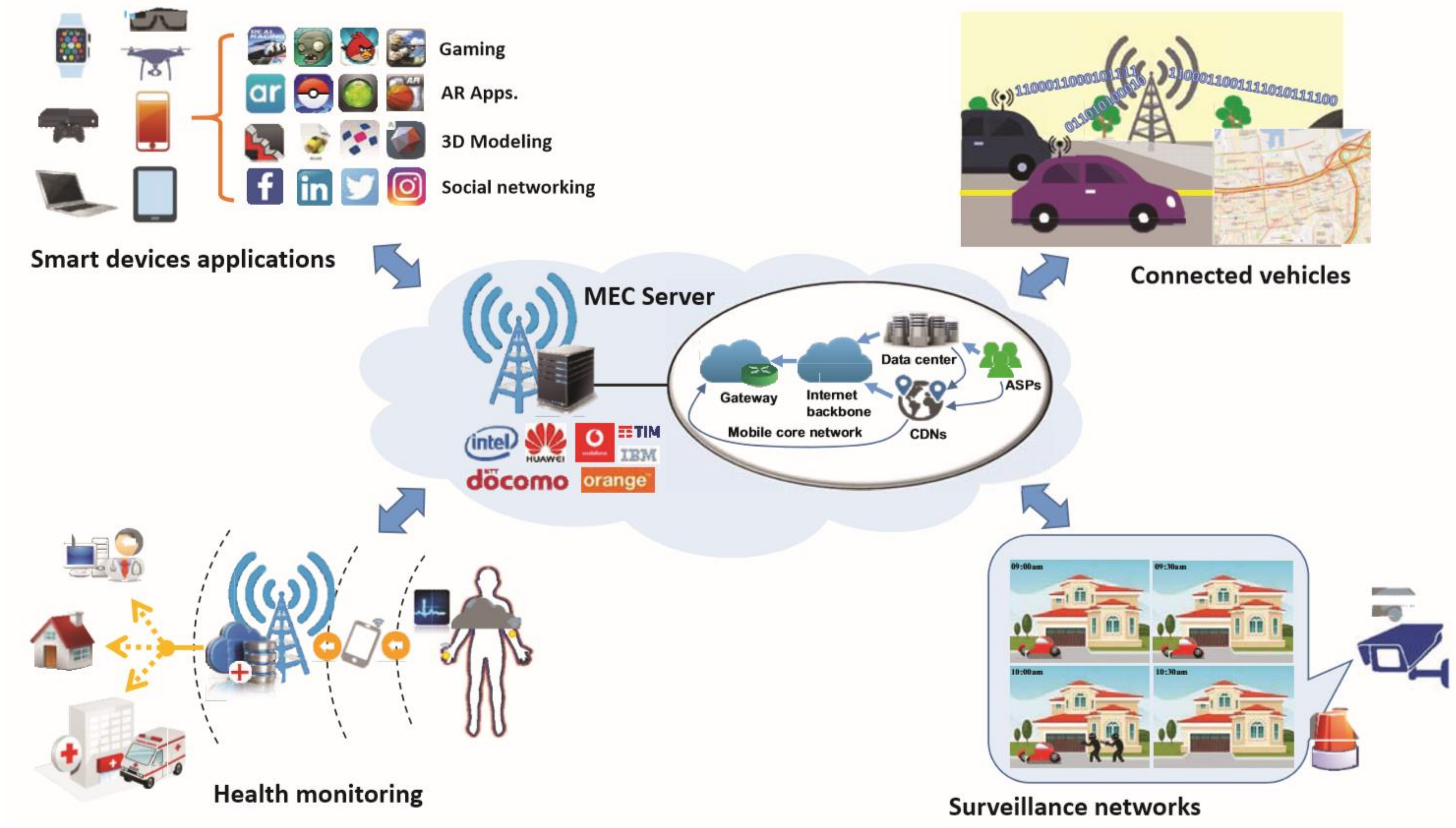
QoE Aware and Cell Capacity Enhanced Computation Offloading for Multi-Server Mobile Edge Computing Systems with Energy Harvesting Devices

Hai-Liang Zhao
`hliangzhao97@gmail.com`

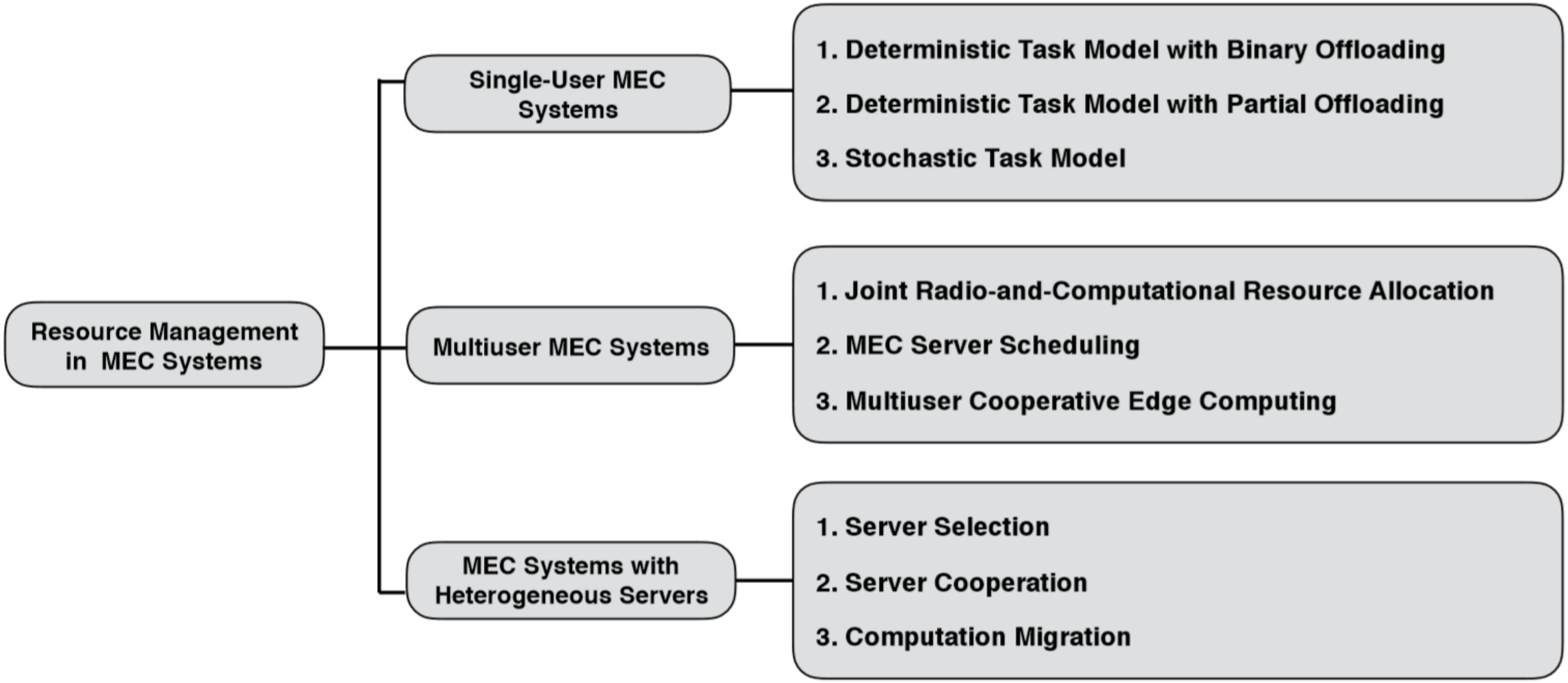
Wuhan University of Technology
2018.10.05

Mobile Cloud Computing (MCC) → Mobile Edge Computing (MEC)



Resource Management in MEC Systems

- My paper focus on the hardest one: **Multi-server Systems with multiple energy harvesting devices**

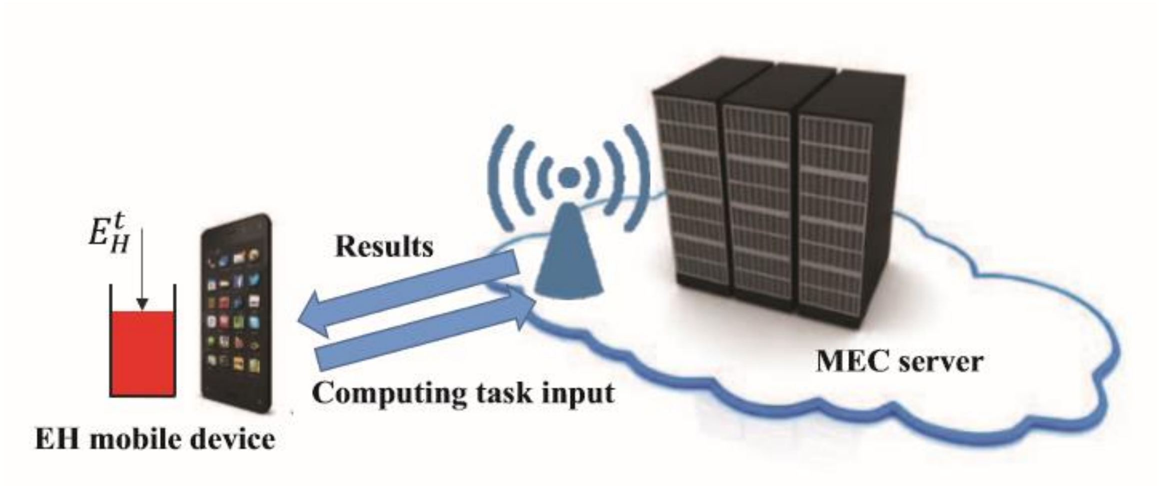


- **Resource Management in MEC Systems**
  - **Single-User MEC Systems**
    - **1. Deterministic Task Model with Binary Offloading**
    - **2. Deterministic Task Model with Partial Offloading**
    - **3. Stochastic Task Model**
  - **Multiuser MEC Systems**
    - **1. Joint Radio-and-Computational Resource Allocation**
    - **2. MEC Server Scheduling**
    - **3. Multiuser Cooperative Edge Computing**
  - **MEC Systems with Heterogeneous Servers**
    - **1. Server Selection**
    - **2. Server Cooperation**
    - **3. Computation Migration**

Obstacles we have encountered

- How to allocate limited resources **among** mobile devices?
- How to **choose a proper server** according to the system optimization metrics or the user's preference?
- How to achieve the tradeoff between cell capacity enhancement and QoE guarantee?
- In the complicated systems, for each mobile device, **offloading or not**, that's the question.
- … …

Most Similar Work to ours (the LODCO algorithm*)

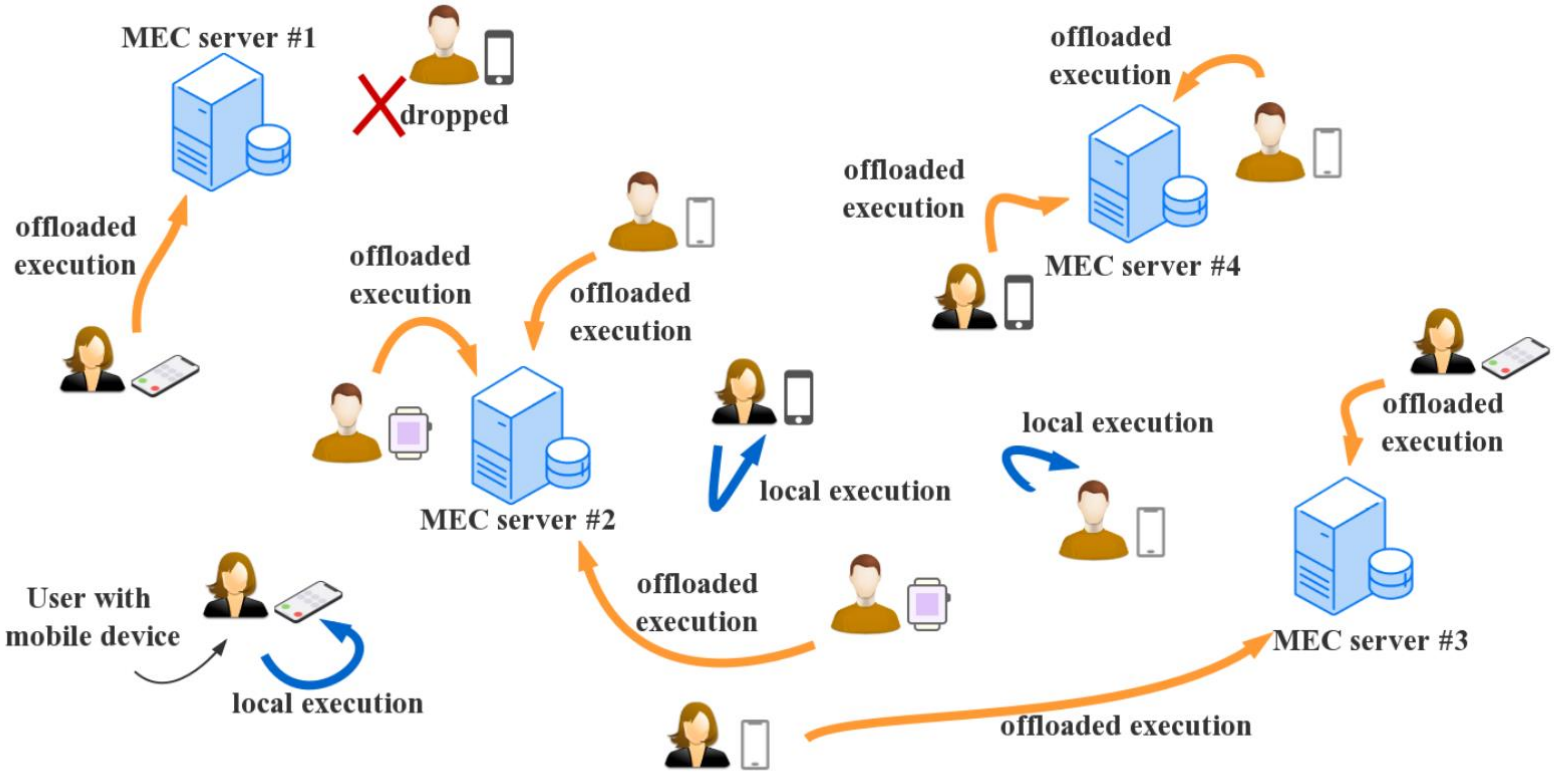


In this paper, the Lyapunov optimization-based dynamic computation offloading (LODCO) algorithm is proposed, which jointly decides the offloading decision, the CPU-cycle frequencies for mobile execution, and the transmit power for computation offloading.

* Yuyi Mao, et al. Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices.

System description

We consider a MEC system consisting of N mobile devices equipped with EH components and M MEC servers.



Model Overview

☐ Computation Task Model

Computation tasks with fixed size modeled as an i.i.d. Bernoulli process at each time slot for each mobile device;

The task can be executed locally at mobile device, or offloaded to MEC server, or dropped.

☐ Offloading Computation Model

Demonstrate the communication details by *Shannon Theorem;*

Computational abilities of MEC servers are constrained.

☐ Local Computation Model

Obtain the execution latency and energy consumption.

☐ Energy Harvesting Model

☐ QoE-Cost Function

User's QoE consists of execution delay and the penalty for dropping the task.

☐ The Optimization Issue (with two goals)

Computation task of the ith mobile device: offloading or not? If yes, to which server? Maximum offloading number?

Problem Formulation

$$\mathcal{P}: \min_{\mathbf{SO}^t} \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\left(\mathcal{D}(\mathbf{I}_i^t, \mathbf{f}_i^t, p_i^t) + \phi\cdot \mathbf{I}(\zeta_i^t \cap I_{i,d}^t)\right) - \psi\cdot\sum_{i=1}^{N}\mathbf{I}(\zeta_i^t \cap I_{i,d}^t)$$

$$
\begin{aligned}
s.t. \quad & I_{i,l}^t + I_{i,r}^t + I_{i,d}^t = 1, t \in \mathcal{T}, i \in \mathcal{N}, \\
& \sum_{j=1}^{M} c_{i,j}^t = 1, t \in \mathcal{T}, i \in \mathcal{N}, j \in \mathcal{M}, \\
& \sum_{i=1}^{N} I_{i,r}^t \cdot c_{ij}^t L X_s \le f_s^{\max}\tau, t \in \mathcal{T}, j \in \mathcal{M}, \\
& \mathcal{E}(\mathbf{I}_i^t, \mathbf{f}_i^t, p_i^t) \le b_i^t, t \in \mathcal{T}, i \in \mathcal{N},
\end{aligned}
$$

... ... a dozen constraints

Problem Analysis

☐ The original problem: Time average expectation form

↓ *Lyapunov Optimization*

☐ The new one: **per-time slot deterministic problem**

$$\mathcal{P}_2 : \min_{\mathbf{SO}^t} \sum_{i=1}^{N} \tilde{b}_i^t \left(e_i^t - \mathcal{E}(\mathbf{I}_i^t, \mathbf{f}_i^t, p_i^t)\right) + V \cdot \mathbf{E}\left[cost_{sum}^t | \tilde{\mathbf{b}}^t\right]$$

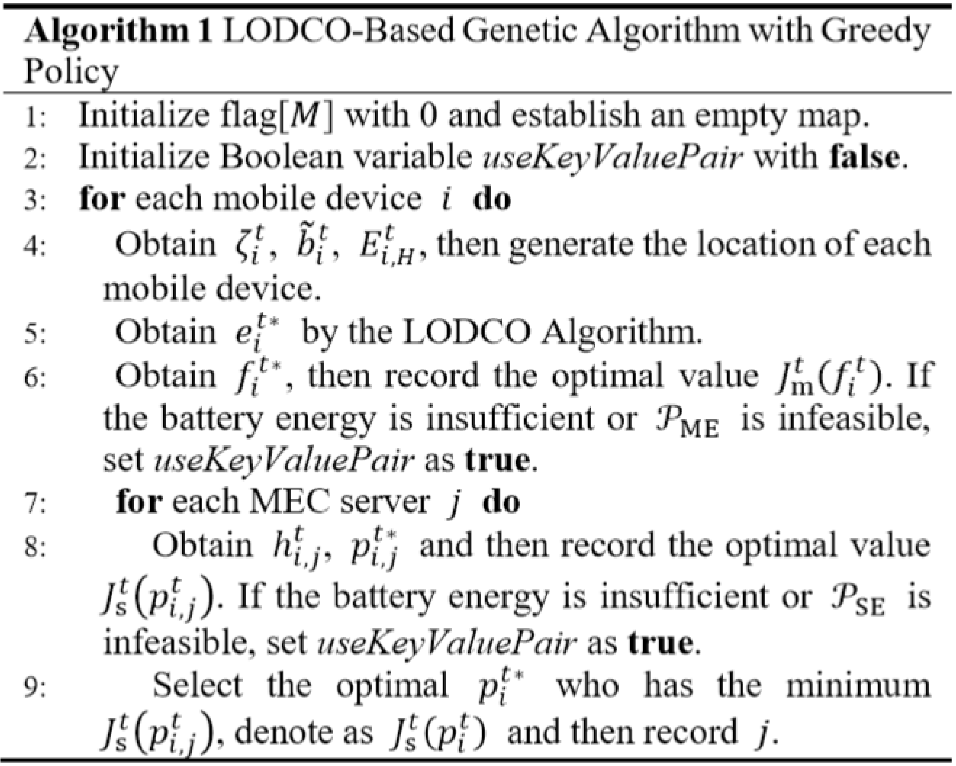
which subjects to every constraints of the original one.

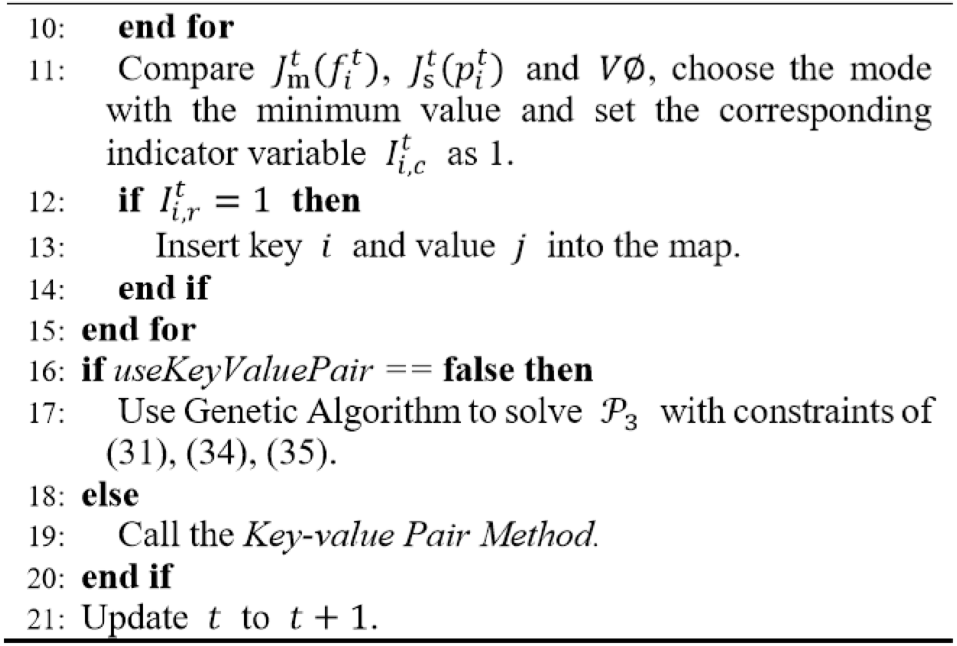
In each time slot, solve the problem above.

☐ Upgrade and reconstructed the LODCO algorithm

☐ Genetic algorithm & Greedy policy for sub problem of `$\mathcal{P}_2$`

LODCO-Based Genetic Algorithm with Greedy Policy

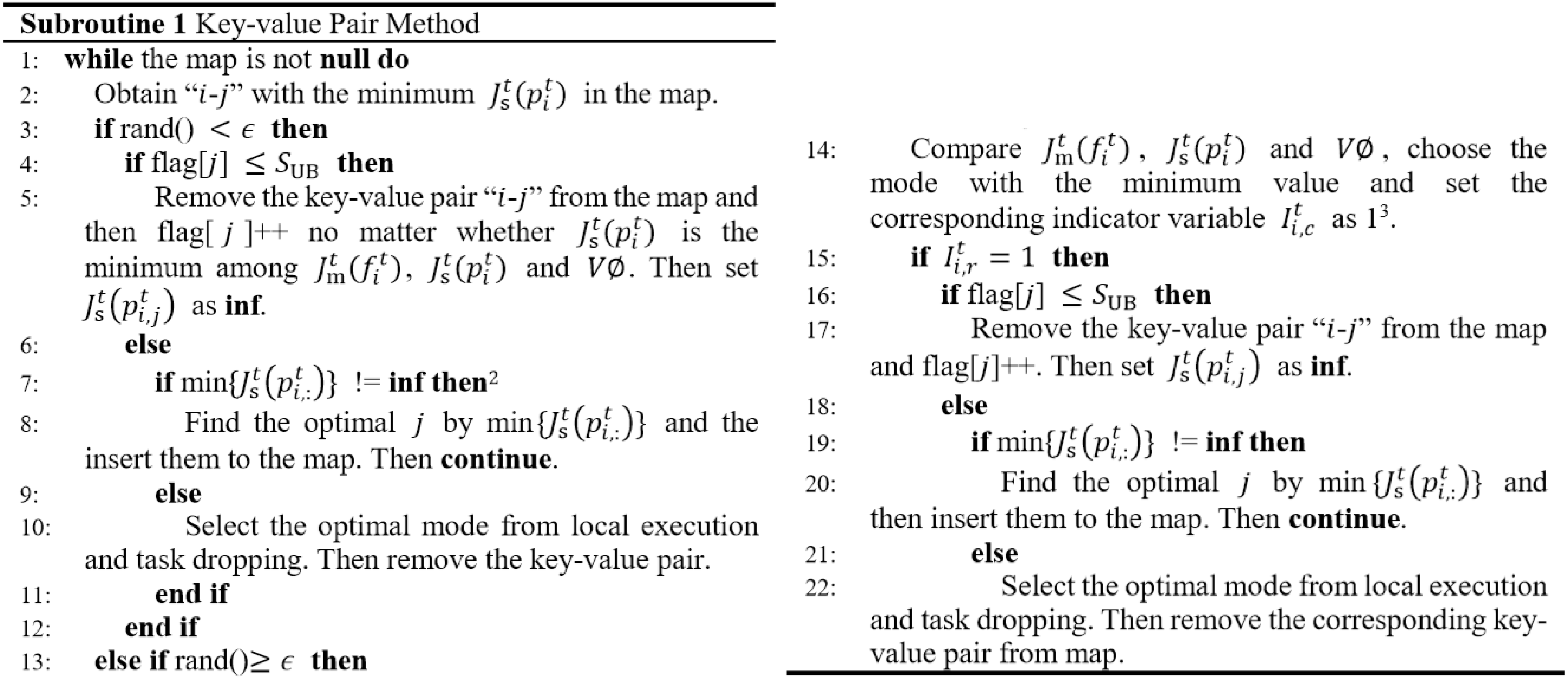




Algorithm 1 LODCO-Based Genetic Algorithm with Greedy Policy

```
Initialize flag[$M$] with 0 and establish an empty map.
Initialize Boolean variable $useKeyValuePair$ with false.
for each mobile device $i$ do
    Obtain $\zeta_i^t$, $\tilde{b}_i^t$, $E_{i,H}^t$, then generate the location of each mobile device.
    Obtain $e_i^{t*}$ by the LODCO Algorithm.
    Obtain $f_i^{t*}$, then record the optimal value $J_{\mathrm{m}}^t(f_i^t)$. If the battery energy is insufficient or $\mathcal{P}_{\mathrm{ME}}$ is infeasible, set $useKeyValuePair$ as true.
    for each MEC server $j$ do
        Obtain $h_{i,j}^t$, $p_{i,j}^{t*}$ and then record the optimal value $J_{\mathrm{s}}^t(p_{i,j}^t)$. If the battery energy is insufficient or $\mathcal{P}_{\mathrm{SE}}$ is infeasible, set $useKeyValuePair$ as true.
        Select the optimal $p_i^{t*}$ who has the minimum $J_{\mathrm{s}}^t(p_{i,j}^t)$, denote as $J_{\mathrm{s}}^t(p_i^t)$ and then record $j$.
    end for
    Compare $J_{\mathrm{m}}^t(f_i^t)$, $J_{\mathrm{s}}^t(p_i^t)$ and $V\emptyset$, choose the mode with the minimum value and set the corresponding indicator variable $I_{i,c}^t$ as 1.
    if $I_{i,r}^t = 1$ then
        Insert key $i$ and value $j$ into the map.
    end if
end for
if $useKeyValuePair$ == false then
    Use Genetic Algorithm to solve $\mathcal{P}_3$ with constraints of (31), (34), (35).
else
    Call the Key-value Pair Method.
end if
Update $t$ to $t+1$.
```

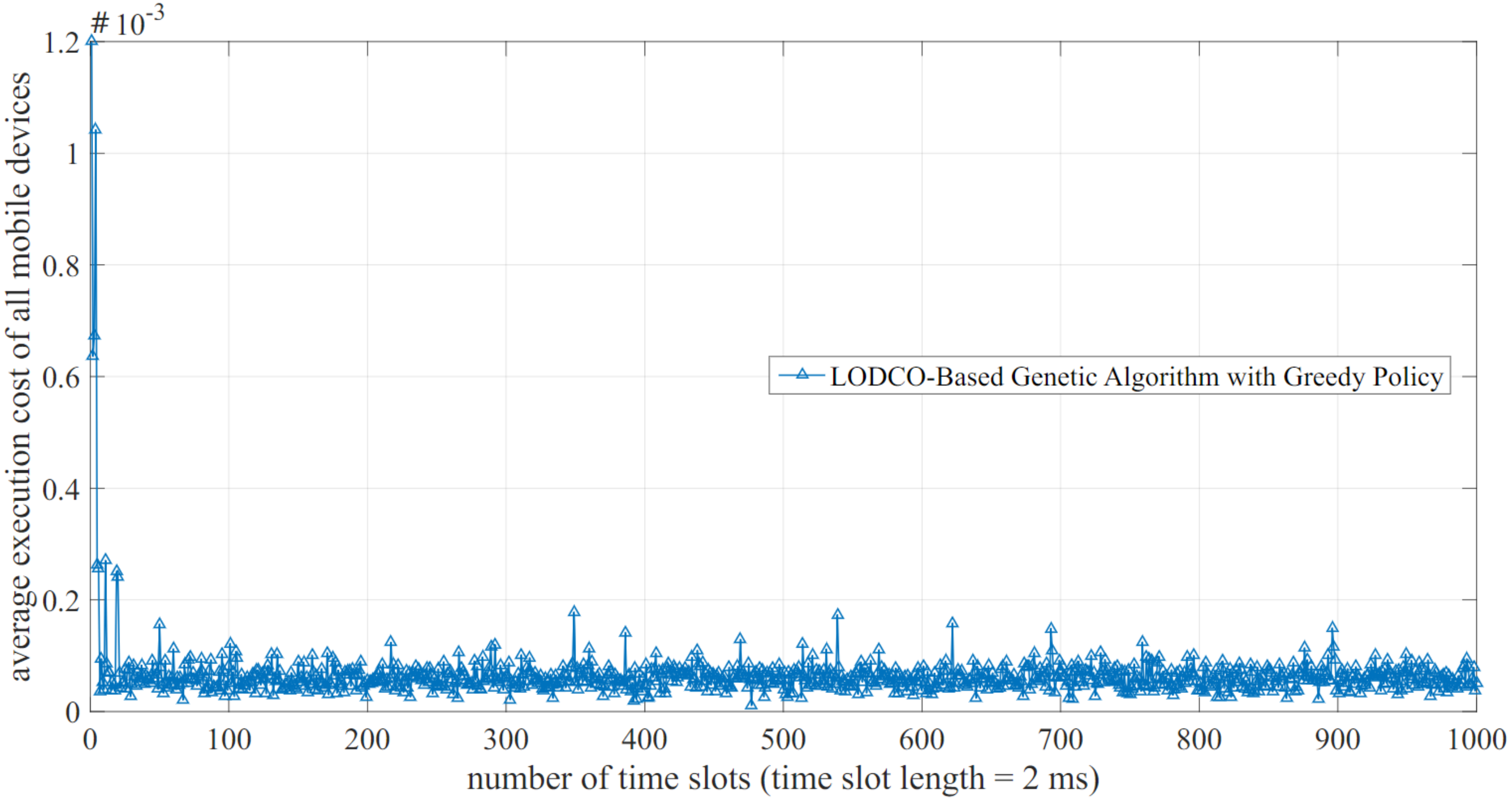
Key-Value Pair Method (subroutine)



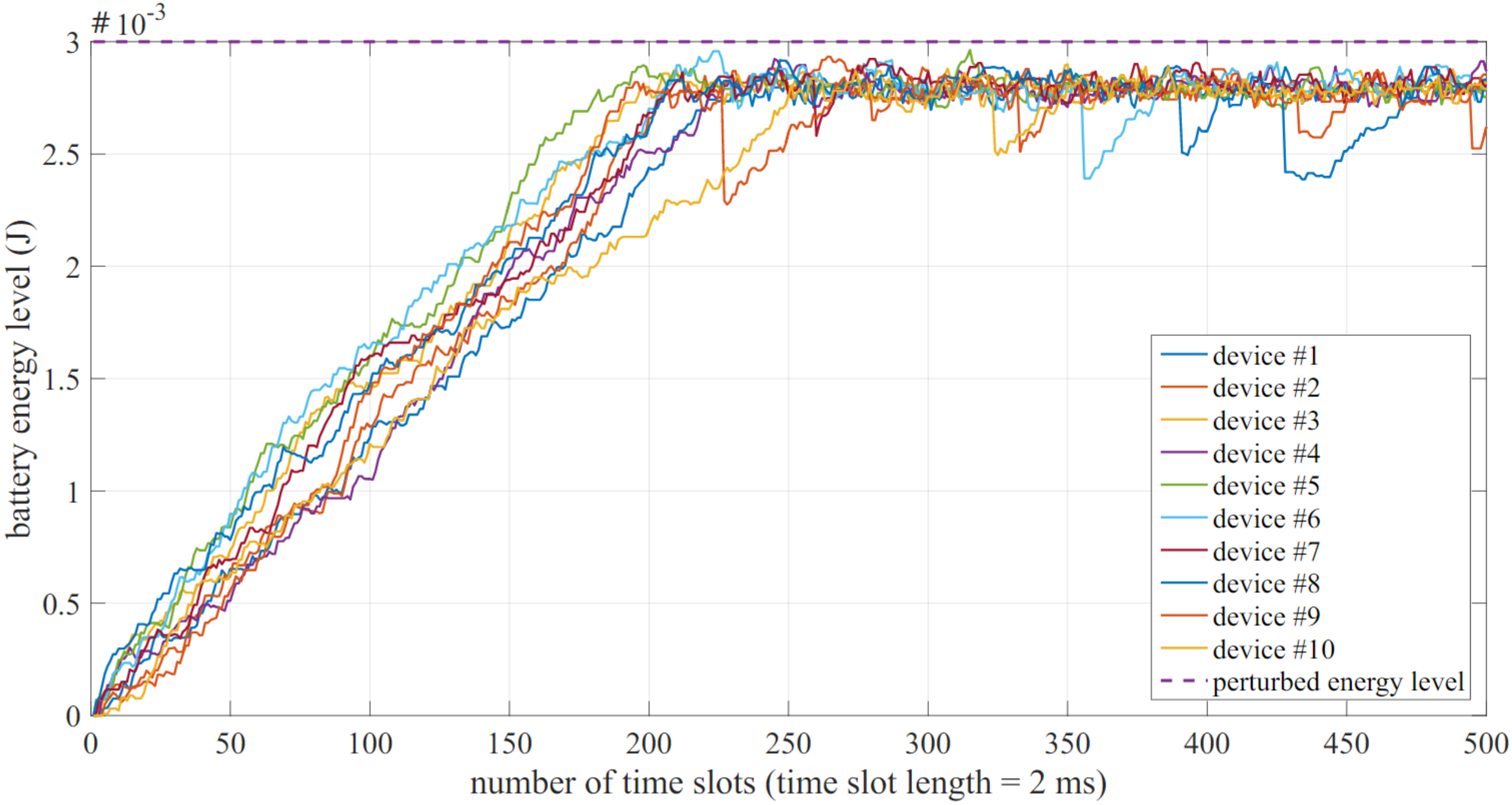
Subroutine 1 Key-value Pair Method

```
while the map is not null do
    Obtain "i-j" with the minimum $J_s^t(p_i^t)$ in the map.
    if rand() $< \epsilon$ then
        if flag[j] $\le S_{UB}$ then
            Remove the key-value pair "i-j" from the map and then flag[ j ]++ no matter whether $J_s^t(p_i^t)$ is the minimum among $J_m^t(f_i^t)$, $J_s^t(p_i^t)$ and $V\emptyset$. Then set $J_s^t(p_{i,j}^t)$ as inf.
        else
            if min{$J_s^t(p_{i,:}^t)$} != inf then[2]
                Find the optimal j by min{$J_s^t(p_{i,:}^t)$} and the insert them to the map. Then continue.
            else
                Select the optimal mode from local execution and task dropping. Then remove the key-value pair.
            end if
        end if
    else if rand()$\ge \epsilon$ then
        Compare $J_m^t(f_i^t)$, $J_s^t(p_i^t)$ and $V\emptyset$, choose the mode with the minimum value and set the corresponding indicator variable $I_{i,c}^t$ as 1[3].
        if $I_{i,r}^t = 1$ then
            if flag[j] $\le S_{UB}$ then
                Remove the key-value pair "i-j" from the map and flag[j]++. Then set $J_s^t(p_{i,j}^t)$ as inf.
            else
                if min{$J_s^t(p_{i,:}^t)$} != inf then
                    Find the optimal j by min {$J_s^t(p_{i,:}^t)$} and then insert them to the map. Then continue.
                else
                    Select the optimal mode from local execution and task dropping. Then remove the corresponding key-value pair from map.
```

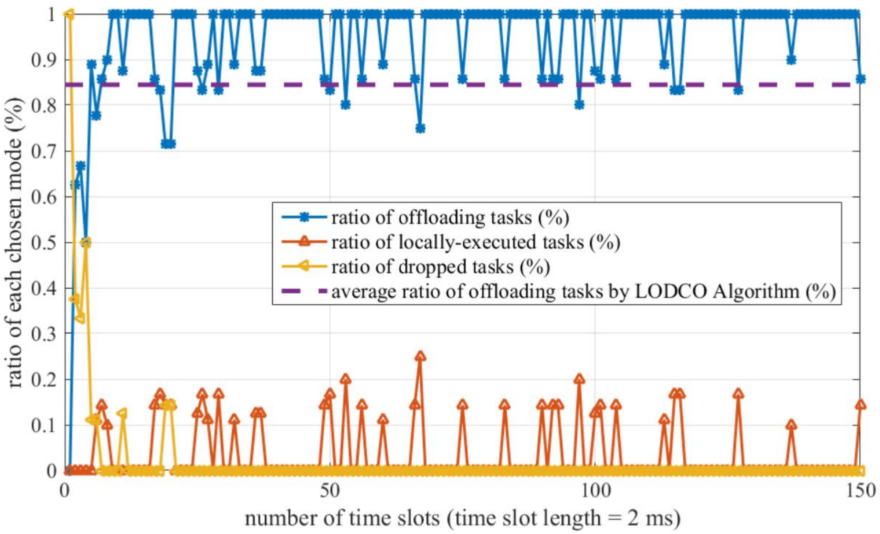
Average QoE-cost of all mobile devices vs. time



Battery energy level of each mobile device vs. time

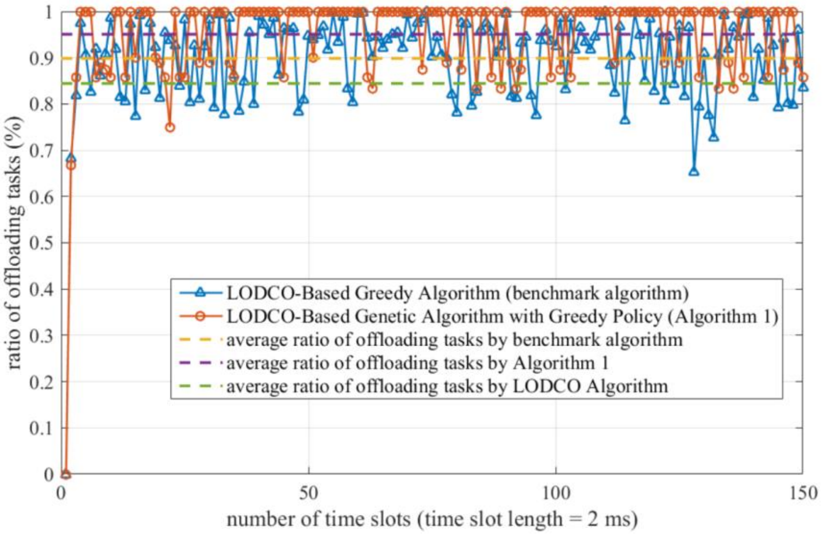


Overall Performance Evaluation



(a)

Ratio of chosen mode vs. time



(b)

Compare with benchmarks

- In our multi-user system, the static bandwidth allocation strategy is adopted (FDMI). What about dynamic allocation?
- What about *intercell interference*?
- Can we find a better algorithm with lower complexity to solve the problem?
- ... ...