Dynamic Resource Admission Control and Scheduling Operation Optimization for 5G Cloud Radio Access Networks

Chiu-Han Hsiao[1] Frank Yeong-Sung Lin[1] Yen-Nun Huang[2] Po-Chuan Chien[2]

[1] [Department of Information Management，National Taiwan University，Taipei 10617，Taiwan（R.O.C.）.]

[2] [Research Center for Information Technology Innovation，Academia Sinica，Taipei 115，Taiwan（R.O.C.）.]

Abstract Cloud computing technologies are established by virtualization technologies to establish a connected resource processing pool in a 5G cloud radio access network（C-RAN）. This enables operators to reduce capital expenses. However，C-RAN operating expenses are typically ignored due to the complex challenges of using limited resources to deliver a satisfactory quality of experience（QoE）to users. Relevant issues of scalability and flexibility in resource management are considered in this dissertation. We considered an operator’s standpoint to focus on communication（network）and computation（system）perspectives； we analyzed the influenced factors，such as task call admission control，resource allocations，scheduling，and server operations in services computing for a sustainable network evolution. The problems were formulated as mathematical programming problems. Approaches based on dynamic programming，bin packing，and Lagrangian relaxations were proposed to determine the operating decisions within several practical strategies. The strategies were not only created to satisfy the QoE requirement of applications，but also to investigate operating servers within a cost-efficient resource pool. The computational experiment results revealed that the compositions of decisions with task admission，resource allocation，scheduling，and server operations were sufficiently supportive to allow operators make decisions efficiently and effectively to achieve near-optimal system revenue by leveraging cloud technology in a 5G C-RAN. The strategies can serve as valuable references or guidelines for the planning and operations of 5G C-RAN network service providers.

Keywords C-RAN；QoE；Task Call Admission Control；Resource Scheduling；Server Operations；Lagrangian Relaxation.

Introduction

A fifth-generation（5G）mobile communication system service will be launched in 2020. Cisco predicts that global IP traffic will triple from 59.9 exabytes（EB）per month in 2014 to 168 EB per month by 2019，as forecasted by the tenth annual Cisco Visual Networking Index Forecast

[1]. Several research projects have been developed based on a wireless service with a high capacity, high data rate, large amount of massive device connectivity, and high energy efficiency, but low end-to-end delay. Another research group proposed an overall system that could technically support the following ideas as improvements over today's networks [2]: （a）a 1000-fold increase in data volume per area, （b）a 10- to 100-fold increase in the number of connected devices, （c）a 10- to 100-fold increase in the typical user data rate, （d）a 10-fold extended battery life for low power massive machinetype communication（mMTC）devices, （e）a five-fold reduction in end-to-end（E2E）latency [3]. These requirements should be fulfilled with cost and energy efficiency similar to current cost and efficiency.

To provide a common connected platform for a variety of applications and requirements for 5G, some technology components should be reconsidered. The traditional role of base stations could be divided into two parts, remote radio heads（RRHs）and baseband units（BBUs）, which would be installed at the fronthaul and backhaul, respectively. At the fronthaul, new mmWave and new transmission and reception transmission technologies with massive multiple-input multiple-output（MIMO）antennae have been developed for multiple access control and radio resource management advanced by complex and multiple radio access technologies（RATs）and internode coordination schemes in heterogeneous networks（HetNets）such as 3G, 4G, 5G, sensors, and device-to-device（D2D）networks. At the backhaul, 5G radio access networks（RANs）have been proposed as a cloud architecture called the cloud radio access network（C-RAN）. The system architecture is illustrated in Figure 1.

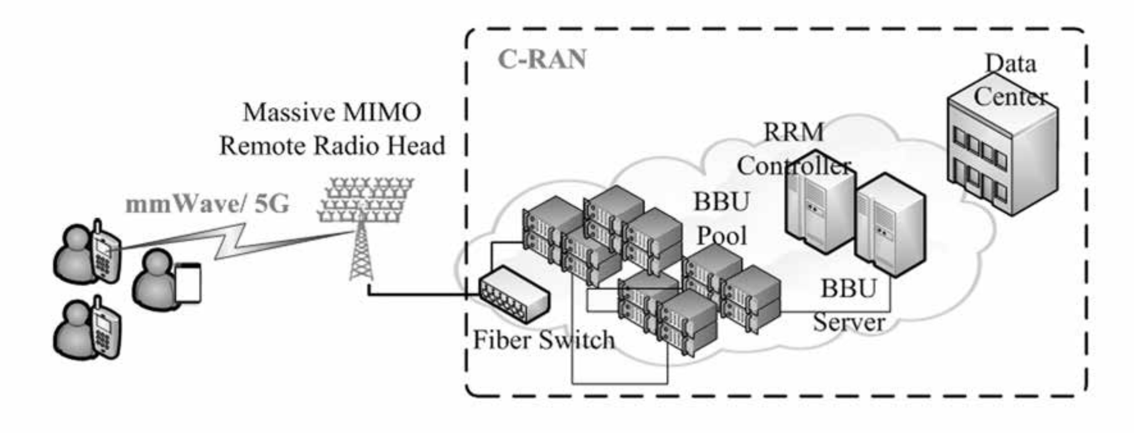


Figure 1 System architecture in 5G

The BBUs are clustered as a BBU pool and centralized to deal with the signal processing resources by being dynamically allocated to servers. Cloud computing technology can provide flexibility and scalability to satisfy user requirements, and the BBUs can share their computing resources in a pool. To provide large-scale processing and management capabilities，centralized BBU management allows the implementation of efficient radio resource management（RRM）algorithms，which have several advantages over traditional cellular architectures，such as increased resource utilization efficiency，and low energy consumption and light interference [4，5]. The issues of scalability and flexibility of RRM in C-RAN are the most pertinent concerns in relation to 5G [4，6，7].

The factors related to sustainable network evolution for new resource management mechanisms influence operations，such as call admission control，resource scheduling，and network operations，are subject to rapidly increasing data traffics，the limited budget of resource pool，and constraints. The cost function is conducted based on CAPEX and OPEX，which can be analyzed as a minimum problem subject to user or BBU allocation constraints [8]. The challenge is that operators must provide a good quality of experience（QoE）to users with affordable costs. Scalability and flexibility in resource management in C-RAN are critical.

In this paper，we focus on resource optimization management issues in operating stages to formulate problems as a mathematical programming form for the maximization of system revenue. A Lagrangian relaxation-based approach is proposed to determine the operating decisions within some designed practical scenarios.

The remainder of this paper is organized as follows：In Section 2，we present the literature review related to the current ideas and mechanisms for the emerging technologies in 5G. Section 3 gives the problem definitions of resource management in C-RAN and the problem is formulated as a mathematical form. The proposed solution approach contains dynamic programming，bin packing algorithms，Lagrangian Relaxation methods，and heuristics are developed to find the primal feasible solutions in Section 4. Section 5 presents various computational experiments and the results are correspondingly discussed and validated. Finally，the conclusions are drawn and the future work are described in Section 6.

Related Work

The major concerns of wireless communication networks inspired from the system architecture in 5G have changed； radio access networks have been adopted. Virtualization technology has been used for cloud computing in the IT field. Various challenges and research problems can be derived and discovered through considering new radio resources'（BBUs，servers in C-RAN，assignment of resources，etc.）compliance with unique requirements such as wireless network virtualization（WNV），software defined networking（SDN），and network function virtualization（NFV）in 5G. This can improve 5G's efficiency and effectiveness compared with existing technologies. From an operator's perspective，cloud computing service providers support share-based services in a pay-as-you-go format. This provides a flexible and lower CAPEX architecture to operators.

However，resource management is a key issue with complex on-demand traffic. New network paradigms may increase the OPEX and decrease the QoE for dissatisfied end-users，unless appropriate resource allocation mechanisms can be provided.

Resource Allocation and Scheduling

Resource allocation and scheduling algorithms are used in numerous research areas，such as transportation management，industrial management，operational research，and computer science，especially in real-time operating systems [9]. Several scheduling algorithms have been proposed in [10]. For example，earliest deadline first（EDF）is a dynamic scheduling algorithm used in real-time operating systems to allocate computing resources in CPUs as a priority queue. The queue is implemented to search for the task that is closest to its deadline； if a task cannot be finished within its deadline，the operating system must release the task. In the case of multiprocessor systems，a proportionally fair scheduling algorithm is a compromise-based scheduling algorithm. The main idea is to maximize the resource utilization for several competing interests while maintaining a load balance. It can be done by assigning a weight priority for each data flow that is inversely proportional to that flow’s anticipated resource consumption [11，12]. Ding et al. proposed a Linux scheduling policy with a priority queue instead of a FIFO queue to improve the performance of a Kernel-based Virtual Machine（KVM）[13]. Online VM placement algorithms to allocate resources to VMs with a cost-efficient way were proposed by Zhao et al. to increase cloud providers’ revenue in a managed server farm. First-Fit（FF），First-fit-migration（FFM），least reliable first（LRF），and decreased density greedy（DDG）algorithms are relevant packing strategies，as those methods can optimize tasks to achieve desirable performance metrics [14].

The growth of IoT traffic has seriously affected the performance of mobile networks [8]. With new mobile services，users of all ages want to share information widely through mobile broadband networks（MBNs）. The consequence is that the quality of experience（QoE）is not consistent； a user might have high QoE with a basic scenario，but QoE might degrade as that user encounters increasingly complex scenarios. The QoE of resource-intensive and latency-sensitive applications，such as interactive high-definition video streaming，online gaming，becomes degraded when users must employ 3G or 4G technologies. It is not sufficiently scalable and flexible to optimize individual scenarios with dynamic traffic loads [15]. To improve mobile users’ QoE，the coordinated allocation of computing resources for 5G wireless communications is a critical issue [16]. Zhai et al. proposed resource allocation algorithms to manage the utilization of servers under bursty and varying traffic [17].

In relation to workload，Ran and Wang proposed a cost-saving scheme for allocating resources. The key idea was to minimize the number of macro BSs to balance the consumption of resources. Minimal power consumption is a cost saving method in CAPEX and OPEX infrastructures [18]. Lu et al. proposed a dynamic resource allocation mechanism to improve the resource pool utilization and spectrum efficiency according to a Karnaugh map and a genetic algorithm [19].

Bin Packing Problems

The previous algorithms of the bin packing problem had been associated with table formatting，prepaging，packing of tracks on a disk，or stock cutting problems encountered in the industry [20，21]. Methods such as next fit，first fit，best fit，and worst fit were analyzed [22]. The problems were considered for finite numbers of tasks running on finite numbers of servers，which are subject to a limited capacity. Namely，the total number of tasks allocated to any server cannot exceed the server's capacity. The objective is to pack the maximum number of taskson the minimum number of servers. The next fit packing rule was proven to assign tasks to servers intuitively and sequentially [22]. In the first fit packing rule，the first server is monitored in relation to an assigned task if the task fits the residual server capacity. Best fit involves choosing the best server，which has the least remaining capacity，and assigning each task iteratively until all servers are full. A new server is switched on to serve the next task that arrives. For the worst fit，the remaining capacities of all servers are considered，and the next task is assigned to the server that possesses the maximum residual capacity. If the task does not fit in any server，then a new server is established. If the input list is in descending order，the algorithms have different behaviors regarding the packing rules. It is difficult to determine an optimal solution for the bin-packing problem due to it being NP-complete problem [23]. In the IT field，Jin et al. proposed a stochastic bin-packing algorithm to overcome resource allocation problems. The objective function was to minimize the number of required servers while satisfying the service-level agreement（SLA）availability guarantee [24]. In considering the resources or tasks that are migrated between hosts，most researchers analyze ideas in the computer science field and consider the practices of data centers. Multicore processor scheduling algorithms are broadly classified into partitioned scheduling，semipartitioned scheduling，and global scheduling [25].

Problems of Cost

Network service providers encounter underprovisioning or overprovisioning problems caused by time-varying demands and seasonal or periodical changes regarding on-demand services. Deployed infrastructure is not easy to change rapidly，due to the expansion cost related to the investment strategy of operators. Therefore，efficient and effective resource scheduling operating mechanisms are expected to conserve the operating costs and to meet the QoE requirements during the interim period [26]. If the resource demands are stochastic，an accurate forecast of the resource demand is difficult. Historical data usage is measured to estimate data demand in the future. Bobroff et al. have proposed a methodology for developing a framework to predict the probability distribution of demand values for multiple intervals ahead more precisely [27，28]. Beloglazov et al. suggested a decentralized architecture for resource management systems in data centers. One can consider network bandwidth and the temperatures of servers to develop policies； one can decide to reallocate resources to reduce data transfer overhead and network devices load [29]. If a system does not have servers that are homogeneous in terms of their CPU，memory，or storage，then，the types of demands can be dynamically simulated as arriving over time. Ghaderi proposed efficient and scalable scheduling algorithms to maximize the average number of users served by a system over time [30]. If the resource routing and placement problem is considered，the long-term-averaged performance can be determined using a Markov decision process，and a real traffic trace can be evaluated and jointly formulated as an optimization problem，as in [31].

Wang et al. considered a global fixed priority scheduling based on a problem window with a preemption threshold in adopting multicore processor scheduling； the optimization objective was minimizing system stack memory requirements [32]. Regarding energy consumption in clouds，minimizing the number of running servers and packing all virtual machines is the optimal solution. Chen et al. designed a practical algorithm in which an initialization of resource allocation consolidates with a spatial and temporal-awareness mechanism to detect the resource utilization patterns to maximize energy efficiency and reduce the overheads involved in migrations [33]. Centralized management is beneficial to control groups of resource pools to reduce operational costs and energy consumption. In relation to dynamic resource demand，a data center must efficiently centralize its management of decisions in a resource pool to optimize system utilities and energy consumption. A pertinent issue is the resource location being allowed to migrate between servers in a pool [34]. Zhang et al. analyzed methods for reducing the computation effort，which were denoted as virtual machine placement problems. They proposed heuristic offline and online algorithms to guarantee QoE and cost savings [35].

These concepts for IT，computer science，and data centers can also be applied to wireless communication. C-RAN enables the centralization of baseband resources，which was proposed as a pooling system on a general purpose processor （GPP）for long-term evolution（LTE）and worldwide interoperability in a microwave access （WiMAX）media access control（MAC） layer [36]. Guan et al. designed a task controller to dynamically assign resources for tasks processing [36]. Agata et al. proposed an algorithm that allows the core network to be designed as a ring topology to decrease the total cable installation cost through shortening the total cable “construction” length between the core network and RRHs [37]. Peng et al. suggested a threshold-based switching strategy to switch on or off some low-power nodes based on traffic loads to increase energy savings [38].

Mathematical Formulation

Software as a service（SaaS）providers must be responsive to users’ requests within an acceptable deadline determined in their service level agreement contracts，whereas infrastructure as a service（IaaS）providers are mostly concerned with infrastructural resources. The goal of SaaS providers is to maximize the revenue from SLAs and minimize the cost of using IaaS resources. IaaS providers want to maximize the utility of their own resources charged to their clients, and minimize the total cost of infrastructural resources at the same time. Each provider's decisions may influence the other providers' strategies[39].

In this section, we analyze an operator perspective, and integrate the roles of SaaS and IaaS providers to propose a centralized computing resource management mechanism in operating stages to optimize the total revenue through admission control, weighted task scheduling, and server operations. An abstract modeling and system framework are separately provided in Figure 2 and 3. At the fronthaul, the tasks are represented as the user computing requirements with values, demands of computing resources, processing, and waiting time requested from RRHs. At the backhaul, the servers have different levels of finite computing resources with costs. The BBUs are managed by resource allocation and server operation mechanisms in a centralized server pool in 5G C-RAN. For example, an application is required by RRHs to request several computing resources（CPU and memory）. If the operator administrator grants the user access to the network, the resources should be reserved as a task to be satisfied. In a C-RAN numerous servers have a number of CPU cores, with processing capabilities for each CPU core, and separate memory capacity.

This problem was formulated as a mathematical programming problem subject to several constraints. Different level of servers have costs related to different levels of computing resource capacities. Generally, all tasks are admitted as soon as possible to maximize the total revenues, but the available supply of resources and the demand for them are usually unbalanced. It is a trade-off to determine which tasks are selected with maximum values subject to assigning tasks to servers efficiently with limited capacity through well-designed operations. The fees of servers heavily depend on the instance type（dependent on CPU and RAM）used. The server cost is calculated by choosing a pay-as-you-go strategy. It is quite similar to Amazon Elastic Compute Cloud（Amazon EC2）for flexibility and scalability to scale up and down by traffic loads. The decision of switching on or off servers is significantly influenced by the cost. The dynamics of the RRHs traffic load is simulated and treated as a package of BBU demands called tasks with CPU processing power and memory requirements. The objective is the maximization of the total profits of tasks against the setup cost of servers by controlling decision variables appropriately. The first decision variable is jointly considered in three dimensions. The first is considering when tasks arrive which one of them should be assigned to which server, and in which time intervals. The second is which server should be turned on or off and in which time intervals. The third is determining how many servers should be switched on during all periods of time to calculate the setup cost.

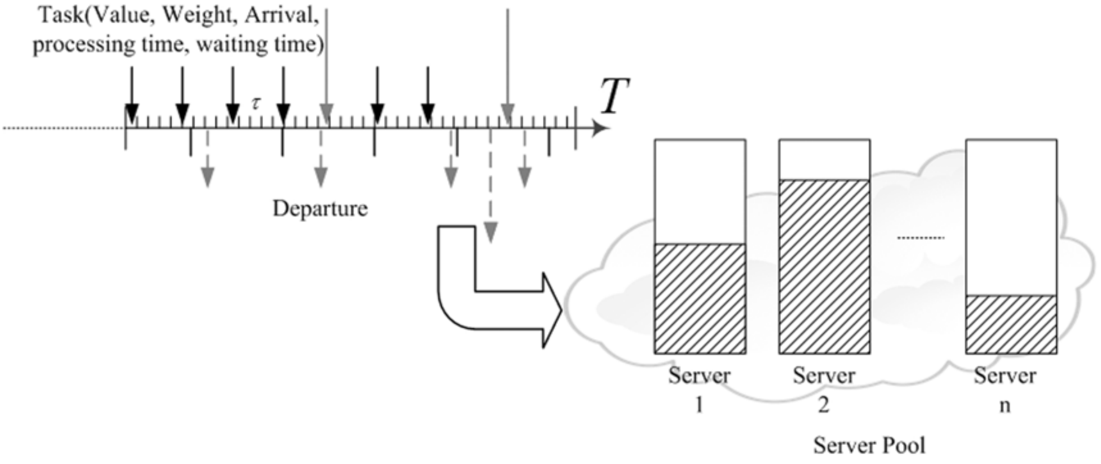


Figure 2 Abstract modeling

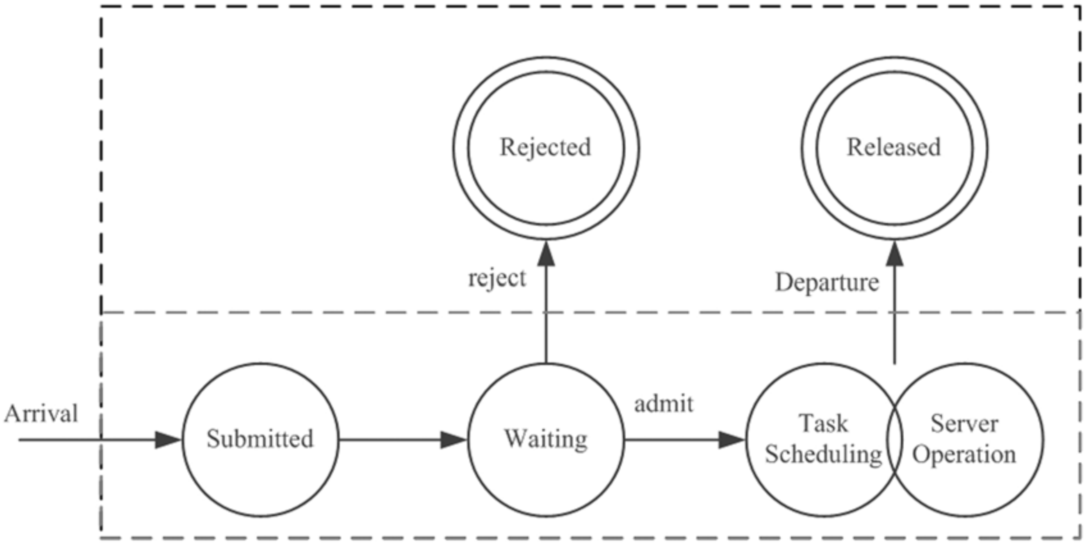


Figure 3 System framework

The constraints are the assignment task constraints where the CPU and memory requirements of each task are not separately divided into different servers. Capacity constraints are the assigned demands，which must not exceed the capacity of the servers. Server switching on or off constraints are related to the operation policy. For example，there should be at least one server operating in the system at any time interval. The number of servers switched on is set to one whereas no demands exist for task arrivals.

The key idea of our resource allocation scheme is to optimize the total profits minus the setup cost of operating servers in C-RAN by choosing decisions that depend on constraints being satisfied. The benefits can be estimated as follows： The maximum resource utilization will be achieved. The fairness of tasks can be achieved which means all tasks are served as soon as possible. The servers can be utilized with great precision，which means an appropriate number of servers are used. The scope and problem definitions are listed in Table 1. The mathematical model formulation was addressed and the given parameters and the decision variables are listed in Table 2 and 3，respectively.

Mathematical Model

BBUs are simulated as tasks to operate as VMs which are deployed on PMs related to the IT field. A limited number of VMs can be served simultaneously with capacity constraints [40]. Operators must have clear admission mechanisms for VM admission control which require a trade-off between assignments and limited resources. Admission control policies are defined according to different requirements such as system utilization，task call-blocking probabilities，or users' expectations in a cloud system. In this section，the call blocking-probability of a requirement is considered. We assumed that incoming requests arrive at the beginning of the admission control process. The inter arrival or departure followed exponential distributions. Operators must utilize appropriate admission control，resource allocation，server operation mechanisms to obtain the maximum system revenue subject to several constraints.

Table 1 Scope and problem definitions of IP

Model： Task scheduling and server operations	
Problem Assumptions	There are finite tasks and servers in our model Each task is simulated as a BBU requesting a package of CPU processing power and memory requirements Each server has a finite capacity of CPU processing power and memory for allocation to BBUs Each server has a setup cost，which has a fee when the server is switched on When an allocating server is switched off，the delay of migration of BBUs from the allocating server to others is ignored At least one server is serving or switched on for receiving BBUs
Given parameters	The set of servers has a capacity of CPU and memory The requirement of each task is generated randomly The setup cost of servers is generated by a fixed cost rate related to the CPU and memory capacity and by the extended setup cost related the number of times a server is re-switched on after initialization The maximum number of servers is predefined
Constraints	Assignment task constraints Server capacity constraints Server switching on or off constraints

Model： Task scheduling and server operations	
Objective	To maximize the total profits of assigned tasks minus the setup cost of servers by determining optimal resource allocation in C-RAN
To determine	Whether or not a task is allocated，and if so，to which server in which time interval Which server should be turned on or off，and in which time interval The total number of servers required to be switched on
Solution Approach	Lagrangian relaxation method

Table 2 Notations of the given parameters in IP

Notation	Description
S	The index set of physical servers in C-RAN，which is {1，2，3，⋯，s}
I	The index set of tasks for processing BBUs，which is {1，2，3，⋯，i}
T	The index set of decision intervals，which is {1，2，3，⋯，τ}
P_s	The number of CPU cores in a server $s\in S$
C_s	The processing capability（GHz）of each CPU core in a server $s\in S$
M_s	The RAM capacity in a server $s\in S$
D_i	Total amount of CPU processing capability rate（GHz）required by task $i\in I$
R_i	Total amount of RAM rate required by task $i\in I$
Vi	The reward rate of task $i\in I$，which is a function of the demands of CPU or memory.
γ_i	The processing time required by task $i\in I$，which is the mean processing time of long-term statistics（length of time per task）.
N_i	A penalty of task $i\in I$ if it is rejected when the requirement is not satisfied.
A_s	Set-up cost rate of server $s\in S$
E_s	Extended set-up cost of server $s\in S$
$\delta_{\tau i}$	Indicator function which is a binary parameter，1 means task $i\in I$ arrives at the time interval $\tau\in T$，0 otherwise.
ε	A tolerance of time delay for a task when it arrivals and obtains services completely.
O	A number of switching on server
t_τ	Time τ
K	Task call blocking requirement

Table 3 Notations of the decision variables in IP

Notation	Description
$a_{\tau is}$	1 if task $i \in I$ is assigned to server $s \in S$ in a time interval $\tau \in T$, and 0 otherwise
b_i	1 if task $i \in I$ is rejected, and 0 otherwise
$x_{\tau s}$	1 if server $s \in S$ is switched on in a time interval $\tau \in T$, and 0 otherwise
$y_{\tau s}$	The decision variable $y_{\tau s}$ is determined by $x_{\tau s} - x_{(\tau-1)s}$, which means $y_{\tau s}$ is set to 1 to mark the server s power-on at time $\tau \in T$ from the previous time $\tau - 1$ is set as power-off, otherwise set to 0
u_{is}	Artificial variable, 1 if task $i \in I$ is assigned to server $s \in S$, and 0 otherwise

To optimize the total revenue of the whole system in operation, the objective function Z_{IP} comprised of the values of maximum tasks that are assigned must subtract the setup cost of servers in the C-RAN. Each setup cost rate depends on the capacities. There are two types of cost; one is setup cost rate A_s; the other is extended setup cost E_s. A_s is counted when the server is switched on per unit of time, and the total setup cost is determined by the on-going time intervals like Amazon EC2. The other cost E_s, is the extended setup cost that is counted when the server is switched on or off one at a time after the server initialization, repeatedly.

The value of E_s was designed to be greater than A_s in the experiments. If tasks were not served, there was a penalty added in the objective function. All terms were composited into the objective function, the decisions are the admission control variable, $a_{\tau is}$, task call blocking variable, b_i, server switch on or off, $x_{\tau s}$, and extended switch on or off, $y_{\tau s}$, correspondingly. There is a trade-off of between the decisions during the network operations.

The optimal value is determined not only by the cost effectiveness of servers offered in services but also the maximization of the total values of tasks assigned efficiently.

Objective function:

$$Z_{IP} = max \sum_{\tau \in T} \sum_{s \in S} \sum_{i \in I} V_i a_{\tau is} - \sum_{i \in I} N_i b_i - \sum_{\tau \in T} \sum_{s \in S} A_s x_{\tau s} - \sum_{\tau \in T} \sum_{s \in S} E_s y_{\tau s} \qquad \text{(IP)}$$

Subject to the following constraints:

Assign task constraints: (1) For each task i in a time interval τ, and the assignment is that task i should be assigned into one of the servers, and 0 otherwise.

$$\sum_{s \in S} a_{\tau is} \leqslant 1 \qquad \forall i \in I, \tau \in T \qquad (1)$$

(2) The constraint is that task i is assigned into any one of the servers during all time intervals.

$$\sum_{s\in S}\left[\frac{\sum_{\tau\in T}a_{\tau is}}{T}\right]=1 \qquad \forall i\in I \qquad (2)$$

（3）The time slots of processing requested by task i is satisfied.

$$\sum_{s\in S}\sum_{\tau\in T}a_{\tau is}\leqslant\gamma_i \qquad \forall i\in I \qquad (3)$$

（4）$a_{\tau is}\leqslant x_{\tau s}$ If only task i is assigned to the server s，the server s must be switched on，that means both $a_{\tau is}$ and $x_{\tau s}$ are set to 1. If task i is not assigned to any servers，$a_{\tau is}$ is set to 0，and $x_{\tau s}$ could be 0 or 1 otherwise.

$$a_{\tau is}\leqslant x_{\tau s} \qquad \forall s\in S, i\in I, \tau\in T \qquad (4)$$

（5）and（6）defined the timing was assigned for task i in chronological order. It should be before the total requested time slots plus a delay tolerance，and after the timing of arrival.

$$\sum_{\tau'\in T}t_{\tau'}\delta_{\tau' i}\leqslant\sum_{s\in S}a_{\tau is}t\tau \qquad \forall i\in I, \tau\in T \qquad (5)$$

$$\sum_{s\in S}a_{\tau is}t_{\tau}\leqslant\sum_{\tau'\in T}t_{\tau'}\delta_{\tau' i}+\gamma_i+\varepsilon \qquad \forall i\in I \qquad (6)$$

（7）and（8）illustrated the blocking decisions are both set to 1，meaning the requested time slots of processing are not satisfied，0 otherwise.

$$\frac{\gamma_i-\sum_{\tau\in T}\sum_{s\in S}a_{\tau is}}{\gamma_i}\leqslant b_i \qquad \forall i\in I \qquad (7)$$

$$bi\leqslant\gamma_i-\sum_{\tau\in T}\sum_{s\in S}a_{\tau is} \qquad \forall i\in I \qquad (8)$$

（9）described the total rate of blocking tasks should not exceed the task blocking rate requirement.

$$\frac{\sum_{i\in I}b_i}{|I|}\leqslant K \qquad (9)$$

Capacity Constraints：（10）and（11）illustrated the capacity constraints would be obtained intuitively that the total demands of tasks are only aggregated and assigned in the server s，and should not exceed the capacity of the server s. In other words，if a new amount of traffic load（CPU or memory）arrives and is assigned in the server s，it should not be larger than either one of the remaining resources of the server s.

$$\sum_{i\in I}a_{\tau is}D_i\leqslant x_{\tau s}P_sC_s \qquad \forall s\in S, \tau\in T \qquad (10)$$

$$\sum_{i\in I}a_{\tau is}R_i\leqslant x_{\tau s}M_s \qquad \forall s\in S, \tau\in T \qquad (11)$$

Switching on or off constraints：（12）For each time interval τ，the status of the server should be at least the number of servers that should be switched on and served.

$$O \leqslant \sum_{s \in S} x_{\tau s} \qquad \forall \tau \in T \qquad (12)$$

（13）The decision variable $y_{\tau s}$ records whether a host is turned on at time τ，determined by $x_{s\tau} - x_{s(\tau-1)}$ when the host s is set to 1 for power-on at time τ and the previous time is set as power-off. Otherwise the host is set to 0，as formulated in（13）.

$$x_{s\tau} - x_{s(\tau-1)} \leqslant y_{\tau s} \qquad \forall s \in S, \tau \in T \qquad (13)$$

（14）For security reasons，the total number of times the host is switched on or off should not be exceed a boundary in time period τ. Each host s requires power-on only when it is powered-off in the previous time slot. Thus，the total number of times the host is switched on is not higher than half of the total time slots.

$$0 \leqslant \sum_{\tau \in T} y_{\tau s} \leqslant \frac{T}{2} \qquad \forall s \in S \qquad (14)$$

Lagrangian Relaxation-based Solution Processes

Relaxation

The solution procedure is based on the LR method. From analyzing（2），the decision variable $a_{\tau is}$ is confined by the ceiling function.（2）was not relaxed into the objective function for the decomposition of variables. To overcome this issue，the problem was reformatted by introducing a new binary decision variable，u_{is}.（2）was replaced by（15）and（16）. The following section provides descriptions of（15）and（16）.

$a_{\tau is} \leqslant u_{is}$ is marked whereas $a_{\tau is}$ is set to 1，u_{is} must be set to 1. The physical meaning is that task i is assigned into the server s at the time interval τ； the assignment is also marked.

$$a_{\tau is} \leqslant u_{is} \qquad \forall s \in S, i \in I, \tau \in T \qquad (15)$$

From a similar consideration of the task assignment constraints，due to the task job not being separable，$\sum_{s \in S} u_{is} \leqslant 1$ means that each task i should be assigned to only one of the servers.

$$\sum_{s \in S} u_{is} \leqslant 1 \qquad \forall i \in I \qquad (16)$$

Z_{IP} was reformatted into standard form through as a minimization problem. Some of the constraints，(4)，(7)，(8)，(10)，(11)，(13) and (15) were relaxed and multiplied by non-negative Lagrangian multipliers and added into the objective functions，respectively. The relaxed problem is called the LR problem in such a way that the corresponding Lagrangian multipliers，$\mu^1_{\tau is}$，$\mu^2_{\tau s}$，$\mu^3_{\tau s}$，$\mu^4_{\tau s}$，μ^5_{i}，μ^6_{i}，$\mu^7_{\tau is}$ and original decision variables. Based on the decision variables，five independent subproblems can be decomposed by the LR problem.

The decomposed subproblems can be adopted through parallel computing and optimally solved. Descriptions of the LR problem and subproblem formulations are provided below.

$$
\begin{aligned}
Z_{LR} = & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{（LR）}\\
& \min - \sum_{\tau\in T}\sum_{s\in S}\sum_{i\in I} V_i a_{\tau is} + \sum_{i\in I} N_i b_i + \sum_{\tau\in T}\sum_{s\in S} A_s x_{\tau s}\\
& + \sum_{\tau\in T}\sum_{s\in S} E_s y_{\tau s} + \sum_{\tau\in T}\sum_{s\in S}\sum_{i\in I} \mu^1_{\tau is}(a_{\tau is} - x_{\tau s})\\
& + \sum_{\tau\in T}\sum_{s\in S} \mu^2_{\tau s}(\sum_{i\in I} a_{\tau is} D_i - x_{\tau s} P_s C_s)\\
& + \sum_{\tau\in T}\sum_{s\in S} \mu^3_{\tau s}(\sum_{i\in I} a_{\tau is} R_i - x_{\tau s} M_s)\\
& + \sum_{\tau\in T}\sum_{s\in S} \mu^4_{\tau s}(x_{s\tau} - x_{s(\tau-1)} - y_{\tau s})\\
& + \sum_{i\in I} \mu^5_i (\frac{\gamma_i - \sum_{\tau\in T}\sum_{s\in S} a_{\tau is}}{\gamma_i} - b_i)\\
& + \sum_{i\in I} \mu^6_i (b_i - \gamma_i + \sum_{\tau\in T}\sum_{s\in S} a_{\tau is})\\
& + \sum_{\tau\in T}\sum_{i\in I}\sum_{s\in S} \mu^7_{\tau is}(a_{\tau is} - u_{is})
\end{aligned}
$$

Subject to constraints（1），（3），（5），（6），（9），（12），（14）and（16）.

Decomposition

Subproblem 1（related to $a_{\tau is}$）

Objective function：

$$
\min \sum_{\tau\in T}\sum_{s\in S}\sum_{i\in I}
\begin{pmatrix}
-V_i + \mu^1_{\tau is} + \mu^2_{\tau s} D_i + \mu^3_{\tau s} R_i \\
-\frac{\mu^5_i}{\gamma_i} + \mu^6_i + \mu^7_{\tau is}
\end{pmatrix} a_{\tau is} \qquad \text{（SUB-1）}
$$

Subject to constraints（1），（3），（5）and（6）.

The objective function of（SUB-1）can be divided into $|T||I||S|$ subproblems，which can be viewed as a minimum cost problem with the coefficients. If the assigned parameter $a_{\tau is}$ is set to 1，when the minimum coefficient is selected，then the optimal solution can be determined.

Running time is O（$|T||I||S|\log|T||I||S|$）. The pseudo code is illustrated as follows：

```
for each task i：
  for each server s：
    for each time t from arrival to deadline：
      calculate the coefficient C[ t ][ i ][ s ]
sort C
select the minimum elements from C
record the result of this server
find the server with the best value and
set decision variable a according to the result
```

Subproblem 2（related to $x_{\tau s}$）

Objective function：

$$\begin{cases} \min \sum_{\tau \in T} \sum_{s \in S} \left[A_s - \sum_{i \in I} \mu^1_{\tau is} - \mu^2_{\tau s} P_s C_s - \mu^3_{\tau s} M_s + \mu^4_{\tau s} - \mu^4_{(\tau+1)s}\right] x_{\tau s} & , \forall \tau < T \\ \min \sum_{\tau \in T} \sum_{s \in S} \left[A_s - \sum_{i \in I} \mu^1_{\tau is} - \mu^2_{\tau s} P_s C_s - \mu^3_{\tau s} M_s + \mu^4_{\tau s}\right] x_{\tau s} & \forall \tau = T \end{cases} \quad \text{(SUB-2)}$$

Subject to constraint（12）.

（SUB-2）can be divided into $|T||S|$ subproblems. The decision variable $x_{\tau s}$ is set to 1 if the coefficient is less than zero in each subproblem. Considering the feasible region of the problem，the constraint（12）is a policy of system design in *O*. There should be at least *O* servers switched on in every time slot τ. If there is no coefficient of SUB2 less than 0，the minimal objective value of SUB2 is determined by there being *O* numbers of $x_{\tau s}$ set to 1，and otherwise set to zero. The running time is O（$|T||S|$）. The pseudo code is illustrated as follows：

```
for each time t：
  for each server s：
    calculate coefficient C given t，s
    if C < 0：
      set x[ t ][ s ] to 1
    else：
      store C
if open too less：
  open servers with maximum C
```

Subproblem 3（*related to* $y_{\tau s}$）

Objective function：

$$\min \sum_{\tau \in T} \sum_{s \in S} (E_s - \mu_{\tau s}^4) y_{\tau s} \qquad \text{(SUB-3)}$$

Subject to constraint（14）.

（SUB-3）is analyzed as a combinatorial optimization problem，so-called a 0/1 knapsack problem. Given a set of items that are the decision variables $y_{\tau s}$，each with a weight of server（$\sum_{\tau \in T} y_{\tau s}$），and a value（$E_s - \mu_{\tau s}^4$），the number of each item to include in a collection is determined so that the total weight is less than or equal to a given limit（$\sum_{\tau \in T} y_{\tau s} \leqslant \frac{T}{2}$），and the total value（Objective value）is as large（minus minimum）as possible. Two solution approaches of SUB-3（dynamic programming and heuristic）are proposed and described as follows：

Dynamic Programming of SUB-3： a dynamic programming technique is traditionally used to solve the 0/1 knapsack problem. The first step is developed from a recurrence relationship derived by mathematical inductions. The second step is the implementation of recursions. The following statements relate to the recurrence relationship and implementation process（SUB-3-DP）.

Step 1： Recurrence Relationship：

$$\mathrm{OPT}[a][b] = 0 \qquad \forall a = 0, b \in TIME$$

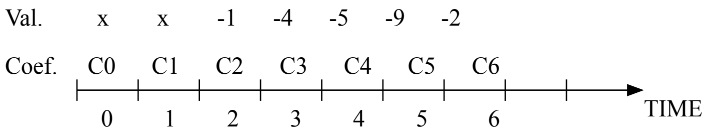
$$\mathrm{OPT}[a][b] = \infty \qquad \forall a > \frac{b}{2}$$

$$\mathrm{OPT}[a][b] = \min\{C_b + \mathrm{OPT}[a-2][b-1], \mathrm{OPT}[a-1][b]\} \qquad \text{,otherwise}$$

Step 2： Implementation of Recursions：

```
SUB-3-DP
for all TIME do
for all SERVER do
Calculate the coefficient of objective function（coef.）
for all SERVER do
Get a column of coef.
Set OPT[a][b]，a is # of take，b is total # of time
for T=0 and 1 do
Mark base
for T=2~TIME do
for Take=1~TIME/2 do
if $C_b$+OPT[a-2][b-1]<OPT[a-1][b] then
Take，Set OPT[a][b]= $C_b$+OPT[a-2][b-1]
else
  Not Take，set OPT[a][b]= OPT[a-1][b]
Find minimum element of OPT[a][b] and get the column and row
Set y[t][s]=1
break；
```

Example：



$$\mathrm{OPT}[a][b]=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ \infty & \infty & \infty & \infty & C_{24} & C_{25} & C_{26} \\ \infty & \infty & \infty & \infty & \infty & \infty & C_{36} \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

C_{12}=－1

C_{13}=min（－1，－4）=－4

C_{14}=min（－1，－4，－5）=－5

C_{15}=min（－1，－4，－5，－9）=－9

C_{13}=min（－1，－4，－5，－9，－2）=－9

C_{24}=min［－5+（－1），－4］=－6

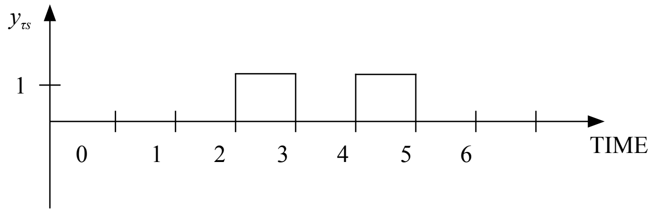
C_{25}=min［－9+（－4），－6］=－13

C_{26}=min［－2+（－9），－13］=－13

C_{36}=min（－2+（－6），∞）=－8

Find minimum element of OPT[a][b] is C_{26}. Mark $y_{3s}=y_{5s}=1$

Result is



Running time： O（T2）

Lemma： *for any number of takes in*（14），*the objective value is the minimum.*

$$\begin{array}{ll} \mathrm{OPT}[a][b]=0 & \forall a=0,b\in TIME \\ \mathrm{OPT}[a][b]=\infty & \forall a>\dfrac{b}{2} \\ \mathrm{OPT}[a][b]=\min\{C_b+\mathrm{OPT}[a-2][b-1],\mathrm{OPT}[a-1][b]\} & ,\text{otherwise} \end{array}$$

Proof：（recursion induction in a proof）

Assume that OPTtake（n）is the minimum value of objective function at the time n. The proof is by deducing which statements match the following：

If the time n is 0 and 1，OPTtake（0）

and OPTtake（1）are both zero. If the time $n > \frac{TIME}{2}$，OPTtake（n）are all set to infinity.

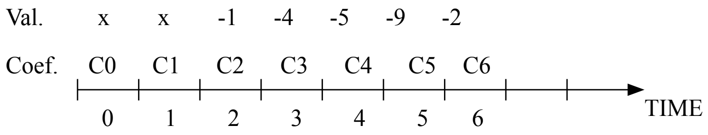
If any of TIME n during 2 to TIME，the OPTtake（n）is the minimum.

If the next time is $n+1$，the decision is not taken，and the optimal value of the objective value is the previous time n，OPTtake（n）. Otherwise，the decision is taken. The time is shifted to $n-2$，and the optimal value is determined by OPTtake（$n-2$）

Heuristic of subproblem 3： from a brief observation of constraint（14），if the decision variable $y_{\tau s}$ is set to 1 at time τ，the time $\tau-1$ and $\tau+1$ should be marked for $y_{(\tau-1)s}$ and $y_{(\tau+1)s}$ and both should be set to 0. The key is the coefficient calculations of the objective function of SUB-3. Choosing which one of $y_{\tau s}$ and setting to 1 on which server s at which time τ is applied through a concept of a sliding window. The following implementation process is described in the heuristic procedure.

```
Algorithm Heuristic
for all TIME do
for all SERVER do
Calculate the coefficient of objective function（coef.）
for all SERVER do
  for all TIME do
   if $C_t > 0$ then
    Mark[t]=−1
    Calculate $\theta_t = C_n - C_{n-1} - C_{n+1}$
  for T=2~TIME do
   if $\theta_t < \theta_{t+1}$ and $C_t < 0$
Mark[t]=1
Mark[t+1]=0
Find t with Mark[t]=1 and set to y[t][s]=1
break；
```

Example：



Steps 3–9：

$\theta_2 = -1 - (-4) = 3$ $\theta_2 > \theta_3$

$\theta_3 = -4 - (-1) - (-5) = 2$ $\theta_3 < \theta_4$

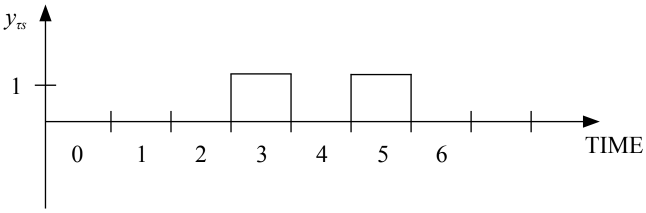
$\theta_4 = -5 - (-4) - (-9) = 8$ $\theta_4 > \theta_5$

$\theta_5 = -9 - (-5) - (-2) = -2$ $\theta_5 < \theta_6$

$\theta_6 = -2 - (-9) = 7$

Steps 10–15： Mark $y_{3s}=1$，$y_{2s}=y_{4s}=0$，and $y_{5s}=1$，and $y_{4s}=y_{6s}=0$

The result is



Running time： O（T）

Subproblem 4（*related to* b_i）

Objective function：

$$\min \sum_{i\in I}(N_i - \mu_i^5 + \mu_i^6)b_i \qquad \text{(SUB-4)}$$

Subject to constraint（9）.

（SUB-4）can be divided into $|I|$ subproblems. In each subproblem，the decision variable b_i is set to 1 whereas the coefficient is less than zero. From the feasible region，the minimal number of b_i that are set to 1 should be equal to $K|I|$ where not enough coefficients are less than 0； otherwise set b_i to zero. The running time is O（$|I|\log|I|$）. The pseudo code is illustrated as follows：

```
for each task i：
find coefficient C[i]
sort C in ascending order
if C[i] < 0，set b[i] to 1
if sum（b）>K|I|
set the exceed indices back to 0
```

Subproblem 5（*related to* u_{is}）

Objective function：

$$\min -\sum_{i\in I}\sum_{s\in S}\sum_{\tau\in T}\mu^7_{\tau is}\mu_{is} \qquad \text{(SUB-5)}$$

Subject to constraint（16）.

（SUB-5）can be divided into $|I||S|$ subproblems. The decision variable u_{is} is always set to 1 from the brief observation of the objective function. However，considering the feasible region，u_{is} is set to 1 with the maximum coefficient $\sum_{\tau\in T}\mu^{7}_{\tau is}$ corresponding to task i and the server s in each subproblem，otherwise u_{is} are set to zero. The running time is O（$|I||S|\log|I||S|$）. The pseudo code is illustrated as follows：

```
for each task $i$：
    for each server s：
        find coefficient C[$i$][$s$] and sort C[$i$][$s$] in descending order
        if find the max C[$i$][$s$]
            set u indices [$s$] to 1，return
        otherwise set to 0
```

Dual Problem and Subgradient Method

According to the weak Lagrangian duality theorem [41]，for the multiples，$\mu^{1}_{\tau is}$，$\mu^{2}_{\tau s}$，$\mu^{3}_{\tau s}$，$\mu^{4}_{\tau s}$，μ^{5}_{i}，μ^{6}_{i}，$\mu^{7}_{\tau is}\geqslant 0$，the objective value of the Lagrangian relaxation problem Z_{LR} is a lower bound of the primal problem，Z_{IP}. Based on the LR problem，the following dual problem Z_D was constructed to calculate the tightest lower bound.

Dual problem（D）：

Objective function：

$$Z_D=\max Z_{LR} \tag{D}$$

subject to $\mu^{1}_{\tau is}$，$\mu^{2}_{\tau s}$，$\mu^{3}_{\tau s}$，$\mu^{4}_{\tau s}$，μ^{5}_{i}，μ^{6}_{i}，$\mu^{7}_{\tau is}\geqslant 0$

There are several methods to solve this dual problem（D）. Among them is the most popular method，the subgradient method proposed in [42，43]. First，let the vector S be a subgradient of Z_D. In iteration k of the subgradient optimization procedure，the multiplier vector π^k=（$\mu^{1}_{\tau is}$，$\mu^{2}_{\tau s}$，$\mu^{3}_{\tau s}$，$\mu^{4}_{\tau s}$，μ^{5}_{i}，μ^{6}_{i}，$\mu^{7}_{\tau is}$）is updated by $\pi^{k+1}=\pi^k+t^kS^k$. The step size t^k is determined by $t^k=\lambda\frac{(Z^h_{IP}-Z_D(\pi^k))}{\|S^k\|^2}.Z^h_{IP}$ is the primal objective value（an upper bound on Z_{IP}）and λ is constant where $0\leqslant\lambda\leqslant 2$.

Getting Primal Feasible Solutions

By applying the Lagrangian relaxation method and subgradient method to solve the subproblems，we can not only determine a theoretically lower bound from the primal feasible solution，but we also identified some helpful hints in relation to the primal feasible solution that are iterated when solving the dual problem. The feasible region of a mathematical programming problem defined by the solutions must be satisfied by all constraints. A set of primal feasible solutions to Z_{IP} is a subset of the solutions to Z_{LR}. Regarding obtaining the primal feasible solutions，several alternative methods based on observations can be used to modify the solutions of Z_{LR} into the feasible region. The process is known as getting primal feasible solutions. The heuristic approach is proposed to find feasible solutions in next sections. To obtain the primal feasible solutions for（IP），the first step is to consider the solutions to the LR. Two major decision variables，$a_{\tau is}$ and $x_{\tau s}$ are taken into consideration.

For the purpose of evaluating our solution of quality，three algorithms，TABLE（task allocation by LR evaluation），ROTATE（reassignment of task aim to equilibrium），and TOWEL（turn off when evaluated lossy）are simultaneously implemented for result comparison. Block diagram is illustrated in Figure 4.

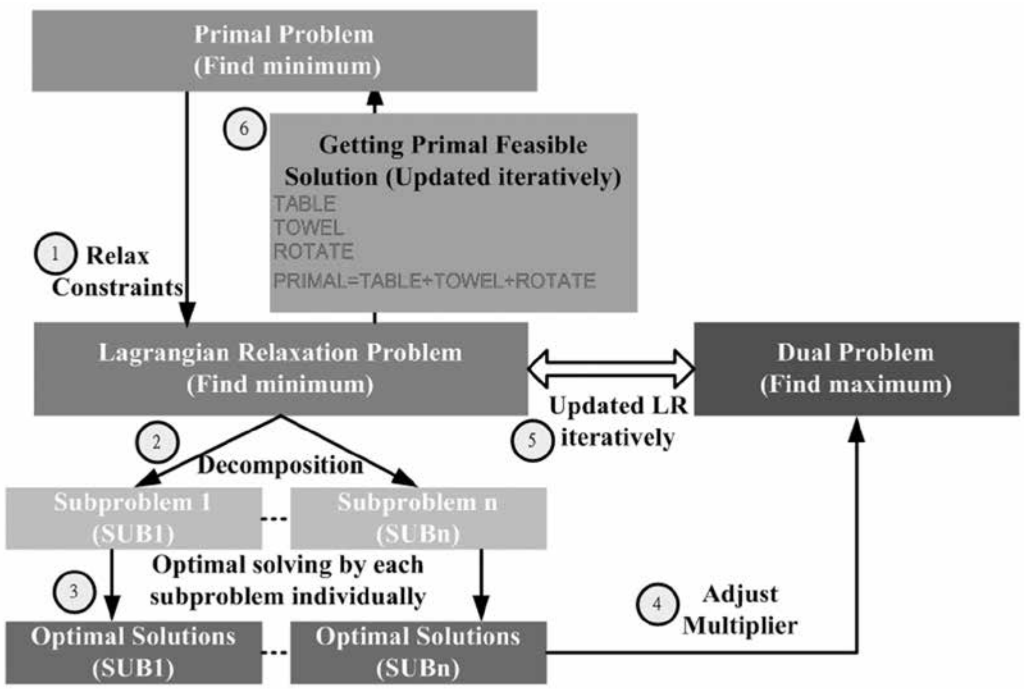


Figure 4　Lagrangian Relaxation-based Solution Processes

Task Assignment and Scheduling

The heuristic is proposed by the values of Lagrangian multipliers determined from the dual problem. The coefficient of the subproblems is the arc weight of importance with respect to the decision variable[43]. Through subproblems，the weighted factor of tasks to servers are represented from（SUB-1），in the form of $\sigma_{\tau is}=\mu^{1}{}_{\tau is}+\mu^{2}{}_{\tau s}D_i+\mu^{3}_{\tau s}R_i-\frac{\mu_i^5}{\gamma_i}+\mu_i^6+\mu_{\tau is}^7$； therefore，the sum of values can be used as an index to interpret the importance of task i to server s at the time τ. The algorithm is called TABLE（Task Allocation by LR Evaluation）shown in（17）.

$$Indicator_i=\frac{V_i\times\sum_{\tau\in T}\sum_{s\in S}\sigma_{\tau is}}{P_sC_s\times R_s\times b^{\beta}}\qquad\forall i\in I\qquad(17)$$

$b$is a residual buffer time of the task i. β is a power value for the evaluation of time which is set to 2 in the experiments. The set of assigning and scheduling decision variables was determined from the corresponding multipliers. For example：$a_{\tau is}\leqslant x_{\tau s}$，（4）indicates that if only task i is assigned to the server s，the server s must be switched on，and bothand are set to 1. If task

i is not assigned to any server，$a_{\tau is}$ is set to 0 and $x_{\tau s}$ could be 0 or 1.A brief illustration is shown in Figure 5，$\mu^1_{\tau is}$ is determined in the updated cases to objective value of dual problem. It interprets a partial weighted objective value to the objective value. TABLE dynamically can make a task selection criterion related the value of multipliers to approaching the optimal objective value.

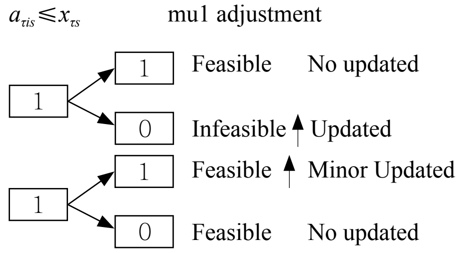


Figure 5　Feasibility Check for Constraint

Server Operations

Regarding the server operation，a utilization factor of revenue was measured. The utilization factor was designed based on the ratio of allocated task value over the server cost at each time τ shown in（18）. The concept was derived from capacity constraints（10）and（11）. The measurement is an index where if the value is greater than 1，the profit can cover the costs，and the server has a positive effect and is turned on，otherwise it is turned off. The algorithm is abbreviated as TOWEL（turn off when a loss is evaluated）.

$$U_{\tau s}=\frac{\sum_{i\in I} V_i a_{\tau is}}{A_s},\qquad \forall s\in S,\tau\in T,x_{\tau s}=1. \tag{18}$$

Time Stamping for Task Reassignments

Facing on the variety of traffic load，the burst traffic occasionally leads to a system resource imbalance with a hard deadline or tight delay tolerance. To overcome this issue，a procedure known as ROTATE（reassignment of task aim to equilibrium）is proposed. The basic idea is moderation of the burst-to-light traffic load with a reasonable delay tolerance. The major function of system utilization is a balanced approach that removes and reassigns some tasks from servers with heavy loads to servers with light loads within an acceptable delay tolerance.

Computational Experiments

Environment

In this section，the experiment environment is initialized according to problems related to C-RAN resource allocation，and a scheduling model is proposed in which tasks represented as BBUs request computing requirements and a resource pool with servers is set up for packing tasks. Algorithms constructed and implemented to analyze the heuristic quality were developed，and performance evaluationswere conducted for several simulation cases. Our experiments were developed using the C++ programming language and implemented in a virtual machine as a platform with a quad-core processor，8 GB of random access memory（RAM），and Ubuntu version 14.04 as the operating system. The given amounts of traffic loads of tasks and arrival time intervals were randomly generated in Table 4 which shows the values of the experimental parameters.

Performance Evaluation Cases

In our experiments，the solution of the dual problem was defined as LR. The initial solutions were determined by examining previous related works and improving them by sorting the weights of items in descending order，denoted as first-fit（FF）and best-fit descending（BF_Descending），respectively[22]. Our methods for obtaining solutions were derived through proposed algorithms such as TABLE，TOWEL，and ROTATE，referred to as selected algorithms（SAs）. Two performance metrics are used to evaluate the solution quality，gap and improvement ratio. The gap and improvement ratios are respectively calculated as follows：

$$GAP=\frac{|SA_i-LR|}{|LR|}\times 100\% \text{ and } IR=\frac{|SA_i-SA_j|}{|SA_j|}\times 100\%.$$

Subsequently，several scenarios were designed for performance evaluations from different perspectives to analyze the decisions that influenced the objective function.

Table 4　Parameters for Computational Experiments

Parameter	Value
Time interval（τ）	3–80
Number of tasks（I）	100–800
Number of hosts（S）	3–8
Host CPU capacity（P_sC_s）	250
Host memory capacity（M_s）	250
Host setup cost rate（A_s）	100
Reopen cost of a host（E_s）	150
Value of each task（V_i）	Random（rand（）%value+1）
CPU requests of a task（D_i）	Random（rand（）%cpu+1）
Memory requests of a task（R_i）	Random（rand（）%memory+1）
A task $i\in I$ arrives at time $\tau\in T$（δ_i）	Random（rand（）%time+1）

Uniform Traffic Load

This experiment was designed to analyze task arrivals with a uniform distribution，which can be interpreted as normal traffic for daily cases. First，we examined the trend of the objective function with the number of tasks as the control variable. The result in Figure 5 indicate that the objective value increased with the number of tasks. However，the curves for FF，ROTATE，and TOWEL decreased when the number of tasks was more than 500，indicating that when the number of arriving tasks exceeded the system capacity，the tasks could not be handled.

The dropping penalty reduced the objective value. The PRIMAL and TABLE methods exhibited increasing trends and moderated the dropping penalties through superior resource allocation and scheduling strategies in which tasks with high values were selected.

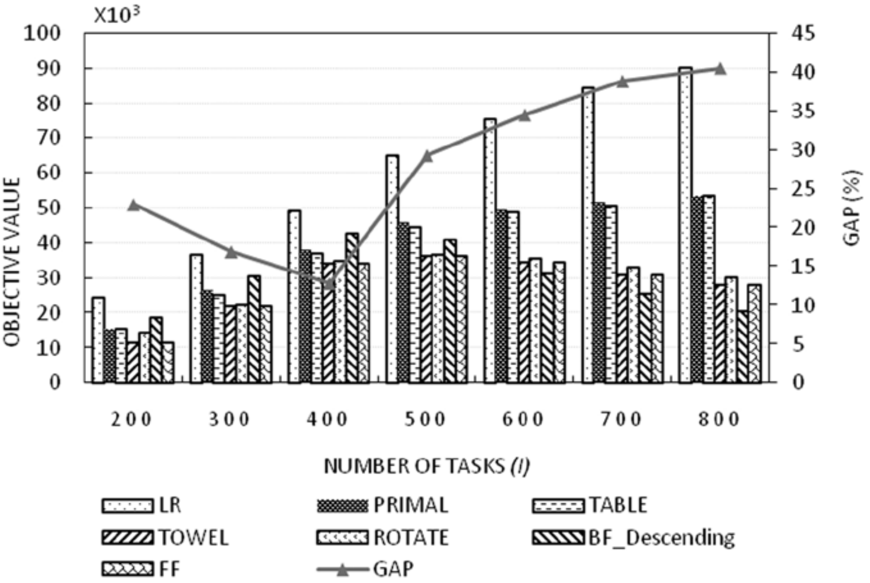


Figure 5 Evaluation of uniform traffic load vs. objective value

Bursty Traffic Load

A bursty pattern is evident when numerous tasks arrive in a short period of time. This scenario was used to test how bursty arrival affected the objective value. Figure 6 shows the results. PRIMAL and TABLE were found to be superior to FF，TOWEL，and ROTATE. This is because when many tasks arrived in a short period of time，PRIMAL and TABLE had sufficient buffer time to reassign tasks，and tasks with high values were selected to be served within the finite server capacity. FF，TOWEL，and ROTATE showed characteristics similar to those seen in Figure 5. In cases where tasks were blocked，leading to a penalty，the total revenue was reduced to lower the objective value.

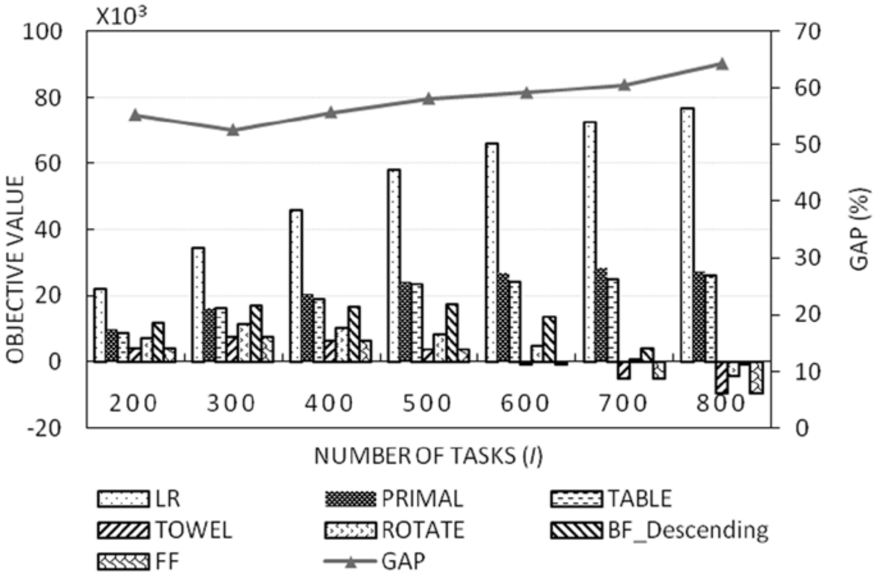


Figure 6 Evaluation of bursty traffic load vs. objective value

Server Deployment

The cost function was considered to be CAPEX and OPEX. CAPEX relates to network infrastructure deployment of appropriate levels of servers，in which the budget was one of the major constraints. The cost difference between high- and low-end servers is usually considerably high. OPEX is related to the cost of network operations and management. Operators generally conduct a comprehensive system analysis to appropriately manage，control，or operate the system（by methods such as turning servers on or off）to provide substantial QoE to users efficiently and effectively. Both factors were emulated in this experiment，and the servers purchased were deployed in three levels with different capacities；this means that there were nonhomogeneous servers deployed under the same limited budget. The rapid increases in data traffic were represented as bursty traffic loads to determine which method delivered superior QoE for users with affordable costs and preserved revenue.

Experiments with different numbers of servers and capacity levels were designed to examine the effects of CAPEX and OPEX. The followings are the experimental scenarios tested.

Large： The system capacity is much higher than the total task demand. The physical meaning is that the CAPEX budget is unlimited，and the pool always has sufficient resource space in the servers. The blocking probability of tasks is almost zero irrespective of the operation method selected.

Medium： The CAPEX budget is limited such that the system capacity is close to the total task demand. The levels of servers are nonhomogeneous，implying that the different levels of servers that are allowed to be deployed tend to approach the limited budget under the maximum server capacity. Bursty task traffic loads may be blocked owing to insufficient resource space when a poor operation method is selected.

Small： The server capacities are much smaller than in the two aforementioned scenarios. All servers face a lack of resources in every experimental time slot. The operating methods become critical in that the bursty traffic loads arrive and decisions must be made to minimize the call-blocking rate of tasks and the resource cost to servers. A bottleneck occurs owing to insufficient resource space in servers，as all servers must generally be switched on all the time to serve tasks.

Table 5　Experimental Results in Case of Server Deployment

Server Capacity	Number of Servers	FF	BF_ Descending	TABLE	TOWEL	ROTATE	PRIMAL	LR	GAP（%）	IR（%）
Large	2	9.62	17.23	17.75	9.62	12.67	18.75	42.20	55.56	8.85
	4	13.89	20.50	20.64	13.89	16.98	21.54	46.23	53.40	5.10
	6	13.95	24.04	22.01	13.95	18.84	22.37	47.47	49.35	0.00
	8	15.77	26.25	22.20	15.77	18.36	22.61	47.71	44.97	0.00
	10	15.30	25.46	22.57	15.30	18.78	23.55	47.95	46.90	0.00

Server Capacity	Number of Servers	FF	BF_ Descending	TABLE	TOWEL	ROTATE	PRIMAL	LR	GAP（%）	IR（%）
Medium	2	1.49	9.55	13.70	1.49	2.37	14.87	40.87	63.62	55.70
	4	2.03	8.78	15.26	2.03	5.98	16.65	42.99	61.26	89.64
	6	4.62	9.48	16.31	4.62	6.83	17.75	43.84	59.51	87.20
	8	4.42	8.50	16.83	4.42	7.00	17.58	43.75	59.82	106.87
	10	4.44	11.46	15.93	4.44	6.69	18.34	43.85	58.18	60.01
Small	2	−7.47	−5.47	8.62	−7.47	−5.52	8.78	28.72	69.45	260.47
	4	−5.46	−4.01	8.52	−5.46	−4.13	10.37	29.59	64.96	358.55
	6	−4.76	−3.68	8.56	−4.76	−2.88	10.56	29.77	64.53	386.59
	8	−3.77	3.99	8.90	−3.77	−1.15	11.14	30.00	62.86	178.99
	10	−4.12	−1.31	7.76	−4.12	0.28	10.47	29.72	64.76	899.47

Table 5 shows the results of these operating methods（LR，GAP，and IRvalues）. Considering GAP，all values in each scenario were calculated to determine the maximum value of the difference between the methods（TABLE，TOWEL，ROTATE，and PRIMAL）and LR. IR，which is represented by the ratio of improvement of feasible solutions，is also calculated as the maximum value difference of methods between FF and BF_Descending

The results revealed that PRIMAL always has the best objective value among all methods in every scenario except in the case of numerous servers（6–10）. When few servers were deployed，IR values were much higher than in the other two scenarios. The physical meaning is that small servers have flexible operating methods to control OPEX and CAPEX with limited budgets. The GAP values are approximately 40%–70%，implying that the solution quality of our heuristics must be improved to make it optimal.

Evaluation of Processing Time

In general，the processing time is an implicit task parameter. There is no way to determine the specific time slots for the processing time of a requested task. However，the solution is usually emulated as a distribution for statistical analysis when the processing time becomes relatively large； proper resource allocation is difficult for such tasks. The occupied time slots for tasks require more system resources and decrease the probability that other tasks can be served with less residual resources. A short processing time means that the task only stays in the system for a short period of time. In this case，the request processing time slots follow a uniform distribution in an interval. For example，64 indicates that the processing time for task *i* varied from 1 to 64 and was distributeduniformly.

Figure 7 shows the results. All objective values increased with the processing time of tasks. The curves of all methods exhibited the same increasing trends，and the objective value was generated by multiplying the task value by the processing time slot. The first observation from the results is reasonable. No significant difference exists between the operating methods when the processing time is relatively short. For the processing time，higher objective values than those of ROTATE and TOWEL should be considered. TABLE and PRIMAL select tasks with high values and assign them to the system.

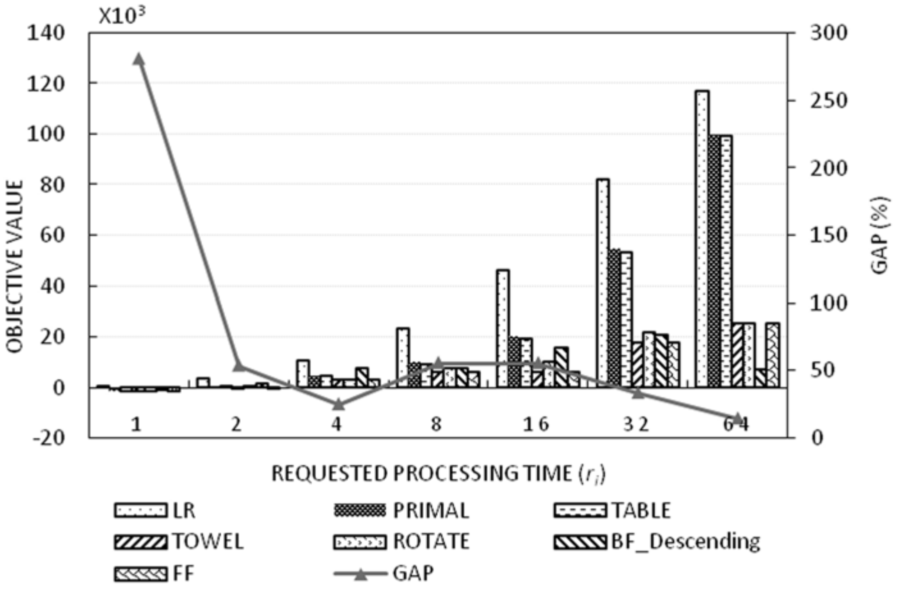


Figure 7　Evaluation of processing time vs. objective value

Evaluation of Allowed Waiting（Buffer）Time

The allowed waiting time for each task is referred to as delay tolerance. The delay tolerance length is related to the type of applications requested by tasks，such as web browsing，video streaming，or voice calls. The correlation between the waiting time and the objective value was considered in this case. Generally，a short waiting time for a task means it has less flexibility for task assignment or rescheduling. It also means that the blocking probability will be higher than for tasks with long delay tolerance.

Figure 8 shows that different waiting times are designed for the tolerance lengths of tasks. Interestingly，the objective value monotonically increased for a buffer time of 1–20.

The objective value increased and then remained constant into the saturation region for a buffer time of 40–80. The permitted buffer time of 40 serves as a threshold for operators to set a QoE metric for the service level agreement of user requests with the maximum acceptable delay tolerance.

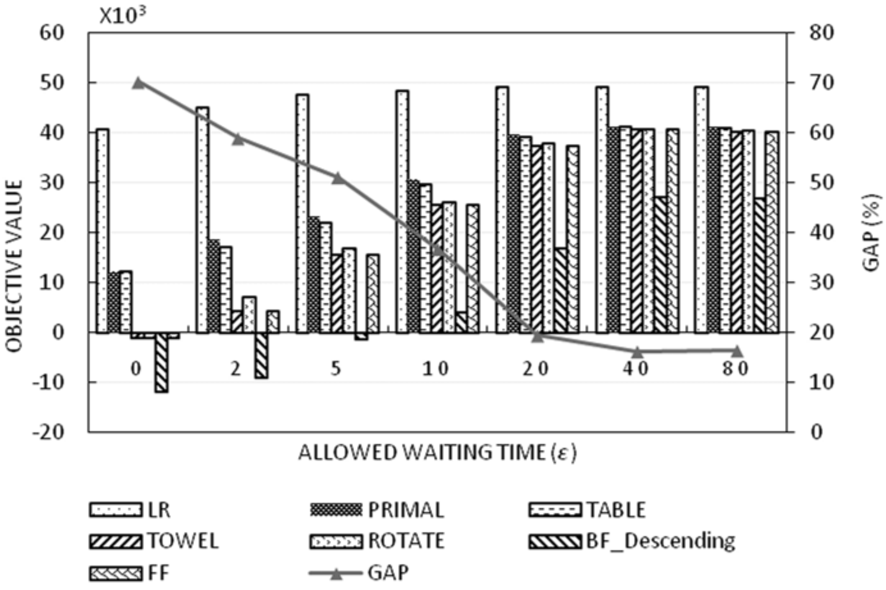


Figure 8 Evaluation of allowed delay tolerance vs. objective value

Evaluation of Task Block Penalty

In this section，we analyze how the task penalty affects the objective value. When the penalty is high，the objective value decreases as the number of tasks is determined； no other action can be performed to achieve the objective value. Figure 9 shows the result. When the task penalty increases linearly，the objective values decrease consistently.

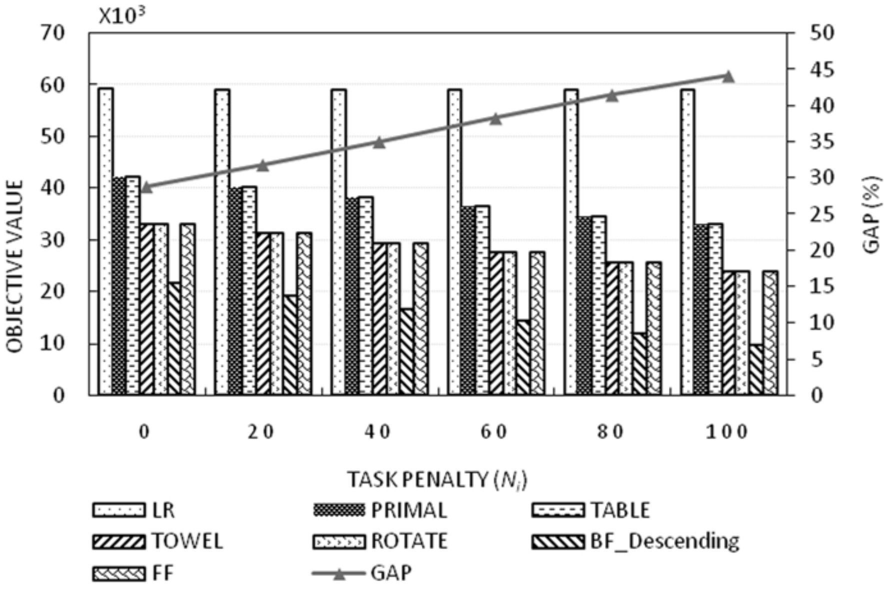


Figure 9 Evaluation of task penalties vs. objective values

Evaluation of Task Revenue Rate

This section examines the relationship between the expected revenue rate and the objective value. If the expected revenue rate is high，tasks with relatively high revenue rates increase the total profits. In this case，the expected task revenue rate was set from 20 to 100. Figure 10 shows the results. When the expected revenue rate is high，the objective value is also high. When the revenue rate is high，the difference between BF_Descending and PRIMAL increases slightly. This is because the curve reported by BF_Descending only showed the loading of tasks and did not show the revenue. As a task with high demand does not guarantee a correspondingly high profit rate，the task with the highest demand is simply assigned as such，and the system will fail to identify the most profitable tasks.

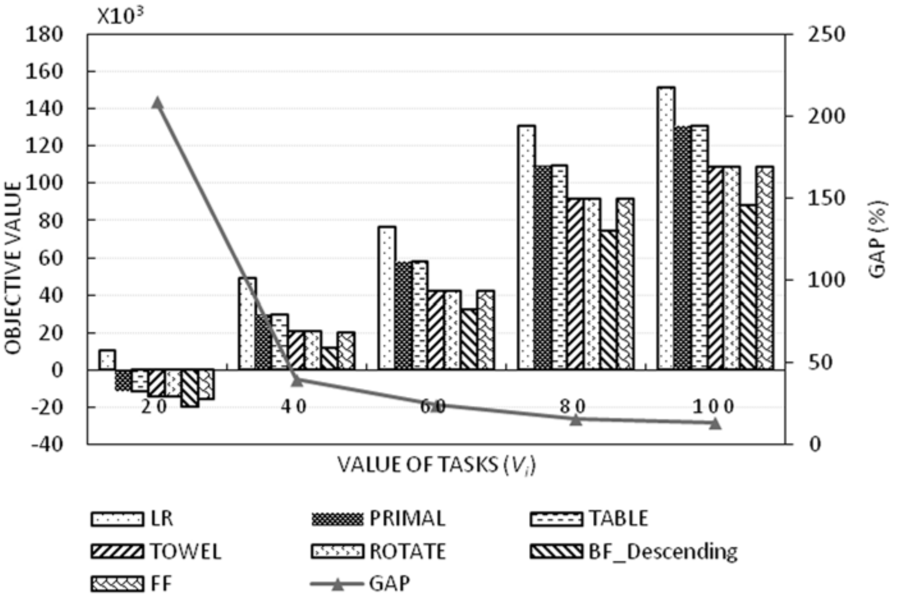


Figure 10 Expected task revenue rate

Evaluation of Server Cost Rate

The last part of our experiments examined the effect of the increasing cost rate of the servers. The trend of the curve was similar to that in the previous case for the task penalty section because when the cost rate was high，the objective value decreased. Figure 11 shows the result. There was a linear relationship between the cost rate and the objective value，as a change in the cost rate reduced the objective value because the number of active servers in the serving status for all tasks under other conditions did not change.

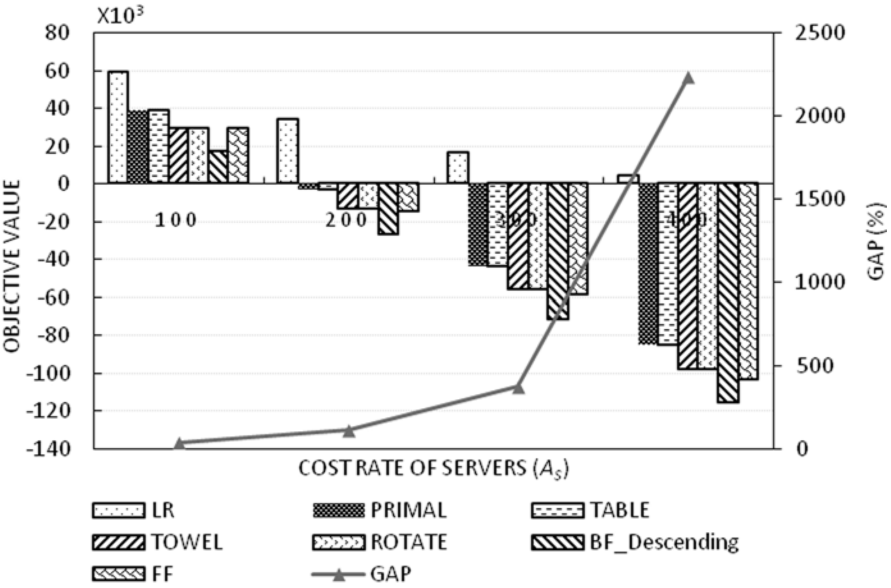


Figure 11 Cost of turning a server on for an interval.

Conclusions

In this paper, we focused on the communication and computation resource management problems in 5G C-RANs. This work can improve QoE at the fronthaul from the customer's perspective. This work can improve the utility of the computing resource pool at the backhaul from a network service provider's perspective.

Resource admission control and scheduling were the goal to apply resource management to offer optimal QoE to users in environments with rapidly changing complex on-demand traffic loads. The influenced factors were studied through a combination of decisions with resource allocation and scheduling by taking both the system perspective and network perspective into account. Lagrangian relaxation was proposed as the near-optimal approach to determine primal feasible solutions. Decision variables were successfully decomposed into subproblems and optimally solved. The results are related to decisions； they significantly influence the system performance； the proposed solutions exhibit the advantages of flexibility and scalability for cloud computing in C-RAN.Near-optimal primal feasible solutions were determined efficiently and effectively； these solutions offered beneficial services that achieved maximum revenue through scalable and flexible strategies in C-RAN. For future research directions，the servers can be shutdown appropriately with controlling task migrations for increasing the energy efficiency to make a green IT system development in 5G.

The contributions of this paper are summarized as follows：

1. We first introduced the cloud resource management problem taking both network （communication） perspectives and system （computation） perspectives into account.

2. A clearly designed mathematical frameworks for resource management problem were proposed to maximize the system revenue of network service providers.

3. Based on dynamic programming，bin packing strategies，and Langrangian relaxation methods，we developed various heuristic approaches to solve these optimization problems.

4. The results of this paper could be used as valuable references or guidelines for the planning and operations of network service providers and researchers in 5G C-RANs.

References

[1] J. Gozalvez，“Tentative 3GPP Timeline for 5G，” *IEEE Vehicular Technology Magazine*，vol. 10，no. 3，pp. 12—18，September 2015.

[2] A. Osseiran，F. Boccardi，V. Braun，K. Kusume，P. Marsch，M. Maternia，O. Queseth，M. Schellmann，H. Schotten，H. Taoka，H. Tullberg，M.A. Uusitalo，B. Timus，and M. Fallgren，“Scenarios for 5G Mobile and Wireless Communications：The Vision of the METIS Project，” *IEEE Communications Magazine*，vol. 52，no. 5，pp. 26—35，May 2014.

[3] A. Gupta and R.K. Jha，“A Survey of 5G Network： Architecture and Emerging Technologies，” *IEEE ACCESS*，vol. 3，pp. 1206—1232，August 2015.

[4] M. Peng，C. Wang，V. Lau，and H.V. Poor，“Fronthaul-constrained Cloud Radio Access Networks： Insights and Challenges，” *IEEE Wireless Communications*，vol. 22，no. 2，pp. 152—160，April 2015.

[5] P. Rost，C.J. Bernardos，A.D. Domenico，M.D. Girolamo，M. Lalam，A. Maeder，D. Sabella，and D. Wübben，“Cloud Technologies for Flexible 5G Radio Access Networks，” *IEEE Communications Magazine*，vol. 52，no. 5，pp. 68—76，May 2014.

[6] A. Checko，H.L. Christiansen，Y. Yan，L. Scolari，G. Kardaras，M.S. Berger，and L. Dittmann，“Cloud RAN for Mobile Networks—A Technology Overview，” *IEEE Communications Surveys & Tutorials*，vol. 17，no. 1，pp. 405—426，Firstquarter 2015.

[7] M. Chen，Y. Zhang，L. Hu，T. Taleb，and Z. Sheng，“Cloud-based Wireless Network：Virtualized，Reconfigurable，Smart Wireless Network to Enable 5G Technologies，” *Mobile Networks and Applications*，vol. 20，no. 6，pp. 704—712，December 2015.

[8] V. Suryaprakash，P. Rost，and G. Fettweis，“Are Heterogeneous Cloud-Based Radio Access Networks Cost Effective?，” *IEEE Journal on Selected Areas in Communications*，vol. 33，no. 10，pp. 2239—2251，October 2015.

[9] R. Dechter，*Constraint Processing*，San Francisco，CA： Morgan Kaufmann Publishers Inc.，2003.

[10] Wikipedia，“Earliest deadline first scheduling，” Internet： https：//en.wikipedia.org/wiki/Earliest_deadline_first_scheduling，March 8，2017，[Accessed November 21，2017]

[11] M.R. Garey and D.S. Johnson，*Computers and Intractability： A Guide to the Theory of NP-Completeness*，New York： W. H. Freeman

& Co.，1990.

［12］Wikipedia，“Proportionally fair，” Internet：https：//en.wikipedia.org/wiki/Proportionally_fair，August 13，2017，[Accessed December 23，2017]

［13］T. Ding，Q.F. Hao，B. Zhang，T.G. Zhang，and L.T. Huai，“Scheduling Policy Optimization in Kernel-Based Virtual Machine，” in *the 2010 International Conference on Computational Intelligence and Software Engineering*（*CiSE 2010*），pp. 1—4，Wuhan，China，December 2010.

［14］L. Zhao，L. Lu，Z. Jin，and C. Yu，“Online Virtual Machine Placement for Increasing Cloud Provider's Revenue，” *IEEE Transactions on Services Computing*，vol. 10，no. 2，pp. 273—285，March-April 2017.

［15］J.Q. Wang，Z.B. Lv，Z.C. Ma，L. Sun，and Y. Sheng，“I-Net：New Network Architecture for 5G Networks，” *IEEE Communications Magazine*，vol. 53，no. 6，pp. 44—51，June 2015.

［16］Y. Liu，M. J. Lee，and Y. Zheng，“Adaptive Multi-Resource Allocation for Cloudlet-Based Mobile Cloud Computing System，” *IEEE Transactions on Mobile Computing*，vol. 15，no. 10，pp. 2398—2410，October 2016.

［17］G. Zhai，L. Tian，Y. Zhou，and J. Shi，“Load Diversity based Processing Resource Allocation for Super Base Stations in Large-scale Centralized Radio Access Networks，” in *the 2014 IEEE International Conference on Communications*（*ICC 2014*），pp. 5119—5124，Sydney，Australia，June 2014.

［18］C. Ran and S. Wang，“Resource Allocation in Heterogeneous Cloud Radio Access Networks：A Workload Balancing Perspective，” in *the 2015 IEEE Global Communications Conference*（*GLOBECOM 2015*），pp. 1—6，San Diego，CA，USA，December 2015.

［19］G. Lu，C. Liu，L. Li，and Q. Yuan，“A Dynamic Allocation Algorithm for Physical Carrier Resource in BBU Pool of Virtualized Wireless Network，” in *the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*（*CyberC 2015*），pp. 434—441，Xi'an，China，September 2015.

［20］A.R. Brown，*Optimum Packing and Depletion*：*The Computer in Space-and Resource-usage Problems*（*Computer monographs*），New York：American Elsevier，1971.

［21］M.R. Garey，R.L. Graham，and J.D. Ullman，“Worst-Case Analysis of Memory Allocation Algorithms，” in *the 4th Annual ACM Symposium on the Theory of Computing*（*STOC 1972*），pp. 143—150，Denver，Colorado，USA，May 1972.

［22］X. Xu，J. Zhang，Y. Ji，H. Li，R. Gu，H. Yu，and J. Zhang，“BBU Aggregation for Maximizing the Resource Utilization in Optical-Enabled Cloud Radio Access Networks，” in *the 2016 15th International Conference on Optical Communications and Networks*（*ICOCN 2016*），pp. 1—3，Hangzhou，China，September 2016.

［23］D.S. Johnson，“Fast Algorithms for Bin Packing，” *Journal of Computer and System*

Sciences，vol. 8，no. 3，pp. 272—314，June 1974.

［24］H. Jin，D. Pan，J. Xu，and N. Pissinou，“Efficient VM placement with Multiple Deterministic and Stochastic Resources in Data Centers，” in *the IEEE Global Communications Conference*（*GLOBECOM 2012*），pp. 2505—2510，Anaheim，California，USA，December 2012.

［25］R.I. Davis and A. Burns，“A Survey of Hard Real-time Scheduling for Multiprocessor Systems，” *ACM Computing Surveys*，vol. 43，no. 4，pp. 35，October 2011.

［26］Y.J. Chiang，Y.C. Ouyang，and C.H. Hsu，“An Optimal Cost-Efficient Resource Provisioning for Multi-servers Cloud Computing，” in *the 2013 International Conference on Cloud Computing and Big Data*（*CloudCom-Asia 2013*），pp. 225—231，Fuzhou，China，December 2013.

［27］N. Bobroff，A. Kochut，and K. Beaty，“Dynamic Placement of Virtual Machines for Managing SLA Violations，” in *the 10th IFIP/IEEE International Symposium on Integrated Network Management*（*IM 2007*），pp. 119—128，Munich，Germany，May 2007.

［28］N. Rameshan，Y. Liu，L. Navarro，and V. Vlassov，“Elastic Scaling in the Cloud：A Multi-tenant Perspective，” in *the IEEE 36th International Conference on Distributed Computing Systems*（*ICDCS 2016*），pp. 25—30，Atlanta，GA，USA，June 2016.

［29］A. Beloglazov and R. Buyya，“Energy Efficient Resource Management in Virtualized Cloud Data Centers，” in *the IEEE/ACM International Conference on Cluster，Cloud and Grid Computing*（*CCGrid 2010*），pp. 826—831，Melbourne，Victoria，Australia，May 2010.

［30］J. Ghaderi，“Randomized Algorithms for Scheduling VMs in the Cloud，” in *the 35th Annual IEEE International Conference on Computer Communications*（*INFOCOM 2016*），pp. 1—9，San Francisco，CA，USA，April 2016.

［31］W.J. Jiang，T. Lan，S. Ha，M.H. Chen，and M. Chiang，“Joint VM Placement and Routing for Data Center Traffic Engineering，” in *the IEEE International Conference on Computer Communications*（*INFOCOM 2012*），pp. 2876—2880，Orlando，Florida，USA，March 2012.

［32］C. Wang，Z. Gu，and H. Zeng，“Global Fixed Priority Scheduling with Preemption Threshold：Schedulability Analysis and Stack Size Minimization，” *IEEE Transactions on Parallel and Distributed Systems*，vol. 27，no. 11，pp. 3242—3255，November 2016.

［33］L.H. Chen and H.Y. Shen，“Consolidating Complementary VMs with Spatial/temporal-awareness in Cloud Datacenters，” in *the IEEE International Conference on Computer Communications*（*INFOCOM 2014*），pp. 1033—1041，Toronto，Canada，May 2014.

［34］Z.H. Han，H.S. Tan，G.H. Chen，R. Wang，Y.F. Chen，and Francis C.M. Lau，“Dynamic Virtual Machine Management via Approximate Markov Decision Process，” in *the 35th Annual IEEE International Conference on Computer Communications*（*INFOCOM*

2016）, pp. 1—9, San Francisco, CA, USA, April 2016.

[35] J. Zhang, Z. He, H. Huang, X. Wang, C. Gu, and L. Zhang, "SLA Aware Cost Efficient Virtual Machines Placement in Cloud Computing, " in *the 2014 IEEE 33rd International Performance Computing and Communications Conference*（*IPCCC 2014*）, pp. 1—8, Austin, Texas, USA, December 2014.

[36] B. Guan, X. Huang, G. Wu, C. Chan, M. Udayan, and C. Neelam, "A Pooling Prototype for the LTE MAC Layer Based on a GPP Platform, " in *the 2015 IEEE Global Communications Conference*（*GLOBECOM 2015*）, pp. 1—7, San Diego, USA, December 2015.

[37] N. Agata, A. Agata, and K. Nishimura, "A Design Algorithm for Ring Topology Centralized-Radio-Access-Network, " in *the 17th International Conference on Optical Network Design and Modeling*（*ONDM 2013*）, pp. 173—178, Bretagne, Brest, France, April 2013.

[38] J. Peng, P. Hong, and K. Xue, "Performance Analysis of Switching Strategy in LTE-A Heterogeneous Networks, " *Journal of Communications and Networks*, vol. 15, no. 3, pp. 292—300, June 2013.

[39] M. Carvalho, D. Menascé, and F. Brasileiro, "Prediction-Based Admission Control for IaaS Clouds with Multiple Service Classes, " in *the 2015 IEEE 7th International Conference on Cloud Computing Technology and Science*（*CloudCom 2015*）, Vancouver, BC, pp. 82—90, December 2015.

[40] Z. Feldman, M. Masin, A.N. Tantawi, and D. Arroyo, "Using Approximate Dynamic Programming to Optimize Admission Control in Cloud Computing Environment, " in *the 2011 Winter Simulation Conference*（*WSC 2011*）, Phoenix, AZ, USA, pp. 3153—3164, December 2011.

[41] M.L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems, " *Management Science*, vol. 50, no. 12, pp. 1861—1871, December 2004.

[42] A.M. Geoffrion, "Lagrangian Relaxation and its Use in Integer Programming, " *Mathematical Programming Study*, vol. 2, pp. 82—114, January 1974.

[43] M.L. Fisher, "An Application Oriented Guide to Lagrangian Relaxation, " *Interfaces*, vol. 15, no. 2, pp. 10—21, April 1985.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Ministry of Science and Technology, Taiwan, R.O.C（grant nos. MOST 106-2221-E-002-032）, and Research Center for Information Technology Innovation, Academia Sinica.