Optimized Resource Sharing for Federated Cloud Services with Desired Performance and Limited OpEx

Raouf Abozariba*, Anas Amjad* and Mohammad N. Patwary†
*School of Creative Arts and Engineering, Staffordshire University, Staffordshire ST4 2DE, UK
†School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7BD, UK

r.abozariba@staffs.ac.uk, *anasamjad@ieee.org*, *mohammad.patwary@bcu.ac.uk*

***Abstract*—The provision of cloud resources to meet user demands in 5G wireless networks is a challenging task due to the high workload predicted to be experienced by cloud service providers (CSPs). Cloud federation has emerged as a paradigm to support CSPs with resource limitations by borrowing surplus resources of other CSPs in periods of high demands. The major concern of each CSP with resource limitations is to borrow resources from other federation participants in such a way that cloud services are provided to the end-users with a desired grade of service (GoS) as well as the overall profit is maximized. This paper proposes an efficient mechanism based on the merchant mode to dynamically facilitate optimal allocation of cloud resources, maximizing the profit of CSPs as well as improving the GoS. The robustness of the proposed optimal scheme is evaluated by comparing it with the heuristic algorithm. The simulation results demonstrate that at each trading window, the proposed optimal scheme outperforms its heuristic counterpart. Moreover, after 50 trading windows, the proposed approach results in 43.5% gain in net profit to CSPs as well as facilitating 3.35% of additional resource.**

***Index Terms*—Cloud computing, cloud federation, internet of things, resource sharing, profit maximization.**

I. Introduction

In future generation of wireless networks, the provision of reliable connectivity is one of the primary concerns due to the exponentially growing number of connected devices [1], [2], [3]. In recent years, the advances in Internet of Things (IoT) have shown promising solutions to dynamically sustain network connectivity over global infrastructures [1], [3]. In order to cope with the limited computational and storage capabilities of diversely located devices, the integration of IoT with cloud computing has shown promising developments for a broad spectrum of applications [4], [5], [6]. Cloud computing facilitates user access to a shared pool of virtually unlimited available resources. Nevertheless, due the growing demand from users to access cloud services, a performance bottleneck is expected to be experienced by the federation participants. In order to guarantee the global success of 5G, the coupling of IoT and cloud computing is required to assure maximum possible service provision reliability. Therefore, assuming unlimited cloud resources for future generation of wireless network design may not be able to meet the users' expectations [7], [8], [9].

Exploiting the potential of cloud computing, cloud service providers (CSPs) have to meet users' requirements by delivering services in a flexible manner with acceptable delay [10]. The end-users are mostly concerned about the cloud services they require, the services cost and the grade of service (GoS) experienced; regardless of the underlying infrastructure [11]. Cloud federations have emerged as an appealing way to isolate the underlying infrastructure from end-users and deliver services with an improved degree of flexibility [12]. In a cloud federation, independent CSPs collaborate dynamically for load balancing, capacity management, efficient utilisation of surplus resources, minimizing failures, improving GoS and enhancing cost efficiency [13], [14]. Within the federation, if a particular CSP's available resources are insufficient to fulfil all the requests due to high workload, it can borrow resources from other CSPs having surplus resources on a pay-per-use basis. Such an approach can minimise the possibility of violating service level agreements (SLAs) and ensures high level of service provision. In such a scenario, one of the primary concerns is to select particular CSPs for borrowing their resources in a cost effective manner as well as without compromising on the target service denial probability.

In recent years, several research studies have focused on cloud resource management and sharing. In [15], a game-theoretic approach is considered to develop a resource sharing model for profit maximization in a federation of cloud providers. Another approach based on game theory for resource sharing in cloud federation leading to profit maximization is presented in [16]. In contrast to [15], the strategy presented in [16] can determine the optimal criteria for service provision to the end-users. Hassan et al. in [17] also considered a game-theoretic approach and investigated the energy consumption issues for federated cloud resources and revenue management. The developed model aims to minimize the energy expenditure by selecting low-energy-cost cloud providers from the federation for resource sharing. In addition to the game-theoretic approaches, auction mode is widely studied in existing literature for resource sharing in cloud

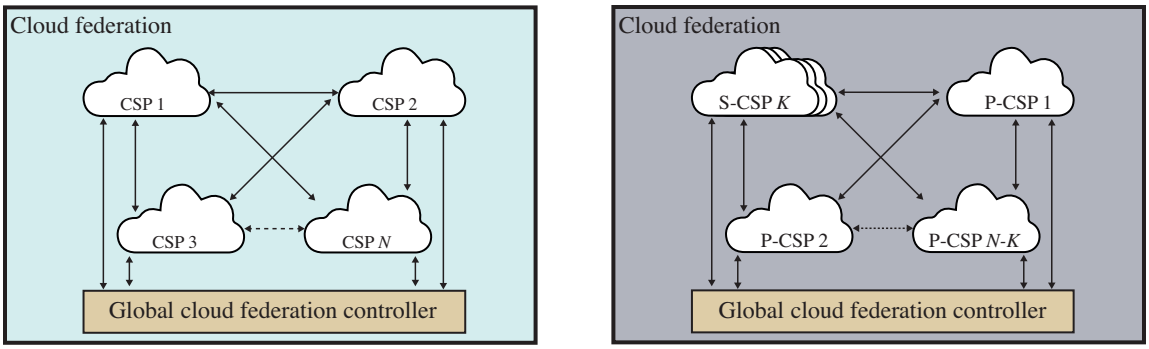

Fig. 1: Cloud services management model (a) generalized model comprising N number of federated cloud service providers (CSP) (b) classification of N CSPs into K secondary cloud service providers (S-CSPs) and $N-K$ primary cloud service providers (P-CSP) at an ith trading window.

federation environments. Considering a broker-based market, a strategy for resource requests outsourcing is proposed in [18] to maximize the social welfare of the cloud federation. In [19], a dynamic trading and scheduling algorithm is proposed for optimal management and allocation of jobs with different SLAs to a suitable cloud resource resulting in net profit maximization of each CSP. Within the context of cloud computing, a comprehensive study on auction mechanisms can be found in [20].

The majority of the existing work study the case where the total demand from overloaded providers is higher than the available resources for lease, thus game theoretic approaches and auction theory was implemented to solve a variety of optimization problems. However, the scenario when there is more resources for secondary access than the secondary demand, was not investigated. In addition, none of the aforementioned studies have modeled the required demand as a function of grade of service (GoS) target. To complement the well established auction models and the existing literature we adopt merchant mode to address how the secondary CSPs borrow the resources from the primary providers to achieve certain GoS considering lower demand than available resources. The main contributions of this paper are as follows:

- A novel purchase approach for dynamic resource management in a cloud federation scenario.
- A formulation of a finite horizon and non-linear optimization problem in merchant mode solved using stochastic modeling.
- A versatile approach to automation of cloud service provision within merchant mode.

Cloud services management model is described next. Section III describes the problem formulations. In Section IV, we present the analytical results. Finally, Section V summarizes our conclusions and future work.

II. Cloud services management model

A cloud services management model is shown in Fig. 1(a), which comprises of N number of federated CSPs; where each cloud provider offers services to a number of end-users. There are two basic classes of workload considered by the cloud providers which can be modeled by: (i) a delay system with arrival (λ^d) where we assume that there is a finite queue length (i.e., $0 < L_q < \infty$ where L_q is the queue length) and (ii) a loss system with arrival (λ^l) with $L_q = 0$, i.e., no buffer capacity. Assume the service rate is represented by μ^l and μ^d for the loss and delay system respectively. Suppose CSPs divide their resources into equal blocks r^l and r^d, where each block can be a computational resource, a storage resource etc. Considering the attainable performance thresholds, cloud providers within the federation are required to dynamically maintain suitable levels of certain parameters such as service denial probability and delay tolerance. However, provision of services by CSPs to the end-users in scenarios with high fluctuating workload is a challenging task. In order to improve the efficiency of cloud resources in such scenarios, CSPs may desire to share resources among multiple providers. In high demand periods, CSPs with limited resources, that are unable to meet users' demands, may borrow sequential (uninterrupted) resource blocks from other providers within the federation to maintain a particular grade of service. The CSPs which borrows resources is referred to as secondary cloud service providers (S-CSPs) while the CSPs which lease resources are referred to as primary cloud service providers (P-CSPs). In such setting, P-CSPs offer resource blocks to S-CSPs in return for monetary rewards. Typically, the price per resource block is expected to vary from one P-CSP to another depending on various factors such as the current workload, the geographical location of servers etc.

Each P-CSP announce information related to its available resources for lease and their respective prices at the start of ith trade window, where $i \in \{1, 2, \ldots, I\}$ and I denotes the total number of trading windows. Suppose $z_{i\hat{p}}$ denote the state of a $\hat{p}$th CSP at the beginning of an ith trading window, where $\hat{p} \in \{1, 2, \ldots, N\}$; $z_{i\hat{p}} = 0$ or 1 refers to being overloaded or underloaded, respectively. Let the announcement for sale of the $\hat{p}$th CSP's resources at the start of an ith trading window is represented by $a_{i\hat{p}}$, where $a_{i\hat{p}} = 1$ refers to the announcement that no resources are available for sale and $a_{i\hat{p}} = 0$ refers to the announcement of available resources for sale. Suppose t_i denote the state of an ith trading window, i.e. $t_i = 0$ means the trading window is deactivated and $t_i = 1$ represents an

activated trading window. The following condition is used to determine the state of an ith trading window

$$t_i = \begin{cases} 1, & \text{if } \left(1 - \prod_{\hat{p}=1}^{N} z_{i\hat{p}}\right)\left(1 - \prod_{\hat{p}=1}^{N} a_{i\hat{p}}\right) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In case of $t_i = 1$, CSPs are classified into P-CSPs and S-CSPs for an activated trading window, as shown in Fig. 1(b). The aforementioned process is repeated by the global cloud federation controller (GCFC) at each trading window. Our merchant market is modeled so that trading is allowed between the participants separately for each class of arrival in each of successive trade windows. Once a S-CSP borrows resources from a P-CSP, the S-CSP would use the purchased blocks for the entire duration of the trading window after which the S-CSP returns the resources to the P-CSP. Due to the availability of multiple CSPs in the federation, each offering resource blocks at different prices, the aim of the S-CSPs is to minimize the total operating expenditure. The incentive to minimize the operating expenditure is not only economical but also due to regulations which could be imposed by the federation arrangement. The federation controller may also consider to set a particular operating expenditure (OpEx) limit to prevent monopoly abuses. Therefore, the objective of our problem is to maximize profit under limited expenditure. In order to dynamically maintain the desired level of performance, we propose an autonomous cloud services management model that is able to make necessary borrowing from other CSPs in the federation, making sure that the profit of S-CSPs is maximized. The next section describes the problem formulation and presents the proposed optimized resource sharing scheme.

III. Problem formulation

For the efficient operation of the cloud federation, the key question for the CSPs is "how they should manage their resources dynamically?", based on time varying demand arising from their users while satisfying a desired grade of service. This is the central focus of this paper. In the previous section we have described the cloud federation which is comprised of $|N|$ ($N \geq 2$) CSPs, where overloaded service providers (with $\xi^c > \xi^d$ or $\epsilon^c > \epsilon^d$) borrow resource blocks from the underloaded providers (with $\xi^c \leq \xi^d$ or $\epsilon^c \leq \epsilon^d$) in the federation facilitated by the global cloud federation controller (GCFC). The GCFC tracks availability of the resources and receives requests from CSPs. As $|K|$ S-CSPs aim to borrow additional resource blocks from $|N-K|$ P-CSPs, the resource borrowing is performed with the objective to either achieve a desired service denial probability (ξ^d) or a desired delay tolerance (ϵ^d) in each trading window. The GCFC perform resource block management by borrowing resources from $|N-K|$ P-CSPs to maximize profit of the S-CSPs under restricted operating expenditure scenario (b) which maybe imposed by the federation. Table I lists the key notations used in the paper. The objectives of the GCFC is to maximize profit and to improve service quality of the cloud providers. At each trade window P-CSP broadcast their available resource blocks $\{\beta^a_{i_jk}\}$ and the corresponding cost per block $\{c_{i_jk}\}$ (where $c > 0$). The GCFC evaluates the expected profit $\{\pi_{i_jk}\}$ which could be made from each resource acquisition. However, to formulate the optimization problem, the we make the following assumptions.

TABLE I: Notation used

Notation	description
i	ith trade window
j	jth type of service
k	kth P-CSP
c_{i_jk}	Cost of resource block
y_{i_jk}	Number of resource blocks to be borrowed
$\xi^d_{i_j}$	Desired denial probability
$\beta^a_{i_jk}$	Number of resource blocks available P-CSPs
$\beta^r_{i_j}$	Number of resource blocks required to satisfy the desired denial probability
π_{i_jk}	Expected profit for borrowing resource blocks

Assumption. *Parameters which are assumed to be known to the GCFC at each trade window are: arrival rates (λ^l and λ^d), service rate (μ^l and μ^d), initial available resources ($r^l_{i_j}$ and $r^d_{i_j}$), borrowing cost c_{i_jk}, budget for borrowing b_{i_j}, available number of resources $\beta^a_{i_jk}$ and expected profit π_{i_jk}.*

Accordingly, profit maximization problem can be formally defined as follows:

$$(\mathcal{P}): \quad \max \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{(N-K)_j} \pi_{i_jk} \cdot y_{i_jk}$$

s.t.

$$(\mathrm{C}_{1A}): \ \arg\min_{y_{i_jk}\forall i,j,k} P\left(\lambda^l, \mu^l, \beta^r_{ijk} + r^l_{i_j}\right) \leq \xi^d_{i_j}, \ \forall i_j, k$$

for a denial system and

$$(\mathrm{C}_{1B}): \ \arg\min_{y_{i_jk}\forall i,j,k} W\left(\lambda^d, \mu^d, \beta^r_{ijk} + r^d_{i_j}, L_q\right) \leq \epsilon^d_{i_j}, \ \forall i_j, k$$

for a delay system.

$$(\mathrm{C}_2): \ y_{i_jk} \leq \beta^a_{i_jk}, \ \forall i_j, k$$

$$(\mathrm{C}_3): \ \sum_{k=1}^{K_{i_j}} y_{i_jk} \leq \beta^r_{i_j}, \ \forall i_j, k$$

$$(\mathrm{C}_4): \ \sum_{k=1}^{K_j} c_{i_jk} \cdot y_{i_jk} \leq b_{i_j}, \ \forall i_j, k,$$

π_{i_jk} in $(\mathcal{P})$ is comprised of the expected revenue R_{i_j} and cost c_{i_jk}, which can be obtained simply by

$$\pi_{i_jk} = R_{i_jk} - c_{i_jk}. \quad (2)$$

Although in some instances the cost of the resource block exceeds the revenue of a S-CSP and this yields a negative profit margin, however, for simplicity we consider $R_{i_j k} > c_{i_j k}$ to generate a positive profit for each resource acquisition. The inequality constraint (C4) implies that the GCFC optimizes profit to S-CSPs not only by taking into account the cost variations of the utility, but also by considering the budget constraint (b_{i_j}) in every trade window.

A. Optimal algorithm

The above non-linear optimization problem is solved by two steps as follows.

Step 1: Queueing analysis: The S-CSPs set their desired denial probability for each upcoming trade window (e.g., $\xi^d = 0.001$ for a denial system) (e.g., $\epsilon^d = 0.01$ sec. for a delay system). The required resources $\beta^r_{i_j}$ of each S-CSP is calculated according to the type of the arrival request to achieve the desired denial probability or the desired delay tolerance for ith trade window. For the zero delay tolerance arrival requests, the well-known denial probability formula [21] is used at the ith trade window of all S-CSPs. The denial probability formula can be written as

$$\xi(r_l) = \frac{\frac{1}{r^l!}\left(\frac{\lambda^l}{\mu^l}\right)^{r_l}}{\sum_{n=0}^{r^l}\frac{1}{n!}\left(\frac{\lambda^l}{\mu^l}\right)^{n}}. \tag{3}$$

In (3) we assume that at every trade window the capacity of a S-CSP belongs to one of the states $\{0, 1, \cdots, n\}$. r^l is the existing cloud computing capacity. Given the existing capacity, the total required resource blocks β_{i_j} to achieve the desired denial probability can be calculated by

$$\beta_{i_j} = g^{-1}\left(P\left(\lambda^l, \mu^l, r^l_{i_j}\right)\right), \tag{4}$$

where $g^{-1}(\cdot)$ the inverse function of $\xi(r^l)$ is used to derive the required capacity over the existing capacity. The problem is solved iteratively by performing a search over different values of β_{i_j} as shown in Algorithm 1. $\beta^r_{i_j}$ is then obtained by subtracting $r^l_{i_j}$ from β_{i_j}.

For the arrival requests with delay tolerance greater than zero, (5) is used for each S-CSP and at each trade window.

$$\epsilon(r^d) = \frac{1}{\lambda^d} \cdot L_q \tag{5}$$

where

$$L_q = \frac{1}{(r^d\mu^d - \lambda^d)^{r^d}} \cdot \frac{(r^d\lambda^d/\mu^d)^{r^d}}{r_d!(1-\lambda^d/\mu^d)} P_0 \tag{6}$$

with

$$P_0 = \left[\sum_{l=0}^{r^d-1}\frac{(r^d\lambda^d/\mu^d)^l}{l!} + \frac{(r^d\lambda^d/\mu^d)^{r^d}}{r_d!(1-\lambda^d/\mu^d)}\right]. \tag{7}$$

Given the existing capacity r^d, the total required resource blocks β_{i_j} to achieve the desired delay tolerance is given by

$$\beta_{i_j} = g^{-1}\left(W\left(\lambda^d, \mu^d, r^d_{i_j}, L_q\right)\right), \tag{8}$$

and the solution is provided iteratively by Algorithm 2.

In a nutshell, the computational approach used in Algorithm 1 and 2 is to evaluate the required resources (of one CSP) of a simulated system, introducing random variations of a specific probability distribution to the parameter values at each trade window.

Step 2: Optimization: Given the information obtained in the previous step, the borrowing selection performed by the GCFC are made subject to achieving the maximum profit to S-CSPs. The spending capacity of the S-CSP is constrained by budget b_{i_j}. If the acquired resources $\beta^a_{i_j k}$ are insufficient to meet the desired denial probability (desired delay tolerance) (i.e., $\beta^r_{i_j k} - \beta^a_{i_j k} > 0$), then the GCFC perform borrowing from the remaining resources from the set $\beta^a_{i_j k} \notin \{\beta^a_{i_j 1}, \beta^a_{i_j 2}, \ldots, \beta^a_{i_j J}\}$ for which the profit is maximum. If the desired performance is met, then the GCFC ends the process of borrowing new resource for the current trading window. Algorithm 3 summarizes the optimal solution.

B. Heuristic algorithm

In this section, the borrowing problem under budget constraint is solved by a random algorithm (Algorithm 4). Algo-

Algorithm 1: Computing required resources for denial system.

```
Initialization: let number of trading windows = I
for i ← 1 : I do
    for j ← 1 : length (ξ^d) do
        ξ^c = P (λ^l, μ^l, r^l_{i_j})
        β_{i_j} = r^l_{i_j}
        while ξ^c > ξ^d do
            ξ(w) = P (λ^l, μ^l, r^l_{i_j})
            β_{i_j} = β_{i_j} + 1
return Required resource blocks
```

Algorithm 2: Computing required resources for delay system.

```
Initialization: let number of trading windows = I
for i ← 1 : I do
    for j ← 1 : length (ε^d) do
        ε^c = W (λ^d, μ^d, r^d_{i_j}, L_q)
        β_{i_j} = r^d_{i_j}
        while ε^c > ε^d do
            ε^c = W (λ^d, μ^d, r^d_{i_j}, L_q)
            β_{i_j} = β_{i_j} + 1
return Required resource blocks
```

Algorithm 3: Optimal resource borrowing under limited budget.

b_{ij}, c_{i_jk}, $\beta^a_{i_jk}$ and $\beta^r_{i_jk}$ are obtained $\forall i, j, k$.
for $i \leftarrow 1 : I$ **do**
  for $k = 1 : (N - K)$ **do**
    Solve the nonlinear stochastic problem ($\mathcal{P}$) s.t. (C_{1A} or C_{1B}), (C_2), (C_3) and (C_4).
return Profit of the cloud service operator

rithm 4 performs borrowing randomly from the set $\{\beta^a_{i_jk}\}$, to satisfy the demand $\beta^r_{i_j}$. The constraints (C_2), (C_3) and (C_4) are satisfied by using

$$y_{i_jk} = \begin{cases} \beta^a_{i_jk}, & \beta^r_{i_j} \geq \beta^a_{i_jk},\ b_{i_j} \geq c_{i_jk} \\ \beta^r_{i_j}, & \beta^r_{i_j} < \beta^a_{i_jk},\ b_{i_j} \geq c_{i_jk} \\ 0, & b_{i_j} < c_{i_jk} \text{ or } \beta^r_{i_j} = 0. \end{cases} \tag{9}$$

Note that when $\sum \beta^a_{i_jk} \leq \beta^r_{i_j}$, the heuristic algorithm performs as well as the optimal algorithm. We also note that when $\sum_{k=1}^{J_{i_j}} \beta^a_{i_jk} > \beta^r_{i_j}$, the heuristic algorithm may still provide outcome in line with the optimal method in terms of achieved profit, however, this is a result of random variation with probability

$$\begin{cases} \dfrac{1}{J}, & \beta^a_{i_jk} \geq \beta^r_{i_j}, \forall i_j \\ \dfrac{1}{D}, & \sum\limits_{m} \{\bar{\beta^a}_{i_jlm},\ \forall l, m\} \geq \beta^r_{i_j}, \forall i_j \end{cases} \tag{10}$$

where $\{\bar{\beta^a}_{i_jlm},\ \forall l, m\} \subset \{\beta^a_{i_jk,\ \forall i_j, k}\}$ and $D = \left|\{\bar{\beta^a}_{i_j}..\}\right|$ is the number of subsets in the set $\{\bar{\beta^a}_{i_j}..\}$ which satisfy the demand. Of course, the probability is 1 if $\sum_{k=1}^{J_{i_j}} \beta^a_{i_jk} \leq \beta^r_{i_j}, \forall i_j$.

Once the problem is solved, either by the optimal or the heuristic algorithm, the new access denial probability can be calculated as

$$\xi^c = P\left(\lambda^l, \mu^l, \left(r^l_{i_j} + \sum_{k=1}^{K_{i_j}} y_{i_jk}\right)\right) \tag{11}$$

and the new delay is calculated by

$$\epsilon^c = W\left(\lambda^d, \mu^d, \left(r^d_{i_j} + \sum_{k=1}^{K_{i_j}} y_{i_jk}\right)\right). \tag{12}$$

To compute the required resource, the non-linear constraints, C_{1A} and C_{1B} are solved iteratively using Algorithm 1 and 2, respectively. The optimization problem is then solved with the remaining constraints by the revised simplex method given in Algorithm 3. Algorithm 3 is clearly a polynomial time ($O(n)$) while the heuristic algorithm (Algorithm 4) is quadratic ($O(n^2)$) with number of P-CSP ($N - K$) or exponential time ($O(2^n)$) with number of trading windows (I).

Algorithm 4: Heuristic resource borrowing under limited budget.

b_{ij}, c_{i_jk}, $\beta^a_{i_jk}$ and $\beta^r_{i_jk}$ are obtained $\forall i, j, k$.
for $i \leftarrow 1 : I$ **do**
  Set $y_{i_jk} \leftarrow \{\phi\}$, where $\{\phi\}$ is an empty set.
  Set counter $\leftarrow \sum_k y_{i_jk}$.
  Choose a random integer $n \in \{1, 2, \ldots, (N - K)\}$.
  for all *P-CSPs* $k = n : (N - K)$ and $1 : (n - 1)$ **do**
    if $(0 < \beta^a_{i_jk}) \leq (\beta^r_{i_j}$ - counter*)* & $(c_{i_jk} * \beta^a_{i_jk}) \leq b_{i_j}$ **then**
      $y_{i_jk} \leftarrow \beta^a_{i_jk}$.
      counter $\leftarrow$ counter $+ \sum y_{i_jk}$.
      $b_{i_j} \leftarrow b_{i_j} - \sum(y_{i_jk} * c_{i_jk})$.
    else if $(\beta^a_{i_jk} > 0)$ & $c_{i_jk} \leq (b_{i_j} -$ counter*)* & $(\beta^a_{i_jk} * c_{i_jk}) \geq b_{i_j}$ **then**
      $y_{i_jk} \leftarrow \left\lfloor \dfrac{b_{i_j}}{c_{i_jk}} \right\rfloor$ (where $\lfloor x \rfloor$ means floor of x.
      counter $\leftarrow$ counter $+ \sum y_{i_jk}$.
      $b_{i_j} \leftarrow b_{i_j} - \sum y_{i_jk}. * c_{i_jk}$.
    else if counter $\leq \beta^r_{i_j}$ & $\beta^a_{i_jk} > 0$ & $\beta^a_{i_jk} \geq (\beta^r_{i_j} -$ counter*)* & $(\beta^a_{i_jk} * c_{i_jk}) \leq b_{i_j}$ **then**
      $y_{i_jk} \leftarrow \beta^r_{i_j} -$ counter.
      counter $\leftarrow$ counter $+ \sum y_{i_jk}$.
      $b_{i_j} \leftarrow b_{i_j} - \sum y_{i_jk}. * c_{i_jk}$.
      break
    else if counter $\leq \beta^r_{i_j}$ & $\beta^a_{i_jk} > 0$ & $\beta^a_{i_jk} \geq (\beta^r_{i_j} -$ counter*)* & $(\beta^a_{i_jk} * c_{i_jk}) \geq b_{i_j}$ **then**
      $y_{i_jk} \leftarrow min\left\{\left\lfloor \dfrac{b_{i_j}}{c_{i_jk}} \right\rfloor\right\}$.
      counter $\leftarrow$ counter $+ \sum y_{i_jk}$.
      $b_{i_j} \leftarrow b_{i_j} - \sum y_{i_jk} * c_{i_jk}$.
    else
      $y_{i_jk} \leftarrow 0$.
return Profit of the cloud service operator

IV. Results and analysis

This section demonstrates the performance of the proposed resource sharing scheme for profit maximization. In order to build the simulation platform, a cloud federation comprising 8 CSPs is considered. CSPs are assumed to have heterogeneous resources and the total resource capacity of each CSP is modeled by unif(30,40); where unif() is a function denoting the uniform random distribution. Similarly, the workload of each CSP is generated as a uniformly distributed random variable, as in [10], [22]. In the simulations, the workload is modeled by unif(8,120). The arrival rate of users' requests for resource blocks is modeled in such a way that it reflects a high workload and, for a given desired service denial probability, at least one CSP within the federation is unable to meet the users' demands. Within a particular trading window,

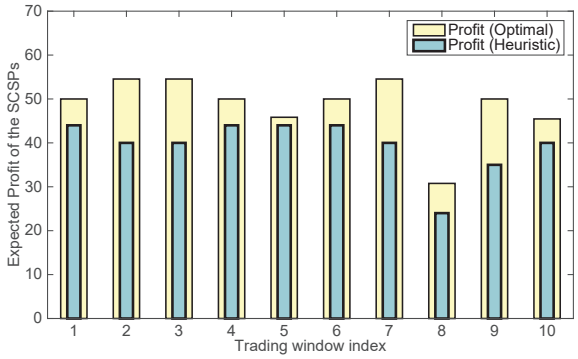

Fig. 2: A comparison of profit made with heuristic and optimal resource selection per trading window.

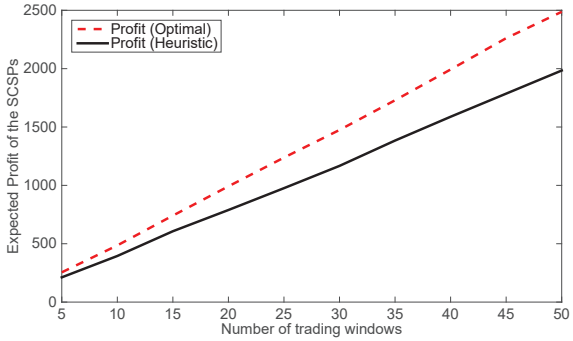

Fig. 3: A comparison of total profit made with heuristic and optimal resource selection for a number of trading windows.

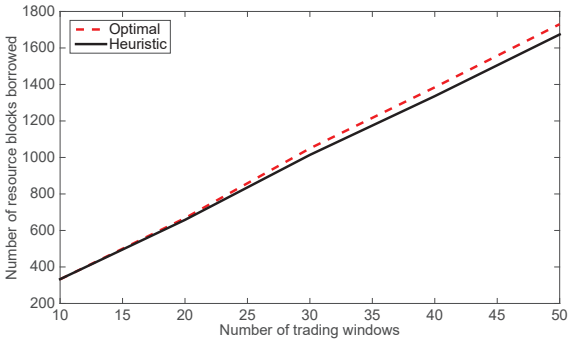

Fig. 4: A comparison of the total number of resource blocks borrowed with random and optimal resource selection for a number of trading windows.

CSPs are categorized into P-CSPs and S-CSPs depending on their workload. CSPs that are unable to meet their users' demands are classified as S-CSPs, whereas, CSPs with surplus resources are categorized into P-CSPs. Two types of systems are considered - denial system and delay system; which have been presented in Algorithm 1 and Algorithm 2 respectively to estimate the number of required resources for a S-CSP to meet its users' demands. In order to evaluate efficiency of the proposed optimal solution presented in Algorithm 3 for profit maximization under restricted budget, we compare the performance of the optimal solution with a heuristic model given in Algorithm 4. In the simulation model, the budget of each CSP is considered to be 50. Although, in the simulations we consider a fixed budget, the proposed model is capable of leading to an optimized solution for dynamically changing available budget of CSPs. Depending on the workload, once the CSPs are categorized into P-CSPs and S-CSPs, the latter are expected to lease resources from the former in order to provide services to the end-users with minimized blocking probability and maximized profit.

Considering the aforementioned simulation settings and 10 trading windows, Fig. 2 illustrates comparative performance analysis of the heuristic and optimal resource selection schemes within the context of the profit made per trading window. It can be observed from the results that the optimal resource allocation scheme results in profit maximization for every trading window as compared to its heuristic counterpart. Moreover, in order to estimate the overall performance gain, 50 trading windows are considered for setting up the simulations and, it is found that on average, the optimal approach leads to 25.35% profit gain per trading window compared to the heuristic solution.

In order to gauge performance of the proposed optimal scheme compared to the heuristic approach in terms of the total profit made for a number of trading windows, simulation results are shown in Fig. 3. Considering a range set of the number of trading windows, that spans from 5 to 50, it can be noticed from the results that at the 50th trading window, S-CSPs can achieve net profit gain of 43.5% by utilizing the proposed strategy for optimal resource allocation.

In the earlier discussion, the superior performance of the proposed optimal scheme over the heuristic approach has been demonstrated in terms of profit maximization. Nevertheless, further investigation is required to strengthen such analysis by taking into account the number of resource blocks borrowed by S-CSPs. Therefore, a comparison of the proposed optimal scheme with heuristic approach in terms of the number of resource blocks leased by S-CSPs from P-CSPs is shown in Fig. 4. It can be observed from the results that the optimal scheme leads to a higher number of borrowed resource blocks under the restricted budget and at the same time maximizes the overall profit. As an example, the optimal solution results in 3.35% additional resource allocations after 50 trading windows compared to the heuristic algorithm.

One of the primary aim of cloud federations is to facilitate resource sharing so that S-CSPs can guarantee minimum possible service denial probability. Therefore, the impact of desired service denial probability and available budget on the expected profit of the S-CSPs is required to be analyzed. Fig. 5 shows such analysis, where the desired service denial probability is considered as $0 \leq \xi^l \leq 0.8$ and available budget is modeled as $0 \leq b \leq 200$. It is found from the analysis that increasing the available budget or decreasing the desired service denial probability can result in a higher profit of the S-CSPs. The analysis presented in Fig. 5 demonstrates that

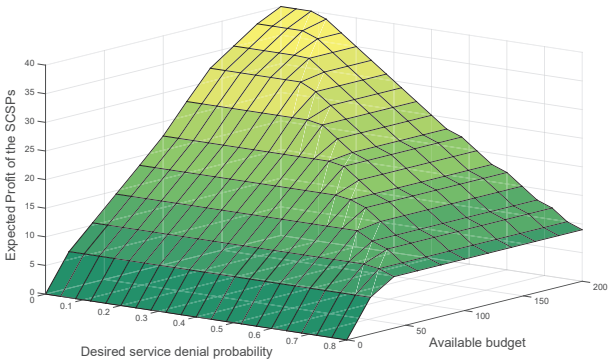

Fig. 5: Expected profit of the S-CSPs for resource borrowing at one trading window with desired service denial blocking probability as a function of budget.

the proposed formulation can assist the CSPs to efficiently choose suitable level of service denial probability and budget for expected revenue generation.

V. Conclusion

As networking has become major innovation driver for the Internet of Things (IoT), cloud computing has attracted a remarkable interest from research community not only because of its ability to lower costs and increase revenue, but also because it can improve existing computing services. In this context we have proposed a framework to model merchant market in a cloud federation scenario. The model categorizes and computes the required resource blocks to meet a certain GoS for cloud federation participants. We have focused primarily on delivering the CSP's demand by using an automated purchasing mechanism with the aim to maximize profit to overloaded service providers. To solve the problem we have developed two algorithms; an optimal and a heuristic solution. The optimal algorithm is evaluated against heuristic model. Our analytical results revealed that our approach leads to higher profit to CSPs, especially when the underutilized computational resources of the P-CSPs are higher than the demand from the S-CSPs. By adding the functionalities of types of demand and types of services provided by the CSP into our framework, we can conclude that our automation model is more versatile than those available in the literature. Although we have focused on maximizing profit, our framework could easily be modified to optimize other measures such as, cost and energy. Another contribution of this paper is that it provides a complementary approach to the well studied auction mode. As a future work, our model could be jointly implemented with auction mode to provide an all-round framework for automation of cloud federation market within 5G.

References

[1] N. Saxena, A. Roy, B. J. R. Sahu, and H. Kim, "Efficient IoT gateway over 5G wireless: A new design with prototype and implementation results," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 97–105, February 2017.

[2] Department for Culture, Media & Sport, "Next Generation Mobile Technologies: A 5G strategy for the UK," https://www.gov.uk/government/publications/next-generation-mobile-technologies-a-5g-strategy-for-the-uk, Policy paper, 2017.

[3] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, March 2016.

[4] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial internet of things (IIoT) - enabled framework for health monitoring," *Computer Networks*, vol. 101, pp. 192–202, 2016.

[5] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62–68, January 2017.

[6] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, April 2014.

[7] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, March 2017.

[8] A. Amjad, F. Rabby, S. Sadia, M. Patwary, and E. Benkhelifa, "Cognitive edge computing based resource allocation framework for Internet of Things," in *2nd International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2017, pp. 1–7.

[9] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2016, pp. 1–8.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[11] K. Chard and K. Bubendorfer, "Co-operative resource allocation: Building an open cloud market using shared infrastructure," *IEEE Transactions on Cloud Computing*, 2016.

[12] A. Celesti, M. Fazio, M. Giacobbe, A. Puliafito, and M. Villari, "Characterizing cloud federation in iot," in *30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, March 2016, pp. 93–98.

[13] M. Assis and L. Bittencourt, "A survey on cloud federation architectures: Identifying functional and non-functional properties," *Journal of Network and Computer Applications*, vol. 72, pp. 51–71, 2016.

[14] D. G. Kogias, M. G. Xevgenis, and C. Z. Patrikakis, "Cloud federation and the evolution of cloud computing," *Computer*, vol. 49, no. 11, pp. 96–99, Nov 2016.

[15] N. Samaan, "A novel economic sharing model in a federation of selfish cloud providers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 12–21, Jan 2014.

[16] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14–27, Jan 2015.

[17] M. M. Hassan, M. Abdullah-Al-Wadud, A. Almogren, B. Song, and A. Alamri, "Energy-aware resource and revenue management in federated cloud: A game-theoretic approach," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–11, 2015.

[18] L. Tang and H. Chen, "Double auction mechanism for request outsourcing in cloud federation," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, June 2015, pp. 1889–1894.

[19] H. Li, C. Wu, Z. Li, and F. C. M. Lau, "Virtual machine trading in a federation of clouds: Individual profit and social welfare maximization," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1827–1840, June 2016.

[20] "A systematic study of double auction mechanisms in cloud computing," *Journal of Systems and Software*, vol. 125, pp. 234–255, 2017.

[21] K. I. Aardal, S. P. Van Hoesel, A. M. Koster, C. Mannino, and A. Sassano, "Models and solution techniques for frequency assignment problems," *Annals of Operations Research*, vol. 153, no. 1, pp. 79–129, 2007.

[22] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 2008, pp. 243–264.