Multiple Granularity Online Control of Cloudlet Networks for Edge Computing

Lei Jiao\textsuperscript{1}, Lingjun Pu\textsuperscript{2}, Lin Wang\textsuperscript{3}, Xiaojun Lin\textsuperscript{4}, Jun Li\textsuperscript{1}

\textsuperscript{1}University of Oregon, USA \quad \textsuperscript{2}Nankai University, China
\textsuperscript{3}TU Darmstadt, Germany \quad \textsuperscript{4}Purdue University, USA

IEEE SECON 2018
Network Scenario

**Figure:** Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

**Users:**
- Connect to a given cloudlet by contracts or principles (i.e., local cloudlet)
- Upload a portion of workloads to process at cloudlet on the fly
Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

Cloudlets:
- Limited processing capacity
- Fast wired connection among cloudlets
- User workloads processed at other cooperative cloudlets, not necessarily the local one
Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

Central Controller:
- From user perspective → Satisfied QoS (i.e., low latency)
- From cloudlet perspective → Satisfied OPEX (i.e., low energy cost)
Network Scenario

Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

Central Controller:
- From user perspective → Satisfied QoS (i.e., low latency)
- From cloudlet perspective → Satisfied OPEX (i.e., low energy cost)

Question: How to design a resource allocation policy to jointly achieve them?
Operating cost of activating the servers in cloudlets for time-varying inputs (i.e., user workloads) + User QoS (i.e., a function of latency)

- Inputs for the current time slot are known; future inputs all unknown
- One-shot optimum
**Operating cost** of activating the servers in cloudlets for time-varying inputs (e.g., user workloads)

- Inputs for the current time slot are known; future inputs all unknown
- Nontrivial to make good decisions, as any decision for the current time slot will affect the switching cost between the current time slot and the next one

**Switching cost** of turning on/off servers in cloudlets

- Server initialization, hardware wear and tear, etc. incurred between two sequential time slots\(^1\)

---

\(^1\)Dynamic right-sizing for power-proportional data centers. INFOCOM 2011 (best paper award).
**Operating cost** of activating the servers in cloudlets for time-varying inputs (e.g., user workloads)

- Inputs for the current time slot are known; future inputs all unknown
- Nontrivial to make good decisions, as any decision for the current time slot will affect the switching cost between the current time slot and the next one

**Switching cost** of turning on/off servers in cloudlets

- Server initialization, hardware wear and tear, etc. incurred between two sequential time slots\(^1\)
- Twisted with the cloudlet switching cost

**Switching cost** of turning on/off cloudlets

- System cooling, network initialization, user authentication, etc. incurred between two sequential time slots
- Small data centers (less than 500 servers) typically have Power Usage Effectiveness (PUEs) of 1.5 to 2.1, while large data centers, such as Google’s, with PUEs as low as 1.1\(^2\)

---

\(^1\)Dynamic right-sizing for power-proportional data centers. INFOCOM 2011 (best paper award).
\(^2\)Shining a Light on Small Data Centers in the US. EEDAL 2017.
Multiple granularity control decisions:

Which cloudlets should be on, how many servers should be on inside each cloudlet, and how much workloads should go to each cloudlet?
Multiple granularity control decisions:

*Which cloudlets should be on, how many servers should be on inside each cloudlet, and how much workloads should go to each cloudlet?*

Straightforward ideas (e.g., one-shot optimization) are inefficient:

**Figure:** The two (extreme) cases of allocating cloudlets and/or servers

(a) Following the workload

(b) Overprovisioning
Problem Formulation and Challenges

The multi-granularity control problem

\[
\begin{align*}
\text{min} & \quad P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& + \sum_t \sum_i c^s_i (y_{it} - y_{it-1})^+ + \sum_t \sum_i c^b_i (z_{it} - z_{it-1})^+ \\
\text{s. t.} & \quad \sum_i x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \quad (1a) \\
& \quad y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \quad (1b) \\
& \quad C_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \quad (1c) \\
& \quad x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \quad (1d) \\
& \quad z_{it} \leq 1, \quad \forall i, \forall t, \quad (1e) \\
& \quad y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f)
\end{align*}
\]
Problem Formulation and Challenges

The multi-granularity control problem

\[ \text{min } P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \]

\[ + \sum_t \sum_i c^s_i (y_{it} - y_{it-1})^+ + \sum_t \sum_i c^b_i (z_{it} - z_{it-1})^+ \]

s.t.

\[ \sum_i x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \quad (1a) \]

\[ y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \quad (1b) \]

\[ C_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \quad (1c) \]

\[ x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \quad (1d) \]

\[ z_{it} \leq 1, \quad \forall i, \forall t, \quad (1e) \]

\[ y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f) \]

- \( \mathcal{I} \): set of cloudlets; \( \mathcal{J} \): set of users
- System time-slotted \( t \in \mathcal{T} \equiv \{1, 2, \ldots, T\} \)
- \( d_{ij} \): delay between cloudlet \( i \in \mathcal{I} \) and user \( j \in \mathcal{J} \)
- \( \lambda_{jt}, j \in \mathcal{J}, t \in \mathcal{T} \): Workload originated from user \( j \) at time \( t \)
- \( \frac{1}{R_i} \): the number of requests handled by a single server of cloudlet \( i \)
- \( C_i \): the total number of servers of cloudlet \( i \)
- \( p^s_{it}, c^s_i, \forall i, \forall t \): the operating cost for operating one server at cloudlet \( i \) at time \( t \), and the switching cost for turning on one sever at cloudlet \( i \)
- \( p^b_{it}, c^b_i, \forall i \): the operating cost for operating cloudlet \( i \) at time \( t \), and the switching cost for turning on cloudlet \( i \)
Problem Formulation and Challenges

The multi-granularity control problem

\[ \begin{align*}
\text{min} & \quad P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_i^s y_{it} + \sum_t \sum_i p_i^b z_{it} \\
& \quad + \sum_t \sum_i c_i^s (y_{it} - y_{it-1})^+ + \sum_t \sum_i c_i^b (z_{it} - z_{it-1})^+ \\
\text{s. t.} & \quad \sum_i x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \quad (1a) \\
& \quad y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \quad (1b) \\
& \quad C_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \quad (1c) \\
& \quad x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \quad (1d) \\
& \quad z_{it} \leq 1, \quad \forall i, \forall t, \quad (1e) \\
& \quad y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f)
\end{align*} \]

Control decisions:

- \( x_{ijt} \geq 0, \forall i, j, t \): the amount of the workload distributed to the cloudlet \( i \) from the user \( j \) at the time slot \( t \)
- \( y_{it} \in \{0, 1, 2, 3, \ldots\}, \forall i, t \): the number of servers activated at the cloudlet \( i \) at the time slot \( t \)
- \( z_{it} \in \{0, 1\}, \forall i, \forall t \): whether to activate cloudlet \( i \) at the time slot \( t \)
Problem Formulation and Challenges

The multi-granularity control problem

\[
\begin{align*}
\text{min} & \quad P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& \quad + \sum_t \sum_i c^s_i (y_{it} - y_{it-1})^+ + \sum_t \sum_i c^b_i (z_{it} - z_{it-1})^+ \\
\text{s.t.} & \quad \sum_j x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \quad (1a) \\
& \quad y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \quad (1b) \\
& \quad C_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \quad (1c) \\
& \quad x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \quad (1d) \\
& \quad z_{it} \leq 1, \quad \forall i, \forall t, \quad (1e) \\
& \quad y_{it} \in \{0, 1, 2, 3, \ldots\}, \quad z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f)
\end{align*}
\]

The problem \( P \) is online.

\[
\sum_t \sum_i c^s_i (y_{it} - y_{it-1})^+ + \sum_t \sum_i c^b_i (z_{it} - z_{it-1})^+, \text{ where } (\tau)^+ \overset{\text{def}}{=} \max\{\tau, 0\},
\]
couples every two sequential time slots \( t - 1 \) and \( t \). At \( t - 1 \), without any knowledge about \( t \), it is nontrivial to make good control decisions.
Problem Formulation and Challenges

The multi-granularity control problem

\[
\min P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
+ \sum_t \sum_i c_i^s (y_{it} - y_{it-1})^+ + \sum_t \sum_i c_i^b (z_{it} - z_{it-1})^+
\]

s.t. \[
\begin{align*}
\sum_i x_{ijt} &\geq \lambda_{jt}, & \forall j, \forall t, \quad (1a) \\
y_{it} &\geq R_i \sum_j x_{ijt}, & \forall i, \forall t, \quad (1b) \\
C_i z_{it} &\geq y_{it}, & \forall i, \forall t, \quad (1c) \\
x_{ijt} &\geq 0, & \forall j, \forall i, \forall t, \quad (1d) \\
z_{it} &\leq 1, & \forall i, \forall t, \quad (1e) \\
y_{it} &\in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f)
\end{align*}
\]

The problem \( P \) is **non-convex** and **intractable**.

\( y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t \) make a NP-hard problem. It is often difficult to design approximation algorithms for an “offline” NP-hard problem, not to mention we are in an “online” setting.
Problem Formulation and Challenges

The multi-granularity control problem

\[
\begin{align*}
\min \quad P &= \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
&\quad + \sum_t \sum_i c_i^s (y_{it} - y_{it-1})^+ + \sum_t \sum_i c_i^b (z_{it} - z_{it-1})^+ \\
\text{s. t.} \quad &\sum_i x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \quad (1a) \\
&y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \quad (1b) \\
&c_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \quad (1c) \\
&x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \quad (1d) \\
&z_{it} \leq 1, \quad \forall i, \forall t, \quad (1e) \\
&y_{it} \in \{0, 1, 2, 3, \ldots\}, \ z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f)
\end{align*}
\]

Main Challenges:

- **Online:** \((y_{it} - y_{it-1})^+ \) and \((z_{it} - z_{it-1})^+\)
- **Non-convex:** \((y_{it} - y_{it-1})^+ \) and \((z_{it} - z_{it-1})^+\)
- **Intractable:** \(y_{it} \in \{0, 1, 2, 3, \ldots\}\) and \(z_{it} \in \{0, 1\}\)
### Problem Formulation and Challenges

The multi-granularity control problem

\[\begin{align*}
\min \quad & P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_i^s y_{it} + \sum_t \sum_i p_i^b z_{it} \\
& + \sum_t \sum_i c_i^s (y_{it} - y_{it-1})^+ + \sum_t \sum_i c_i^b (z_{it} - z_{it-1})^+
\end{align*}\]

\[\begin{align*}
s.t. \quad & \sum_i x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \quad (1a) \\
& y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \quad (1b) \\
& C_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \quad (1c) \\
& x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \quad (1d) \\
& z_{it} \leq 1, \quad \forall i, \forall t, \quad (1e) \\
& y_{it} \in \{0, 1, 2, 3, \ldots\}, \ z_{it} \in \{0, 1\}, \forall i, \forall t. \quad (1f)
\end{align*}\]

**Main Challenges:**

- **Online:** \((y_{it} - y_{it-1})^+\) and \((z_{it} - z_{it-1})^+\)
- **Non-convex:** \((y_{it} - y_{it-1})^+\) and \((z_{it} - z_{it-1})^+\)
- **Intractable:** \(y_{it} \in \{0, 1, 2, 3, \ldots\}\) and \(z_{it} \in \{0, 1\}\)

Covering chain of control variables (i.e., \(1 \rightarrow z \rightarrow y \rightarrow x \rightarrow \lambda\))
The original problem

\[
\begin{align*}
\min & \quad P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& \quad + \sum_t \sum_i c_i^s (y_{it} - y_{it-1})^+ + \sum_t \sum_i c_i^b (z_{it} - z_{it-1})^+ \\
\text{s.t.} & \quad (1a) \sim (1e), \\
& \quad y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t.
\end{align*}
\]

**Non-convex** (taking \((y_{it} - y_{it-1})^+\) as the example):

- \((y_{it} - y_{it-1})^+\) can be approximately interpreted as the L1-distance
- **The relative entropy is an efficient alternative regularizer to the L1-distance in online learning problems**
- The relative entropy \((y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it}, \text{ which is convex,} \) is introduced to substitute \((y_{it} - y_{it-1})^+\)
  - \((\varepsilon \text{ is an arbitrary positive value to guarantee the non-zero denominator})\)
The regularized problem $\tilde{P}$

$$
\begin{align*}
\min \quad & \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& + \sum_t \sum_i \frac{c_i^s}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& + \sum_t \sum_i \frac{c_i^b}{\sigma_i} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\
\text{s.t.} \quad & (1a) \sim (1e), \\
& y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t.
\end{align*}
$$

Non-convex (taking $(y_{it} - y_{it-1})^+$ as the example):

- $(y_{it} - y_{it-1})^+$ can be approximately interpreted as the L1-distance.
- The relative entropy is an efficient alternative regularizer to the L1-distance in online learning problems.
- The relative entropy $(y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it}$, which is convex, is introduced to substitute $(y_{it} - y_{it-1})^+$.
  ($\varepsilon$ is an arbitrary positive value to guarantee the non-zero denominator.)
- $\sigma_i$ set to $\ln(1 + \frac{C_i}{\varepsilon})$ and $\sigma'$ set to $\ln(1 + \frac{1}{\varepsilon})$ which are used in the performance analysis.
Basic Ideas (Online)

The regularized problem $\tilde{P}$

\[
\begin{align*}
\min \quad & \tilde{P} = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& + \sum_t \sum_i \frac{c^s_i}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it} - 1 + \varepsilon} - y_{it} \right) \\
& + \sum_t \sum_i \frac{c^b_i}{\sigma_i} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it} - 1 + \varepsilon} - z_{it} \right) \\
\text{s.t.} \quad & (1a) \sim (1e), \\
& y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t.
\end{align*}
\]

Online:

If we can optimally solve the one-shot regularized problem at any a time slot, then we can prove that $\sum_t \tilde{P}^*_t \leq r_1 P_{OPT}$ ($r_1$ is competitive ratio)
The regularized problem $\tilde{P}_t$, $\forall t$

$$\begin{align*}
\min \quad & \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& + \sum_i \frac{c_i^s}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& + \sum_i \frac{c_i^b}{\sigma_i} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right)
\end{align*}$$

s. t. (1a) $\sim$ (1e), without "$\forall t$"

$y_{it} \in \{0, 1, 2, 3, \ldots\}$, $z_{it} \in \{0, 1\}$, $\forall i$.

Online:

- If we can **optimally** solve the **one-shot** regularized problem, then we can **prove** that $\sum_t \tilde{P}_t^* \leq r_1 P_{OPT}$ ($r_1$ is competitive ratio)

- But how to (optimally or approximately) solve that problem in polynomial time??
Basic Ideas (Intractable)

The regularized problem $\tilde{P}_t, \forall t$

$$\begin{aligned}
\min & \quad \tilde{P}_t = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& \quad + \sum_i \frac{c_i^s}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& \quad + \sum_i \frac{c_i^b}{\sigma_r} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\
\text{s. t.} & \quad (1a) \sim (1e), \text{ without } \forall t \\
& \quad y_{it} \in \{0, 1, 2, 3, \ldots\}, \ z_{it} \in \{0, 1\}, \forall i.
\end{aligned}$$

Intractable:

- Relax the integer variables $y, z$ to take real values
Basic Ideas (Intractable)

The regularized and relaxed problem $\tilde{P}_t^\prime$, $\forall t$

$$
\min \quad \tilde{P}_t^\prime = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
+ \sum_i \frac{c^s_i}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
+ \sum_i \frac{c^b_i}{\sigma_i} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right)
$$

s. t. $(1a) \sim (1e)$, without “$\forall t$”

$y_{it} \geq 0$, $z_{it} \in [0, 1]$, $\forall i$.

Intractable:

- Relax the integer variables to real ones
- Invoke interior point methods to “optimally” solve the relaxed convex problem $\{\tilde{x}_t, \tilde{y}_t, \tilde{z}_t\}$ in polynomial time
Basic Ideas (Intractable)

The regularized and relaxed problem $\tilde{P}_t$, $\forall t$

$$\min \tilde{P}_t = \sum_i \sum_j d_{ij} x_{ij} + \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it}$$
$$+ \sum_i \frac{c_i^s}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right)$$
$$+ \sum_i \frac{c_i^b}{\sigma_i} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right)$$

s. t. $(1a) \sim (1e)$, without $\forall t$
$$y_{it} \geq 0, z_{it} \in [0, 1], \forall i.$$

Intractable:

- Relax the integer variables $y, z$ to take real values
- Invoke interior point methods to "optimally" solve the relaxed convex problem $\{\tilde{x}_t, \tilde{y}_t, \tilde{z}_t\}$ in polynomial time
- Rounding the fractional $z$ and $y$ sequentially to generate the final solution $\{x^{**}_t, \bar{y}_t, \bar{z}_t\}$
Online Algorithm

The regularized and relaxed problem $\tilde{P}_t'$, $\forall t$

$$\begin{align*}
\min & \quad \tilde{P}_t' = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& + \sum_i \frac{c_i^s}{\sigma_i} \left( (y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& + \sum_i \frac{c_i^b}{\sigma_i} \left( (z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\
\text{s. t.} & \quad (1a) \sim (1e), \text{ without } \forall t \\
& \quad y_{it} \geq 0, z_{it} \in [0, 1], \forall i.
\end{align*}$$

Algorithm 1: Online algorithm, $\forall t$

1. Solve $\tilde{P}_t'$ to obtain its solution $(\tilde{x}_t, \tilde{y}_t, \tilde{z}_t)$;
2. Invoke Algorithm 2 to round $(\tilde{x}_t, \tilde{y}_t, \tilde{z}_t)$ to $(\bar{x}_t, \bar{y}_t, \bar{z}_t)$;
3. Fix $(\tilde{z}_t)$, solve $\tilde{P}_t'$ to obtain its solution $(x_t^*, y_t^*, \bar{z}_t)$;
4. Invoke Algorithm 2 to round $(x_t^*, y_t^*, \bar{z}_t)$ to $(x_t^*, \bar{y}_t, \bar{z}_t)$;
5. Fix $(\bar{y}_t, \bar{z}_t)$, solve $\tilde{P}_t'$ to obtain its solution $(x_t^{**}, \bar{y}_t, \bar{z}_t)$. 
Rounding each control variable independently is not a good choice:

- all variables are rounded up → inefficient
- all variables are rounded down → infeasible
- all variables are rounded with their fractional values → maybe infeasible
Rounding each control variable *independently* is not a good choice:
- all variables are rounded up $\rightarrow$ inefficient
- all variables are rounded down $\rightarrow$ infeasible
- all variables are rounded with their fractional values $\rightarrow$ maybe infeasible

A feasible and efficient rounding algorithm is required!
Rounding each control variable independently is not a good choice:
- all variables are rounded up $\rightarrow$ inefficient
- all variables are rounded down $\rightarrow$ infeasible
- all variables are rounded with their fractional values $\rightarrow$ maybe infeasible

We introduce a randomized dependent rounding algorithm:
- Basic idea: compensate the round-down variables with the round-up ones
- Require to round the outermost variables sequentially, due to the covering chain of control variables
We introduce a randomized dependent rounding algorithm:

- Basic idea: compensate the round-down variables with the round-up ones
- Require to round the outermost variables (i.e., $z$), due to the covering chain of control variables

Take $\theta_1 = 0.8$, $\theta_2 = 0.6$ as example:

- we want $\theta_1 = 1$, $\theta_2 = 0.4$ with a given probability $p$ or $\theta_1 = 0.4$, $\theta_2 = 1$ with the probability $1 - p$
- we do not want $\theta_1 = 1.4$, $\theta_2 = 0$ or $\theta_1 = 0$, $\theta_2 = 1.4$
We introduce a randomized **dependent** rounding algorithm:

- Compensate the round-down variables with the round-up ones
- Require to round the outermost variables (i.e., $z$), due to the covering chain of control variables

Take $\theta_1 = 0.8$, $\theta_2 = 0.6$ as example:

- we **want** $\theta_1 = 1$, $\theta_2 = 0.4$ with a given probability $p$ or $\theta_1 = 0.4$, $\theta_2 = 1$ with the probability $1 - p$
- we **do not want** $\theta_1 = 1.4$, $\theta_2 = 0$ or $\theta_1 = 0$, $\theta_2 = 1.4$

**Figure:** Illustration of Algorithm 2
Algorithm 2: Randomized dependent rounding, $\forall t$

1. To round $\tilde{z}_t$, replace $\bar{u}_{it}$ by $\tilde{z}_it$, $\hat{u}_{it}$ by $\tilde{z}_it$, and $U_i$ by $C_i$, $\forall i$;
2. To round $y^*_t$, replace $\bar{u}_{it}$ by $\tilde{y}_it$, $\hat{u}_{it}$ by $\tilde{y}_it^*$, and $U_i$ by $\frac{1}{R_i}$, $\forall i$;
3. $\theta_{it} \overset{\text{def}}{=} \hat{u}_{it} - \lfloor \hat{u}_{it} \rfloor$, $\forall i$, $\mathcal{I}'_t \overset{\text{def}}{=} \mathcal{I} \setminus \{i \mid \theta_{it} \in \{0, 1\} \}$;
4. while $|\mathcal{I}'_t| > 1$ do
   5. Select $i_1, i_2 \in \mathcal{I}'$, where $i_1 \neq i_2$;
   6. $\omega_1 \overset{\text{def}}{=} \min\{1 - \theta_{i_1t}, \frac{U_{i_2}}{U_{i_1}}\theta_{i_2t}\}$, $\omega_2 \overset{\text{def}}{=} \min\{\theta_{i_1t}, \frac{U_{i_2}}{U_{i_1}}(1 - \theta_{i_2t})\}$;
   7. With the probability $\frac{\omega_2}{\omega_1 + \omega_2}$, set $\theta'_{i_1t} = \theta_{i_1t} + \omega_1$, $\theta'_{i_2t} = \theta_{i_2t} - \frac{U_{i_1}}{U_{i_2}}\omega_1$;
   8. With the probability $\frac{\omega_1}{\omega_1 + \omega_2}$, set $\theta'_{i_1t} = \theta_{i_1t} - \omega_2$, $\theta'_{i_2t} = \theta_{i_2t} + \frac{U_{i_1}}{U_{i_2}}\omega_2$;
   9. Set $\bar{u}_{i_1t} = \lceil \hat{u}_{i_1t} \rceil + \theta'_{i_1t}$, $\mathcal{I}'_t = \mathcal{I}'_{t_1} \setminus \{i_1\}$, if $\theta'_{i_1t} \in \{0, 1\}$;
10. Set $\bar{u}_{i_2t} = \lceil \hat{u}_{i_2t} \rceil + \theta'_{i_2t}$, $\mathcal{I}'_t = \mathcal{I}'_{t_2} \setminus \{i_2\}$, if $\theta'_{i_2t} \in \{0, 1\}$;
11. end
12. if $|\mathcal{I}'_t| = 1$ then
   13. Set $\bar{u}_{it} = \lceil \hat{u}_{it} \rceil$ for the only $i \in \mathcal{I}'_t$;
14. end
We can establish the following:

\[
E(P(\{x_t^*, y_t, z_t, \forall t\})) \leq r_2 P(\{\tilde{x}_t, y_t, z_t, \forall t\}) \quad \text{← Rounding} \\
\leq r_1 r_2 D(\{\pi(\tilde{x}_t, y_t, z_t), \forall t\}) \quad \text{← Regularization} \\
\leq r_1 r_2 P(\{\tilde{x}_t, \tilde{y}_t, \tilde{z}_t, \forall t\}) \quad \text{← Weak duality} \\
\leq r_1 r_2 P_{OPT} \quad \text{← Relaxation}
\]

- “E” refers to expectation, as we use randomized rounding.
- \(r_2\) is the multiplicative approximation ratio due to dependent rounding.
- \(r_1\) is the multiplicative approximation ratio due to regularization.
- \(r_1 r_2\) is the competitive ratio.
Theorem 1: We can prove $P(\{\tilde{x}_t, \tilde{y}_t, \tilde{z}_t, \forall t\}) \leq r_1 D(\{\pi(\tilde{x}_t, \tilde{y}_t, \tilde{z}_t), \forall t\})$, 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 sketch: using $\tilde{P}_t$’s KKT conditions to bound the static (i.e., delay plus operation) cost and the dynamic (i.e., switching) cost respectively.

Theorem 2: We can prove $E(P(\{x_{t^*}, y_{t^*}, z_{t^*}, \forall t\})) \leq r_2 P(\{\tilde{x}_t, \tilde{y}_t, \tilde{z}_t, \forall t\})$, where $r_2 = \delta_x + \delta_y + \delta_z + \delta_w + \delta_v$, $\kappa = \max_t \frac{\max_i C_i}{\min_i R_i \sum_j \lambda_{jt}}$, and

$$
\delta_x = (1 + \kappa) \frac{\max_i, j d_{ij}}{\min_i R_i} \max_i, t \frac{C_i}{p_{it}}^{b},
\delta_y = (1 + \kappa) \max_i, t p_{it}^{s} \max_i, t \frac{C_i}{p_{it}}^{b},
\delta_z = (1 + \kappa) \max_i, t \frac{p_{it}^{b}}{C_i} \max_i, t \frac{C_i}{p_{it}}^{b},
\delta_w = (1 + \kappa) \max_i C_i^{s} \max_i, t \frac{C_i}{p_{it}}^{b},
\delta_v = (1 + \kappa) \max_i \frac{C_i^{b}}{C_i} \max_i, t \frac{C_i}{p_{it}}^{b}.
$$

Proof sketch: using the definition of Algorithm 2 to show $(x_{t^*}, y_{t^*})$ always exists, given $\tilde{z}_t$; $x_{t^*}$ always exists, given $(\tilde{y}_t, \tilde{z}_t)$. 
Numerical Study: Settings

**Cloudlets and Delay**
- Envisage cloudlet deployments at London underground stations
- Use 100 largest stations based on annual passenger count
- Use geographic distance to represent delay

**Workload**
- Quarterly (i.e., 15 min.) passenger numbers at each station obtained from Transport for London for Nov. 2016

![Dynamic inputs](image)

**Figure:** Dynamic inputs

**Electricity Price (Unit Operating Cost)**
- European Electricity Index (ELIX) reported by EPEX SPOT for Monday, Nov. 14 through Sunday, Nov. 20, 2016.
Cloud Capacity

- Use the workload to estimate the cloudlet capacity

Algorithms for Comparison

- \texttt{reg+r}: (our algorithm) regularization, randomized pairwise rounding;
- \texttt{lcp+r}: the existing Lazy Capacity Provisioning algorithm, randomized pairwise rounding;
- \texttt{grb}: Gurobi, the state-of-the-art mixed integer linear program solver (one-shot optimum)
- \texttt{grb(s)}: Gurobi for server control (i.e., single granularity)—an cloudlet is on if the number of servers is non-zero, and is off otherwise.
For combinations of different fractional online algorithms and rounding algorithms, we further compare our algorithm \texttt{reg+r} to

- \texttt{ipt+d}: IPOPT, deterministic rounding (rounding all variables up)
- \texttt{reg+d}: regularization, deterministic rounding
- \texttt{ipt+r}: IPOPT, randomized pairwise rounding

where IPOPT is the state-of-the-art interior point convex program solver.

**Weights and PUE**

- We vary the weight $\chi$ of the switching cost for both cloudlets and servers. Specifically, we vary $\log \chi$ as an integer in $[0, 4]$.
- We vary the PUE in $[1, 2]$ for the cloudlet operating cost; we always set 1 as the weight of the server operating cost.
**Numerical Study: Results**

**Figure:** Impact of switching cost

- *reg+r* incurs $15\% \sim 65\%$ **less cost** than *lcp+r*, *grb*, and *grb(s)*.
  - As the weight grows, the gap between *reg+r* and others expands.
  - As the PUE grows, the gap between *reg+r* and others shrinks.
  - *lcp+r* does not do well, as its Lazy Capacity Principle cannot suit well for the multi-granularity control.

**Figure:** Impact of the PUE
reg+r incurs $5\% \sim 25\%$ less cost than the next best algorithm.

- For all rounding algorithms, our regularization algorithm reg is better.
- For all fractional online algorithms, our randomized rounding r is better.

grb is rather unscalable; ipt+r and our reg+r scale much better and the execution time grows more slowly.
Different from large data centers, turning on/off cloudlets or small data centers makes sense to save their energy consumption.
Take away messages

- Different from large data centers, turning on/off cloudlets or small data centers makes sense to save their energy consumption.

- Proposing an online approximation algorithm with regularization and dependent rounding technique, which is a general tool to deal with the optimization problem including the “ramp objective” such as \([x_t - x_{t-1}]^+\)
Take away messages

- Different from large data centers, turning on/off cloudlets or small data centers makes sense to save their energy consumption.

- Proposing an online approximation algorithm with regularization and dependent rounding technique, which is a general tool to deal with the optimization problem including the “ramp objective” such as $[x_t - x_{t-1}]^+$

- Exploiting the real traces of city underground stations or Starbuck locations may be an alternative choice for cloudlets or edge clouds simulations.
Take away messages

- Different from large data centers, turning on/off cloudlets or small data centers makes sense to save their energy consumption.

- Proposing an online approximation algorithm with regularization and dependent rounding technique, which is a general tool to deal with the optimization problem including the “ramp objective” such as $[x_t - x_{t-1}]^+$

- Exploiting the real traces of city underground stations or Starbuck locations could be a choice for cloudlets or edge clouds simulations

- Tighter theoretical bound will be explored (e.g., using different regularizer and/or designing novel rounding policies)
Different from large data centers, turning on/off cloudlets or small data centers makes sense to save their energy consumption.

Proposing an online approximation algorithm with regularization and dependent rounding technique, which is a general tool to deal with the optimization problem including the “ramp objective” such as \([x_t - x_{t-1}]^+\)

Exploiting the real traces of city underground stations or Starbucks locations could be a choice for cloudlets or edge clouds simulations.

Tighter theoretical bound will be explored (e.g., using different regularizer and/or designing novel rounding policies)

Ramp objective + Ramp constraints???