Cost-Efficient Resource Scheduling under QoS Constraints for Geo-Distributed Data Centers

Mirza Mohd Shahriar Maswood*[1], Robayet Nasim*[2], Andreas J. Kassler[2], Deep Medhi[1]
[1]University of Missouri–Kansas City, USA.
[2]Department of Mathematics and Computer Science, Karlstad University, Sweden.

***Abstract*—Geo-distributed Data Centers (DCs) are increasingly common in order to provide scalability for increasing compute demands of modern applications. When multiple geo-distributed DCs serve user requests, it is important to determine which DC and server to select to fulfill the demand at minimum cost, given that enough resources are available in terms of e.g., CPU and bandwidth. This is a complex task since every DC has different operational costs due to e.g. energy, carbon emission, and bandwidth costs. In this paper, we develop a novel mathematical optimization model that guides the decision maker which DC to select, which server to use, and which DC gateway and network path to use to route the user demand in order to satisfy the time varying compute, bandwidth, and latency demands. Our model is based on the concept of virtual networks, which have different requirements in terms of e.g. latency, and we model the queuing delay as a function of the traffic load. Our extensive numerical evaluation, which is based on real-world DC locations, their resource provision costs, and typical demand patterns, shows how operational costs increase with the traffic load, and we analyze the impact of different latency bounds on the performance of different virtual networks.**

***Index Terms*—Geo-distributed data centers; Energy efficiency; Virtual networks; Dynamic resource management; QoS; Latency.**

I. Introduction

There is a growing trend towards large scale and geo-distributed cloud data centers (DCs) in order to support enterprise customers who require virtual network (VN) services in a reservation-oriented mode for both computation- and communication-related resources. However, the operation of such DCs is raising severe concerns about their power consumption and carbon footprints. For example, all the DCs distributed over the USA generated approximately 200 million metric tons of carbon dioxide and were responsible for around 3% of the global power consumption in 2014 [1]. In consequence, electricity costs dominate the overall operational costs for big cloud providers such as Google and Microsoft [2].

To reduce operational costs, both the energy efficiency of the servers inside DCs and the location-based price diversities such as electricity costs, carbon taxes, resource provision costs, etc. need to be considered. Given the heterogeneity of servers in terms of power consumption and cost differences across regions, the key idea [3] is to shift resource allocation (1) to energy-efficient servers and (2) to locations associated with comparatively lower operational costs. In addition to cost savings, DC operators also try to serve as many customers' demands as possible while meeting Quality of Service (QoS) requirements such as latency. For example, when a DC operator allocates resources mostly from a less expensive DC to multiple organizations or groups of customers, sharing resources may result in reduced operational costs but may adversely affect service times. Several issues like spatial diversity of operational costs, available network infrastructure, heterogeneity of servers and resource demand patterns have an impact on the overall performance of the cloud, and often performance metrics such as operational cost and QoS contradict each other. Therefore, a flexible resource allocation strategy is required that meets different business requirements.

Recently, researchers tackled the problem of resource allocation for geographically distributed DCs to achieve different objectives [4], [5], [6]. However, most of them consider east-west traffic (intra DC traffic between hosts). On the contrary, we focus on enterprise customers' requests that result in north-south traffic in DCs requiring both computational and communication-related resources. We study the problem of optimizing the operational cost of the dynamic, multi-cloud-based infrastructures over consecutive time periods where demand varies while ensuring QoS requirements. We aim at managing resources efficiently for DCs with heterogeneous servers and serving diverse request demand profiles for different customer groups using different VN classes.

Our contributions are three-fold. First, we develop a mixed-integer linear programming (MILP) formulation that optimally allocates resources to customers while minimizing location dependent costs such as carbon emission costs and resource provision costs. Unlike the previous contribution [7], our formulation explicitly considers idle power consumption of the servers to reflect the heterogeneous nature of the servers' power consumption. Further, a penalty cost is introduced in the objective function to keep track of the requests that are blocked due to a shortage of resources or not meeting the desired QoS. Second, we link QoS of different VN customers with their latency requirements. Our formulation considers both load-independent propagation latency and load-dependent queuing latency for the links. Finally, we perform an extensive numerical evaluation of the proposed approach using real-world DC locations, demand patterns, and resource provision costs, which presents an insight in to the relationship among request arrival rates, applications QoS requirements, and operating costs of the cloud providers.

The remainder of the paper is structured as follows. Sec-

*Both authors contributed substantially, and share the first authorship. The names are ordered alphabetically.

tion II summarizes the related works. In Section III, we first present our system model for dynamic resource allocation in geo-distributed DCs while in Section IV, we present the MILP model. Sections V and VI present the setup for the numerical evaluation and results of our analysis. Finally, Section VII concludes the paper.

II. Related Work

Several works address the problem of dynamic resource allocation among geo-distributed DCs. [17] presented a detailed survey on cloud resource scheduling techniques, and emphasized that resource scheduling in a geo-distributed environment is a challenging task due to heterogeneity of servers, resources, demands and their associated costs, and discussed the difficulty to solve this with traditional resource management techniques in cloud environments. Towards optimizing energy cost, instead of focusing on servers' power consumption, [18], [19], [20], [2], [21] focused on Geographical Load Balancing (GLB) by exploiting the difference in electricity cost due to location diversities. [19] identifies the importance of utilizing both spatial and temporal diversity in electricity prices and suggested to select a comparatively cheap DC to allocate resources. However, they do not consider servers' energy consumption and hence, may lead to comparatively higher energy consumption. [19] formulated an optimization model for minimizing the DCs electricity costs by considering multi-electricity-market environment. However, they do not consider bandwidth provision cost. [20] presented a scheduling algorithm for distributing workload from the front end servers to back end servers while reducing power costs. However, they only consider delay-tolerant workloads. [21] suggested to use a budget on monthly electricity bill to minimize the network-related energy cost for DCs such as networking devices. They presented an algorithm by dynamically dispatching requests towards multiple DCs but during high workload only guarantee QoS to the premium customers.

On the other hand, some works such as [4], [5], [6], [3] illustrated the potential of balancing load among multiple DCs. [4] proposed a fuzzy logic based load balancing algorithm to reduce operational cost and increase renewable energy consumption without having prior knowledge about future demands. [5] proposed a scheduling approach to distribute incoming workloads to multiple DCs based on local renewable availability, carbon efficiency, and electricity prices. However, it only considered propagation latency, and didn't consider any strict QoS requirements. [6] proposed a load balancing architecture for geo-distributed cloud considering service delay for the applications. The authors proposed to use a traffic migration strategy when a DC gets overloaded. However, the architecture is just a prototype and still requires proper implementation. [3] addressed the GLB problem but with heterogeneity issues such as DCs are constructed with heterogeneous servers and workload are heterogeneous. However, the authors didn't consider the network topology for the DCs. A closely related but slightly different model is presented in [15], where a hierarchical approach is proposed to combine both inter-DC and intra-DC request routing. However, it only exploits the routing problem and does not consider the three-tier network design within a DC. In our previous work [7], we proposed a MILP model for dynamic traffic engineering which allocates resource optimally to VN customers in a multi-location DC environment. However, the model in [7] does not keep track of the blocked requests, does not consider carbon emission taxes and is not able to handle latency requirements.

In contrast to the related works presented above, we consider the DC with full network topology to allocate network resources such as bandwidth to the customers. Further, in our approach each request comes with three attributes, where the first two define computational and network resource demands and the last one defines the maximum tolerable delay. Furthermore, regarding operational costs of the DCs, we consider three different components due to carbon tax, power consumption of the severs and bandwidth provision. In a concurrent work [22], we focused on proportional distribution of load among geo-distributed DCs, while this work focuses on maximum tolerable delay as a QoS constraint.

III. System Model and Assumptions

Our proposed approach aims for a joint DC, gateway, path, and server selection for dynamic request scheduling among geographically distributed DCs. Each request consists of a 3-tuple $\langle r, h, \widehat{\psi} \rangle$ and has a specific duration, where r denotes the computational requirement, h the bandwidth demand, and $\widehat{\psi}$ the latency bound (upper bound on the sum of the delays associated with the links used to fulfill a request). Requests arrive dynamically and the proposed system allocates resources in a DC where the DC and server are jointly selected to minimize costs including location based carbon tax, servers power cost, and resource provision cost. Resource allocation decisions are updated optimally at each review point $t \in T$, where T is a discrete temporal window consisting of review points. As DCs are aimed for serving VN customers, at any time instant, VN tunnels (a set of links from the DC entry point towards the server) and server resources are reserved for serving prior requests. Therefore, any (micro-)workload that needs instant access to resources can be fulfilled through existing VN tunnels and server resources that are already allocated at earlier review points and still active.

From a set of D DCs in distinct geographical regions, each DC d is equipped with J_d number of servers and L_d number of links to satisfy the compute and network demands. Servers are heterogeneous in terms of the power consumption model and run at a particular frequency $f \in F$ to provide a particular processing capacity denoted by a^d_{jf}. Hence, power consumption of every server depends on two factors, the load independent idle consumption (ζ^d_j) and the load dependent current operating frequency (b^d_{jf}). Further, every link has a certain bandwidth capacity. The link delay is composed of a static propagation delay, which depends on the length of the link and a dynamic queuing delay, which depends on the current traffic on the link and the buffer size. The average queuing latency for the link l can be approximated by the

TABLE I: Input Parameters Used in the Formulation

Input Parameters:
D = Set of data centers, $N = \#(D)$
J_d = Set of servers in data center d
I_d = Set of entry points in data center d
V = Set of virtual networks
F = Set of frequencies in which server j can run
L_d = Set of links in data center d
K = Set of line segments of the convex delay curve
$P_{ij}^{vd}(t)$ = Set of paths from entry point i to server j in DC d for VN v
M = A large positive number
ε = A very small positive number
ζ_j^d = Power consumption of idle server j of DC d
b_{jf}^d = Power consumption in server j of data center d at frequency f
$h^v(t)$ = Bandwidth demand for VN v at review point t
$r^v(t)$ = CPU processing demand for VN v at review point t
a_{jf}^d = Capacity of server j of data center d at frequency f
$c_l^d(t)$ = Available capacity on link l of data center d at review point t
$\overline{\phi}_l^d$ = A fixed constant propagation delay of link l of data center d
$\widehat{\phi}_l^d$ = Maximum queuing delay of link l of data center d
$\widehat{\psi}^v(t)$ = Maximum allowable latency for VN v at review point t
$\delta_{ijpl}^{vd}(t)$ = Link-path indicator: 1 if path p from i to j uses l of DC d for VN v at review point t, 0 otherwise
$\overline{\rho}_l^d(t)$ = Utilization of link l in DC d at previous review point t
θ_k^1, θ_k^2 = Coefficients of the linear function that approximate the convex delay curve for k^{th} line segment
β^d = Normalized cost of data center d due to carbon emission tax
Ω^{vd} = Bandwidth pricing per request from VN v served by DC d
α, μ, γ = weight parameters related to 3 optimization objectives

TABLE II: Decision Variables Used in the Formulation

Decision Variables:
$u^{vd}(t)$ = Binary variable to choose DC d for VN v at review point t
$s^{vd}(t)$ = Bandwidth allocation from DC d for VN v at review point t
$\widetilde{s}^v(t)$ = Artificial bandwidth allocation for VN v
$q^v(t)$ = Binary variable to choose real allocation for VN v
$\widetilde{f}^v(t)$ = Binary variable for artificial allocation with penalty for VN v
$y_{ij}^{vd}(t)$ = Bandwidth allocation for VN v from i to j of DC d
$\widetilde{y}_{ij}^{vd}(t)$ = Binary variable that parallels $y_{ij}^{vd}(t)$
$x_{ijp}^{vd}(t)$ = Bandwidth allocation on path p form i to j, if used by VN v
$z_l^{vd}(t)$ = Bandwidth needed on link l of DC d for VN v
$\widetilde{z}_l^{vd}$ = Binary variable to indicate if link l of DC d is used by VN v
$\widetilde{a}_l^d(t)$ = Binary variable to indicate if link l of DC d is used
$\rho_l^d(t)$ = Utilization of link l of DC d at review point t
$\phi_l^{vd}(t)$ = Total link delay of a request from VN v for using link l of data center d
$\overline{\tau}_l^d(t)$ = Queuing delay of link l of data center d due to buffering
$\widetilde{\tau}_l^d(t)$ = Total delay of link l of data center d
$e_j^{vd}(t)$ = CPU processing capacity requirement from server j of datacenter d to satisfy the request coming from VN v at review point t
$g_{ij}^{vd}(t)$ = Server resource (CPU processing capacity) allocation for VN v through entry point i to server j of data center d at review point t
$\widetilde{g}^v(t)$ = Artificial server resource (CPU processing capacity) allocation for virtual network v at review point t
$w_{jf}^{vd}(t)$ = Binary variable to choose the optimum frequency f from the range of available frequencies of server j of DC d to meet the required demand of CPU processing capacity for VN v at review point t
$\widetilde{w}_{jf}^d(t)$ = Binary variable to select the optimum frequency f in which server j of DC d needs to run to meet the required CPU processing capacity for all requests served by that server at review point t
$\xi_l^d(t)$ = This variable indicates the total utilization of link l of DC d including an existing utilization at review point t
$\widetilde{b}(t)$ = This variable indicates the total idle server cost for all server used

M/M/1/K queuing system as follows [8]:

$$d_l = \frac{\frac{x}{b} \cdot (1 + k \cdot (\frac{x}{b})^{k+1} - (k+1) \cdot (\frac{x}{b})^k)}{\frac{x}{e} \cdot (1 - \frac{x}{b}) \cdot (1 - (\frac{x}{b})^k)} \tag{1}$$

where k, b, and e denote the buffer size, the link capacity, and the average packet size, respectively. Furthermore, the entry points I_d to DC d are at the north end and the servers are at the south end of the north-south traffic. There are a $P_{ij}(t)$ available paths from the entry points to the servers, which can be different at the review point t. Additionally, our approach allows for dividing the different enterprise customers groups using VN classes, $v \in V$ depending on their demands.

The proposed approach is based on several assumptions. First, the bandwidth demands can be split and routed over multiple paths. Second, one central controller is used to run the optimization model at each review point to allocate resources to the consumers. For instance, a software-defined network (SDN) based approach can be applied, where traffic can be load balanced on a subflow level [9]. Third, as the queueing delay is nonlinear, a piecewise linear (PWL) approximation is applied to estimate the delay curve (1) [10]. The notations used are summarized in Tables I-II.

IV. Mathematical Formulation

In this section, the optimization model for the dynamic scheduling problem is presented. First, at most, only one DC out of the N DCs can be selected to meet the request for a VN v at review point t:

$$\sum_{d \in D} u^{vd}(t) \leq 1, \quad v \in V \tag{2}$$

For the given DC, to satisfy the bandwidth demand from a VN, the DC must ensure bandwidth requirements for that VN:

$$s^{vd}(t) = h^v(t) u^{vd}(t), \quad v \in V, d \in D \tag{3}$$

Next, either the total link bandwidth demand must be served by the chosen DC or if there is not enough bandwidth to serve a request from a particular VN, then this request will be chosen as an artificial allocation, $\widetilde{s}^v(t)$, to keep a count on blocked requests:

$$\sum_{d \in D} s^{vd}(t) + \widetilde{s}^v(t) = h^v(t), \quad v \in V \tag{4}$$

A binary variable is used to indicate an artificial allocation if a request cannot be served by limited resources:

$$\widetilde{s}^v(t) \leq M \widetilde{f}^v(t), \quad v \in V \tag{5}$$

A request from a VN can only be considered for either a real or artificial allocation but not for both at review point t:

$$\widetilde{f}^v(t) + q^v(t) = 1, \quad v \in V \tag{6}$$

If a request is considered for a real allocation, the total link bandwidth demand must be served by the chosen DC:

$$\sum_{d \in D} s^{vd}(t) = h^v(t) q^v(t), \quad v \in V \tag{7}$$

The total amount of the bandwidth demand from VN v which will be served by DC d, is the summation of the bandwidth that is allocated from all chosen entry points i to all chosen servers j of DC d at review point t:

$$\sum_{i \in I_d} \sum_{j \in J} y_{ij}^{vd}(t) = s^{vd}(t), \quad v \in V, d \in D \tag{8}$$

Now, we introduce a binary shadow variable $\widetilde{y}_{ij}^{vd}(t)$ corresponding to $y_{ij}^{vd}(t)$ to track one-to-one mapping from i to j at t using a large and small positive number M and ε, respectively:

$$y_{ij}^{vd}(t) \leq M \widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \tag{9}$$

$$y_{ij}^{vd}(t) \geq \varepsilon \widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \tag{10}$$

The bandwidth allocated to path p from entry point i to server j of DC d is given by using the path flow variables x_{ijp}^{vd}:

$$\sum_{p \in P_{ij}^{vd}(t)} x_{ijp}^{vd}(t) = y_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \tag{11}$$

If any bandwidth is allocated on path p to satisfy a portion of h^v of a request from VN v, then all the links associated with that path have to carry that portion of h^v. Therefore, we can determine the flow on link l for tuple $\langle v,d \rangle$:

$$\sum_{i \in I} \sum_{j \in J} \sum_{p \in P_{ij}^{vd}(t)} \delta_{ijpl}^{vd}(t) x_{ijp}^{vd}(t) = z_l^{vd}(t) \quad d \in D, l \in L_d, v \in V \tag{12}$$

while the total amount of bandwidth required in link l of DC d to satisfy the requests of all VNs must not exceed the capacity of that link:

$$\sum_{v \in V} z_l^{vd}(t) \leq c_l^d(t), \quad l \in L_d, d \in D \tag{13}$$

Constraint (14) is used to calculate the utilization of each link l of each DC d. The utilization on each link is computed as the sum of the demands forwarded through it and normalized to the total capacity of the link.

$$\rho_l^d(t) = (\sum_{v \in V} z_l^{vd}(t))/c_l^d(t), \quad l \in L_d, d \in D \tag{14}$$

(15) calculates the total utilization of link l of DC d at review point t by summing up existing utilization from the previous review point and current utilization.

$$\xi_l^d(t) = \overline{\rho}_l^d(t) + \rho_l^d(t), \quad l \in L_d, d \in D \tag{15}$$

(16) and (17) are used to identify the links which are used to satisfy a request from VN v.

$$z_l^{vd}(t) \leq M \widetilde{z}_l^{vd}(t), \quad l \in L_d, v \in V, d \in D \tag{16}$$

$$z_l^{vd}(t) \geq \varepsilon \widetilde{z}_l^{vd}(t), \quad l \in L_d, v \in V, d \in D \tag{17}$$

(18) to (20) are used to identify all the links of available DCs which are used to satisfy all requests at review point t.

$$\widetilde{u}_l^d(t) \geq \widetilde{z}_l^{vd}(t), \quad l \in L_d, v \in V, d \in D \tag{18}$$

$$\widetilde{u}_l^d(t) \leq \sum_{v \in V} \widetilde{z}_l^{vd}(t), \quad l \in L_d, d \in D \tag{19}$$

$$\widetilde{u}_l^d(t) \leq 1, \quad l \in L_d, d \in D \tag{20}$$

(21) and (22) are used to calculate the piecewise linear queueing delay on the links using the coefficients θ_k^1 and θ_k^2.

$$\theta_k^1 + \theta_k^2 \ \xi_l^d(t) \leq \overline{\tau}_l^d(t) + (1 - \widetilde{u}_l^d(t)) \ \widehat{\phi}_l^d, \quad k \in K, l \in L_d, d \in D \tag{21}$$

$$\overline{\tau}_l^d(t) \leq \widehat{\phi}_l^d \ \widetilde{u}_l^d(t), \quad l \in L_d, d \in D \tag{22}$$

Constraint (23) is used to calculate the total delay of link l of DC d which is the sum of the queuing latency and propagation latency of that link.

$$\widehat{\tau}_l^d(t) = \overline{\tau}_l^d(t) + \overline{\phi}_l^d, \quad l \in L_d, d \in D \tag{23}$$

(24) to (27) are used to satisfy that only the delay of those links are considered to calculate the total link delay of a request which are used to satisfy that request.

$$\phi_l^{vd}(t) \leq M \widetilde{z}_l^{vd}(t), \quad l \in L_d, v \in V, d \in D \tag{24}$$

$$\phi_l^{vd}(t) \leq \widehat{\tau}_l^d(t), \quad l \in L_d, v \in V, d \in D \tag{25}$$

$$\phi_l^{vd}(t) \geq \widehat{\tau}_l^d(t) - (1 - \widetilde{z}_l^{vd}(t))M, \quad l \in L_d, v \in V, d \in D \tag{26}$$

$$\phi_l^{vd}(t) \geq 0, \quad l \in L_d, v \in V, d \in D \tag{27}$$

(28) is used to calculate the total link delay for each request which must be less than or equal to the maximum allowable latency of that request.

$$\sum_{d \in D} \sum_{l \in L_d} \phi_l^{vd}(t) \leq \widehat{\psi}^v(t), \quad v \in V \tag{28}$$

Furthermore, we must determine whether a request can be served with limited server resources or not. If there is a resource limitation to serve a particular request from a VN at review point t, then the binary variable to choose an artificial allocation for that request will be 1:

$$\sum_{d \in D} \sum_{i \in I_d} \sum_{j \in J_d} g_{ij}^{vd}(t) + \widetilde{g}^v(t) = r^v(t), \quad v \in V \tag{29}$$

$$\widetilde{g}^v(t) \leq M \widetilde{f}^v(t), \quad v \in V \tag{30}$$

$$\sum_{d \in D} \sum_{i \in I_d} \sum_{j \in J_d} g_{ij}^{vd}(t) = r^v(t) \, q^v(t), \quad v \in V \tag{31}$$

Next, we address resource allocation of $r^v(t)$ to the appropriate tuple $\langle d,i,j \rangle$, in accordance with shadow variable $\widetilde{y}$.

$$g_{ij}^{vd}(t) \leq M \widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \tag{32}$$

$$g_{ij}^{vd}(t) \geq \varepsilon \widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \tag{33}$$

$$\sum_{i \in I_d} g_{ij}^{vd}(t) = e_j^{vd}(t), \quad j \in J_d, v \in V, d \in D \tag{34}$$

In constraint (34), $e_j^{vd}(t)$ represents the total amount of resources required from server j to satisfy a request from VN v at time t that uses the server coming through all entry points of a particular data center. The total resources allocated to each request from a particular server must be less than or equal to the available resources of that server of a data center:

$$e_j^{vd}(t) \leq \sum_{f \in F} a_{jf}^d w_{jf}^{vd}(t), \quad j \in J_d, v \in V, d \in D \tag{35}$$

Server j running at frequency f can produce capacity a_{jf}^d. However, a server can run only at one frequency option for a request from VN v (36). (37) calculates the total capacity required from server j of DC d to satisfy all the requests forwarded to that server at review point t. (38) satisfies that a server cannot run at more than one frequency. Constraint (39) calculates the idle power consumption by server j of DC d.

$$\sum_{f \in F} w_{jf}^{vd}(t) \leq 1, \quad j \in J_d, d \in D, v \in V \tag{36}$$

$$\sum_{f \in F} \sum_{v \in V} a_{jf}^d w_{jf}^{vd}(t) = \sum_{f \in F} a_{jf}^d \widetilde{w}_{jf}^d(t), \quad j \in J_d, d \in D \tag{37}$$

$$\sum_{f \in F} \widetilde{w}_{jf}^d(t) \leq 1, \quad j \in J_d, d \in D \tag{38}$$

$$\sum_{f\in F}\sum_{j\in J}\sum_{d\in D}\widetilde{w}^{d}_{jf}(t)*\zeta^{d}_{j}=\widetilde{b} \quad (39)$$

To achieve the goal of the optimization problem, four cost components are considered in the objective function: the bandwidth, energy consumption, carbon emission based on DC location, and the penalty cost for those requests which are not satisfied by the limited resources identified through the artificial allocation. Furthermore, since resources are of different types, a utility function-based approach is taken by assigning weights to different components from the objective function. The first three sources of costs are assigned different weight parameters, α, μ, γ, to understand the influence of each term on the overall decision, while the penalty term is assigned through parameter M. Thus, the goal is to accommodate as many requests as possible by minimizing the amount of resources used, leading to the following objective function:

$$\min \alpha \sum_{d\in D}\sum_{v\in V}\Omega^{vd}*s^{vd}(t)+\mu\Big(\sum_{d\in D}\sum_{j\in J}\sum_{f\in F}b^{d}_{jf}\widetilde{w}^{d}_{jf}(t)+\widetilde{b}(t)\Big) +\gamma\sum_{d\in D}\sum_{v\in V}\beta^{d}u^{vd}(t)+M\sum_{v\in V}\widetilde{f}^{v}(t) \quad (40)$$

To summarize, our unified formulation jointly addresses decision choices at four different levels: data center, entry point, which links to route the traffic over and then the destination server. Secondly, we take power consumption into account in determining the right frequency for operating a server and traffic and link latency for determining the paths. Finally, we consider four cost components in the composite objectives.

V. Implementation and Evaluation Design

We used AMPL [12] and IBM ILOG CPLEX 12.6.0 [13] to solve the multi-objective MILP at each review point. CPLEX option of node limits is set to 10000 to obtain near optimal solutions without consuming too much time.

A. Parameter Settings

In the evaluation, we use the DC topology shown in Fig. 1. We use a maximum of three DCs ($N = 3$). Each DC is considered to be identical in terms of the number of available servers (16 in each DC) and number of links (56 in each DC) inside the DC. While these values for the number of servers are low by today's DCs, we use these values since the primary focus of our work here is to understand and reveal relationship among the arrival rate of requests, applications QoS requirements, and operating cost for cloud providers with multiple DCs.

We consider the maximum capacity of each link inside the DCs to be 1 Gbps with a propagation latency of 1 ms and the maximum normalized capacity of each server is 1.

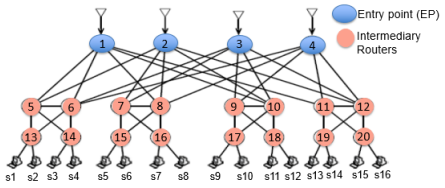

Fig. 1: Data Center Topology [11]

Regarding queuing delay, we linearize and approximate (1) by using 7 line segments. However, every DC is constructed from non-identical server classes with different specifications on power consumption. We consider three different types of servers, Dell R515, HP DL380 G8, and HP DL585 G7, and their idle power consumption [14] is used (Table III). Each server has 10 different frequency options to run with associated capacity and power consumption. The normalized cost, β_d of using each DC is different as mentioned in Table III. This cost depends on the carbon emission rate (CER). The DCs that rely on nuclear and hydro power for electricity generation have very low CERs compared to the DCs that use coal and natural gas. For a detailed explanation, please see [15]. We also assume that the DCs are located in Ontario, Britain, and Kansas and their associated costs are denoted by β_1, β_2, and β_3, respectively [15].

We use 3 different VN classes. The bandwidth provision cost per request for each VN class is different, and it also depends on the DC from where the request is served. This price list is presented in table IV, which is collected from the real world trace [15]. Further, we set 4 paths $P^{vd}_{ij}(t)$ from an entry point to a server to allocate bandwidth in order to satisfy a specific request for the duration of that request. In order to have a balanced impact of different cost components on the objective function, we scale up the bandwidth provision costs (Ω^{vd}) and carbon costs (β_d) by multiplying by 100,000 and 25, respectively. Further, through initial experimentation, we determined the weight factors for each term in the objective function (40) and set them as $\alpha = 0.1, \gamma = 0.3, \mu = 0.6$ to understand the influence of the three cost components on the overall operational cost. They were chosen to give higher importance on the energy consumption cost, followed by the carbon emission cost and finally, bandwidth cost.

To quantify the effectiveness of our approach, demand requests from customers need to follow a realistic pattern. We used the traffic pattern generated for US core gateways network by [16] and the requests' arrival rate (demand intensity) is divided into four slots (Table V) [16]. The request arrival from different VN classes are random and follows a Poisson distribution. Further, the service duration of the request arrival is assumed to follow the negative exponential distribution with an average value of 5 time units measured in terms of the number of discrete review points [7]. If the request arrival rate

TABLE III: Values for Different Input Parameters

Number of DCs #(D)	3
Number of Servers in each DC #(J_d)	16
Number of Entry points in each DC #(I_d)	4
Number of VN classes #(V)	3
ζ^1_j(DC 1), ζ^2_j(DC 2), ζ^3_j(DC 3)	213, 109, 258
Servers' frequency options#(F)	10
Number of links in each DC #(L_d)	56
Capacity of each link (c^d_l)	1
Propagation delay of each link ($\overline{\phi}^d_l$)	1
Carbon emission cost ($\beta_1, \beta_2, \beta_3$)	0.1, 0.5, 1.0
Weight factors (α, γ, μ)	0.1, 0.3, 0.6

TABLE IV: Bandwidth Cost per Request ($/request)

	Ontario (DC 1)	Britain (DC 2)	Kansas (DC 3)
VN 1	0.0010	0.0015	0.0012
VN 2	0.0008	0.0011	0.0009
VN 3	0.0006	0.0008	0.0007

TABLE V: Specifications of the Scenarios Used in This Study

Scenarios	Requests (r, h, ψ)			Demand Intensity
	VN 1	VN 2	VN 3	
Scenario 1	0.3, 0.45, 10	0.3, 0.45, 10	0.3, 0.45, 10	0.20, 0.45, 0.70, 0.80
Scenario 2	0.3, 0.45, 5	0.3, 0.45, 10	0.3, 0.45, 10	0.20, 0.45, 0.70, 0.80
Scenario 3	0.3, 0.45, 4	0.3, 0.45, 4	0.3, 0.45, 10	0.20, 0.45, 0.70, 0.80
Scenario 4	0.6, 0.9, 10	0.6, 0.9, 10	0.6, 0.9, 10	2.0
Scenario 4a	0.6, 0.9, 4	0.6, 0.9, 6	0.6, 0.9, 8	2.0
Scenario 4b	0.3, 0.8, 10	0.6, 0.9, 10	0.9, 1.0, 10	2.0
Scenario 5	0.6, 0.9, 10	0.6, 0.9, 10	0.6, 0.9, 10	2.0

is too high, the DCs may not have enough capacity to fulfill all demands. Thus, our approach keeps track of the blocked requests to determine the blocking rate.

B. Scenario Setup

For evaluating our proposed approach, all the computations are performed on an Intel i7-4770k CPU@ 3.40GHz machine with 32 GB RAM. A total number of seven scenarios are investigated. In scenario 1, homogeneous requirements for all the VN classes are generated, i.e., bandwidth, CPU demands, and latency bound for all the VN classes are similar. In scenario 2, we lowered the latency bound for VN class 1. In scenario 3, we further reduced the limit on the worst case latency for both VN classes 1 and 2. The goal of scenario 2 and 3 is to investigate the impact of strict latency bounds on operational costs of DCs. In scenario 4, we used a higher demand intensity and also increased both compute and bandwidth demands for all VN classes. The focus of this scenario is to reveal the cost and blocking relation under heavy demand. Further, we extended scenario 4 in two directions to explore the impact of heterogeneity on the proposed resource allocation scheme. In the first extension (scenario 4a), heterogeneity is introduced in terms of latency bound, i.e., different VN classes have different upper bounds on their tolerable latency. As a second extension (scenario 4b), heterogeneity is added by setting different resource requirements for different VN classes. Finally, in order to investigate the impact of the heterogeneous server type together with DCs' location cost, we have interchanged the server types between the first and second DCs in the scenario 5. Table V summarizes all the scenarios used in our study. For all the experiments, we first determined the warm-up time and then collected the data for a steady-state region after the warm-up time. Further, for each demand intensity, we used 10 independent seeds and reported the average value.

VI. Numerical Results

A. Cost Analysis for Homogeneous Resource Requirement

The goal of scenario 1 is to understand how the allocation cost changes with increasing demand intensity. From Fig. 2, we observe that the cost increased significantly, approximately 500%, when the demand intensity increased from 0.2 to 0.7.

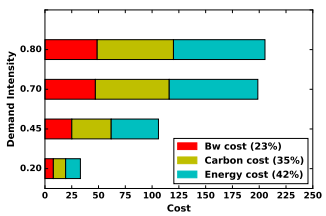

Fig. 2: Demand Intensity Vs Total Cost for Scenario 1

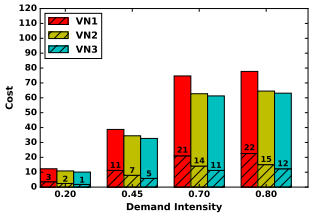

Fig. 3: Demand Intensity Vs Per VN Cost for Scenario 1

TABLE VI: Avg. Latency of VNs and DCs for each Scenario

Demand Intensity	VN 1	VN 2	VN 3	DC 1	DC 2	DC 3
Scenario 1 (Max. Latency Bound: VN 1 → 10, VN 2 → 10, VN 3 → 10)						
0.20	4.71	4.91	4.60	0	4.74	0
0.45	5.64	5.48	5.43	0	5.52	0
0.70	6.13	5.82	6.27	0	6.07	0
0.80	6.36	6.66	6.82	0	6.61	0
Scenario 2 (Max. Latency Bound: VN 1 → 5, VN 2 → 10, VN 3 → 10)						
0.20	3.83	5.08	4.72	0	4.55	0
0.45	3.97	6.16	5.90	0	5.25	0
0.70	4.03	6.21	6.57	0	5.71	0
0.80	4.19	6.98	6.93	0	5.86	0
Scenario 3 (Max. Latency Bound: VN 1 → 4, VN 2 → 4, VN 3 → 10)						
0.20	3.10	3.35	5.34	0	3.93	0
0.45	3.45	3.44	6.51	0	4.37	0
0.70	3.49	3.46	7.14	0	4.66	
0.80	3.57	3.49	7.69	0	4.87	0
Scenario 4 (Max. Latency Bound: VN 1 → 10, VN 2 → 10, VN 3 → 10)						
2.00	7.66	8.59	8.00	8.02	8.80	7.41
Scenario 4a (Max. Latency Bound: VN 1 → 4, VN 2 → 6, VN 3 → 8)						
2.00	3.73	5.27	6.72	4.88	5.59	4.69
Scenario 4b (Max. Latency Bound: VN 1 → 10, VN 2 → 10, VN 3 → 10)						
2.00	8.39	8.72	9.16	8.34	9.11	7.88
Scenario 5 (Max. Latency Bound: VN 1 → 10, VN 2 → 10, VN 3 → 10)						
2.00	8.28	7.78	7.82	8.88	7.88	7.44

Then, the cost increased slightly around 3.36% for the demand intensity of 0.8. The nature of change in the operational cost with respect to the demand intensity indicates that initially, the cost increased sharply due to the fixed carbon cost and lower utilization of the servers, but when the DC utilization reached approximately 50% then cost increases slightly. For demand intensity of 0.2, on average 2 servers were powered on to accommodate all the requests, but it increased to 8 when the demand intensity increases to 0.7 or 0.8. Among our three considered cost factors that affected the overall operational cost for a DC, energy contributed to the highest proportion (around 42%), followed by the carbon cost (around 35%), and the bandwidth provision cost (around 23%).

Fig. 3 shows the costs for different VN classes. The bars illustrate the total cost for different levels of demand intensity where as the bottom layers of the bars (stripped lines) show the bandwidth costs. Even though all the VN classes are homogeneous in terms of resource demand, the bandwidth cost varies significantly. This is because the bandwidth pricing scheme varies depending on the VN type and the DC where the demand is allocated. The average cost for VN 1 is always higher than the cost for VN 2 or 3 as the bandwidth price for VN 1 is the highest. For all the VNs, although DC 1 and 3 offer a comparatively lower bandwidth price, all the demands are still allocated to DC 2. Since, for the demand intensity used in this scenario, DC 2 had enough resources to satisfy all requests, no request is allocated to DC 1 or 3 as DC 2 is the overall cheapest option. Therefore, the cost of different VN classes with homogeneous requirements can vary because of being served by a heterogeneous system environment. Finally,

TABLE VII: Average Number of Active Links and Link Utilization of DCs for Different Scenarios

Scenario 1						
	Average Active Links			Average Link Utilization		
Demand Intensity	DC 1	DC 2	DC 3	DC 1	DC 2	DC 3
0.20	0	8.80	0	0	46.07%	0
0.45	0	19.65	0	0	47.99%	0
0.70	0	31.98	0	0	48.45%	0
0.80	0	32.13	0	0	49.84%	0
Scenario 2						
0.20	0	8.01	0	0	48.02%	0
0.45	0	14.17	0	0	49.40%	0
0.70	0	23.23	0	0	51.07%	0
0.80	0	29.53	0	0	52.85%	0
Scenario 3						
0.20	0	7.10	0	0	49.51%	0
0.45	0	13.68	0	0	51.56%	0
0.70	0	21.09	0	0	53.36%	0
0.80	0	26.53	0	0	54.68%	0
Scenario 4						
2.0	46.93	52.73	40.93	80.02%	81.39%	74.23%
Scenario 4a						
2.0	40.93	48.65	30.56	86.57%	88.53%	84.64%
Scenario 4b						
2.0	48.24	54.19	39.51	78.12%	79.31%	78.07%
Scenario 5						
2.0	54.08	49.78	37.55	79.87%	78.75%	75.62%

from these two figures, we observe that cost increases non-linearly with increasing demand intensity.

B. Impact on Latency

Here, we investigate how the avg. VN latency changes with increasing demand intensity. Demand intensity is increased until the system gets so overloaded that it cannot accommodate all the requests. We analyze the impact of introducing stricter latency bounds . Additionally, different latency bounds or different resource demands for different VN classes allow us to understand the impact of heterogeneity in our resource allocation scheme. The results of are summarized in Table VI.

For scenario 1, we observe that the avg. latency for all VN classes increases non-linearly with increasing demand intensity and follows a pattern similar to the previously discussed cost of resource allocation. Here, all three VN classes have a similar latency bound of 10, and their average latency is also close to each other. This further implies that no VN class is penalized in terms of latency due to sharing network paths with other VN classes. In fact, the most common tendency is to allocate the shortest path to all of the VNs, which may in turn create unexpected latency for some VNs because of network congestion. However, using splittable routing and distributing traffic, we can maintain a balance for network resource allocation among different VN customers. For instance, for a demand intensity of 0.2, the average latency for all VN classes is 4.74 with a deviation of 0.15 and the latency shows a similar behavior for all other demand intensities.

Next, in scenario 2, as we reduced the latency bound for VN 1 from 10 to 5, we found the avg. latency always within the bound for all demand intensities. However, this had an impact on the avg. latency of the other two VN classes. For instance, the avg. latency for VN 1 was 4.05 for all demand intensities which reduced approximately 29% compared to the scenario 1, but the avg. latency for VN 2 and VN 3 increased around 7% and 5%, respectively. However, the avg. latency of all three VNs reduced slightly. In scenario 3, we decreased the latency bound to 4 for both VN 1 and 2, which restricted the average latency for VN 1 and 2 within 4. However, the avg. latency for VN 3 increased more compared to scenario 1 and 2, which is comparatively 16% and 11% higher. Further, the avg. latency for all VN classes was lowest among all these three scenarios. For these scenarios, the avg. latency in DC 1 and 3 was 0, as all the requests was served by DC 2 because of its cheapest allocation cost. When a VN class has more strict latency bound, the solution forces other VNs to increase their avg. latency, since traffic related to the higher latency VN classes is routed over different paths to reduce traffic volume on the path used by low latency VN classes.

In scenario 4, the demand intensity is increased to 2.0. Here, all three DCs were used to satisfy customers' requests. The key findings are: (i) the VN wise avg. latency is much higher compared to the previous scenarios as the demand intensity is 2.5 times higher than the max. intensity for the previous scenarios; (ii) the avg. latency is showing a downward trend from the cheapest DC to the most expensive one as the cheapest DC is comparatively congested because of admitting higher number of requests. For instance, the avg. latency for DC 2 is 8.80, which has the lowest allocation cost, then the latency decreases approximately 9% for DC 1 (next cheaper DC), and around 16% for DC 3 (the most expensive one). Further, scenario 5 showed the same behavior as scenario 4. Because of exchanging server types between DC 1 and DC 2, now DC 1 is the cheapest option in terms of operational cost and has experienced the highest latency of 8.88. When different VN classes have different max. tolerable latency (scenario 4a), the resource allocation scheme satisfied all the latency bounds for the VNs. Thus, VN 1 experienced the lowest avg. latency whereas VN 3 faced the highest. However, the comparatively lower latency for scenario 4a than 4 can be explained by the higher blocking rate of 4a (see Section VI-D). Finally, when the heterogeneity is imposed on VNs in terms of resource requirements (scenario 4b), all the VN classes experienced the highest latency. Thus, heterogeneity has an impact on the resource allocations scheme and further, traffic of all VN classes are routed over higher number of links as they have higher tolerable latency bounds.

C. Average Number of Active Links and their Utilization

The impact of demand diversity, latency bounds and operational costs on link utilization and number of active links is summarized in Table VII. In scenario 1, we observed that the number of active links and their utilization increased non-linearly with demand intensity where the utilization increased slowly. As the demand intensity increases, the latency bounds coupled with our splittable flow routing model require more links to activate until the latency bound is reached, which then requires more load balancing and thus more links to activate.

For scenario 2, when the max. tolerable delay for VN 1 is reduced, we observe that the avg. number of links is also reduced. However, the avg. link utilization is increased compared to s-1. In the topology, there are several possible paths from entry points to servers and we use a splittable flow routing. If latency bounds of some VNs are more strict (e.g. VN 1), less paths can serve the traffic under the stricter

TABLE VIII: Blocking Rate

Scenario	Overall	VN 1	VN 2	VN 3
4a	7.5%	11.15%	7.92%	3.47%
4b	8.75%	5.26%	7.15%	13.83%

latency bounds (the low latency traffic will be more likely routed along the shortest hop paths). Consequently, the link util. increases as less links need to be activated. The same behavior is visible for scenario 3. Here, the average number of active links is the lowest and the average link utilization is the highest among these three scenarios. As we continue to reduce the maximum tolerable delay for low demands, the average number of active links reduces and their utilization increases because adding extra links causes additional delay.

Once we increase the demands for scenario 4 and have a less latency bound of 10 which is the same for all VNs, the avg. used links and util. increases significantly. Also, since DC 2 has the cheapest allocation cost, it serves the highest number of requests and hence, the link util. is the highest for this DC. In s-5, DC 1 has the highest number of used links and utilization as it is the cheapest DC now. This implies that our approach can successfully use the cheapest DC among all available ones in order to reduce the resource provision cost. In 4a and b, some traffic cannot be admitted and blocked because the latency bounds are already so tight for some VN traffic that no available path is found to satisfy the required bound.

D. Impact of Demand Heterogeneity on Blocking Rate

As blocking we refer to a situation when a request does not get enough resources to be satisfied both for compute and network. In scenarios 1 to 3, the demand intensity is too low to create any blocking. Therefore, to understand the impact of blocking, we have analyzed the results of scenarios 4, 4a, and 4b, which are presented in Table VIII. In scenario 4, when all the VNs have similar resource demands and latency bounds, we found an overall blocking rate of 4.50%. In scenario 4a, where VNs are heterogeneous in terms of max. tolerable delay, the overall blocking rate is 7.5% which is significantly higher than scenario 4. Further, we notice that the VN class with the lowest delay bound has the highest blocking rate and vice versa. For instance, VN class 1 has the most strict latency bound of 4 and hence, its blocking rate is approximately 11.15%, while the blocking rate for VN class 3 is the lowest of 3.47% due to the higher tolerable latency of 10. Finally, in scenario 4b the VN classes are heterogeneous in terms of resource requirement. Consequently, it shows a higher blocking rate (overall 8.75%) compared to the previous two scenarios. The VN class with the highest resource requirements faces the maximum blocking rate and the VN class with lowest resource requirements (VN 1) faces the lowest blocking rate (5.26%). We notice that heterogeneity has an impact on the resource allocation scheme in terms of higher blocking rate which may lead to significant revenue losses for the cloud providers. Additionally, the VN class with the lowest resource requirement or with the highest upper bound on worst case latency, has comparatively higher admission rate to fulfill their resource demands.

E. Impact of Server Heterogeneity on Energy Consumption

In order to clearly identify the impacts of server types on DCs' operational cost, we have interchanged the servers between DC 1 and 2 in scenario 5. In both scenarios (4 and 5), resource requirements for all VNs are the same and the carbon cost and the bandwidth cost of VNs are always lowest for DC 1. However, the energy efficiency of the servers, specially their idle power consumption, influences the resource allocations. For instance, in scenario 4, servers of DC 2 have on avg. 45% lower idle consumption than servers of DC 1 and hence, allocate a higher number of requests. In this case, DC 2 has on avg. 11 active servers, DC 1 and 3 has on avg. 10 and 6 powered on servers. On the other hand, scenario 5 shows the opposite trend. For example, now, DC 1 has on avg. 11 active severs and DC 2 and 3 have 10 and 6 active servers, respectively. When energy efficient servers are combined with cheapest carbon cost and lowest bandwidth cost, the overall operational cost for resource allocation becomes comparatively cheaper. For instance, scenario 5 has 5% lower operational cost compared to scenario 4 for the same resource demand.

VII. Conclusions and Future Directions

In this paper, we tackled the problem of allocating a given workload in terms of a time varying compute and bandwidth demands, to diverse geo-located DCs to minimize total costs such as bandwidth costs, carbon taxes, etc. We presented a novel MILP optimization model that is solved at each review point of a dynamic traffic engineering environment to allocate resources to the best servers over the best network paths while maintaining a strict QoS constraint on maximum tolerable latency and bandwidth capacity. We consider a fixed and a variable delay due to the queuing caused by traffic routed over the links and DC gateways. Through a systematic numerical analysis, we show the dependency among resource requirement, resource availability, and blocking, for both homogeneous and heterogeneous resource requirements. We study how the average number of links used and their utilization varies as we reduce the latency bounds. We noticed that bursty traffic resulted in denial of service (blocking) for several requests under limited resources. This is an indicator for the DC providers to be prepared for the temporal diversity of the network load. We observed that our approach reduced the provisioning cost by using the spatial diversity of bandwidth, energy and carbon emission cost of geo-distributed DCs.

While our optimization model is too complex to optimize the resource allocation in realtime, it serves as an important benchmark against which any fast solution heuristic can be compared against. In the future, we plan to develop a fast online algorithm and study the behavior of a large-scale system with several DCs that consist of many servers. Using our scheme, we further plan to study how different performance matrices vary for DCs with different architectures.

ACKNOWLEDGMENT

Part of this work has been funded by the Knowledge Foundation of Sweden through READY and HITS Profile and by the National Science Foundation Grant # 1526299, USA.

REFERENCES

[1] B. Lavallée. Undertaking the Challenge to Reduce the Data Center Carbon Footprint. [Online]. Available: http://www.datacenterknowledge.com/archives/2014/12/17/undertaking-challenge-reduce-data-center-carbon-footprint

[2] Y. Guo, Z. Ding, Y. Fang, and D. Wu, "Cutting down electricity cost in internet data centers by using energy storage," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Dec 2011, pp. 1–5.

[3] X. Lu, F. Kong, J. Yin, X. Liu, H. Yu, and G. Fan, "Geographical job scheduling in data centers with heterogeneous demands and servers," in *2015 IEEE 8th International Conference on Cloud Computing*, June 2015, pp. 413–420.

[4] A. N. Toosi and R. Buyya, "A fuzzy logic-based controller for cost and energy efficient load balancing in geo-distributed data centers," in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, Dec 2015, pp. 186–194.

[5] A. H. Mahmud and S. S. Iyengar, "A distributed framework for carbon and cost aware geographical job scheduling in a hybrid data center infrastructure," in *2016 IEEE International Conference on Autonomic Computing (ICAC)*, July 2016, pp. 75–84.

[6] J. Srinivas, A. A. M. Qyser, and B. E. Reddy, "Exploiting geo distributed datacenters of a cloud for load balancing," in *2015 IEEE International Advance Computing Conference (IACC)*, June 2015, pp. 613–616.

[7] M. M. S. Maswood, C. Develder, E. Madeira, and D. Medhi, "Dynamic virtual network traffic engineering with energy efficiency in multi-location data center networks," in *2016 28th International Teletraffic Congress (ITC 28)*, vol. 01, Sept 2016, pp. 10–17.

[8] O. Dobrijevic, A. J. Kassler, L. Skorin-Kapov, and M. Matijasevic, *Q-POINT: QoE-Driven Path Optimization Model for Multimedia Services*. Springer International Publishing, 2014, pp. 134–147. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-13174-0_11

[9] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "Hula: Scalable load balancing using programmable data planes," in *Proceedings of the Symposium on SDN Research*, ser. SOSR '16. New York, NY, USA: ACM, 2016, pp. 10:1–10:12. [Online]. Available: http://doi.acm.org/10.1145/2890955.2890968

[10] A. Imamoto and B. Tang, "A recursive descent algorithm for finding the optimal minimax piecewise linear approximation of convex functions," in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, Oct 2008, pp. 287–293.

[11] M. A. Owens and D. Medhi, "Temporal bandwidth-intensive virtual network allocation optimization in a data center network," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3493–3497.

[12] R. Fourer, D. M. Gay, and B. W. Kernighan, "A modeling language for mathematical programming," *Manage. Sci.*, vol. 36, no. 5, pp. 519–554, May 1990. [Online]. Available: http://dx.doi.org/10.1287/mnsc.36.5.519

[13] CPLEX 12, IBM ILOG CPLEX. [Online]. Available: User'sManual, [Online].Available:http://gams.com/dd/docs/solvers/cplex.pdf.

[14] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[15] X. Xiang, C. Lin, F. Chen, and X. Chen, "Greening geo-distributed data centers by joint optimization of request routing and virtual machine scheduling," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, Dec 2014, pp. 1–10.

[16] A. Basta, A. Blenk, M. Hoffmann, H. J. Morper, K. Hoffmann, and W. Kellerer, *SDN and NFV Dynamic Operation of LTE EPC Gateways for Time-Varying Traffic Patterns*. Springer International Publishing, 2015, pp. 63–76. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16292-8_5

[17] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges," *Journal of Grid Computing*, vol. 14, no. 2, pp. 217–264, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10723-015-9359-2

[18] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09. New York, NY, USA: ACM, 2009, pp. 123–134. [Online]. Available: http://doi.acm.org/10.1145/1592568.1592584

[19] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Proceedings of the 29th Conference on Information Communications*, ser. INFOCOM'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 1145–1153. [Online]. Available: http://dl.acm.org/citation.cfm?id=1833515.1833689

[20] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 1431–1439.

[21] Y. Zhang, Y. Wang, and X. Wang, "Electricity bill capping for cloud-scale data centers that impact the power markets," in *2012 41st International Conference on Parallel Processing*, Sept 2012, pp. 440–449.

[22] M. M. S. Maswood and D. Medhi, "An adaptive allocation scheme for load balancing and sla maintenance in multi-location data center networks," Under Submission.