Multiple Granularity Online Control of Cloudlet Networks for Edge Computing

Lei Jiao[1], **Lingjun Pu**[2], Lin Wang[3], Xiaojun Lin[4], Jun Li[1]

[1]University of Oregon, USA [2]Nankai University, China
[3]TU Darmstadt, Germany [4]Purdue University, USA

IEEE SECON 2018

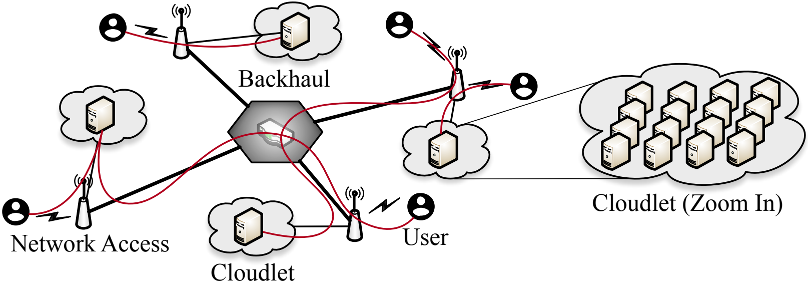


Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

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

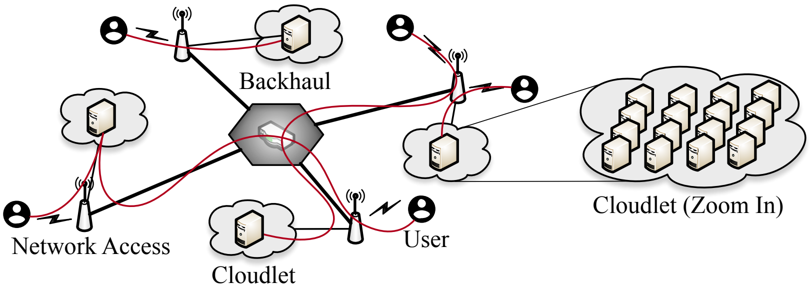


Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

Cloudlets:

- Limited processing capacity
- Fast wired connection among cloudlets
- User workloads processed at other cooperative cloudlets, not necessarily the local one

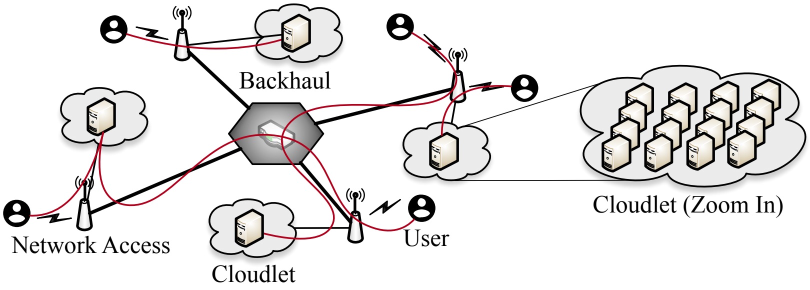


Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

Central Controller:

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

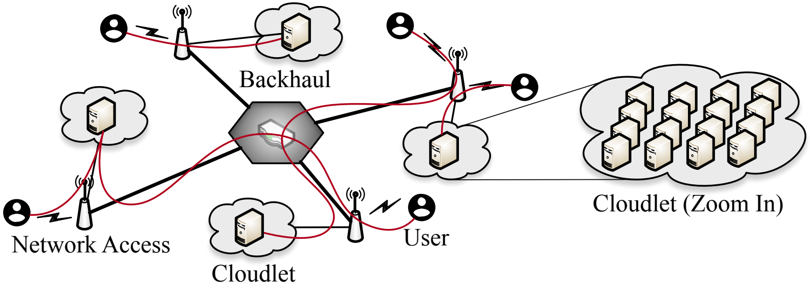


Figure: Typical cloudlet network structure (e.g., Base Stations as cloudlets in C-RAN)

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?

Is It A Novel Question?

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

Is It A Novel Question?

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).

Is It A Novel Question?

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.

It Is A Novel Question!

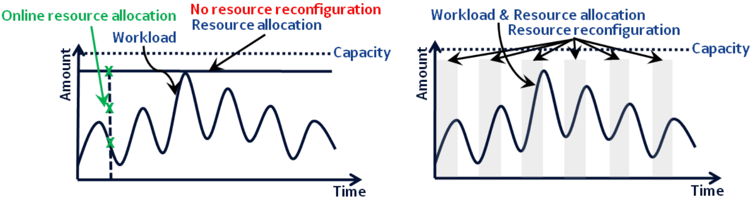
Multiple granularity control decisions:

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:



(a) Following the workload **(b)** Overprovisioning

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

The multi-granularity control problem

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

The multi-granularity control problem

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

$$\text{s. t.} \quad \sum_i x_{ijt} \geq \lambda_{jt}, \quad \forall j, \forall t, \tag{1a}$$

$$y_{it} \geq R_i \sum_j x_{ijt}, \quad \forall i, \forall t, \tag{1b}$$

$$C_i z_{it} \geq y_{it}, \quad \forall i, \forall t, \tag{1c}$$

$$x_{ijt} \geq 0, \quad \forall j, \forall i, \forall t, \tag{1d}$$

$$z_{it} \leq 1, \quad \forall i, \forall t, \tag{1e}$$

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

- $\mathcal{I}$: set of cloudlets; $\mathcal{J}$: set of users
- System time-slotted $t \in \mathcal{T} \stackrel{\text{def}}{=} \{1, 2, ..., T\}$
- d_{ij}: delay between cloudlet $i \in \mathcal{I}$ and user $j \in \mathcal{J}$
- λ_{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

The multi-granularity control problem

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

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
- $y_{it} \in \{0, 1, 2, 3, ...\}, \forall i, t$: 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

The multi-granularity control problem

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

The problem P is online.

$\sum_t \sum_i c^s_i \left(y_{it} - y_{it-1}\right)^+ + \sum_t \sum_i c^b_i \left(z_{it} - z_{it-1}\right)^+$, where $(\tau)^+ \stackrel{\text{def}}{=} \max\{\tau, 0\}$, 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

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

The problem **P** is non-convex and intractable.

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

The multi-granularity control problem

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

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**: $y_{it} \in \{0, 1, 2, 3, ...\}$ and $z_{it} \in \{0, 1\}$

The multi-granularity control problem

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

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., $1 \to z \to y \to x \to \lambda$)

The original problem

$$\begin{aligned} \min \quad & P = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\ & + \sum_t \sum_i c^s_i (y_{it} - y_{it-1})^+ + \sum_t \sum_i c^b_i (z_{it} - z_{it-1})^+ \\ \text{s. t.} \quad & (1a) \sim (1e), \\ & y_{it} \in \{0, 1, 2, 3, ...\}, z_{it} \in \{0, 1\}, \forall i, \forall t. \end{aligned}$$

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 $(y_{it} - y_{it-1})^+$
  (ε is an arbitrary positive value to guarantee the non-zero denominator)

The regularized problem $\widetilde{\mathbf{P}}$

$$\begin{aligned}
\min \quad & \widetilde{\mathrm{P}} = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p_{it}^s y_{it} + \sum_t \sum_i p_{it}^b z_{it} \\
& + \sum_t \sum_i \frac{c_i^s}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it}+\varepsilon}{y_{it-1}+\varepsilon} - y_{it} \right) \\
& + \sum_t \sum_i \frac{c_i^b}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it}+\varepsilon}{z_{it-1}+\varepsilon} - z_{it} \right) \\
\text{s. t.} \quad & (1a) \sim (1e), \\
& y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t.
\end{aligned}$$

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 $(y_{it} - y_{it-1})^+$
  (ε is an arbitrary positive value to guarantee the non-zero denominator)
- σ_i set to $\ln(1 + \frac{C_i}{\varepsilon})$ and σ' set to $\ln(1 + \frac{1}{\varepsilon})$ which are used in the performance analysis

Basic Ideas (Online)

The regularized problem $\widetilde{\mathrm{P}}$

$$
\begin{aligned}
\min \quad & \widetilde{\mathrm{P}} = \sum_t \sum_i \sum_j d_{ij} x_{ijt} + \sum_t \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& + \sum_t \sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& + \sum_t \sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\
\text{s. t.} \quad & (1a) \sim (1e), \\
& y_{it} \in \{0, 1, 2, 3, \ldots\}, z_{it} \in \{0, 1\}, \forall i, \forall t.
\end{aligned}
$$

Online:

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

Basic Ideas (Online)

The regularized problem $\widetilde{\mathbf{P}}_\mathrm{t}$, $\forall t$

$$\begin{aligned} \min \quad & \widetilde{\mathrm{P}}_\mathrm{t} = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\ & + \sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\ & + \sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\ \text{s. t.} \quad & (1a) \sim (1e), \text{without } \text{“}\forall t\text{”} \\ & y_{it} \in \{0, 1, 2, 3, ...\}, z_{it} \in \{0, 1\}, \forall i. \end{aligned}$$

Online:

- If we can **optimally** solve the **one-shot** regularized problem, then we can **prove** that $\sum_t \widetilde{\mathrm{P}}^*_\mathrm{t} \leq r_1 \mathrm{P}_{OPT}$ (r_1 is competitive ratio)
- **But how to (optimally or approximately) solve that problem in polynomial time**??

The regularized problem $\widetilde{\mathbf{P}}_{\mathbf{t}}$, $\forall t$

$$\min \quad \widetilde{\mathrm{P}}_{\mathrm{t}} = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it}$$
$$+\sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it}+\varepsilon}{y_{it-1}+\varepsilon} - y_{it} \right)$$
$$+\sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it}+\varepsilon}{z_{it-1}+\varepsilon} - z_{it} \right)$$
$$\text{s. t.} \quad (1a) \sim (1e), \text{without "}\forall t\text{"}$$
$$y_{it} \in \{0, 1, 2, 3, ...\}, z_{it} \in \{0, 1\}, \forall i.$$

Intractable:

- Relax the integer variables y, z to take real values

Basic Ideas (Intractable)

The regularized and relaxed problem $\widetilde{\mathbf{P}}'_{\mathbf{t}}$, $\forall t$

$$\begin{aligned}
\min \quad & \widetilde{\mathrm{P}}'_{\mathrm{t}} = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& + \sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it}+\varepsilon}{y_{it-1}+\varepsilon} - y_{it} \right) \\
& + \sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it}+\varepsilon}{z_{it-1}+\varepsilon} - z_{it} \right) \\
\text{s. t.} \quad & (1a) \sim (1e), \text{without ``}\forall t\text{''} \\
& y_{it} \geq 0, z_{it} \in [0,1], \forall i.
\end{aligned}$$

Intractable:

- Relax the integer variables to real ones
- Invoke interior point methods to "optimally" solve the relaxed convex problem $\{\widetilde{\mathbf{x}}_{\mathbf{t}}, \widetilde{\mathbf{y}}_{\mathbf{t}}, \widetilde{\mathbf{z}}_{\mathbf{t}}\}$ in polynomial time

The regularized and relaxed problem $\widetilde{\mathbf{P}}'_{\mathbf{t}}$, $\forall t$

$$
\begin{aligned}
\min \quad & \widetilde{\mathrm{P}}'_{\mathrm{t}} = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& + \sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it}+\varepsilon}{y_{it-1}+\varepsilon} - y_{it} \right) \\
& + \sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it}+\varepsilon}{z_{it-1}+\varepsilon} - z_{it} \right) \\
\text{s. t.} \quad & (1a) \sim (1e), \text{without "}\forall t\text{"} \\
& y_{it} \geq 0, z_{it} \in [0, 1], \forall i.
\end{aligned}
$$

Intractable:

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

The regularized and relaxed problem $\widetilde{\mathbf{P}}'_{\mathbf{t}}$, $\forall t$

$$\begin{aligned}
\min \quad & \widetilde{\mathrm{P}}'_{\mathrm{t}} = \sum_i \sum_j d_{ij} x_{ijt} + \sum_i p^s_{it} y_{it} + \sum_t \sum_i p^b_{it} z_{it} \\
& + \sum_i \frac{c^s_i}{\sigma_i} \left((y_{it} + \varepsilon) \ln \frac{y_{it} + \varepsilon}{y_{it-1} + \varepsilon} - y_{it} \right) \\
& + \sum_i \frac{c^b_i}{\sigma'} \left((z_{it} + \varepsilon) \ln \frac{z_{it} + \varepsilon}{z_{it-1} + \varepsilon} - z_{it} \right) \\
\text{s. t.} \quad & (1a) \sim (1e), \text{without } \text{“}\forall t\text{”} \\
& y_{it} \geq 0, z_{it} \in [0, 1], \forall i.
\end{aligned}$$

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 $\widetilde{\mathbf{P}}'_{\mathbf{t}}$ to obtain its solution $(\mathbf{x}^*_{\mathbf{t}}, \mathbf{y}^*_{\mathbf{t}}, \bar{\mathbf{z}}_{\mathbf{t}})$;
4 Invoke **Algorithm 2** to round $(\mathbf{x}^*_{\mathbf{t}}, \mathbf{y}^*_{\mathbf{t}}, \bar{\mathbf{z}}_{\mathbf{t}})$ to $(\mathbf{x}^*_{\mathbf{t}}, \bar{\mathbf{y}}_{\mathbf{t}}, \bar{\mathbf{z}}_{\mathbf{t}})$;
5 Fix $(\bar{\mathbf{y}}_{\mathbf{t}}, \bar{\mathbf{z}}_{\mathbf{t}})$, solve $\widetilde{\mathbf{P}}'_{\mathbf{t}}$ to obtain its solution $(\mathbf{x}^{**}_{\mathbf{t}}, \bar{\mathbf{y}}_{\mathbf{t}}, \bar{\mathbf{z}}_{\mathbf{t}})$.

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

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

A feasible and efficient rounding algorithm is required!

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

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 $\theta_1 = 0$, $\theta_2 = 1.4$

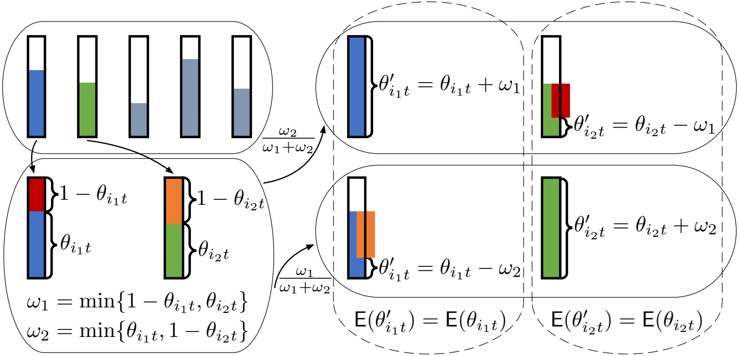


Figure: Illustration of Algorithm 2

Nontrivial Rounding Algorithm (i.e., Algorithm 2)

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

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, $\forall i$;
To round $\mathbf{y}^*_{\mathbf{t}}$, replace $\bar{u}_{it}$ by $\bar{y}_{it}$, $\widehat{u}_{it}$ by y^*_{it}, and U_i by $\frac{1}{R_i}$, $\forall i$;
$\theta_{it} \overset{\text{def}}{=} \widehat{u}_{it} - \lfloor \widehat{u}_{it} \rfloor$, $\forall i$, $\mathcal{I}'_t \overset{\text{def}}{=} \mathcal{I} \setminus \{i \mid \theta_{it} \in \{0,1\}\}$;
while $|\mathcal{I}'_t| > 1$ **do**
    Select i_1, $i_2 \in \mathcal{I}'$, where $i_1 \neq i_2$;
    $\omega_1 \overset{\text{def}}{=} \min\{1 - \theta_{i_1 t}, \frac{U_{i_2}}{U_{i_1}} \theta_{i_2 t}\}$, $\omega_2 \overset{\text{def}}{=} \min\{\theta_{i_1 t}, \frac{U_{i_2}}{U_{i_1}}(1 - \theta_{i_2 t})\}$;
    With the probability $\frac{\omega_2}{\omega_1 + \omega_2}$, set $\theta'_{i_1 t} = \theta_{i_1 t} + \omega_1$, $\theta'_{i_2 t} = \theta_{i_2 t} - \frac{U_{i_1}}{U_{i_2}} \omega_1$;
    With the probability $\frac{\omega_1}{\omega_1 + \omega_2}$, set $\theta'_{i_1 t} = \theta_{i_1 t} - \omega_2$, $\theta'_{i_2 t} = \theta_{i_2 t} + \frac{U_{i_1}}{U_{i_2}} \omega_2$;
    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 $\theta'_{i_1 t} \in \{0,1\}$;
    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 $\theta'_{i_2 t} \in \{0,1\}$;
end
if $|\mathcal{I}'_t| = 1$ **then**
    Set $\bar{u}_{it} = \lceil \widehat{u}_{it} \rceil$ for the only $i \in \mathcal{I}'_t$;
end

We can establish the following:

$$
\begin{aligned}
& E(P(\{\mathbf{x}_t^{**}, \bar{\mathbf{y}}_t, \bar{\mathbf{z}}_t, \forall t\})) && && \text{(6a)} \\
\le\ & r_2 P(\{\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{y}}_t, \widetilde{\mathbf{z}}_t, \forall t\}) && \leftarrow \textbf{Rounding} && \text{(6b)} \\
\le\ & r_1 r_2 D(\{\pi(\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{y}}_t, \widetilde{\mathbf{z}}_t), \forall t\}) && \leftarrow \textbf{Regularization} && \text{(6c)} \\
\le\ & r_1 r_2 P(\{\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{y}}_t, \widetilde{\mathbf{z}}_t, \forall t\}) && \leftarrow \textbf{Weak duality} && \text{(6d)} \\
\le\ & r_1 r_2 P_{OPT} && \leftarrow \textbf{Relaxation} && \text{(6e)}
\end{aligned}
$$

- "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 $P(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\}) \le r_1 D(\{\pi(\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}), \forall t\})$, 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 sketch: using $\widetilde{\mathbf{P}}_\mathbf{t}$'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 $E(P(\{\mathbf{x}_\mathbf{t}^{**}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\})) \le r_2 P(\{\widetilde{\mathbf{x}}_\mathbf{t}, \widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}, \forall t\})$, where $r_2 = \delta_x + \delta_y + \delta_z + \delta_w + \delta_v$, $\kappa = \max_t \frac{\max_i C_i}{\min_i R_i \sum_j \lambda_{jt}}$, 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 $(\mathbf{x}_\mathbf{t}^*, \mathbf{y}_\mathbf{t}^*)$ always exists, given $\bar{\mathbf{z}}_\mathbf{t}$; $\mathbf{x}_\mathbf{t}^{**}$ always exists, given $(\bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t})$.

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

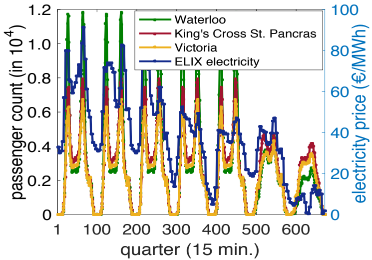


Figure: Dynamic inputs

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

- `reg+r`: **(our algorithm) regularization, randomized pairwise rounding;**
- `lcp+r`: the existing Lazy Capacity Provisioning algorithm, randomized pairwise rounding;
- `grb`: Gurobi, the state-of-the-art mixed integer linear program solver (one-shot optimum)
- `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.

Numerical Study: Settings (Cont.)

For combinations of different fractional online algorithms and rounding algorithms, we further compare our algorithm `reg+r` to

- `ipt+d`: IPOPT, deterministic rounding (rounding all variables up)
- `reg+d`: regularization, deterministic rounding
- `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 χ 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

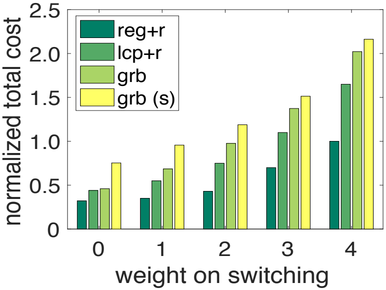


Figure: Impact of switching cost

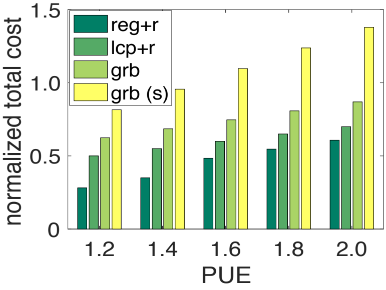


Figure: Impact of the PUE

- `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.

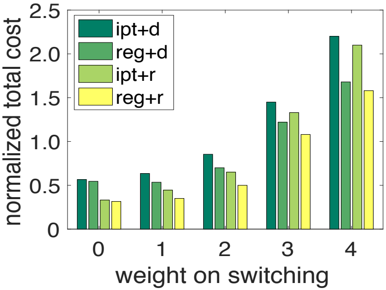


Figure: Algorithm combinations

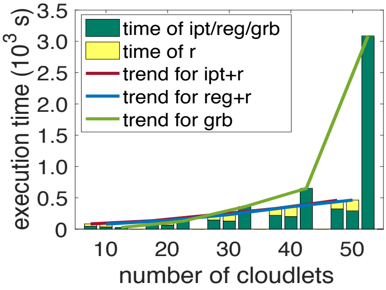


Figure: Execution time

- `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**.

Take away messages

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

- 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 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)
- Ramp objective + Ramp constraints???