jt}+\gamma_{ijt}) + \frac{C_i}{R_i}\sum_i \mu_{it}\right) \quad (12g)$$
$$= \eta' \sum_t \sum_{i\in\mathcal{I}'''_{t,z}} \left(\frac{C_i}{R_i}\alpha_{jt} + \frac{C_i}{R_i}\sum_i \mu_{it}\right) \quad (12h)$$
$$\leq \eta' \sum_t \sum_{i\in\mathcal{I}'''_{t,z}} \left(\frac{C_i}{R_i}\lambda_{jt}\alpha_{jt} + \frac{C_i}{R_i}\sum_i (\sum_j \lambda_{jt} - \frac{C_i}{R_i})\mu_{it}\right) \quad (12i)$$
$$\leq (1+\varepsilon)\ln(1+\tfrac{1}{\varepsilon})\sum_i \frac{C_i}{R_i}\mathsf{D} \quad (12j)$$

(12a) is by the definition of $\mathcal{I}'_{t,z}$. (12b) follows from (11b). (12c) follows, because of $\widetilde{z}_{it} \leq 1$. (12d) follows from (7c). (12e) follows from (7b). (12f) follows, due to $\widetilde{y}_{it} > \widetilde{y}_{it-1}$, which is further because of the following: in (12e), we have $\rho_{it}>0$; due to (7f), we have $\widetilde{y}_{it} = C_i\widetilde{z}_{it} > C_i\widetilde{z}_{it-1} = \widetilde{y}_{it-1}$. (12g) follows from (7a). (12h) follows, because of $\gamma_{ijt}=0$. This is further because of the following. Due to $\mathcal{I}'_{t,z}$, $\rho_{it}>0$ and (7f), we get $\widetilde{y}_{it} = C_i\widetilde{z}_{it} > 0$. Using $\beta_{it}>0$ and (7e), we reach $\sum_j \widetilde{x}_{ijt} = \frac{\widetilde{y}_{it}}{R} > 0$, and thus there exists at least one j such that $\widetilde{x}_{ijt} > 0$. With this $\widetilde{x}_{ijt}$, we reach $\gamma_{ijt}=0$ following (7g). (12i) follows, due to $\lambda_{jt} \geq 1$ and $\sum_j \lambda_{jt} - \frac{C_i}{R_i} \geq 1$. Note that this is because we have required λ_{jt}, $\forall j$, $\forall t$ and $\frac{C_i}{R_i}$, $\forall i$ to be integral; also note that even if $\sum_j \lambda_{jt} - \frac{C_i}{R_i} \leq 0$ is the case, then D changes correspondingly and we still reach (12j), following the definition of the new D.

Secondly, we have $\eta_{\max} = \max_i\{(C_i+\varepsilon)\sigma_i\}$, and define $\mathcal{I}'_{t,y} \stackrel{\text{def}}{=} \{i \,|\, \widetilde{y}_{it} > \widetilde{y}_{it-1}\}$ and $\mathcal{I}''_{t,y} \stackrel{\text{def}}{=} \mathcal{I}'_{t,y} \cap \{i \,|\, \beta_{it}>0\}$. We bound the switching cost incurred by $\widetilde{\mathbf{y}}_\mathbf{t}$:

$$\sum_t \sum_i c_i^s \left(\widetilde{y}_{it} - \widetilde{y}_{it-1}\right)^+$$
$$\leq \eta_{\max} \sum_t \sum_{i\in\mathcal{I}'_{t,y}} \frac{c_i^s}{\sigma_i} \ln \frac{\widetilde{y}_{it}+\varepsilon}{\widetilde{y}_{it-1}+\varepsilon} \quad (13a)$$
$$\leq \eta_{\max} \sum_t \sum_{i\in\mathcal{I}''_{t,y}} \frac{\beta_{it}}{R_i} \quad (13b)$$
$$\leq \max_i\{(C_i+\varepsilon)\ln(1+\tfrac{C_i}{\varepsilon})\} \sum_i \tfrac{1}{R_i}\mathsf{D} \quad (13c)$$

(13a)$\sim$(13c) omit the details, as they follow highly analogous derivations as (12c)$\sim$(12j). In particular, the involved KKT conditions are (7a), (7b), (7e), and (7g). □

B. Integrality Gap

We establish $\mathsf{E}(\mathsf{P}(\{\mathbf{x}_\mathbf{t}^{**}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\})) \leq r_2 \mathsf{P}(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\})$ and derive r_2 in this section. We firstly exhibit the existence of $\{\mathbf{x}_\mathbf{t}^{**}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\}$ via Lemmas 1 and 2. Afterwards, we present Lemma 3, following which we show Theorem 2 to derive r_2.

Lemma 1. *$\mathbf{P_t}$ is feasible, i.e., $(\mathbf{x}_\mathbf{t}^*, \mathbf{y}_\mathbf{t}^*)$ exists, given $\bar{\mathbf{z}}_\mathbf{t}$.*

Proof. As $\widetilde{\mathbf{z}}_\mathbf{t}$ exists, $\bar{\mathbf{z}}_\mathbf{t}$ always exists as a result of executing Algorithm 2. In Algorithm 2, in each iteration, no matter Line 10 or 12 is actually executed, the sum of $C_{i_1}\widetilde{z}_{i_1t} + C_{i_2}\widetilde{z}_{i_2t}$ stays constant. For example, in the case of Line 10, we have $C_{i_1}(\widetilde{z}_{i_1t}+\omega_1) + C_{i_2}(\widetilde{z}_{i_2t} - \omega_1\frac{C_{i_1}}{C_{i_2}}) = C_{i_1}\widetilde{z}_{i_1t} + C_{i_2}\widetilde{z}_{i_2t}$. That is, we have $\sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} C_i\bar{z}_{it} = \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} C_i\widetilde{z}_{it}$, after the loop of Line 5 through 15. Executing Line 16 through 18 can only lead to $\sum_{i\in\mathcal{I}} C_i\bar{z}_{it} = \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} C_i\bar{z}_{it} + \sum_{i\in\mathcal{I}'_t} C_i \geq \sum_{i\in\mathcal{I}} C_i\widetilde{z}_{it}$. Consequently, the constraints (1a)$\sim$(1e) remain feasible. The constraints (2a)$\sim$(2d) are always feasible. □

Lemma 2. *$\mathbf{P_t}$ is feasible, i.e., $\mathbf{x}_\mathbf{t}^{**}$ exists, given $(\bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$.*

Proof. As $\mathbf{y}_\mathbf{t}^*$ exists due to the previous lemma, $\bar{\mathbf{y}}_\mathbf{t}$ always exists as a result of executing Algorithm 2. Firstly, let us see $\bar{\mathbf{y}}_\mathbf{t}$ and $\bar{\mathbf{z}}_\mathbf{t}$ satisfy (1c). Note $y_{it}^* \leq C_i\bar{z}_{it}$, because $\mathbf{y}_\mathbf{t}^*$ is solved from $\mathbf{P_t}$ given $\bar{\mathbf{z}}_\mathbf{t}$; then, note no matter how $\mathbf{y}_\mathbf{t}^*$ is rounded, $\bar{y}_{it} \leq C_i\bar{z}_{it}$ always holds because C_i, $\forall i$ and $\bar{\mathbf{z}}_\mathbf{t}$ are integral. Secondly, let us see (1a) and (1b) are feasible due to Algorithm 2. For example, for Line 10, we have $\frac{1}{R_{i_1}}\theta'_{i_1t} + \frac{1}{R_{i_2}}\theta'_{i_2t} = \frac{1}{R_{i_1}}(\theta_{i_1t}+\omega_1) + \frac{1}{R_{i_2}}(\theta_{i_2t} - \omega_1\frac{R_{i_2}}{R_{i_1}}) = \frac{1}{R_{i_1}}\theta_{i_1t} + \frac{1}{R_{i_2}}\theta_{i_2t}$. Therefore, we have $\sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} \frac{1}{R_i}\bar{y}_{it} = \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} \frac{1}{R_i}(\lfloor y_{it}^*\rfloor + \theta'_{it}) = \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} \frac{1}{R_i}(\lfloor y_{it}^*\rfloor + \theta_{it}) = \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} \frac{1}{R_i} y_{it}^*$. Then, executing Line 16 through 18, we have $\sum_{i\in\mathcal{I}} \frac{1}{R_i}\bar{y}_{it} = \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} \frac{1}{R_i}\bar{y}_{it} + \sum_{i\in\mathcal{I}'_t} \frac{1}{R_i}\lceil y_{it}^*\rceil \geq \sum_{i\in\mathcal{I}} \frac{1}{R_i} y_{it}^*$, i.e., (1a) and (1b) are feasible. The constraints (2a)$\sim$(2d) are always feasible. □

Bounding. The objective of our relaxed problem $\mathbf{P}$ is the sum of five terms. When putting $(\mathbf{x}_\mathbf{t}^{**}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$ into P, we bound the expectation of each of the five terms. We connect each of them to the one term in particular, i.e., $\sum_t \sum_i p_{it}^b \widetilde{z}_{it}$, which is part of, and no greater than, $\mathsf{P}(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\})$.

Lemma 3. *For the random variable $\bar{z}_{it}$, $\forall i$, $\forall t$ and for every specific value it takes, we have*

$$\sum_t \sum_i C_i\bar{z}_{it} \leq (1+\kappa) \sum_t \sum_i C_i\widetilde{z}_{it},$$

where $\kappa = \max_t \kappa_t$, and $\kappa_t = \frac{\max_i C_i}{\min_i R_i \sum_j \lambda_{jt}}$, $\forall t$.

Proof.

$$\sum_t \sum_i C_i\bar{z}_{it}$$
$$= \sum_t \left(\sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} C_i\bar{z}_{it} + \sum_{i\in\mathcal{I}'_t} C_i\right) \quad (14a)$$
$$\leq \sum_t \left(\sum_{i\in\mathcal{I}} C_i\widetilde{z}_{it} + \max_{i\in\mathcal{I}} C_i\right) \quad (14b)$$
$$= \sum_t \left(\sum_i C_i\widetilde{z}_{it} + \kappa_t \min_i R_i \sum_j \lambda_{jt}\right) \quad (14c)$$

$$\leq \sum_t \left(\sum_i C_i \widetilde{z}_{it} + \kappa_t \sum_i \sum_j R_i \widetilde{x}_{ijt}\right) \quad (14d)$$
$$\leq \sum_t \left(\sum_i C_i \widetilde{z}_{it} + \kappa_t \sum_i \widetilde{y}_{it}\right) \quad (14e)$$
$$\leq \sum_t \left(\sum_i C_i \widetilde{z}_{it} + \kappa_t \sum_i C_i \widetilde{z}_{it}\right) \quad (14f)$$
$$\leq (1+\kappa) \sum_t \sum_i C_i \widetilde{z}_{it}$$

(14a) uses the index set $\mathcal{I}'_t$, which contains no more than one element after executing Algorithm 2, to expand the derivation. If $\mathcal{I}'_t = \varnothing$, then we will not have $\sum_{i\in\mathcal{I}'_t} C_i$ in (14a), but have $\sum_i C_i \bar{z}_{it} = \sum_i C_i \widetilde{z}_{it}$ due to Algorithm 2. We can reach (14b). If $\mathcal{I}'_t \neq \varnothing$, we will have $\sum_{i\in\mathcal{I}'_t} C_i \leq \max_{i\in\mathcal{I}} C_i$ due to $|\mathcal{I}'_t| = 1$, and also we will have $\sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} C_i \bar{z}_{it} \leq \sum_{i\in\mathcal{I}} C_i \widetilde{z}_{it}$ due to $\mathcal{I}\setminus\mathcal{I}'_t \in \mathcal{I}$ and Algorithm 2. We can then still reach (14b). (14c) simply uses the definition of κ. (14d), (14e), and (14f) use the constraints (1a), (1b), and (1c), respectively. □

Theorem 2. $\mathsf{E}(\mathsf{P}(\{\mathbf{x}^{**}_{\mathbf{t}}, \bar{\mathbf{y}}_{\mathbf{t}}, \bar{\mathbf{z}}_{\mathbf{t}}, \forall t\})) \leq r_2 \mathsf{P}(\{\widetilde{\mathbf{x}}_{\mathbf{t}}, \widetilde{\mathbf{y}}_{\mathbf{t}}, \widetilde{\mathbf{z}}_{\mathbf{t}}, \forall t\})$, *where* $r_2 = \delta_x + \delta_y + \delta_z + \delta_w + \delta_v$, *and*

$$\delta_x = (1+\kappa)\tfrac{\max_{i,j} d_{ij}}{\min_i R_i} \max_{i,t} \tfrac{C_i}{p^b_{it}},$$
$$\delta_y = (1+\kappa) \max_{i,t} p^s_{it} \max_{i,t} \tfrac{C_i}{p^b_{it}},$$
$$\delta_z = (1+\kappa) \max_{i,t} \tfrac{p^b_{it}}{C_i} \max_{i,t} \tfrac{C_i}{p^b_{it}},$$
$$\delta_w = (1+\kappa) \max_i c^s_i \max_{i,t} \tfrac{C_i}{p^b_{it}},$$
$$\delta_v = (1+\kappa) \max_i \tfrac{c^b_i}{C_i} \max_{i,t} \tfrac{C_i}{p^b_{it}}.$$

Proof. We exhibit δ_z, δ_y, δ_x, δ_v, and δ_w, respectively. Firstly, consider $\sum_t \sum_i p^b_{it} \bar{z}_{it}$:

$$\sum_t \sum_i p^b_{it} \bar{z}_{it}$$
$$= \sum_t \sum_i C_i \bar{z}_{it} \tfrac{p^b_{it}}{C_i} \quad (16a)$$
$$\leq \max_{i,t} \tfrac{p^b_{it}}{C_i} \sum_t \sum_i C_i \bar{z}_{it} \quad (16b)$$
$$\leq (1+\kappa) \max_{i,t} \tfrac{p^b_{it}}{C_i} \sum_t \sum_i C_i \widetilde{z}_{it} \quad (16c)$$
$$= (1+\kappa) \max_{i,t} \tfrac{p^b_{it}}{C_i} \sum_t \sum_i p^b_{it} \widetilde{z}_{it} \tfrac{C_i}{p^b_{it}} \quad (16d)$$
$$\leq \delta_z \sum_t \sum_i p^b_{it} \widetilde{z}_{it}$$

(16c) follows from Lemma 3. The above holds for any possible value that the random variable $\bar{z}_{it}$ takes, so we can directly add the expectation to the left-hand side:

$$\mathsf{E}\left(\sum_t \sum_i p^b_{it} \bar{z}_{it}\right) \leq \delta_z \sum_t \sum_i p^b_{it} \widetilde{z}_{it}.$$

Next, we consider the expectation of $\sum_t \sum_i p^s_{it} \bar{y}_{it}$:

$$\mathsf{E}(\textstyle\sum_t \sum_i p^s_{it} \bar{y}_{it})$$
$$= \textstyle\sum_t \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} p^s_{it} \Big(\frac{\omega_2}{\omega_1+\omega_2}(\theta_{it} - \frac{R_i}{R_{i'}}\omega_1) + \frac{\omega_1}{\omega_1+\omega_2}(\theta_{it} + \frac{R_i}{R_{i'}}\omega_2)\Big)$$
$$+ \textstyle\sum_t \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} p^s_{it} \lfloor y^*_{it} \rfloor + \sum_t \sum_{i\in\mathcal{I}'_t} p^s_{it} \lceil y^*_{it} \rceil \quad (17a)$$
$$= \textstyle\sum_t \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} p^s_{it} y^*_{it} + \sum_t \sum_{i\in\mathcal{I}'_t} p^s_{it} \lceil y^*_{it} \rceil \quad (17b)$$
$$\leq \max_{i,t} p^s_{it} \left(\textstyle\sum_t \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} y^*_{it} + \sum_t \sum_{i\in\mathcal{I}'_t} \lceil y^*_{it} \rceil\right) \quad (17c)$$
$$\leq \max_{i,t} p^s_{it} \left(\textstyle\sum_t \sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t} C_i \bar{z}_{it} + \sum_t \sum_{i\in\mathcal{I}'_t} C_i \bar{z}_{it}\right) \quad (17d)$$
$$\leq (1+\kappa) \max_{i,t} p^s_{it} \textstyle\sum_t \sum_i C_i \widetilde{z}_{it} \quad (17e)$$
$$\leq \delta_y \textstyle\sum_t \sum_i p^b_{it} \widetilde{z}_{it} \quad (17f)$$

(17a) shows the case of i_2 in Lines 10 and 12 of Algorithm 2, and it can lead to the same (17b) if i_1 is used. (17b) is the result of (17a) following Algorithm 2. (17c) and (17d) use the constraint (1c), and use $\lceil y^*_{it} \rceil \leq C_i \bar{z}_{it}$, $\forall i$, $\forall t$. (17e) follows from Lemma 3. (17f) follows analogously from (16c)∼(16d).

Afterwards, we consider $\sum_t \sum_i \sum_j d_{ij} x^{**}_{ijt}$ below. Because $\mathbf{x}^{**}_{\mathbf{t}}$ is not randomized, its "expectation" is itself.

$$\sum_t \sum_i \sum_j d_{ij} x^{**}_{ijt}$$
$$\leq \tfrac{\max_{i,j} d_{ij}}{\min_i R_i} \textstyle\sum_t \sum_i \bar{y}_{it} \quad (18a)$$
$$\leq \tfrac{\max_{i,j} d_{ij}}{\min_i R_i} \textstyle\sum_t \sum_i C_i \bar{z}_{it} \quad (18b)$$
$$\leq \delta_x \textstyle\sum_t \sum_i p^b_{it} \widetilde{z}_{it} \quad (18c)$$

(18a) and (18b) leverage the constraints (1b) and (1c), respectively. (18c) uses Lemma 3, and follows from an analogous derivation to (17d) ∼ (17f).

Finally, let us prove δ_v and δ_w:

$$\mathsf{E}\left(\textstyle\sum_t \sum_i c^b_i \left(\bar{z}_{it} - \bar{z}_{it-1}\right)^+\right)$$
$$\leq \mathsf{E}\left(\textstyle\sum_t \sum_i c^b_i \bar{z}_{it}\right) \quad (19a)$$
$$\leq (1+\kappa) \max_{i,t} \tfrac{C_i}{p^b_{it}} \max_i \tfrac{c^b_i}{C_i} \textstyle\sum_t \sum_i p^b_{it} \widetilde{z}_{it} \quad (19b)$$
$$\mathsf{E}\left(\textstyle\sum_t \sum_i c^s_i \left(\bar{y}_{it} - \bar{y}_{it-1}\right)^+\right)$$
$$\leq \mathsf{E}\left(\textstyle\sum_t \sum_i c^s_i \bar{y}_{it}\right) \quad (20a)$$
$$\leq (1+\kappa) \max_i c^s_i \max_{i,t} \tfrac{C_i}{p^b_{it}} \textstyle\sum_t \sum_i p^b_{it} \widetilde{z}_{it} \quad (20b)$$

(19a) and (20a) follow analogously from (12a). (19b) and (20b) omit all the details, and follow analogously from (16a)∼(16d) and (17a)∼(17f). We embed $v^{**}_{it} = \left(\bar{z}_{it} - \bar{z}_{ijt-1}\right)^+$ and $w^{**}_{it} = \left(\bar{y}_{it} - \bar{y}_{it-1}\right)^+$, $\forall i$, $\forall t$ into the above derivations. □

V. Experimental Study

A. Data and Settings

Cloudlets and Delay. We envisage the cloudlets deployment at London's underground stations. We assume all passengers can connect to the underground cloudlet network via their local WiFi. A passenger's requests may be served by any cloudlet, and if served by a remote one in the network, the service delay is approximated as the geographical distance between the local station and that remote station [20]. Out of London's 268 underground stations [21], we assume the cloudlets are located at the largest 100 stations based on the annual passenger count.

Workload and Processing. We use the dynamic passenger numbers at a station to represent the workload originated from that station. From Transport for London [21], we acquire such passenger data for every underground station, measured for every quarter (15 minutes) for a weekday, a Saturday, and a Sunday around Nov. 16, 2016. We consider a 1-week period of 672 quarters or time slots. As an example, Figure 3 depicts the workloads of the largest 3 stations, where we repeat the weekday data for 5 days to mimic Monday through Friday. Without loss of generality, we assume servers are homogenous and a server can process 1000 requests at a time slot.

Resource Price. We assume that all London's underground cloudlets are powered by the same wholesale electricity market. We use the hourly electricity price of European Electricity Index (ELIX) reported by EPEX SPOT [15] for Monday, Nov. 14 through Sunday, Nov. 20, 2016, also shown in Figure 3.

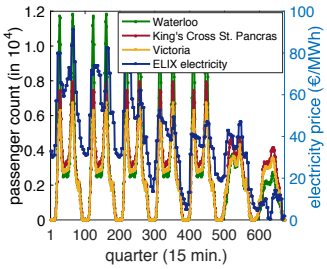

Fig. 3: Dynamic inputs

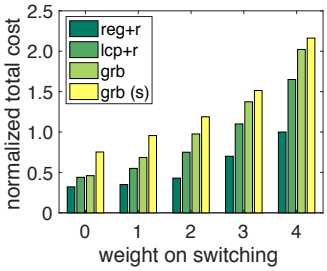

Fig. 4: Impact of switching cost

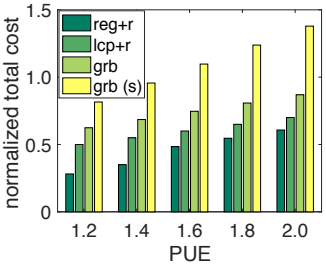

Fig. 5: Impact of the PUE

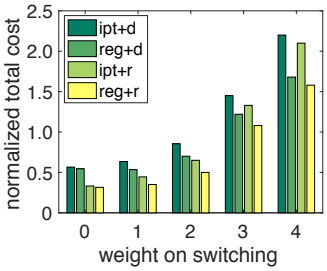

Fig. 6: Algorithm combinations

Cloudlet Capacity. We sum up the peak values of all the workloads from all the underground stations, divide that sum by the total number of cloudlets, translate that quotient into the number of servers, and use it as the cloudlet capacity. We use the workload to estimate the data center capacity [8].

Algorithms. We implement multiple algorithms. We also compose a *deterministic*, *best-effort rounding* algorithm below. Given the fractional solution $(\mathbf{x_t}, \mathbf{y_t}, \mathbf{z_t})$, for each cloudlet i, we use $\lfloor y_{it} \rfloor$ as the number of servers if it is a sufficient number for the workload $R_i \sum_j x_{ijt}$; otherwise, we use $\lceil y_{it} \rceil$. If the number of servers is zero, we use $\lfloor z_{it} \rfloor$, and the cloudlet is off; otherwise, we use $\lceil z_{it} \rceil$, and the cloudlet is on.

We adopt the two state-of-the-art solvers in our evaluations: IPOPT [23], for solving convex programs via barrier methods, and Gurobi [24], for solving mixed-integer linear programs via simplex, branch-and-bound, and other heuristics. We use "`grb`" to refer to using Gurobi to solve $\mathbf{P_t}$ with (1f), i.e., the integer constraint, at every t, "`ipt`" to refer to using IPOPT to solve $\mathbf{P_t}$ at every t, and "`reg`" to refer to using our regularized online algorithm that solves $\widetilde{\mathbf{P}}_\mathbf{t}$ at every t, with $\varepsilon = 0.001$.

To highlight the benefit of the multi-granularity control, we compare our algorithm "`reg+r`" to multiple others as follows, including the single granularity control "`grb(s)`":

- `reg+r`: regularization, randomized pairwise rounding;
- `lcp+r`: the Lazy Capacity Provisioning algorithm [3], randomized pairwise rounding;
- `grb`: Gurobi;
- `grb(s)`: Gurobi for server control only—an cloudlet is on if servers are non-zero, and is off otherwise.

To compare different combinations of fractional online algorithms and rounding algorithms, we further have the following:

- `ipt+d`: IPOPT, deterministic rounding as above;
- `reg+d`: regularization, deterministic rounding as above;
- `ipt+r`: IPOPT, randomized pairwise rounding.

We do not consider the offline optimal integral solution. It is not practical, as it takes an unacceptably long time for Gurobi to find it for the (even small) problem instances in our case.

Weights and PUE. We vary the weight of the switching cost to obtain a spectrum of results, so that we avoid interpreting the concrete metric represented by the switching cost, as it can capture a range of metrics as stated earlier. Given the weight χ, we vary $\log \chi$ as an integer in $[0, 4]$; we abuse the term "weight" and use it to refer to $\log \chi$ in our results. For the weight of the delay, we set it less than the weight of the switching cost, as cloudlets are close to users and connected via high-speed networks. We vary the PUE in $[1, 2]$, which contributes to the weight of the operational cost of the cloudlet; we fix the weight of the operational cost of the server as 1.

B. Evaluation Results

Figure 4 contrasts `reg+r` with the multi-granularity `lcp+r` and `grb`, and the single granularity `grb(s)`, as the weight of the switching cost increases. `reg+r` incurs about $15\%\sim40\%$, $30\%\sim50\%$, and $40\%\sim60\%$ less total cost than `lcp+r`, `grb`, and `grb(s)`, respectively. As the weight increases, the gap between `reg+r` and others expands, since the former handles the switching cost well; the gap between `grb(s)` and others shrinks, since it is more inclined to leave the servers always on and the cloudlet on, and whether to control the cloudlets separately becomes less influential. `lcp+r` does not do well, as its lazy capacity principle, designed for server control only, cannot suit well when controlling both servers and cloudlets.

Figure 5 compares the same group of algorithms as the PUE grows. `reg+r` incurs about $15\% \sim 65\%$ less total cost than all other algorithms. The cost incurred by `grb(s)` increases, and goes further away from other algorithms. This is because it does not control cloudlets and leaves more cloudlets on, and thus incurs more cost as the PUE grows. The cost incurred by `reg+r` increases as well, and gets closer to those of `lcp+r` and `grb`. This is because, despite `reg+r` handles the switching cost well, the operational cost becomes more important as the PUE grows, and thus its advantage decreases.

Figure 6 checks the performance of different combinations of the fractional online algorithms and the rounding algorithms. `reg+r` is the best, and has about $5\%\sim25\%$ less total cost than the next best algorithm in each case. As the weight of the switching cost grows, for all rounding algorithms, our `reg` is better and the gap between it and `ipt` becomes larger; for all fractional online algorithms, our randomized rounding `r` is better and the gap between it and the deterministic rounding `d` becomes smaller. Regularization handles the switching cost better, but the randomized rounding tends to sacrifice that advantage due to its random selection of fractional decisions.

Figures 7 visualizes the number of the active cloudlets as time goes. Single granularity `grb(s)` tends to involve more cloudlets. Multi-granularity `reg+r` controls the cloudlets as well and uses our iterative, randomized rounding process to aggregate the workloads to a fewer number of cloudlets.

Figure 8 focuses on the execution time of the algorithms on a MacBook Pro laptop with a 2.6 GHz CPU. Passengers from the 100 largest stations issue requests, and the $10\sim50$ largest

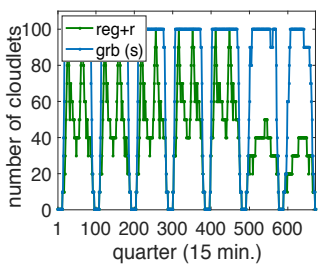

Fig. 7: Cloudlets usage

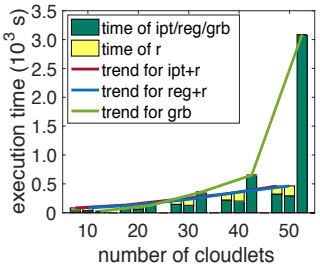

Fig. 8: Execution time

stations host cloudlets. `grb` is rather unscalable; `ipt+r` and `reg+r` scale much better and the execution time grows more slowly. Our randomized rounding `r` occupies about $30\% \sim 55\%$ of the total time of `ipt+r` and `reg+r`.

VI. Related Work

Previous works have studied the online control of servers in cloud data centers. Lin et al. [3] proposed the lazy capacity provisioning algorithm and proved its competitiveness. Lu et al. [4] controlled multiple resources such as on-site electricity generators and servers via online algorithms based on the "ski-rental" idea. Tu et al. [9] investigated the joint server control and job scheduling, and designed online algorithms also based on "ski-rental". Jiao et al. [10] studied the multi-tier clouds and devised regularization-based online algorithms to control the servers in clouds and the networks across clouds.

Such works either make assumptions that do not hold for our case (e.g., fractional server numbers, fixed or bounded prices), or lack of considerations of the challenges in our case (e.g., workload distribution, joint server and cloudlet control). Their algorithms are not directly adaptable to our problem.

There also exists research on edge cloud resource allocation and control. Hou et al. [11] worked on a cloudlet-cloud joint architecture and developed online algorithms to download services to cloudlets to handle the varying user requests. Chen et al. [2] focused on computation offloading for edge computing, and made both job offloading and resource allocation decisions. Wang et al. allocated resources and migrated workloads to accommodate user mobilities [12] and also allocated social-network-based service entities [13] at edge clouds.

Such works have not explored the potential of the multi-granularity control for cloudlets and are thus complementary to our work. Their solutions fall insufficient for our scenario.

VII. Conclusion

We propose and study the multiple granularity control of cloudlet networks to push the limits of edge computing beyond the current single granularity server control paradigm. We design an online algorithmic framework to make control decisions for servers, cloudlets, and workload distribution on the fly, with theoretically provable performance guarantees towards the offline optimum. We also conduct extensive experiments using large-scale real-world data to demonstrate and validate the practical advantages of our proposed approach.

Acknowledgement

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. CNS 1564348 and CNS 1703014, by the National Natural Science Foundation of China (NSFC) under Grant No. 61702287 and 61761136014, by the German Research Foundation (DFG) under Grant No. 392046569, and also by the DFG Collaborative Research Center 1053 MAKI. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, NSFC, or DFG.

References

[1] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet enhanced fiber-wireless access networks for mobile-edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3601–3618, 2017.

[2] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *IEEE INFOCOM*, 2017.

[3] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, 2013.

[4] L. Lu, J. Tu, C.-K. Chau, M. Chen, and X. Lin, "Online energy generation scheduling for microgrids with intermittent energy sources and co-generation," in *ACM SIGMETRICS*, 2013.

[5] M. T. Chaudhry, T. C. Ling, A. Manzoor, S. A. Hussain, and J. Kim, "Thermal-aware Scheduling in Green Data Centers," *ACM Computing Surveys*, vol. 47, no. 3, 2015.

[6] M. Ganeshalingam, A. Shehabi, and L.-B. Desroches, "Shining a Light on Small Data Centers in the US," in *EEDAL*, 2017.

[7] "Efficiency: How we do it - Data Centers - Google," https://www.google.com/about/datacenters/efficiency/internal.

[8] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening Geographical Load Balancing," in *ACM SIGMETRICS*, 2011.

[9] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *ACM e-Energy*, 2013.

[10] L. Jiao, A. Tulino, J. Llorca, Y. Jin, and A. Sala, "Smoothed Online Resource Allocation in Multi-tier Distributed Cloud Networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2556–2570, 2017.

[11] I.-H. Hou, T. Zhao, S. Wang, and K. Chan, "Asymptotically optimal algorithm for online reconfiguration of edge-clouds," in *ACM MOBIHOC*, 2016.

[12] L. Wang, L. Jiao, J. Li, and M. Mühlhäuser, "Online resource allocation for arbitrary user mobility in distributed edge clouds," in *IEEE ICDCS*, 2017.

[13] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in *IEEE INFOCOM*, 2018.

[14] M. C. Calzarossa, L. Massari, and D. Tessera, "Workload Characterization: A Survey Revisited," *ACM Computing Surveys*, vol. 48, no. 3, 2016.

[15] "EPEX SPOT SE: European Electricity Index (ELIX)," https://www.epexspot.com/en/market-data/elix.

[16] "Amazon EC2 Spot Instances Pricing," https://aws.amazon.com/ec2/spot/pricing/.

[17] N. Buchbinder, S. Chen, and J. S. Naor, "Competitive Analysis via Regularization," in *ACM-SIAM SODA*, 2014.

[18] A. A. Ageev and M. I. Sviridenko, "Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 307–328, 2004.

[19] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent Rounding and Its Applications to Approximation Algorithms," *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[20] "List of London Underground Stations - OpenStreetMap Wiki," http://wiki.openstreetmap.org/wiki/List_of_London_Underground_stations.

[21] "Our open data - Transport for London," https://tfl.gov.uk/info-for/open-data-users/our-open-data.

[22] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips, "Strengthening Integrality Gaps for Capacitated Network Design and Covering Problems," in *ACM-SIAM SODA*, 2000.

[23] "Interior Point Optimizer," https://projects.coin-or.org/Ipopt.

[24] "Gurobi Optimization - The State-of-the-Art Mathematical Programming Solver," http://www.gurobi.com.