# An Admission Control Algorithm in Cloud Computing Systems


Authors:

Frank Yeong-Sung Lin Department of Information Management National Taiwan University Taipei, Taiwan, R.O.C. yslin@im.ntu.edu.tw

Yingjie Lan Management Science and Information Systems Guanghua School of Management Beijing, China ylan@gsm.pku.edu.cn

Yu-Wen Chiou Department of Information Management National Taiwan University Taipei, Taiwan, R.O.C. r02029@im.ntu.edu.tw

Yu-Shun Wang Department of Information Management National Taiwan University Taipei, Taiwan, R.O.C. d98002@im.ntu.edu.tw

Presenter: Yeong Sung Lin

# Agenda

I. Introduction
II. Solution Approach
III. Experiment Results
IV. Conclusions

# I. Introduction

*A. Description and Scenario*

*B. Mathematical Formulation*

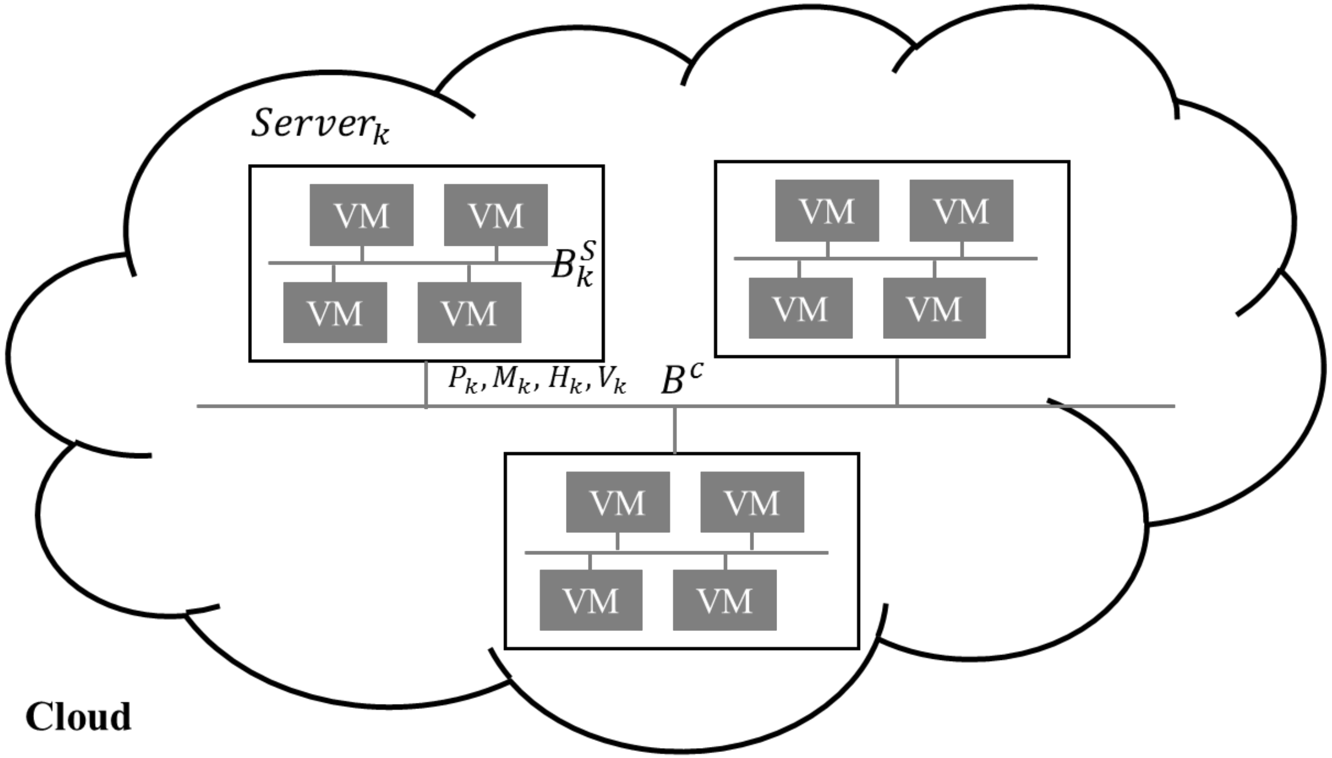
# A. Description and Scenario

- End users of cloud computing hope cloud service providers can offer high quality cloud environments that satisfy their individual requirements. Cloud computing service providers wish to *maximize their resource utilization*, while *maximizing their profits*. To make efficient use of their cloud computing system resources while ensuring those resources' availability to the end users, service providers need to adopt a suitable load balance technique. Hence, to provide an **efficient cloud service to users while maximizing profit margins at the same time** is imperative to cloud service providers.

# A. Description and Scenario (Cont'd)

- We proposed a **generic mathematical programming model** that can be used for developing an algorithm for *allocating resources* and *virtual machines*. The algorithm simulates the roles of the cloud service provider and users in a cloud computing system. The Lagrangean Relaxation Method was adopted here to obtain the optimal solution for the problem of allocation.

# A. Description and Scenario (Cont'd)




| Notation | Description |
|---|---|
| $B_k^s$ | internal communication bandwidth of server $k \in S$ |
| $B^c$ | shared communication bandwidth within the cloud computing system |
| $P_k$ | total number of CPU cores in server $k \in S$ |
| $M_k$ | total amount of RAM capacity in server $k \in S$ |
| $H_k$ | total amount of HD capacity in server $k \in S$ |
| $V_k$ | the maximum number of VMs allowable on server $k \in S$ |

# B. Mathematical Formulation

**Problem Assumptions**

- Virtual machines leased by specific users cannot be shared with others.
- Rewards gained from each user's satisfaction is not in complete accord.
- Some users' virtual machines should not be in the same physical machine due to privacy considerations.
- The CPU core capability of each physical machine varies.
- The number of virtual machines each user needs varies.

# Given Parameter

| **Given Parameter** | |
|---|---|
| Notation | Description |
| $S$ | the index set of physical servers in the cloud computing system, which is equal to $\{1,2,3,\ldots,s\}$ |
| $B_k^s$ | internal communication bandwidth of server $k \in S$ |
| $B^c$ | shared communication bandwidth within the cloud computing system |
| $P_k$ | total number of CPU cores in server $k \in S$ |
| $\pi_k$ | processing capability of each CPU core in server $k \in S$ |
| $M_k$ | total amount of RAM capacity in server $k \in S$ |
| $H_k$ | total amount of HD capacity in server $k \in S$ |
| $V_k$ | the maximum number of VMs allowable on server $k \in S$ |

| Given Parameter | |
|---|---|
| Notation | Description |
| $D$ | the index set of demands, hereafter referred to as either VPCs or users interchangeably, onto the cloud computing system, which is equal to $\{1,2,3,\ldots,d\}$ |
| $\gamma_i$ | the reward of admitting user $i \in D$ (user $i$ can be admitted only if his/her demands on all types of resources be fully satisfied) |
| $W_i$ | the index set of VMs required by user $i \in D$, which is equal to $\{1,2,3,\ldots,w_i\}$ ($w_i$ is also referred to as the total number of VMs required by user $i \in D$) |
| $p_{ij}$ | total amount of CPU processing capability required by user $i \in D$ on VM $j \in W_i$ |
| $m_{ij}$ | total amount of RAM capability required by user $i \in D$ on VM $j \in W_i$ |
| $h_{ij}$ | total amount of HD capability required by user $i \in D$ on VM $j \in W_i$ |
| $c_{ijk}$ | total amount of communication channel capacity required by user $i \in D$ between VMs $j$ and $k \in W_i$ |
| $a_{ijkl}$ | the indicator function which is 1 if VM $j \in W_i$ and VM $l \in W_k$ are allowed to be allocated on the same physical server, where users $i$ and $k$ are in $D$, and 0 otherwise |

# Decision Variables

| **Decision Variables** | |
|---|---|
| Notation | Description |
| $y_i$ | 1 if user $i \in D$ is admitted to the cloud computing system, and 0 otherwise |
| $x_{ijk}$ | 1 if VM $j \in W_i$ of user $i \in D$ is allocated to server $k \in S$, and 0 otherwise |
| $t_{ijk}$ | the number of CPU cores allocated to VM $j$ of user $i$ on server $k$ |
| $f_{ijkl}$ | 1 if VM $j \in W_i$ of user $i \in D$ and VM $k \in W_i$ of user $i \in D$ are allocated to server $l \in S$, and 0 otherwise |

# Objective Function

- The problem is then formulated as the following problem to maximize the total revenue:

$$\max \sum_{i=1}^{d} \gamma_i y_i \qquad \text{(IP 1)}$$

# Constraints

$$y_i w_i \le \sum_{\forall j \in W_i} \sum_{\forall k \in S} x_{ijk}, \forall i \in D \qquad (c.1)$$

$$x_{ijk} p_{ij} \le t_{ijk} \pi_k, \forall k \in S \qquad (c.2)$$

$$x_{ijm} + x_{klm} \le a_{ijkl} + 1, \forall i \in D, \forall j \in W_i, \\ \forall k \in D, \forall l \in W_k, \forall m \in S \qquad (c.3)$$

$$\sum_{\forall i \in D} \sum_{\forall j \in W_i, k \in X_i, k \ne j} c_{ijk} f_{ijkl} \le B_l^S \qquad (c.4)$$

$$\sum_{\forall i \in D} \sum_{\forall j \in W_i} t_{ijk} \le P_k, \forall k \in S \qquad (c.5)$$

$$\sum_{\forall i \in D} \sum_{\forall j \in W_i} m_{ij} x_{ijk} \le M_k, \forall k \in S \qquad (c.6)$$

$$\sum_{\forall i \in D} \sum_{\forall j \in W_i} h_{ij} x_{ijk} \le H_k, \forall k \in S \qquad (c.7)$$

$$\sum_{\forall i \in D} \sum_{\forall j \in W_i} x_{ijk} \le V_k, \forall k \in S \qquad (c.8)$$

$$\sum_{\forall i \in D} \sum_{\forall j \in W_i, k \in W_i, k \ne j} c_{ijk} (1 - f_{ijkl}) \le B^c, \forall l \in S \qquad (c.9)$$

$$\sum_{\forall k \in S} x_{ijk} \le 1, \forall i \in D, \forall j \in W_i \qquad (c.10)$$

$$y_i = 0 \ or \ 1, \forall i \in D$$
$$x_{ijk} = 0 \ or \ 1, \forall i \in D, \forall j \in W_i, \forall k \in S$$
$$t_{ijk} = 0 \ or \ 1, \forall i \in D, \forall j \in W_i, \forall k \in S$$
$$f_{ijkl} = 0 \ or \ 1, \forall i \in D, \forall j \in W_i, \forall k \in W_i, \forall l \in S \qquad (c.11)$$

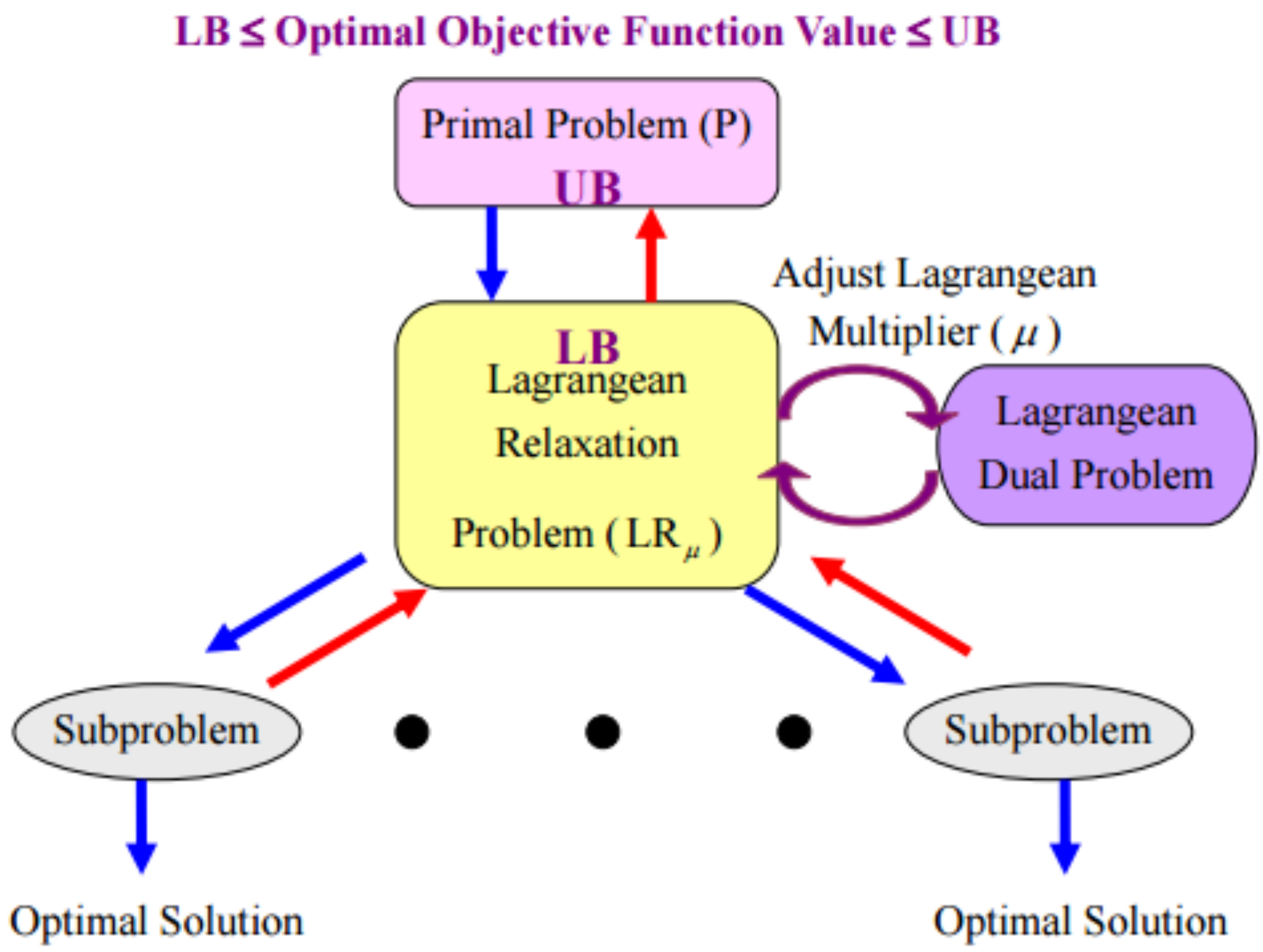
$$x_{ijl} + x_{ikl} \le f_{ijkl} + 1, \forall i \in D, \forall j \in W_i, \forall k \in W_i, \forall l \in S \qquad (c.12)$$

# II. Solution Approach

*A. Lagrangean Relaxation*

*B. Getting Primal Feasible Solutions*

# Lagrangean Relaxation



# Lagrangean Relaxation (Cont'd)

$$\begin{aligned}
Z_D(\mu^1,\mu^2,\mu^3,\mu^4,\mu^5,\mu^6,\mu^7,\mu^8,\mu^9,\mu^{10},\mu^{11}) = \min \sum_{i=1}^{d} -\gamma_i y_i \\
+\sum_{\forall i\in D} \mu_i^1 [y_i w_i - (\sum_{\forall j\in W_i} \sum_{\forall k\in S} x_{ijk})] + \sum_{\forall i\in D} \sum_{\forall j\in W_i} \sum_{\forall k\in S} \mu_{ijk}^2 [x_{ijk} p_{ij} - t_{ijk}\pi_k] \\
+\sum_{\forall i\in D} \sum_{\forall j\in W_i} \sum_{\forall k\in D} \sum_{\forall l\in W_k} \sum_{\forall m\in S} \mu_{ijklm}^3 [x_{ijm} + x_{klm} - a_{ijkl} - 1] \\
+\sum_{\forall l\in S} \mu_l^4 [(\sum_{\forall i\in D} \sum_{\forall j\in W_i, k\in W_i, k\neq j} c_{ijk} f_{ijkl}) - B_l^S] \\
+\sum_{\forall k\in S} \mu_k^5 [(\sum_{\forall i\in D} \sum_{\forall j\in W_i} t_{ijk}) - P_k] + \sum_{\forall k\in S} \mu_k^6 [(\sum_{\forall i\in D} \sum_{\forall j\in W_i} m_{ij} x_{ijk}) - M_k] \\
+\sum_{\forall k\in S} \mu_k^7 [(\sum_{\forall i\in D} \sum_{\forall j\in W_i} h_{ij} x_{ijk}) - H_k] + \sum_{\forall k\in S} \mu_k^8 [(\sum_{\forall i\in D} \sum_{\forall j\in W_i} x_{ijk}) - V_k] \\
+\sum_{\forall l\in S} \mu_l^9 [(\sum_{\forall i\in D} \sum_{\forall j\in W_i, k\in W_i, k\neq j} c_{ijk}(1 - f_{ijkl})) - B^c] \\
+\sum_{\forall i\in D} \sum_{j\in W_i} \mu_{ij}^{10} [(\sum_{\forall k\in S} x_{ijk}) - 1] + \sum_{\forall i\in D} \sum_{\forall j\in W_i} \sum_{\forall k\in W_i} \sum_{\forall l\in S} \mu_{ijkl}^{11} [x_{ijl} + x_{ikl} - f_{ijkl} - 1]
\end{aligned} \qquad \text{(LR 1)}$$

# Lagrangean Relaxation (Cont'd)

**Subject to:**

$y_i = 0 \; or \; 1, \forall i \in D$ (LR 1.1)

$t_{ijk} \geq 0, \forall i \in D, \forall j \in W_i, \forall k \in S$ (LR 1.2)

$x_{ijk} = 0 \; or \; 1, \forall i \in D, \forall j \in W_i, \forall k \in S$ (LR 1.3)

$f_{ijkl} = 0 \; or \; 1, \forall i \in D, \forall j \in W_i, \; \forall k \in W_i, \mathrm{k} \neq \mathrm{j}, \forall l \in S$ (LR 1.4)

# Lagrangean Relaxation (Cont'd)

- The Lagrangean multipliers $\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7, \mu^8, \mu^9, \mu^{10}, \mu^{11}$ are the vectors of $\{\mu_i^1\}$, $\{\mu_{ijk}^2\}$, $\{\mu_{ijklm}^3\}$, $\{\mu_l^4\}$, $\{\mu_k^5\}$, $\{\mu_k^6\}$, $\{\mu_k^7\}$, $\{\mu_k^8\}$, $\{\mu_l^9\}$, $\{\mu_{ij}^{10}\}$ and $\{\mu_{ijkl}^{11}\}$, respectively, where $\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7, \mu^8, \mu^9, \mu^{10}, \mu^{11}$ are non-negative. In order to solve (LR 1), it is decomposed into four independent and easily solvable subproblems, as shown below.

# Subproblem 1.1

**related to decision variable $\boldsymbol{y_i}$:**

$$Z_{sub1.1}(\mu^1) = \min \sum_{\forall i \in D} y_i(-\gamma_i + \mu_i^1 w_i) \qquad (sub\,1.1)$$

**Subject to:**

$$y_i = 0 \; or \; 1, \forall i \in D \qquad (sub\,1.1.1)$$

# Lagrangean Relaxation (Cont’d)

- Subproblem 1.1 (sub 1.1) is further decomposed into $|D|$ independent minimization subproblems. In Subproblem 1.1 (sub 1.1), decision variable $y_i$ has two options. The $y_i$ can be decided by examining the coefficient $-\gamma_i + \mu_i^1 w_i$. When $-\gamma_i + \mu_i^1 w_i$ is negative or zero, we set $y_i$ to 1. Otherwise, we set $y_i$ to 0.
- The time complexity of Subproblem 1.1 (sub 1.1) is O($|D|$).

# Subproblem 1.2

**related to decision variable $\boldsymbol{t_{ijk}}$:**

$$Z_{sub1.2}(\mu^2, \mu^5) = \min \sum_{\forall i \in D} \sum_{\forall j \in W_i} \sum_{\forall k \in S} t_{ijk}(\mu_k^5 - \mu_{ijk}^2 \pi_k) - \sum_{\forall k \in S} \mu_k^5 P_k \qquad (sub\,1.2)$$

**Subject to:**

$$t_{ijk} \geq 0, \forall i \in D, \forall j \in W_i, \forall k \in S \qquad (sub\,1.2.1)$$

# Lagrangean Relaxation (Cont’d)

- Subproblem 1.2 (sub 1.2) can be further decomposed to $|D|\times|W_i|\times|S|$ subproblems. This subproblem can be simply and optimally solved by examining the coefficient of decision variable $t_{ijk}$. If the coefficient $\mu_k^5 - \mu_{ijk}^2\pi_k$ is negative or zero, the value of $t_{ijk}$ is set the biggest value; conversely, if $\mu_k^5 - \mu_{ijk}^2\pi_k$ is bigger than zero, $t_{ijk}$ is set to zero.
- The time complexity of Subproblem 1.2 (sub 1.2) is $O(|D|\times|W_i|\times|S|)$.

# Subproblem 1.3

**related to decision variable $x_{ijk}$:**

$$Z_{sub1.3}(\mu^1,\mu^2,\mu^3,\mu^6,\mu^7,\mu^8,\mu^{10},\mu^{11})$$

$$= \min \sum_{\forall i\in D}\sum_{\forall j\in W_i}\sum_{\forall k\in S}(-\mu_i^1 x_{ijk} + \mu_{ijk}^2 x_{ijk} p_{ij} + \mu_k^6 m_{ij} x_{ijk} + \mu_k^7 h_{ij} x_{ijk} + \mu_k^8 x_{ijk} + \mu_{ij}^{10}\ x_{ijk} - \mu_{ij}^{10})$$

$$\boxed{+\sum_{\forall i\in D}\sum_{\forall j\in W_i}\sum_{\forall k\in W_i}\sum_{\forall l\in S}\mu_{ijkl}^{11}(x_{ijl}+x_{ikl})} + \sum_{\forall i\in D}\sum_{\forall j\in W_i}\sum_{\forall k\in D}\sum_{\forall l\in W_k}\sum_{\forall m\in S}\mu_{ijklm}^{3}(x_{ijm}+x_{klm}-a_{ijkl}-1)$$

$$-\sum_{\forall k\in S}(\mu_k^6 M_k + \mu_k^7 H_k + \mu_k^8 V_k) \qquad (sub\,1.3)$$

**Subject to:**

$$x_{ijk} = 0\ or\ 1, \forall i \in D, \forall j \in W_i, \forall k \in S \qquad (sub\,1.3.1)$$

# Lagrangean Relaxation (Cont’d)

- In order to solve the subproblem 1.3 (sub 1.3), we first reformulate (sub 1.3) by applying [1], as shown below.

[1] Frank Yeong-Sung Lin, “Quasi-static Channel Assignment Algorithms for Wireless Communications Networks,” Information Networking, 1998. (ICOIN-12) Proceedings, Twelfth International Conference, pp. 434-437, Jan 1998.

# Subproblem 1.3’

**related to decision variable $\boldsymbol{x_{ijk}}$:**

$$
\begin{aligned}
&Z_{sub1.3}(\mu^1,\mu^2,\mu^3,\mu^6,\mu^7,\mu^8,\mu^{10},\mu^{11}) \\
&= \min \sum_{\forall i\in D}\sum_{\forall j\in W_i}\sum_{\forall k\in S}(-\mu_i^1 x_{ijk} + \mu_{ijk}^2 x_{ijk} p_{ij} + \mu_k^6 m_{ij} x_{ijk} + \mu_k^7 h_{ij} x_{ijk} + \mu_k^8 x_{ijk} + \mu_{ij}^{10}\ x_{ijk} - \mu_{ij}^{10}) \\
&\boxed{+\sum_{\forall j\in W_i}\sum_{\forall i\in D}\sum_{\forall l\in S}\mu_{ijkl}^{11} x_{ijl}} + \sum_{\forall i\in D}\sum_{\forall j\in W_i}\sum_{\forall k\in D}\sum_{\forall l\in W_k}\sum_{\forall m\in S}\mu_{ijklm}^{3}(x_{ijm} + x_{klm} - a_{ijkl} - 1) \\
&-\sum_{\forall k\in S}(\mu_k^6 M_k + \mu_k^7 H_k + \mu_k^8 V_k) \qquad (sub\,1.3')
\end{aligned}
$$

**Subject to:**

$$x_{ijk} = 0\ or\ 1, \forall i \in D, \forall j \in W_i, \forall k \in S \qquad (sub\,1.3.1')$$

# Lagrangean Relaxation (Cont’d)

- In Subproblem 1.3’ (sub 1.3’), decision variable $x_{ijk}$ has two options. As a result, the value of $x_{ijk}$ can be determined by applying exhaustive search.
- The time complexity of Subproblem 1.3’ (sub 1.3’) is **O(|$\boldsymbol{D}$|×|$\boldsymbol{W_i}$|×|$\boldsymbol{S}$|).**

# Subproblem 1.4

**related to decision variable $\boldsymbol{f_{ijkl}}$:**

$$Z_{sub1.4}(\mu^4,\mu^9,\mu^{11}) = \min \sum_{\forall l\in S}\sum_{\forall i\in D}\sum_{\forall j\in W_i, k\in W_i, k\neq j}(\mu_l^4 c_{ijk} f_{ijkl} + \mu_l^9 c_{ijk}(1-f_{ijkl}))$$

$$-\sum_{\forall i\in D}\sum_{\forall j\in W_i}\sum_{\forall k\in W_i}\sum_{\forall l\in S}\mu_{ijkl}^{11}(f_{ijkl}+1) - \sum_{\forall l\in S}(\mu_l^4 B_k^S + \mu_l^9 B^c) \qquad (sub1.4)$$

**Subject to:**

$$f_{ijkl} = 0 \; or \; 1, \forall i\in D, \forall j\in W_i, \forall k\in W_i, \forall l\in S \qquad (sub1.4.1)$$

# Lagrangean Relaxation (Cont'd)

- Subproblem 1.4 (sub 1.4) can be optimally solved through analyzing the composition of (sub 1.4). When $\sum_{\forall l \in S} \sum_{\forall i \in D} \sum_{\forall j \in W_i, k \in W_i, j \neq k} (\mu_l^4 c_{ijk} f_{ijkl} + \mu_l^9 c_{ijk} (1 - f_{ijkl}))$ of subproblem 1.4 (sub 1.4) has minimum value and $\sum_{\forall i \in D} \sum_{\forall j \in W_i} \sum_{\forall k \in W_i} \sum_{\forall l \in S} \mu_{ijkl}^{11} (f_{ijkl} + 1) - \sum_{\forall l \in S} (\mu_l^4 B_k^S + \mu_l^9 B^c)$ of subproblem 1.4 (sub 1.4) has maximum value, (sub 1.4) is minimized.
- The time complexity of Subproblem 1.4 (sub 1.4) is **O(|*D*|×|$\boldsymbol{W_i}$|×|$\boldsymbol{W_i}$|×|*S*|).**

# Getting Primal Feasible Solutions

- By applying Lagrangean Relaxation method, a theoretical lower bound on primal objective function can be found. Moreover, it provides some suggestions for obtaining primal feasible solutions.
- However, the result of the dual problem may be invalid to the original problem since some important and complex constraints are relaxed. Therefore, a heuristic is needed here to tune infeasible solutions feasible.
- In order to obtain primal feasible solutions and an upper bound of (IP 1), the outcome of (LR 1) and Lagrangean multipliers are used as hints for deriving solutions. The concept of the proposed heuristic is described below.

# Getting Primal Feasible Solutions (Cont'd)

- Recall that subproblem 1.1 is related to decision variable $y_i$, which determines whether the user is admitted by cloud service provider or not. The concept used to develop a Drop and Add Heuristic to obtain primal feasible solutions by dropping and adding users to cloud environment.
- Initially, we check whether the outcome of (LR 1) can satisfy all constraints of (IP 1). If one of the constraints of (IP 1) is not satisfied, Drop and Add Heuristic will be used. There are two steps in Drop and Add Heuristic and the first step is dropping step.

# Getting Primal Feasible Solutions (Cont'd)

- The main purpose of the dropping step is to drop users one by one through adjusting the number of users or particular users we admit. By using the hint provided by subproblem 1.1, **we propose three strategies**.
  - In the first strategy, subproblem 1.1 has the minimum value when $\mu_i^1 w_i$ is minimized. Hence, we drop the biggest $\mu_i^1 w_i$ one after another.
  - In the second strategy, once $-\gamma_i$ has the smallest value, the subproblem 1.1 can obtain the optimal solution. Therefore, we drop the biggest $-\gamma_i$ one by one.
  - The third strategy is similar to the first and second strategy. When $-\gamma_i + \mu_i^1 w_i$ is minimized, subproblem 1.1 is also minimized. As a result, we drop the biggest $-\gamma_i + \mu_i^1 w_i$ one after another.

# Getting Primal Feasible Solutions (Cont’d)

- The dropping process keeps dropping users sequentially according the given strategies mentioned above, and **it stops while all the decision variables meet the constrains of (IP 1)**.
- The process in dropping step reallocates the remaining users’ VMs as much centralized as possible (on the same physical machine) under the premise that all constrains of (IP 1) are met. This reduces the severity of internal fragmentation, and we named such process as **compression**.

# Getting Primal Feasible Solutions (Cont’d)

- When we drop users sequentially, the degree of violation of constraints are also reduced. However, when the number of users is decreased, the rewards are also reduced. So, when we reduce the number of users that satisfy all constraints of (IP 1), we should consider about admitting more users to gain more rewards.

# Getting Primal Feasible Solutions (Cont'd)

- The second step of Drop and Add Heuristic is **adding step**. There are also three strategies.
  - The first strategy considers the **rewards** that users give to cloud service provider. We add users with the biggest reward one by one.
  - In the second strategy, the cloud service provider has to **spend resources** for admitting users into cloud environment. Consequently, we admit users who has the smallest requirement one after another.
  - At last, we use the **net profit** to decide the order of admitting users. The users with the biggest net profits are added into our system one by one.

# Getting Primal Feasible Solutions (Cont’d)

- The adding process considers the resource limits of physical machines and the upper limit of VM instances. When the given constrains are met, the cloud service provider allocates the VM $j$ of user $i$ on physical machine $k$, and check over all constrains of (IP 1).
- Then executes these actions repeatedly until all constrains of (IP 1) are satisfied. Every user requests will perform the compression process to minimize internal fragmentation.

# Getting Primal Feasible Solutions (Cont'd)

Table 3 Drop and Add Heuristic

```
// Step1: dropping process
switch (dropping strategy) {
case 1:
      sort admitted users by μ_i^1 w_i value in descending order;
      break;
case 2:
      sort admitted users by −γ_i value in descending order;
      break;
case 3:
      sort admitted users by −γ_i + μ_i^1 w_i value in descending order;
      break;
}
```

```
while (decision variables not satisfy constraints) {
        remove users one by one according to the dropping strategy;
}
do the compression process;

// Setp2: adding process
switch (adding strategy) {
case 1:
        sort not admitted users by their rewards in descending order;
        break;
case 2:
        sort not admitted users by their costs in ascending order;
        break;
case 3:
        sort not admitted users by their net profits in descending order;
        break;
}
```

```
while (decision variables satisfy constraints) {
    for each (not admitted users in order) {
        assign proper VMs to users;
        allocation user's VMs to proper server;
        if (decision variables satisfy constraints) {
            admit the user into cloud environment;
            do the compression process;
        }
    }
}
```

# III. Experiment Results

# Experiment Results

- To prove the **LR algorithm** and the proposed heuristic are effective, we used the **optimal software Lingo** for comparison purposes.
- Before that, we compared the different drop and add strategy combinations in the LR algorithm to determine which drop and add combination is more effective.

# Experiment Results (Cont’d)

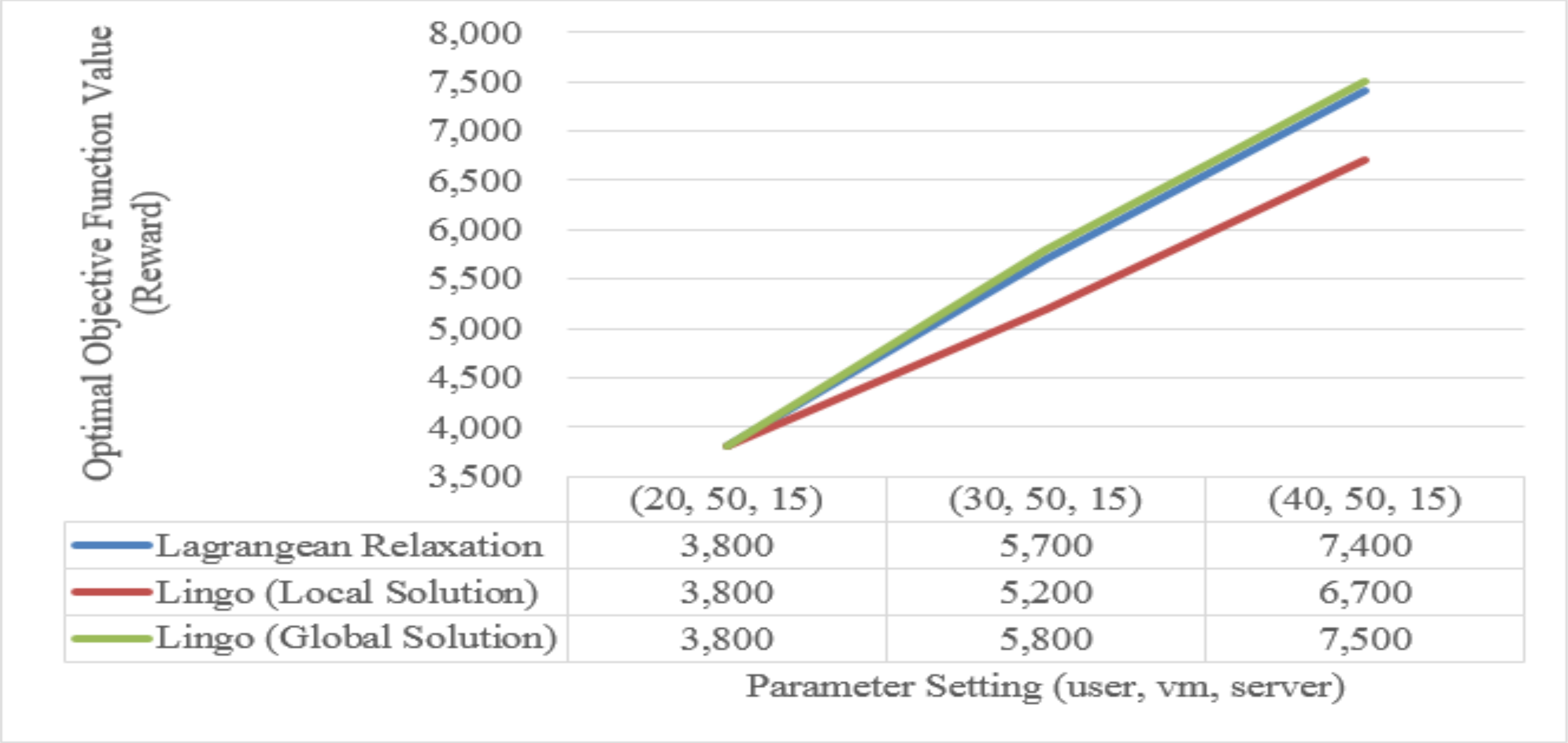
- Parameter setting
  - User demand (including CPU, RAM, HD respectively)
    - Level 1: Highest demand (150)
    - Level 2: Medium demand (80)
    - Level 3: Lowest demand (60)
  - Cloud service provider
    - Total number of VM: 50
    - Total number of physical server: 50
    - Total number of CPU cores, computing capability, RAM and HD: 3,000

# Experiment Results (Cont'd)

| | Case 1: drop users by $\mu_i^1 w_i$ | | Case2: drop users by $-\gamma_i$ | | Case3: drop users by $-\gamma_i + \mu_i^1 w_i$ | |
|---|---|---|---|---|---|---|
| Case 1: add users by their rewards ($\gamma_i$) | Reward | 7,100 | Reward | 6,800 | Reward | 7,100 |
| | Time (m) | 168.2 | Time (m) | 201.6 | Time (m) | 189.7 |
| Case 2: add users by their costs ($\mu_i^1 w_i$) | Reward | 7,100 | Reward | 6,800 | Reward | 7,100 |
| | Time (m) | 168.2 | Time (m) | 177.3 | Time (m) | 191.3 |
| Case 3: add users by their net profits ($-\gamma_i + \mu_i^1 w_i$) | Reward | 7,400 | Reward | 7,100 | Reward | 7,400 |
| | Time (m) | 185.3 | Time (m) | 177.1 | Time (m) | 190.9 |

# Experiment Results (Cont’d)

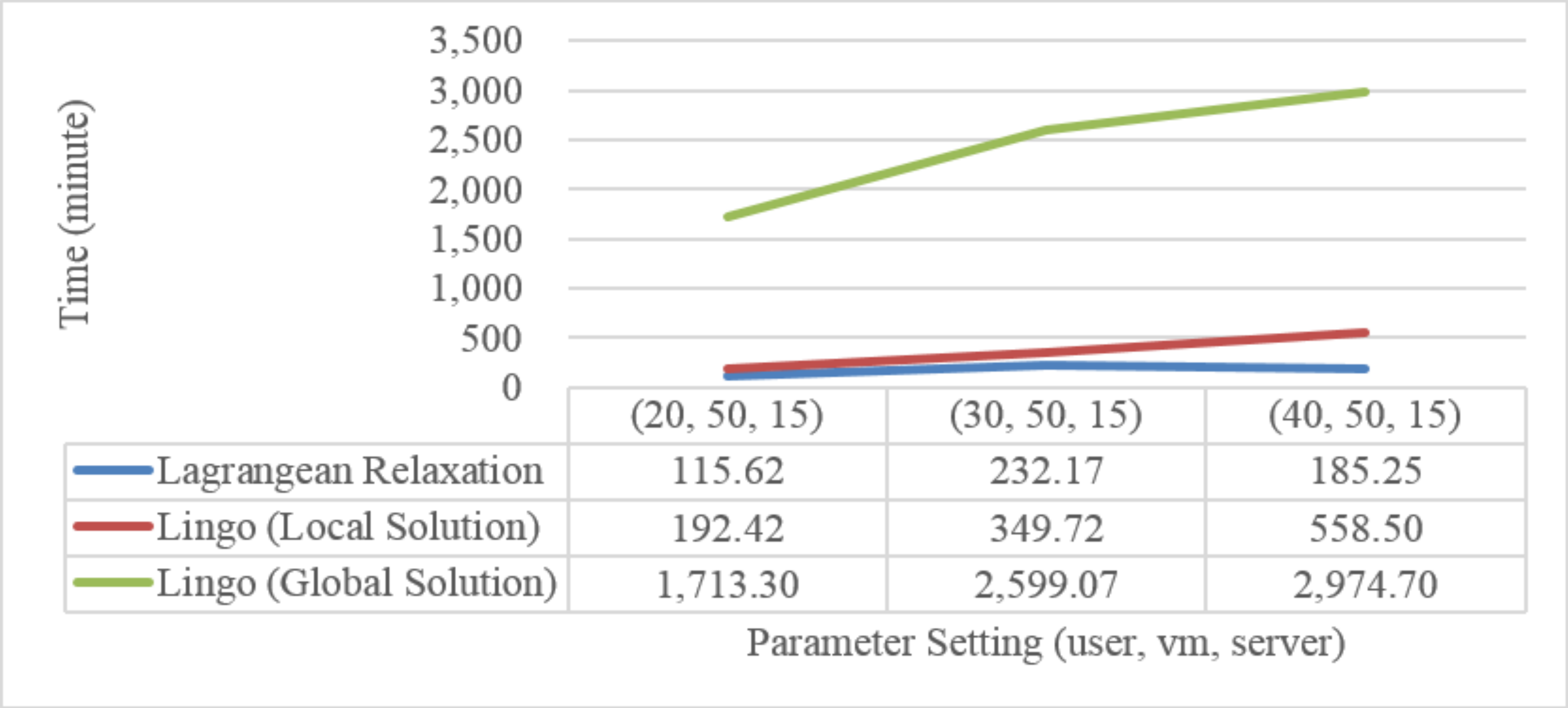
Reward of Different Algorithm




| | (20, 50, 15) | (30, 50, 15) | (40, 50, 15) |
|---|---|---|---|
| Lagrangean Relaxation | 3,800 | 5,700 | 7,400 |
| Lingo (Local Solution) | 3,800 | 5,200 | 6,700 |
| Lingo (Global Solution) | 3,800 | 5,800 | 7,500 |

# Experiment Results (Cont’d)

Time Consumption of Different Algorithm




| | (20, 50, 15) | (30, 50, 15) | (40, 50, 15) |
|---|---|---|---|
| Lagrangean Relaxation | 115.62 | 232.17 | 185.25 |
| Lingo (Local Solution) | 192.42 | 349.72 | 558.50 |
| Lingo (Global Solution) | 1,713.30 | 2,599.07 | 2,974.70 |

# Experiment Results (Cont’d)

Gap of Different Algorithms

| | (20, 50, 15) | (30, 50, 15) | (40, 50, 15) |
|---|---|---|---|
| **Lagrangean Relaxation** | 0% | 4.17% | 5.71% |
| **Lingo (Local Solution)** | 0% | 10.34% | 10.67% |
| **Lingo (Global Solution)** | 0% | 0% | 0% |

# IV. Conclusions

# Conclusions

- The main contribution of this research is the production of a **generic mathematical model** that can develop an algorithm for allocating resources and virtual machines.
- From the experiment result, we found that our proposed Lagrangean relaxation-based algorithm and drop and add heuristic show excellent performance in **effectiveness** and **efficiency** in comparison with Lingo.
- From the outcomes of these experiments, we conclude that our research can be effectively applied in real-world situations.

# Thanks for your listening