Utility-based Resource Allocation and Pricing for Serverless Computing with Dependencies

Vipul Gupta[1] Soham Phade[2] Yigit Efe Erginbas[3] Thomas Courtade[3]
Kannan Ramchandran[3]

[1]Coinbase [2]Wayve [3]UC Berkeley

Abstract

Serverless computing platforms currently employ static pricing schemes that often lead to inefficiencies. To address this, our prior work introduced a novel market-based scheduler that utilizes user utility functions to optimize resource allocation and maximize social welfare based on delay-sensitivity. This paper extends that framework to tackle a critical challenge in cloud computing: the efficient scheduling of jobs with inter-dependencies. We propose an enhanced scheduler capable of allocating resources for serverless computing tasks with finish-to-start dependencies, ensuring that the overall system utility is maximized while respecting these constraints. Our approach retains the dynamic pricing mechanism derived from the dual problem and the decentralized feedback mechanisms for handling private user information, now incorporating the complexities introduced by job dependencies. Simulations demonstrate that our extended framework can effectively manage dependent tasks, track market demand, and achieve significantly higher social welfare compared to existing schemes that do not account for these dependencies.

1 Introduction

Resource allocation in cloud computing environments is a critical area of research, aiming to efficiently manage shared resources to maximize system performance and user satisfaction. Various scheduling frameworks have been proposed to address this challenge, often considering factors such as user preferences and job characteristics. In our prior work, we introduced a novel decentralized framework for resource allocation in multi-agent scheduling scenarios, focusing on maximizing social welfare through a dynamic multi-tier pricing scheme that incentivizes users to bid optimally based on their delay-sensitivity. This framework, detailed in the introduction provided below, effectively addressed the complexities of scheduling delay-sensitive jobs, particularly within the burgeoning field of serverless computing.

Building upon this foundation, the current paper extends our previous work by incorporating the crucial aspect of job inter-dependencies. In many real-world cloud computing systems, jobs are not isolated tasks but rather form complex workflows with dependencies, where the execution of one job can be contingent on the completion of others. These dependencies significantly impact scheduling decisions and resource allocation strategies. To address this, we introduce an extension of our resource allocation and pricing framework to handle finish-to-start relationships between multiple pairs of jobs. By representing the dependency structure as a directed acyclic graph (DAG), we can explicitly model the constraints imposed by these inter-dependencies within our optimization problem. This extension allows for a more realistic and comprehensive approach to resource allocation in cloud environments where job dependencies are prevalent.

2 Problem Formulation

This paper extends the resource allocation framework for multi-agent scheduling in serverless computing environments that we introduced in our prior work. In that work, we focused on optimizing resource allocation based on user utility functions that capture their delay-sensitivity for independent jobs. Building upon this foundation, the current paper incorporates the crucial aspect of job inter-dependencies, specifically focusing on finish-to-start relationships.

Serverless (also called Function-as-a-service) as a cloud computing framework is ideal for "simple" jobs where each *user* submits a *function* to be executed on serverless workers. Each user can request for a job comprising of any number of executions of her function at any time, which could be triggered due to external events. For example, the users can provide the conditions under which they require the execution of a certain number of instances of their function. The users provide these *trigger event* details along with their function submissions. Our goal here is to design an efficient real-time job scheduler to allocate resources to the jobs that have been triggered across multiple users and a corresponding pricing scheme for the cloud provider.

We envision a job scheduler that operates periodically and schedules the jobs that are currently in its queue. This queue consists of all the jobs that have been triggered and are ready to be executed but have not been scheduled yet. Thus, it consists of previously unscheduled jobs (complete or partial) plus any new jobs that arrived since the previous run of the scheduler.

Our model assumes that the user derives utility only when her job is completed, that is, when all the functions comprising her job are executed where each function execution requires one serverless worker[1]. Let the respective delay-sensitivity for user i be captured by a utility function $U_i : [0, \infty) \to \mathbb{R}$. That is, user i obtains utility $U_i(\tau)$ if her job is completed at time instant $\tau (> 0)$, where $U_i(\cdot)$ is non-increasing. Let J_i denote the number of function executions needed to complete the job of user i. We call this the size of user i's job. For example, it could be a single function instance in which case $J_i = 1$ or a batch job of size $J_i > 1$.

We think of the scheduler as allocating resources to the users in different *service tiers* based on their execution times—some jobs will be scheduled for immediate execution, whereas others will be scheduled for execution at later times in the future. The jobs that are not scheduled will remain in the queue to be scheduled by the scheduler at later operations. Note that the jobs that have been scheduled for execution at a future point are removed from the queue. Let T be the maximum number of service tiers offered by the cloud provider, and let the t-th tier be characterized by the end time of this tier given by τ_t. This is a useful feature because it allows our scheduler to plan over longer time horizons with a limited number of tiers. The pricing for these different tiers of service should ideally decrease as the job completion time increases. We refer to the intervening time between the adjacent tier end times as *service intervals* and we let M_t be the constraint on the number of machines available for the scheduler to allocate resources in service interval t.

System Problem without Dependencies: In our prior work, at each scheduler implementation, we assumed access to unscheduled jobs (size J_i for user i), their utility functions $U_{i,t} := U_i(\tau_t)$, and machine availability M_t. We formulated the system problem SYS to maximize the sum utility as follows:

SYS

$$\underset{x_{i,t} \geq 0}{\text{Maximize}} \quad \sum_{i=1}^{n} U_{i,T_i}$$

[1]Without loss of generality, we make a simplifying assumption that all users' functions are identical in terms of computation costs. Thus, the pricing scheme in a given tier charges users only on the basis of their job sizes.

$$\text{subject to} \quad \sum_{i=1}^{N} x_{i,t} \leq M_t, \forall\, t \in [T], \text{ and} \tag{2.1}$$

$$T_i = \min\{t \in [T] : \sum_{s=1}^{t} x_{i,s} \geq J_i\}, \ \forall\, i \in [N]. \tag{2.2}$$

In this paper, we extend the aforementioned framework to account for finish-to-start dependencies between jobs. We represent the dependency structure as a directed acyclic graph (DAG) $G = ([N], E)$, where $[N]$ is the set of jobs and a directed edge $(i, j) \in E$ indicates that job j can only start after job i is completed.

To incorporate these dependencies into our system problem, we define the starting time S_i and completion time T_i for each job i. The completion time T_i of job i is defined as

$$T_i := \min\{t \in [T] : \sum_{s=1}^{t} x_{i,s} \geq J_i\}, \forall i \in [N],$$

and the starting time is defined as

$$S_i := \min\{t \in [T] : \sum_{s=1}^{t} x_{i,s} > 0\}, \forall i \in [N]. \tag{2.3}$$

System Problem with Dependencies: The system problem is now modified to maximize the total utility while respecting both resource capacity constraints and job dependencies:

SYS-DEP

$$\begin{aligned}
\underset{x_{i,t} \geq 0}{\text{Maximize}} \quad & \sum_{i=1}^{N} U_{i,T_i} \\
\text{subject to} \quad & \sum_{i=1}^{N} x_{i,t} \leq M_t, \forall t \in [T] \\
& T_i = \min\{t \in [T] : \sum_{s=1}^{t} x_{i,s} \geq J_i\}, \forall i \in [N] \\
& S_i = \min\{t \in [T] : \sum_{s=1}^{t} x_{i,s} > 0\}, \forall i \in [N] \\
& T_i < S_j, \forall (i, j) \in E
\end{aligned}$$

We can formulate an equivalent ILP problem. Using the utility increments $u_{i,t} := U_{i,t} - U_{i,t+1}$ and introducing binary indicator variables $y_{i,t}$ (job i completed by time t) and $z_{i,t}$ (job i started by time t), the problem becomes:

SYS-DEP-ILP

$$\begin{aligned}
\underset{x_{i,t}\geq 0, y_{i,t}\in\{0,1\}, z_{i,t}\in\{0,1\}}{\text{Maximize}} \quad & \sum_{i=1}^{N}\sum_{t=1}^{T} u_{i,t}y_{i,t} \\
\text{subject to} \quad & \sum_{i=1}^{N} x_{i,t} \leq M_t, \forall t \in [T] \\
& y_{i,t} \leq \frac{\sum_{s=1}^{t} x_{i,s}}{J_i}, \forall i \in [N], t \in [T] \\
& \frac{\sum_{s=1}^{t} x_{i,s}}{J_i} \leq z_{i,t}, \forall i \in [N], t \in [T] \\
& y_{i,t-1} \geq z_{j,t}, \forall (i,j) \in E, t \in [T]
\end{aligned}$$

with the convention that $y_{i,0} = 0$.

3 Analyzing SYS-DEP-ILP

Since the problems SYS-DEP and SYS-DEP-ILP are NP-hard in general, we consider an approximation by relaxing the integer constraints. Accordingly, we replace the indicator constraints $y_{i,t} \in \{0,1\}$ by $0 \leq y_{i,t} \leq 1$ and $z_{i,t} \in \{0,1\}$ by $0 \leq z_{i,t} \leq 1$.

We note that the objective value for the relaxed problem only depends on $y_{i,t}$ values. Therefore, for any feasible solution, there exists an equivalent feasible solution (with the same objective value) that satisfies $y_{i,t} = \sum_{s=1}^{t} x_{i,s}/J_i$. Hence, we can substitute $y_{i,t} = \sum_{s=1}^{t} x_{i,s}/J_i$ in the objective function and dependency constraints. Note that we also need to satisfy $\sum_{t=1}^{T} x_{i,t} \leq J_i$ since $0 \leq y_{i,t} \leq 1$. Furthermore, by reducing $z_{i,t}$ values, we can also find an equivalent feasible solution that satisfies $z_{i,t} = \sum_{s=1}^{t} x_{i,s}/J_i$. So, we can substitute $z_{i,t} = \sum_{s=1}^{t} x_{i,s}/J_i$ in the objective function and dependency constraints.

Using the definition of utility per unit $F_{i,t} = U_{i,t}/J_i$, we obtain the relaxed problem

SYS-DEP-LP

$$\begin{aligned}
\underset{x_{i,t}\geq 0}{\text{Maximize}} \quad & \sum_{i=1}^{N}\sum_{t=1}^{T} x_{i,t}F_{i,t} \\
\text{subject to} \quad & \sum_{t=1}^{T} x_{i,t} \leq J_i, \forall i \in [N] \\
& \sum_{i=1}^{N} x_{i,t} \leq M_t, \forall t \in [T] \\
& \frac{\sum_{s=1}^{t-1} x_{i,s}}{J_i} \geq \frac{\sum_{s=1}^{t} x_{j,s}}{J_j}, \forall (i,j) \in E, t \in [T].
\end{aligned}$$

where empty sums (sums with no summands) are assumed to be equal to 0.

In order to formulate a decentralized algorithm that will solve this problem, we continue with writing the Lagrangian corresponding to the optimization problem SYS-DEP-LP:

$$
\begin{aligned}
L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\rho}) &= \sum_{i=1}^{N}\sum_{t=1}^{T} x_{it}F_{it} + \sum_{i=1}^{N}\lambda_i\left(J_i - \sum_{t=1}^{T} x_{it}\right) \\
&\quad + \sum_{t=1}^{T}\mu_t\left(M_t - \sum_{i=1}^{N} x_{it}\right) \\
&\quad + \sum_{(i,j)\in E}\sum_{t=1}^{T}\rho_{ij}^{t}\left(\frac{\sum_{s=1}^{t-1} x_{is}}{J_i} - \frac{\sum_{s=1}^{t} x_{j,s}}{J_j}\right)
\end{aligned}
\tag{3.1}
$$

where $\boldsymbol{x} = [x_{it}]$, $\boldsymbol{\lambda} = [\lambda_i]$, $\boldsymbol{\mu} = [\mu_t]$, $\boldsymbol{\rho} = [\rho_{ij}^t]$. Here, $\boldsymbol{x}$ represents the primal variables that correspond to the number of function calls made for each user at each time. $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ are Lagrangian multipliers and they correspond to the dual variables for the problem. By manipulating the given expression for the Lagrangian, we can write it as

$$
\begin{aligned}
L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\rho}) &= \sum_{i=1}^{N}\sum_{t=1}^{T} x_{it}F_{it} + \sum_{i=1}^{N}\lambda_i\left(J_i - \sum_{t=1}^{T} x_{it}\right) \\
&\quad + \sum_{t=1}^{T}\mu_t\left(M_t - \sum_{i=1}^{N} x_{it}\right) \\
&\quad + \sum_{(i,j)\in E}\left(\frac{1}{J_i}\sum_{t=1}^{T}\left(\sum_{s=t+1}^{T}\rho_{ij}^{s}\right)x_{it} - \frac{1}{J_j}\sum_{t=1}^{T}\left(\sum_{s=t}^{T}\rho_{ij}^{s}\right)x_{jt}\right)
\end{aligned}
\tag{3.2}
$$

Next, we decompose SYS-DEP-LP into a user problem per job as well as a cloud problem.

We first analyze the problem from the i-th user's perspective. Similar to the problem without dependency constraints, the cloud service provider sets base prices q_t for each service interval t. Additionally, due to dependency constraints, there also are associated fees for each dependency relation between jobs. If job i directly depends on the completion of another job j, then i-th user is required to pay an extra fee θ_{ij}^t for its job to be scheduled in service interval t or before. On the other hand, if another job j directly depends on the completion of another job i, then i-th user is offered a discount bonus θ_{ji}^t for scheduling before service interval t. Formally, the user-specific prices for each service interval are

$$
z_{it} = q_t + \sum_{j:(j,i)\in E}\frac{1}{J_i}\left(\sum_{s=t}^{T}\rho_{ji}^{s}\right) - \sum_{j:(i,j)\in E}\frac{1}{J_i}\left(\sum_{s=t+1}^{T}\rho_{ij}^{s}\right). \tag{3.3}
$$

Then, the user feedback is provided in the form of user budgets defined as $m_{it} = x_{it}z_{it}$ and computed by the user problem

USER-DEP(i)

$$
\underset{m_{it}\geq 0}{\text{Maximize}} \quad \sum_{t=1}^{T}\frac{m_{it}}{z_{it}}\left(F_{i,t} - z_{it}\right) \tag{3.4}
$$

$$
\text{subject to} \quad \sum_{t=1}^{T}\frac{m_{it}}{z_{it}} \leq J_i \tag{3.5}
$$

Cloud provider then receives budgets from all users and solves the cloud problem defined as

CLOUD-DEP

$$\underset{x_{i,t}\geq 0}{\text{Maximize}} \quad \sum_{i=1}^{N}\sum_{t=1}^{T} m_{it}\log x_{it} \tag{3.6}$$

$$\text{subject to} \quad \sum_{i=1}^{N} x_{i,t} \leq M, \ \forall\, t \in [T] \tag{3.7}$$

$$\frac{\sum_{s=1}^{t-1} x_{i,s}}{J_i} \geq \frac{\sum_{s=1}^{t} x_{j,s}}{J_j}, \ \forall (i,j) \in E, \forall t \in [T]. \tag{3.8}$$

Theorem 3.1. *There exist an equilibrium allocation matrix* $\mathbf{x} = (x_{i,t}, i \in [N],\ t \in [T])$ *with associated budgets* $\mathbf{m} = (m_{i,t}, i \in [N], t \in [T])$*, prices* $\mathbf{q} = (q_t, t \in [T])$*, and dependency fees* $\boldsymbol{\theta} = (\theta_{ij}^t, (i,j) \in E, t \in [T])$ *such that*

(i) $\mathbf{m}_i = (m_{i,t}, t \in [T])$ *solves the problem USER-DEP(i) ,* $\forall\, i \in [N]$,

(ii) $\mathbf{x} = (x_{i,t}, i \in [N], t \in [T])$ *solves the problem CLOUD-DEP,*

(iii) $m_{i,t} = x_{i,t}\left(q_t + \sum_{j:(j,i)\in E}\frac{1}{J_i}\left(\sum_{s=t}^{T}\theta_{ji}^s\right) - \sum_{j:(i,j)\in E}\frac{1}{J_i}\left(\sum_{s=t+1}^{T}\theta_{ij}^s\right)\right), \forall i \in [N], t \in [T]$,

(iv) $(M_t - \sum_i x_{i,t})q_t = 0, \forall t$,

(v) $\left(\frac{\sum_{s=1}^{t-1} x_{is}}{J_i} - \frac{\sum_{s=1}^{t} x_{js}}{J_j}\right)\theta_{ij}^t = 0, \forall t$ *and* $\forall (i,j) \in E$.

Further, if any matrix $\mathbf{x}$ *is at equilibrium, i.e. has corresponding* $\mathbf{m}$*,* $\mathbf{q}$*, and* $\boldsymbol{\theta}$ *that together satisfy (i), (ii), (iii), (iv), and (v), then* $\mathbf{x}$ *solves the system problem SYS-DEP-LP.*

A Proofs

Proof of Theorem 3.1. Note that we can write the Lagrangian for SYS-DEP-LP as

$$\begin{aligned} L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\rho}) &= \sum_{i=1}^{N}\sum_{t=1}^{T} x_{it}F_{it} + \sum_{i=1}^{N}\lambda_i\left(J_i - \sum_{t=1}^{T} x_{it}\right) \\ &\quad + \sum_{t=1}^{T}\mu_t\left(M_t - \sum_{i=1}^{N} x_{it}\right) \\ &\quad + \sum_{t=1}^{T}\sum_{(i,j)\in E}\rho_{ij}^t\left(\frac{\sum_{s=1}^{t-1} x_{is}}{J_i} - \frac{\sum_{s=1}^{t} x_{js}}{J_j}\right) \end{aligned} \tag{A.1}$$

Now, let $\mathbf{x}$ be an optimal solution to SYS-DEP-LP and let $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and $\boldsymbol{\rho}$ be the dual variables corresponding to this solution. We know that these satisfy the Karush-Kuhn-Tucker (KKT) conditions [1] given as

$$\mu_t + \lambda_i + \frac{\sum_{j:(j,i)\in E}\sum_{s=t}^{T}\rho_{ji}^s - \sum_{j:(i,j)\in E}\sum_{s=t+1}^{T}\rho_{ij}^s}{J_i} \begin{cases} = F_{i,t}, & \text{if } x_{i,t} > 0, \\ \geq F_{i,t}, & \text{if } x_{i,t} = 0, \end{cases} \qquad \forall i, t, \tag{A.2}$$

$$\sum_{i=1}^{N} x_{i,t} \begin{cases} = M_t, & \text{if } \mu_t > 0, \\ \leq M_t, & \text{if } \mu_t = 0, \end{cases} \qquad \forall t, \tag{A.3}$$

$$\sum_{t=1}^{T} x_{i,t} \begin{cases} = J_i, & \text{if } \lambda_i > 0, \\ \leq J_i, & \text{if } \lambda_i = 0, \end{cases} \qquad \forall i \tag{A.4}$$

$$\frac{\sum_{s=1}^{t-1} x_{is}}{J_i} - \frac{\sum_{s=1}^{t} x_{js}}{J_j} \begin{cases} = 0, & \text{if } \rho_{ij}^t > 0, \\ \geq 0, & \text{if } \rho_{ij}^t = 0 \end{cases} \qquad \forall (i,j) \in E, \forall t \tag{A.5}$$

Now, set $\mathbf{q} = \boldsymbol{\mu}$, $\boldsymbol{\theta} = \boldsymbol{\rho}$, and $m_{it} = x_{it} z_{it}$ where the user-specific prices are given by

$$z_{it} = q_t + \frac{\sum_{j:(j,i)\in E}\sum_{s=t}^{T}\rho_{ji}^s - \sum_{j:(i,j)\in E}\sum_{s=t+1}^{T}\rho_{ij}^s}{J_i}.$$

for all i, t.

We will now show that $\mathbf{m}_i$ solves USER-DEP(i) for this $\mathbf{q}$ and $\boldsymbol{\rho}$. Observe that $m_{i,t} = 0$ if $z_{i,t} = 0$ by definition. Thus, it is enough to look at the user-tier pairs for which $z_{i,t} \neq 0$. Hence, without loss of generality, we will assume that $z_{i,t} \neq 0$ for all i and t. Consider the Lagrangian for the user problem USER-DEP(i),

$$L(\mathbf{m}_i, p_i) = \sum_{t=1}^{T} \frac{m_{it}}{z_{it}}(F_{i,t} - z_{i,t}) + p_i \left(J_i - \sum_{t=1}^{T} \frac{m_{it}}{z_{it}} \right)$$

where p_i is the dual variable corresponding to the job size constraint (3.5). Thus, the KKT conditions for USER-DEP(i) can be written as

$$z_{it} + p_i \begin{cases} = F_{i,t}, & \text{if } m_{i,t} > 0 \\ \geq F_{i,t}, & \text{if } m_{i,t} = 0, \end{cases} \qquad \forall i, t, \tag{A.6}$$

$$\sum_{t=1}^{T} \frac{m_{i,t}}{z_{it}} \begin{cases} = J_i, & \text{if } p_i > 0 \\ \leq J_i, & \text{if } p_i = 0, \end{cases} \qquad \forall t. \tag{A.7}$$

Taking $p_i = \lambda_i$, we observe that these KKT conditions are satisfied. Thus, $\mathbf{m}_i$ is an optimal solution to USER-DEP(i) with $\mathbf{q} = \boldsymbol{\mu}$ and $\boldsymbol{\theta} = \boldsymbol{\rho}$.

Next, we will show that $\mathbf{m}$ is an optimal solution for the CLOUD-DEP problem. The Lagrangian for CLOUD-DEP is given by

$$L(\mathbf{x}, \tilde{\mathbf{q}}, \tilde{\boldsymbol{\rho}}) = \sum_{i=1}^{N}\sum_{t=1}^{T} m_{it} \log x_{it} + \tilde{q}_t \left(M - \sum_{i=1}^{N} x_{it} \right) + \sum_{(i,j)\in E}\sum_{t=1}^{T} \tilde{\rho}_{ij}^t \left(\frac{\sum_{s=1}^{t-1} x_{is}}{J_i} - \frac{\sum_{s=1}^{t} x_{j,s}}{J_j} \right)$$

where $\tilde{\mathbf{q}} = (\tilde{q}_t, t \in [N])$ is the dual variable corresponding to the load constraint (3.7) and $\tilde{\boldsymbol{\rho}} = (\tilde{\rho}_{ij}, (i,j) \in E)$ is the dual variable corresponding to the dependency constraint (3.8) in the CLOUD-DEP problem.

Consequently, the KKT conditions for the CLOUD-DEP problem can be written as

$$\frac{m_{it}}{x_{it}} - \tilde{q}_t - \frac{\sum_{j:(j,i)\in E}\sum_{s=t}^{T}\rho_{ji}^{s} - \sum_{j:(i,j)\in E}\sum_{s=t+1}^{T}\rho_{ij}^{s}}{J_i} \begin{cases} = 0, & \text{if } x_{i,t} > 0 \\ \leq 0, & \text{if } x_{i,t} = 0, \end{cases} \qquad \forall i, t, \tag{A.8}$$

$$\sum_{i=1}^{N} x_{it} \begin{cases} = M, & \text{if } \tilde{q}_t > 0 \\ \leq M, & \text{if } \tilde{q}_t = 0, \end{cases} \qquad \forall t. \tag{A.9}$$

$$\frac{\sum_{s=1}^{t-1} x_{is}}{J_i} - \frac{\sum_{s=1}^{t} x_{js}}{J_j} \begin{cases} = 0, & \text{if } \tilde{\rho}_{ij}^{t} > 0, \\ \geq 0, & \text{if } \tilde{\rho}_{ij}^{t} = 0 \end{cases} \qquad \forall (i,j) \in E, \forall t \tag{A.10}$$

Now, setting $\tilde{\mathbf{q}} = \boldsymbol{\mu}$, $\tilde{\boldsymbol{\rho}} = \boldsymbol{\rho}$ we satisfy the KKT conditions (A.9) and (A.10). If $m_{it} > 0$ for some i and t, then $x_{it} > 0$ and $\frac{m_{it}}{x_{it}} - z_{it} = 0$. On the other hand, if $m_{it} = 0$, then $x_{it} = 0$ and we have $-z_{it} \leq 0$. Hence $\mathbf{x}$ is an optimal solution to CLOUD-DEP with $\mathbf{m}$. Thus we have showed statements (i) and (ii) in Theorem 3.1. Statement (iii) follows from construction, statement (iv) follows from (A.3), and statement (v) follows from (A.5).

We now prove the later assertion, namely, if we have an equilibrium solution $\mathbf{x}, \mathbf{m}, \mathbf{q}, \boldsymbol{\rho}$ that satisfy (i)-(v), then $\mathbf{x}$ solves the system problem SYS-DEP-LP. To see this, take $\boldsymbol{\mu} = \mathbf{q}$ and $\boldsymbol{\rho} = \boldsymbol{\rho}$. Since $\mathbf{m}_i$ is an optimal solution to USER-DEP(i) with $\mathbf{q}$ and $\boldsymbol{\rho}$, there exists dual a variable p_i corresponding to the constraint (3.5). Take $\lambda_i = p_i$ for all i. We can verify that $\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\rho}$ satisfy the KKT consitions (A.2), (A.3), (A.5), and (A.5), and hence, form an optimal solution to SYS-DEP-LP. This completes the proof of the theorem.

□

References

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.