Data Center Network Design for Internet-Related Services and Cloud Computing

(Authors' names blinded for peer review)

Data center networks provide the physical infrastructure that hosts Internet-related services and cloud computing. Designing data center networks properly is imperative for Internet-related service and cloud computing providers to gain competitive edges through cost efficiency and service quality. In this paper, we formulate a mathematical programming model to address the data center network design problem, in which the objective is to minimize total operating cost and the service delay penalty by optimizing data center location, footprint allocation, and resource provisioning decisions, while incorporating essential features, such as latency, power, multiple resources, configuration limits, and interdependent footprints. We employ a queueing model to approximate the service latency and provide tractable reformulations. To enhance computational efficiency for large-scale problems, we further develop Lagrangian relaxation methods and generate strengthening cuts by exploiting the structural properties of the problem. Our numerical studies demonstrate that the proposed model, which jointly optimizes location, allocation, and resource provisioning, can achieve significant cost reductions and improvements in service quality compared with a hierarchical approach that optimizes these decisions sequentially. Moreover, our proposed solution methods outperform state-of-the-art commercial software in terms of computational efficiency. Based on real-world datasets, the proposed model selects data centers that have been chosen by major cloud computing infrastructure providers. We also draw managerial insights that can be used as design guidelines in practice.

Key words: Data centers; Cloud computing; Service operations; Network design; Resource provisioning

1. Introduction

Over the past two decades, the booming demand for Internet-related services and cloud computing has called for massive infrastructure construction of warehouse-scale computer systems hosted in data centers. Data center investments by major service providers in North America have amounted to more than $20 billion in 2017 (CBRE 2018). In the recent 2018Q3 earnings report, Alphabet Inc. (the parent company of Google) reported that the nine-month capital expenditures of the firm more than doubled year-over-year to $18.6 billion (compared to a 25% increase in operating expenses), while the bulk of it went to data center constructions (https://abc.xyz/investor/pdf/20181025_alphabet_10Q.pdf).

Nourished by exploding market opportunities, Internet giants, including Google, Amazon, Facebook, and Microsoft, have started to open their data centers to provide cloud computing services to businesses ranging from small startups to unicorn companies, such as Uber, or even multi-billion

companies, such as Netflix. Intense competition pushes infrastructure providers to offer services with better quality and cost effectiveness. To achieve this goal, infrastructure providers indubitably need better data center network designs.

Data center network design in the Internet and cloud computing industry is no less crucial than supply chain and logistics network design in the manufacturing and retailing industry. However, the latter has received far more attention from academia compared with the former. Although data center network and supply chain network designs share many common features—for example, both have large and expensive facilities and operational costs that are sensitive to facility location and the distance between facilities and demand points—data center network design is very different from and more challenging than supply chain network design. First, unlike traditional distribution centers, data centers consume a tremendous amount of power (Vaughan 2016). As a result, the location-dependent costs of power significantly impact data center network designs. Second, in data center networks, there exist multiple types of resources that share a limited power capacity. Furthermore, different demand points may have heterogeneous requirements for the variety of resource types. Consequently, the resource provisioning and demand allocation problem is significantly more challenging in data center networks than in supply chain networks. Third, two types of delays may occur in data center networks, namely, the delay within a center (i.e., endhost latency) and the delay between the demand and its assigned center (i.e., network fabric latency). The endhost latency depends nonlinearly on resource provisioning and demand allocation decisions and is as critical as network fabric latency. By contrast, supply chain network design usually focuses on transit times (which is analogous to network fabric latency) but ignores the processing times within a distribution center (which is analogous to endhost latency). In addition to the challenges mentioned above, many data center networks have special features that further complicate the design problem, for instance, there can be interdependent footprints as required by synchronization or redundancy. In this case, a job at one data center will initiate another job at a different data center with a certain probability, thereby adding extra workload and traffic between data centers. Another feature is the existence of nonlinear dependencies between power consumption and average workload, as well as between fabric latency and average traffic.

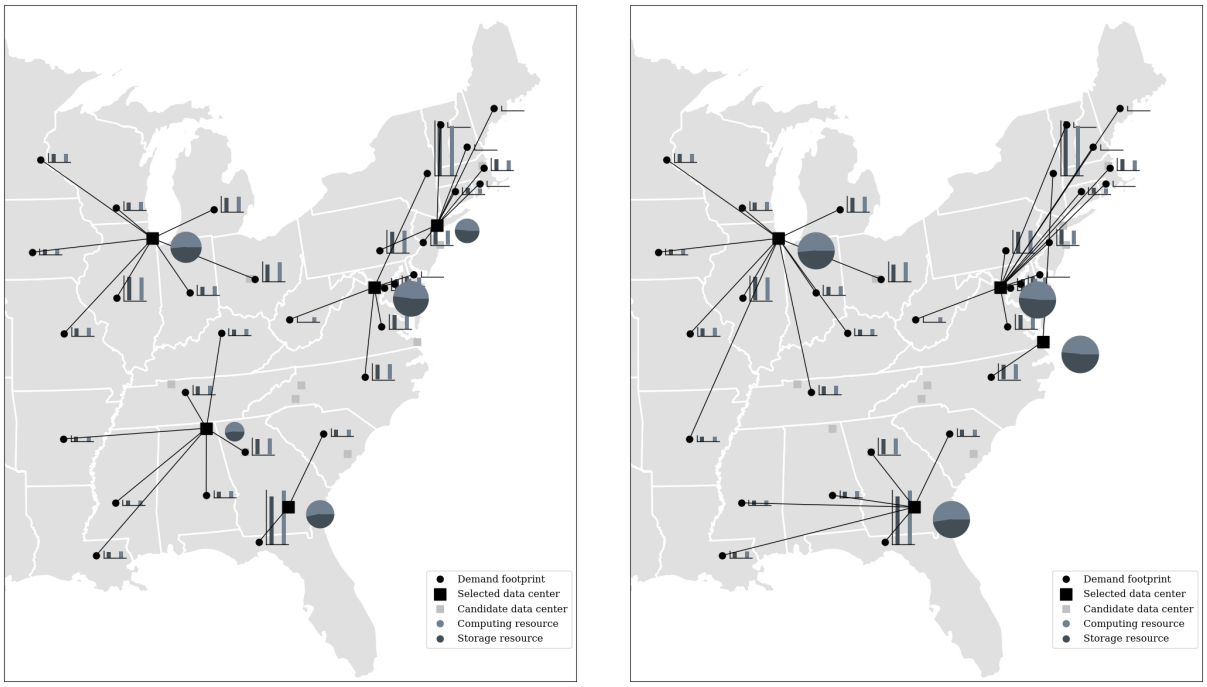
Failing to recognize any of the aforementioned ingredients can result in decisions that are highly suboptimal and undesirable. Therefore, in this paper, we aim to provide a decision-making tool for the data center network design problem, which is concerned with decisions such as the location of data centers, the allocation of computing footprints, and the provisioning (i.e., capacity planning, or the allocation of different resources to satisfy demands) of computing and storage resources. We focus on the objective of minimizing total cost and latency penalties. We remark that this idea of using models that jointly optimize strategical and operational decisions is not new. Daskin

et al. (2002) and Shen et al. (2003) were the first to propose joint location-inventory models with nonlinear safety stock levels to provide both location and inventory-related decisions. In a similar vein, a variety of integrated models have been proposed in the literature, see, for example, Geunes et al. (2011) in supply chain management, Taaffe et al. (2008) in marketing, and Geunes and Pardalos (2003) for a review of similar models applied in supply chain management and financial engineering.

The model we present in this paper is motivated by the authors' experience working with one of the leading cloud computing infrastructure providers in the world. Given the complexity of the problem, the current practice of data center network design resorts to simplification techniques, such as the hierarchical approach. Specifically, the design problem is decomposed into multiple sequential stages for tractability at the cost of suboptimality. For example, the locations of new data centers of the aforementioned cloud infrastructure provider are determined separately based on certain factors, such as local electricity prices, temperature (which affects the need for cooling), and access to renewable energy sources. However, other factors such as footprint allocation, resource provisioning, and quality of service, are not considered. Subsequently, the requirement for resources, such as central processing units (CPUs) and storage, is estimated from demand forecasts and allocated to different locations according to the available power capacities. We refer the readers to Barroso et al. (2013) for details of the current practice. By contrast, the models presented in this paper are highly integrated, thereby making them more suitable for data center network design compared with most existing models.

We use a brief numerical example to demonstrate the necessity and efficacy of our model. Suppose one needs to choose from a set of candidate locations to build data centers serving a set of users. By applying our model, one can jointly minimize total cost and latency. Alternatively, we consider a benchmark model that resorts to a hierarchical approach by first solving a reduced model to obtain the facility location and allocation decisions without considering endhost latency. Then, the benchmark model solves for the resource provisioning decisions to minimize the endhost latency and power cost given the selected data centers and demand allocation. The detailed formulation of the hierarchical model is provided in Online Appendix EC.3.1. The location, allocation, and resource provisioning results given by the benchmark and the proposed models are illustrated in Figure 1. The demand mix for different types of resources varies by location, as depicted by the bar charts next to the demand points. We use pie charts to illustrate the amount of provisioned resources at each data center, and the sizes of these pie charts reflect the total power consumption. The benchmark model (left panel) not only uses one more data center than the proposed model, but also incurs more than double the endhost latency cost, thereby resulting in a total cost that is 33% higher compared to the proposed model (right panel). These results suggest that the proposed

model can be of great value in the long term in a competitive market by striking a balance between cost efficiency and service quality. We also numerically verify that the resulting strategic decisions are robust with respect to changing parameters, such as demand growth. In the rest of this section, we briefly review the related literature and summarize our contributions by comparing our study with existing research on the topic.



(a) Benchmark hierarchical model (b) Proposed integrated model

Figure 1 **Comparison of data center network designs from two models. The proposed integrated model saves more than half in endhost latency and cuts a quarter of the total cost.**

1.1. Related Literature and Contribution

The importance of data center network design has been recognized by both researchers and practitioners. Iyoob et al. (2013) review the challenges and opportunities in cloud computing and data center operations management and propose a hierarchy of important operational problems from the service provider's perspective, with data center location and capacity planning (i.e., resource provisioning) being the most fundamental problems in this hierarchy. Greenberg et al. (2008) also identify data center placement (i.e., location) and sizing (i.e., resource provisioning) as being among the most important and challenging problems in data center management. This problem has recently attracted attention from the computer science community (see, for example, Larumbe and Sansò (2012) and Larumbe and Sansò (2013), and the references therein), with most of the research

considering the limited operational features of data center networks and relying on computationally cumbersome methods that generate local optimal solutions.

A related stream of research that has recently attracted growing attention from the academic community has focused on the resource management problem of cloud computing services. The focal problem in this stream is to improve operational efficiency by optimizing server-workload and service allocations under certain service level agreement (SLA) constraints (see, for example, Goudarzi et al. (2012) and Verma et al. (2008), and the references therein). Meanwhile, our work is concerned with the problem at the data center level instead of the server level and considers additional decisions, such as resource provisioning and colocation. The proposed model also incorporates detailed modeling on both fabric and endhost latencies, introduces nonlinearity in power consumption, and relaxes assumptions to achieve improved model accuracy.

Within the OR/MS literature, our model falls into the category of facility location problems with congestion and immobile service resources. As a major departure from conventional models, facility location problems with congestion and immobile service resources assume that facilities employ limited resources to fulfill those demands with stochastic service times. We refer to Berman and Krass (2015) for an excellent review and classification of literature in this area. Following their classification, our model falls into the category where customers do not control the demand allocation and where the demand rates are fixed. Along this line of research, our model combines the costs borne by both the customers and the data center network to strike a balance between the two. Our work contributes to the rapidly growing research on facility location problems with congestion and immobile service resources by proposing a novel model that adapts to the aforementioned challenges introduced by data center network designs, such as the consideration of power cost as a function of utilization, capacitated resource provisioning, fabric and endhost latencies and their dependencies on resource provisioning, and interdependence between demand footprints. This paper also contributes to the literature by exploring the structural properties of the model to provide efficient and effective solution approaches.

In particular, our model is closely related to those with *balanced objectives* as surveyed by Berman and Krass (2015), including Wang et al. (2004), Elhedhli (2006a), Aboolian et al. (2008), Castillo et al. (2009), Zhang et al. (2009, 2010), Abouee-Mehrizi et al. (2011), and Paraskevopoulos et al. (2016), among others. For instance, Zhang et al. (2009) optimize the location decisions in a preventive healthcare facility network to maximize client participation in the presence of congestion, which is modeled with $M/M/1$ queues. Zhang et al. (2010) study the same problem while incorporating additional capacity allocation decisions; that is, they model congestion at facilities with $M/M/s$ queues and use the number of servers s as a decision variable. Compared with other studies in this area, our model differs mainly in the following aspects. First, we adopt a queueing

network model with the processor sharing scheduling discipline, which better fits the application of Internet-related services and cloud computing, and enables us to model multiple resource types and interdependent footprints. Second, in addition to endhost congestion (i.e., congestion at the facilities), we also consider network congestion and other crucial features to achieve more accurate modeling of data center operations. Incorporating these features in existing models can introduce significant challenges in both modeling and computation. Third, we provide efficient mixed-integer second-order cone program (MISOCP) reformulations and develop tailored solution methods based on the structural properties of the problem, whereas most existing studies depend on more general methods such as linearization (e.g., Aboolian et al. 2007), column generation (e.g., Elhedhli 2006b), or heuristics (e.g., Zhang et al. 2009, 2010), to solve nonlinear optimization problems.

Recently, MISOCP has been widely applied to solve many traditional or emerging OM problems, including robust facility location with production decisions (Baron et al. 2011), integrated supply chain network design (Atamtürk et al. 2012), electric vehicle battery swapping infrastructure planning (Mak et al. 2013), appointment scheduling (Kong et al. 2013, Mak et al. 2015, Kong et al. 2017), joint planning of energy storage and transmission for wind power generation (Qi et al. 2015), vehicle sharing system design (He et al. 2017, 2018), and assortment optimization (Sen et al. 2017). Most of these models focus on modeling demand variance/covariance or probabilistic constraints. We extend the application domain to the congestion of stochastic service systems. Moreover, instead of directly solving MISOCP, we develop efficient Lagrangian relaxation solution approaches.

Our main contributions are summarized as follows:

1. We propose a novel data center network design model (Section 2) where we consider contributory factors of great practical value, including multiple resource types, heterogeneous demand mix, fabric and endhost latency, resource ratio constraints induced from limited server configurations, colocation, interdependent footprints, convex power consumption, and network congestion. To the best of our knowledge, this is the first model that jointly considers these factors, thereby making this model appropriate for practical data center network designs.

2. We introduce non-trivial MISOCP reformulations to the proposed models (Section 3.1) and develop Lagrangian relaxation solution approaches (Section 3.2) enhanced with extended extremal polymatroid cuts (Section 3.3). Our numerical results indicate that these approaches outperform commercial software in terms of computational efficiency (Section 4.4), thereby making the proposed models and the solution approaches valuable to firms expanding their current data center networks (see Miller 2018, for a Facebook example) or starting to build data center networks (see Sverdlik 2018, for an example where Oracle plans to launch its own data centers in new regions).

3. We conduct numerical studies by using several real-world datasets to address important design questions and draw the following design guidelines (Section 4.1): (a) high fabric latency costs always result in more data centers, whereas high endhost latency costs can lead to either fewer or more data centers, depending on the power capacities and resource ratio constraints; (b) colocation is preferred for regions with a moderate demand volume, low power cost, and long distance from regular sites; and (c) a high level of interdependence between footprints leads to data centers being located closer to one another in terms of demand-weighted fabric latency. These numerical studies also demonstrate the importance of integrated modeling in data center network design.

2. Model and Formulation

In this section, we present an optimization model for designing the data center backbone network of a large Internet-related services or cloud computing company. First, we formulate the base model that determines data center locations, demand allocation, and resource provisioning. Second, we introduce a queueing model to approximate the endhost latency cost. Finally, we study several model extensions.

2.1. Base Model

We assume that demands originate from a set $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$ of nodes, also known as demand footprints, or *footprints* for brevity. Each footprint $i \in \mathcal{I}$ can be considered as an aggregated user of all services (e.g., email, social networking, and video streaming) provided by the company within a geographical region. Assume that these demands are encapsulated in packets (also known as service requests or jobs) that arrive according to independent Poisson processes with an arrival rate of d_i for footprint $i \in \mathcal{I}$. Processing a job requires a set of resources denoted by $\mathcal{K} = \{1, \ldots, |\mathcal{K}|\}$, with each resource serving a specific need. For instance, CPUs serve computing needs, whereas disk drives provide storage iput/output (I/O). For ease of exposition, we assume $|\mathcal{K}| = 2$ throughout this paper and directly refer to resources as computing and storage, although the generalization to more resource types is straightforward. We refer to the amounts of resources required by a job as its job size, and assume that average job sizes are known. Let u_{ik} denote the average amount of resource $k \in \mathcal{K}$ required to fulfill a job from footprint i. The mix between the required amounts of different resources, also known as the shape of a footprint, can vary across different footprints.

The network design problem consists of the following decisions. We select locations to build data centers from a set $\mathcal{J} = \{1, \ldots, |\mathcal{J}|\}$ of candidate locations. Let $\mathbf{x} = (x_1, \ldots, x_{|\mathcal{J}|})$ denote the vector of data center locations, where $x_j = 1$ if a data center is built at location j, and $x_j = 0$ otherwise. The customers' service requests (jobs) are routed to (and back from) data centers via backbone networks. We assume that each footprint is assigned to one of the available data centers as its

dedicated site. Let $\mathbf{y}_{\cdot j} = (y_{1j}, \ldots, y_{|\mathcal{I}|j})$ denote the footprint assignment decision to data center j, with $y_{ij} = 1$ if footprint i is assigned to data center j, and $y_{ij} = 0$ otherwise.

We also decide the levels of resource provisioning. Let $\mathbf{z}_{j\cdot} = (z_{j1}, \ldots, z_{j|\mathcal{K}|})$ denote the resource provisioning decision, where z_{jk} is the amount of resource k built at data center j. The total amount of resources that can be built at data center j is constrained by its power capacity p_j and by the minimum and maximum feasibility ratios between resources. The latter is a common constraint in practice because resources are typically deployed via a limited number of machine configurations. Let $\bar{r}_{kl}$ denote the maximum ratio between resources k and l that can be deployed at the same site.

We consider minimizing the following costs. Building a data center at candidate site j incurs fixed cost f_j, which can be measured by its annual amortization. Meanwhile, consuming electricity at data center j incurs power costs at rate c_j. The site-dependent unit power cost not only represents the geographical difference in electricity price but also subsumes different cooling and maintenance needs, and other site-dependent costs. To incorporate service quality into the model, we consider both network fabric and endhost latency costs, which can be jointly considered along the same lines as in Shen et al. (2003) and Berman and Krass (2015), etc. Let t_{ij} denote the unit network fabric latency cost between footprint i and let data center j, and $L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$ denote the total endhost latency costs incurred at data center j given footprint allocation $\mathbf{y}_{\cdot j}$ and resource provisioning $\mathbf{z}_{j\cdot}$. We summarize the major notations in Table 1.

Table 1 List of main notations.

Notation	Description
$\mathcal{I}$	Set of demand footprints, $\mathcal{I}$
$\mathcal{J}$	Set of candidate data center sites, $\mathcal{J}$
$\mathcal{K}$	Set of resources, $\mathcal{K}$
d_i	Footprint i's demand arrival rate
u_{ik}	Unit requirement of resource k for footprint i
p_j	Site j's power capacity
c_j	Site j's unit power cost
f_j	Fixed cost of building a data center at candidate site j
t_{ij}	Unit network fabric latency cost between footprint i and data center j
$t_{jj'}$	Unit network fabric latency cost between candidate data centers j and j'
τ_i	Unit endhost latency cost for footprint i
w_k	Peak power consumption per unit of resource k
α	Average proportion of peak power consumed
$\bar{r}_{kl}$	Maximum ratio between resources k and l at the same site
x_j	Whether data center j is built
y_{ij}	Whether footprint i is assigned to center j
z_{jk}	Total amount of resource k built at center j

We also model the use of *micro colocation data centers* (also known as Colo). In practice, service providers may contract with local providers for colocations to serve users from a specific region

to avoid enormous capital expenditures or to cut fabric latencies (Greenberg et al. 2008). To incorporate colocation decisions, we add a corresponding set of decision variables $\{\tilde{\mathbf{x}}, \tilde{\mathbf{z}}\}$. We use $\tilde{\cdot}$ for parameters and decision variables associated with colocations hereinafter. We assume that colocation only serves local demand and is available in all locations. Therefore, $\tilde{x}_i = 1$ if footprint i is served by its own colocation. Similar to the fixed costs of locating data centers, the costs associated with colocations can be measured in either annual amortization or rent, and thus can be jointly considered with other costs in the objective functions. Note that to avoid an *all-or-nothing* colocation decision for any footprint i, we can further partition i's demand into several subsets and determine the colocation decision for each subset.

The *base* model for the data center network design is formulated as follows:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z},\tilde{\mathbf{x}},\tilde{\mathbf{z}}} \quad \underbrace{\sum_{j\in\mathcal{J}} f_j x_j + \alpha \sum_{j\in\mathcal{J},k\in\mathcal{K}} c_j w_k z_{jk}}_{\text{fixed and power costs at data centers}} + \underbrace{\sum_{i\in\mathcal{I},j\in\mathcal{J}} d_i t_{ij} y_{ij} + \sum_{j\in\mathcal{J}} L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})}_{\text{fabric and endhost latency costs}}$$
$$+ \underbrace{\sum_{i\in\mathcal{I}} \tilde{f}_i \tilde{x}_i + \alpha \sum_{i\in\mathcal{I},k\in\mathcal{K}} \tilde{c}_i w_k \tilde{z}_{ik} + \sum_{i\in\mathcal{I}} L_i(\tilde{x}_i, \tilde{\mathbf{z}}_{i\cdot})}_{\text{costs associated with colocations}} \tag{1a}$$

$$\text{s.t.} \quad y_{ij} \le x_j, \quad \forall i \in \mathcal{I},\ j \in \mathcal{J} \tag{1b}$$

$$\sum_j y_{ij} + \tilde{x}_i \ge 1, \quad \forall i \in \mathcal{I} \tag{1c}$$

$$\sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij} \le z_{jk}, \quad \forall j \in \mathcal{J},\ \forall k \in \mathcal{K} \tag{1d}$$

$$\sum_{k\in\mathcal{K}} w_k z_{jk} \le p_j, \quad \forall j \in \mathcal{J} \tag{1e}$$

$$z_{jk} \le \bar{r}_{kl} z_{jl}, \quad \forall j \in \mathcal{J},\ \forall k,l \in \mathcal{K} \tag{1f}$$

$$d_i u_{ik} \tilde{x}_i \le \tilde{z}_{ik}, \quad \forall i \in \mathcal{I},\ \forall k \in \mathcal{K} \tag{1g}$$

$$\sum_k w_k \tilde{z}_{ik} \le \tilde{p}_i, \quad \forall i \in \mathcal{I} \tag{1h}$$

$$\tilde{z}_{ik} \le \bar{r}_{kl} \tilde{z}_{il}, \quad \forall i \in \mathcal{I},\ \forall k,l \in \mathcal{K} \tag{1i}$$

$$x_j, \tilde{x}_i, y_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J} \tag{1j}$$

$$z_{jk} \ge 0, \tilde{z}_{ik} \ge 0 \quad \forall j \in \mathcal{J},\ \forall i \in \mathcal{I},\ \forall k \in \mathcal{K}, \tag{1k}$$

where w_k is the peak power consumption per unit of resource k, and parameter α is the average proportion of peak power consumed. The objective (1a) consists of the fixed cost for building data centers, average power cost, network fabric, and endhost latency costs. In the base model, we assume that the power cost is proportional to the average power consumption at a data center. Later, we will relax this assumption by allowing the power cost to be dependent on the workload. Constraints (1b) and (1c) ensure that each footprint is assigned to one opened data center. Constraint (1d) and

(1g) ensure that the total amount of required resources is less than or equal to the total resource provisioning. Constraint (1e) and (1h) ensure that the total power provisioned is sufficient for the deployed resources. Constraint (1f) and (1i) define the maximum (and minimum) ratios between any pair of resource types.

We present above a static model in which no data center exists and a new network is designed from scratch. In practice, a data center network may need to be dynamically expanded and/or redesigned over time. Dynamic location models are generally much harder to solve, and their applicability is further limited by the high level of uncertainty over the long planning horizon. Nonetheless, our model can be modified and extended to address one-time network expansion/redesign problems for an existing network. Some additional model features need to be incorporated, including the closing of existing data centers, the moving or recycling of deployed servers, and the migration of computation footprints. Such a model can then be applied in a rolling horizon manner to provide plans for network expansion and reconfiguration.

From the modeling perspective, all components associated with colocation show no structural difference compared to those with data centers. Therefore, for clarity, we omit colocation decisions in the subsequent analyses, but will revisit colocation in numerical studies. In the next subsection, we derive the endhost latency cost term $L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$ in the objective function.

2.2. Endhost Latency Cost

Queueing models have been widely applied in modeling the performance of computer systems (see Harchol-Balter 2013). In this subsection, we measure endhost latency using the mean sojourn time experienced by jobs at data centers and analyze a queueing model to approximate endhost latency costs.

To facilitate the analysis, we make the following assumptions. First, we assume that data centers operate under the *processor sharing* discipline, such that any incoming job is immediately dispatched to be processed and the total capacity is shared equally among all jobs present. Processor sharing has been among the most widely used paradigms in the cloud computing literature (e.g., Altman et al. 2011). In practice, industry leaders such as Google have also been using management systems that process jobs in a processor sharing manner (Verma et al. 2015, Silberschatz et al. 1998). More advanced scheduling and priority policies can be used to shorten latency time (Pinedo 2016, Wan et al. 2007), however, in a strategic planning model, we do not model the detailed scheduling of jobs, but rather use the processor sharing discipline to provide a tractable approximation of the expected endhost latency.

Recall that processing a job requires several types of resources, and a job seizes one type of resource at a time. For example, almost all programs have alternating cycles of computing and

storage I/O (Silberschatz et al. 1998). We differentiate job processing with different resources as separate *service stages*, and further assume that the service stages form a tandem queue. Consequently, the total service time is defined as the spread from the start of the computing stage to the end of the storage stage. We also assume that the service times of a job at different stages are independent. In Online Appendix EC.1.1, we generalize the tandem queue setting where each job joins a *open queueing network* consisting of a computing and a storage service stage. Specifically, the job is randomly routed through the service stages in an alternating fashion before eventually departing from the network. For clarity, we still follow the tandem queue assumption in the rest of the paper, and all theoretical results and proposed solution approaches apply to the general case.

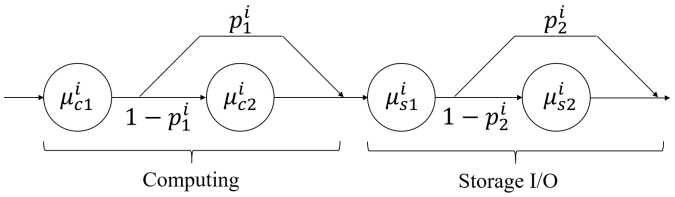


Figure 2 **An approximation of mean sojourn time by a tandem queue of two service stages with Coxian-2 distributed stage service times.**

We model demand heterogeneity by using multiple customer classes with different mean service times. Suppose for the moment that data center j serves only footprint i. Given that the amount of resource k at data center j is z_{jk}, the mean service rate for a job originating from footprint i in stage k of data center j is z_{jk}/u_{ik}. We further assume that the *stage service time* follows the *Coxian* distribution. Figure 2 illustrates an approximation of the sojourn time at a data center with a tandem queue of two stages, where the service time of each stage is further approximated by a Coxian-2 distribution. Parameters $(\mu_{c1}^i, \mu_{c2}^i, p_c^i)$ and $(\mu_{s1}^i, \mu_{s2}^i, p_s^i)$ of the Coxian-2 distributions are chosen such that the first three moments of the stage service times match those of the approximations. These parameters are functions of resource provisioning decisions $\mathbf{z}$ and are useful in deriving the queueing metrics as explained in details in Online Appendix EC.1.1. This assumption is non-restrictive as the Coxian distribution can approximate by moment matching an arbitrary distribution, provided that this distribution has a rational Laplace-Stieltjes transformation (Cox 1955).

Under the above assumptions, each data center consists of a tandem queue with $|\mathcal{K}|$ stages, and each stage forms a multi-class single-server processor sharing queue with Coxian service times. In Proposition 1, we provide the class-specific mean sojourn time used to approximate the endhost latency cost. It is noteworthy that the subsequent analyses can be generalized to the case with alternative occupancy of different resources by using the broader class of phase-type distributions.

PROPOSITION 1. *(Mean Sojourn Time) The average time spent by a job at data center j, $W_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$, is given by*

$$W_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}) = \sum_{k\in\mathcal{K}} \frac{\mathbb{E}[S_{jk}]}{1-\lambda_j \mathbb{E}[S_{jk}]},$$

where $\mathbb{E}[S_{jk}] = \frac{\sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij}}{z_{jk} \sum_{i\in\mathcal{I}} d_i y_{ij}}$ denotes the expected service time of a job at stage k of data center j averaged over all demand classes allocated to j, and $\lambda_j = \sum_i d_i y_{ij}$ denotes the demand arrival rate at data center j.

Moreover, the sojourn time of a job originating from footprint i at data center j, $W_j^i(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$ satisfies

$$W_j^i(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}) = \sum_{k\in\mathcal{K}} \frac{u_{ik} y_{ij}}{z_{jk}(1-\lambda_j \mathbb{E}[S_{jk}])}.$$

Proposition 1 implies the intuitive result that the mean sojourn time of a job at a service stage of a data center is inversely proportional to the amount of provisioned resource for that service stage, and the percentage of time that the service stage is empty at the data center. The proof of Proposition 1 stems from the seminal work of Baskett et al. (1975). We also briefly discuss the generalization of Proposition 1 for all phase-type distributed service times at the end of Online Appendix EC.1.1.

The second half of Proposition 1 enables the calculation of endhost latency costs for heterogeneous customers. In particular, we consider user heterogeneity in latency sensitivity, which is measured by the unit endhost latency cost denoted by τ_i. The total expected endhost latency cost at data center j, $L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$, given footprint allocation $\mathbf{y}_{\cdot j}$ and resource provisioning $\mathbf{z}_{j\cdot}$, can be described as follows:

$$L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}) = \sum_i \tau_i d_i W_j^i(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}) = \sum_{k\in\mathcal{K}} \frac{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}{z_{jk} - \sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij}}. \tag{2}$$

By substituting (2) into the base model, we obtain the following data center network design model with heterogeneous users:

$$\textbf{(P)} \quad \min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad \sum_{j\in\mathcal{J}} f_j x_j + \alpha \sum_{j\in\mathcal{J},k\in\mathcal{K}} c_j w_k z_{jk} + \sum_{i\in\mathcal{I},j\in\mathcal{J}} d_i t_{ij} y_{ij} + \sum_{j\in\mathcal{J},k\in\mathcal{K}} \frac{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}{z_{jk} - \sum_i d_i u_{ik} y_{ij}}$$
$$\text{s.t.} \quad \text{Constraints (1b)} - \text{(1f)}, \ \text{(1j)} - \text{(1k)}.$$

Problem **(P)** is a mixed-integer nonlinear programming (MINLP) problem owing to the linear-fractional term in the objective function. The problem is clearly NP-hard by reducing from the Uncapacitated Facility Location problem. In Online Appendix EC.2, we discuss the approximability of the proposed model. Before introducing the solution approach, we illustrate several important model extensions in the following subsections.

2.3. Interdependent Footprints

We first consider the interdependence between footprints. In cloud computing, jobs occasionally need to be rerouted to different data centers from their designated ones for further services. Such routing may result from the need for synchronization, redundancy, or additional data access among footprints (Greenberg et al. 2008).

Without loss of generality, we assume that a proportion of jobs, upon service completion at their designated data centers, are routed to another data center for additional service and then leave the system. Let $P_{ii'}$ be the probability that a job from footprint i is routed to footprint i''s dedicated data center for further services. Thus, the demand arrival rate at data center j due to demands originating from footprint i is given by

$$\lambda_{ij} = d_i y_{ij} + \sum_{i' \neq i} d_i P_{ii'} y_{i'j} \triangleq d_i \sum_{i'} P_{ii'} y_{i'j}, \tag{3}$$

where $P_{ii} \triangleq 1$, for all $i \in \mathcal{I}$. Note that $P_{ii'}$ differs from the routing probability in queueing networks in that $\sum_{i' \neq i} P_{ii'}$ can be different from one. For example, a job from footprint i may need to be further processed simultaneously at multiple data centers serving several other footprints, and thus $\sum_{i' \neq i} P_{ii'} > 1$.

For each job that requires additional service, two extra costs are incurred, namely, the extra endhost latency cost associated with the additional service and the extra fabric latency cost between data centers when the additional service is processed at a different data center. The expected endhost latency cost at data center j with heterogeneous latency cost is then given by

$$L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}) = \sum_k \frac{\sum_i \left(\sum_{i'} \tau_{i'} d_{i'} u_{i'k} P_{i'i}\right) y_{ij}}{z_{jk} - \sum_i \left(\sum_{i'} d_{i'} u_{i'k} P_{i'i}\right) y_{ij}}. \tag{4}$$

Given that the traffic intensity of footprint i's jobs from data center j to data center j' is $d_i y_{ij} \sum_{i'} P_{ii'} y_{i'j'}$, the total fabric latency cost associated with traffic between data centers is given by

$$\sum_{i,i' \in \mathcal{I}} \sum_{j,j' \in \mathcal{J}} \psi_i t_{jj'} d_i P_{ii'} y_{ij} y_{i'j'} \ , \tag{5}$$

where $t_{jj'}$ is the network latency cost from data center j to j', and ψ_i is the scaling factor for the latency cost associated with the rerouting of footprint i. The model can be readily generalized to allow for job size changes after rerouting. To illustrate, let $\Gamma_{ii'}$ be the scaling factor associated with job traffic from footprint i to i' with $\Gamma_{ii} = 1$; job size changes only affect the extra endhost latency cost terms as follows:

$$L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}) = \sum_k \frac{\sum_i \left(\sum_{i'} \tau_{i'} d_{i'} u_{i'k} P_{i'i} \Gamma_{i'i}\right) y_{ij}}{z_{jk} - \sum_i \left(\sum_{i'} d_{i'} u_{i'k} P_{i'i} \Gamma_{i'i}\right) y_{ij}}.$$

We omit job size changes for notation brevity. By adding (5) to the objective function of problem **(P)** and by substituting the endhost latency term with (4), we obtain the following model for data center network design that considers traffic between data centers:

$$\begin{aligned}
\textbf{(P-ID)} \quad \min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \sum_{j\in\mathcal{J}} f_j x_j + \alpha \sum_{j\in\mathcal{J},k\in\mathcal{K}} c_j w_k z_{jk} + \sum_{i\in\mathcal{I},j\in\mathcal{J}} d_i t_{ij} y_{ij} + \sum_{i,i'\in\mathcal{I}} \sum_{j,j'\in\mathcal{J}} \psi_i t_{jj'} d_i P_{ii'} y_{ij} y_{i'j'} \\
& + \sum_{j\in\mathcal{J},k\in\mathcal{K}} \frac{\sum_i \left(\sum_{i'} \tau_{i'} d_{i'} u_{i'k} P_{i'i}\right) y_{ij}}{z_{jk} - \sum_i \left(\sum_{i'} d_{i'} u_{i'k} P_{i'i}\right) y_{ij}} \\
\text{s.t.} \quad & \sum_i \Big(\sum_{i'} d_{i'} u_{i'k} P_{i'i}\Big) y_{ij} \le z_{jk}, \quad \forall j \in \mathcal{J}\ , k \in \mathcal{K} \qquad (6) \\
& \text{Constraints } (1b) - (1c),\ (1e) - (1f),\ (1j) - (1k).
\end{aligned}$$

Problem **(P-ID)** is a generalization of the single-allocation hub location problem, and is a computationally more challenging MINLP.

2.4. Convex Power Consumption in Utilization

Studies have shown that a higher utilization rate at data centers may lead to a lower power usage effectiveness (PUE). Consequently, power consumption is typically increasing and convex in utilization (Chen et al. 2013). To capture the decrement in power efficiency as utilization increases, we assume that the average power consumption per unit of resource k at data center j, denoted by ω_{jk}, is a general convex function in the form of a sum of power functions—also known as an exponential polynomial—of the utilization (load) ρ_{jk}, where $\rho_{jk} = \sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij} / z_{jk}$. In other words, let

$$\omega_{jk} = \sum_l a^l_{jk} \rho^{\sigma^l}_{jk} + b_{jk}, \quad \rho_{jk} = \sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij} / z_{jk}, \qquad (7)$$

where a^l_{jk}, b_{jk}, and σ^l are the parameters that can be estimated from historical power consumption, and σ^l can be any rational number greater than or equal to one (σ^l can also vary across different data centers and resources, and the polynomial function becomes a special case when σ^l is a positive integer). In addition, we assume $a^l_{jk} \ge 0, b_{jk} \ge 0$, because even if the data center has a zero-utilization rate, its power consumption is non-negative. Then, the average power cost at data center j consumed by resource k can be represented as $c_j z_{jk} \omega_{jk}$. To linearize this term, we use $\alpha \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} c_j w'_{jk}$ in the objective function and add the following constraint:

$$w'_{jk} \ge \omega_{jk} z_{jk}. \qquad (8)$$

By applying constraints (7) and (8), we formulate the extension with convex power consumption in utilization as the following problem:

$$\begin{aligned}
\textbf{(P-CP)} \quad \min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \sum_{j\in\mathcal{J}} f_j x_j + \alpha \sum_{j\in\mathcal{J},k\in\mathcal{K}} c_j w'_{jk} + \sum_{i\in\mathcal{I},j\in\mathcal{J}} d_i t_{ij} y_{ij} + \sum_{j\in\mathcal{J},k\in\mathcal{K}} \frac{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}{z_{jk} - \sum_i d_i u_{ik} y_{ij}} \\
\text{s.t.} \quad & \text{Constraints } (1b) - (1f),\ (1j) - (1k),\ (7),\ (8).
\end{aligned}$$

Note that constraint (7) contains an exponential polynomial function of linear-fractionals, and constraint (8) requires further linearization, thereby making problem **(P-CP)** difficult to solve. Finally, it should be noted that despite being widely used in the literature, the quadratic function is not necessarily the best way to characterize the convexity of power consumption. By contrast, the proposed model allows the exponent σ^l to take value of any rational number greater than or equal to one.

2.5. Network Congestion

Intense data exchanges may cause congestions in the network. Theoretical models for network congestion typically assume that fabric latency is convexly increasing in traffic intensity and depends on the fabric capacity (Spiess 1990, Luna and Mahey 2000).

To characterize network congestion, we adopt the well-known BPR delay function introduced by the U.S. Bureau of Public Roads. Specifically, the fabric latency cost i to j is given by

$$d_i y_{ij} t_{ij} \left(1 + \left(\frac{d_i}{\chi_{ij}}\right)^{\sigma_{ij}}\right),$$

where a new decision variable, χ_{ij}, is introduced to represent the capacity of the link from i to j. Parameter $\boldsymbol{\sigma}$ can be estimated from historical data. We assume $\sigma_{ij} > 1$ to capture the convexity of fabric latency in traffic intensity. To linearize the objective function, we use s_{ij} to denote network fabric latency and add the following constraint:

$$s_{ij} \geq y_{ij} + \left(\frac{d_i}{\chi_{ij}}\right)^{\sigma_{ij}} y_{ij}. \tag{9}$$

When demand interdependence is observed, the congestion between data centers can be formulated in a similar way. Specifically, the fabric latency from j to j' is

$$t_{jj'} d_{jj'} \left(1 + \left(\frac{d_{jj'}}{\chi_{jj}}\right)^{\sigma_{jj'}}\right),$$

where $d_{jj'} = \sum_{i,i'} \psi_i d_i y_{ij} P_{ii'} y_{i'j'}$ denotes the traffic intensity from data center j to j'. We use $s_{jj'}$ to denote the network latency between data centers and add the following constraints:

$$s_{jj'} \geq d_{jj'} + \frac{d_{jj'}^{\sigma_{jj'}+1}}{\chi_{jj'}^{\sigma_{jj'}}}. \tag{10}$$

By using constraints (9) and (10), we obtain the following comprehensive model that incorporates all the extensions:

$$\begin{aligned}
\textbf{(P-CC)} \quad \min_{\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\chi}} \; & \sum_{j\in\mathcal{J}} f_j x_j + \alpha \sum_{j\in\mathcal{J},k\in\mathcal{K}} c_j w'_{jk} + \sum_{i\in\mathcal{I},j\in\mathcal{J}} (d_i t_{ij} s_{ij} + \kappa_{ij}\chi_{ij}) \\
& + \sum_{jj'\in\mathcal{I},j\in\mathcal{J}} (d_{jj'} s_{jj'} + \kappa_{jj'}\chi_{jj'}) + \sum_{j\in\mathcal{J},k\in\mathcal{K}} \frac{\sum_i \left(\sum_{i'} \tau_{i'} d_{i'} u_{i'k} P_{i'i}\right) y_{ij}}{z_{jk} - \sum_i \left(\sum_{i'} d_{i'} u_{i'k} P_{i'i}\right) y_{ij}} \\
\text{s.t.} \quad & \text{Constraints (1b)} - \text{(1c)},\ \text{(1e)} - \text{(1f)},\ \text{(1j)} - \text{(1k)},\ \text{(6)} - \text{(10)},
\end{aligned}$$

where κ_{ij} and $\kappa_{jj'}$ are the unit costs associated with the capital expenditure on fabric capacity. Note that constraints (9) and (10) are mixed integer nonlinear constraints that introduce further complications to the solution approaches.

3. Analyses and Solution Approach

In this section, we first construct equivalent formulations and analyze some structural properties. Second, we develop a Lagrangian relaxation solution approach. Third, we provide an alternative Lagrangian relaxation approach with extremal extended polymatroid cuts.

3.1. Equivalent Formulation

All models developed in the previous section can be reformulated as equivalent mixed-integer second-order cone programs (MISOCP). We first show that the linear-fractional terms in the objectives of problem **(P)** are second-order cone (SOC) representable. This result is formally stated in the following proposition:

PROPOSITION 2. *(Equivalent Formulation for Endhost Latency) Problem* $(\boldsymbol{P})$ *(i.e., the data center network design base model with heterogeneous footprints) is equivalent to the following problem:*

$$
\begin{aligned}
(\bar{\boldsymbol{P}}) \quad \min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}} \quad & \sum_{j\in\mathcal{J}} f_j x_j + \alpha \sum_{j\in\mathcal{J},k\in\mathcal{K}} c_j w_k z_{jk} + \sum_{i\in\mathcal{I},j\in\mathcal{J}} d_i t_{ij} y_{ij} + \sum_{j\in\mathcal{J},k\in\mathcal{K}} v_{jk} \\
s.t. \quad & \left\| \begin{pmatrix} 2\Lambda_k \mathbf{y}_{\cdot j} \\ v_{jk} - z_{jk} + \sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij} \end{pmatrix} \right\|_2 \leq v_{jk} + z_{jk} - \sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij}, \ \forall j\in\mathcal{J}, k\in\mathcal{K} \qquad (11) \\
& v_{jk} \geq 0, \quad \forall j\in\mathcal{J}, k\in\mathcal{K} \\
& \textit{Constraints } (1b)-(1k),
\end{aligned}
$$

where $\Lambda_k \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ *is a diagonal matrix with* $(\Lambda_k)_{ii} = \sqrt{\tau_i d_i u_{ik}}$. *Problem* $(\bar{\boldsymbol{P}})$ *is an MISOCP.*

The resulting problem $(\bar{\boldsymbol{P}})$ has a linear objective and a set of (convex) second-order cone constraints. Therefore, we can employ methods that solve MISOCP to address this reformulated problem directly. However, these methods become ineffective for problems with moderate to large scales.

Next, in problem **(P-ID)** that models interdependence between footprints, the fabric latency cost associated with rerouting in (5) can be linearized. Several linearization techniques have been proposed in the literature, including a standard technique (Balas and Mazzola 1984), a tightened standard technique, and a more compact technique that has been developed recently for 0-1 quadratic programs by only introducing a linear number of additional variables and constraints (Chaovalitwongse et al. 2004, Sherali and Smith 2007). We numerically compare these techniques and find that the compact technique outperforms the other two. Therefore, we adopt the compact linearization technique to linearize the model. By introducing auxiliary variables w_{ij} and v_{ij}, we

substitute $\sum_{i\in\mathcal{I},j\in\mathcal{J}} w_{ij}$ for the rerouting fabric latency cost in the objective function and add the following set of constraints:

$$\sum_{i'}\sum_{j'} d_i\psi_i t_{jj'}P_{ii'}y_{i'j'} - v_{ij} = w_{ij}, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J} \tag{12a}$$

$$v_{ij} \le \sum_{i'}\sum_{j'} t_{jj'}d_i\psi_i P_{ii'}(1-y_{ij}), \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J}, \tag{12b}$$

$$w_{ij}\ge 0, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J}. \tag{12c}$$

In problem **(P-CP)** that describes the convex power consumption in utilization, the non-convex constraints (7) and (8) are again SOC-representable.

PROPOSITION 3. *(Equivalent Formulation for Power) In problem **(P-CP)**, constraints (7) and (8) are equivalent to the following constraints:*

$$w'_{jk} \ge \sum_l a^l_{jk}\tilde{w}^l_{jk} + b_{jk}z_{jk}, \quad \forall j\in\mathcal{J}, k\in\mathcal{K}\ , \tag{13}$$

$$\tilde{w}^l_{jk} z_{jk}^{\sigma^l-1} \ge (\sum_{i\in\mathcal{I}} d_i u_{ik} y_{ij})^{\sigma^l}, \quad \forall j\in\mathcal{J}, k\in\mathcal{K}, \forall l, \tag{14}$$

where (13) is a linear constraint and (14) is either linear (when $\sigma^l = 1$) or SOC-representable (when σ^l is a rational number greater than 1).

Proposition 3 asserts that the non-convex constraints (7) and (8) are SOC-representable. Throughout the rest of this paper, we assume for clarity that power consumption is a quadratic function of utilization.

Finally, in problem **(P-CC)** that incorporates network congestion, non-convex constraints (9) and constraint (10) are SOC-representable.

PROPOSITION 4. *(Equivalent Formulation for Network Congestion) In problem **(P-CC)**, constraint (9) is equivalent to*

$$\begin{aligned} s_{ij} &\ge y_{ij} + \pi_{ij} \\ d_i^{\sigma_{ij}} y_{ij}^{\sigma_{ij}+1} &\le \chi_{ij}^{\sigma_{ij}}\pi_{ij}, \end{aligned} \tag{15}$$

whereas constraint (10) is equivalent to

$$\begin{aligned} s_{jj'} &\ge d_{jj'} + \pi_{jj'} \\ d_{jj'}^{\sigma_{jj'}+1} &\le \chi_{jj'}^{\sigma_{jj'}}\pi_{jj'}. \end{aligned} \tag{16}$$

Both constraints (15) and (16) are SOC-representable when σ_{ij} ($\sigma_{jj'}$) is positive and rational.

In summary, problem **(P)** and all of the extensions can be reformulated into equivalent MISOCP problems, which in turn can be solved directly with off-the-shelf software packages, such as CPLEX and Gurobi. However, solving instances of practical scale may still be computationally expensive, if not completely intractable.

Before developing tailored solution approaches, note that the term for endhost latency cost $L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$ plays an important role in all of the aforementioned models. Recall that a set function $g(\cdot)$ is said to be supermodular if for any two sets $S,\ T \subset \mathcal{V}$, $g(S \cup T) + g(S \cap T) \geq g(S) + g(T)$, and the definition of supermodularity can be extended to lattice domains. Proposition 5 states that $L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot})$ possesses supermodularity.

PROPOSITION 5. *(Supermodularity for Endhost Latency) For each $j \in \mathcal{J}$, define the function $\tilde{L}_j : \{0,1\}^{|\mathcal{I}|} \times \mathbb{R}_-^{|\mathcal{K}|} \to \mathbb{R}_+ \cup \{+\infty\}$ as*

$$\tilde{L}_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}^-) = \begin{cases} L_j(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}), \ \textit{if } \tilde{z}_{jk} + \sum_i d_i u_{ik} y_{ij} < 0, \ \forall k \in \mathcal{K}; \\ +\infty, \ \textit{otherwise}, \end{cases}$$

where $\mathbf{z}^- = -\mathbf{z}$. Then, function $\tilde{L}_j$ is increasing and supermodular in $(\mathbf{y}_{\cdot j}, \mathbf{z}_{j\cdot}^-)$ on $\{0,1\}^{|\mathcal{I}|} \times \mathbb{R}_-^{|\mathcal{K}|}$.

Proposition 5 sheds light on the marginal effects of footprint allocation and resource provisioning decisions. The implications are two-fold: (1) when a footprint is pending assignment to a set of near-capacitated data centers, all of which cannot be further expanded due to power capacity constraints or limited budget, it is desirable from the endhost latency perspective to assign the footprint to the data center that has the lightest load; and (2) when the allocation decision is rigid, possibly due to dominating fabric latency costs, and there exists extra budget to invest in resource provisioning, we shall invest in data centers with heavier loads. We leverage these properties to design accompanying heuristics for our proposed Lagrangian relaxation methods, which will be presented in the next subsection.

3.2. A Lagrangian Relaxation Algorithm

Lagrangian relaxation has been proven efficient in solving large-scale capacitated facility location (CFL) problems (Cornuéjols et al. 1991). This algorithm is mainly applied in two ways, namely, by relaxing either the capacity constraint (1e) or the assignment fulfillment constraint (1c) (Beasley 1993, Sridharan 1995). Although the latter leads to easier knapsack subproblems for the CFL, it fails for the proposed model due to the endhost latency cost. Therefore, in this subsection, we relax the power capacity constraints (1e) to solve MISOCP problem ($\bar{\boldsymbol{P}}$). This approach also applies to the extensions. Let $\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{J}|}$ be the vector of Lagrangian multipliers associated with the set of

constraints (1e). After relaxing constraints (1e), the Lagrangian subproblem, which is again an MISOCP, can be formulated as follows:

$$\textbf{(P-L)}\ Z_D(\boldsymbol{\lambda}) = \min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}} \sum_j f_j x_j + \alpha \sum_{j,k} c_j w_k z_{jk} + \sum_{i,j} d_i t_{ij} y_{ij} + \sum_{j,k} v_{jk} + \sum_j \lambda_j (\sum_k w_k z_{jk} - p_j)$$
$$\text{s.t. (1b) – (1d), (1f), (1j), (1k), (11)}$$

Intuitively, the Lagrangian subproblem **(P-L)** becomes easier to solve due to the withdrawal of *packing* of resources into data centers. Given the Lagrangian multipliers $\boldsymbol{\lambda}^{(m)}$, where superscript (m) denotes the iteration index, the Lagrangian subproblem's objective value $Z_D(\lambda^{(m)})$ generates a sequence of lower bounds for problem **(P)**. If the solution to the Lagrangian subproblem **(P-L)** is feasible to problem **(P)**, then this solution yields an upper bound of $\bar{Z}$. We terminate the Lagrangian iterations when $\bar{Z}$ and $Z_D(\lambda^{(m)})$ are close enough. We follow the commonly used rules to update the Lagrangian multiplier $\boldsymbol{\lambda}$ (see Fisher 1981). For notation convenience, we drop the iteration index (m) in the subsequent discussion.

Since the solution to the Lagrangian subproblem **(P-L)** is not always feasible to the primal problem **(P)**, we cannot guarantee consistent updates of the upper bound $\bar{Z}$. To accelerate convergence, we develop a heuristic algorithm that generates feasible solutions from the solution to problem **(P-L)**. The core idea of the heuristic is to find over-provisioned data centers, choose certain footprints assigned to them, and swap the host data centers for these footprints.

Specifically, for each solution that is primal infeasible, we divide the facilities into three sets according to the power capacity constraints. Set $\mathcal{A}$ consists of selected data centers that are over-provisioned; set $\mathcal{B}$ consists of selected data centers that are under-provisioned; and set $\mathcal{C}$ consists of unselected data centers. The heuristic attempts to sequentially re-assign footprints served by data centers in set $\mathcal{A}$ to those in $\mathcal{B}$, and if none of the data centers in $\mathcal{B}$ have sufficient power, then the heuristic opens a new data center in $\mathcal{C}$. The heuristic continues this swapping process until $\mathcal{A}$ is empty.

The heuristic is outlined in Algorithm 1. Suppose that in the optimal solution to the Lagrangian subproblem **(P-L)**, footprint s is assigned to an over-provisioned data center t. In this case, we shall swap the host of s. Lines 7 and 16 of Algorithm 1 are crucial steps that identify the candidate host data center t', either from $\mathcal{B}$ or $\mathcal{C}$. The key in the swapping process is estimating the cost increment associated with each swap. Denote $\delta(s,t,t')$ to represent the upper bound on the cost increment imposed by swapping the host of footprint s from t to t'. Proposition 6 quantifies $\delta(s,t,t')$. Since $\delta(s,t,t')$ bounds the duality gap from above, Proposition 6 is useful for checking whether the incumbent feasible solution is close enough to the optimal solution.

PROPOSITION 6. *(Bounds of the Incumbent Feasible Solution) Let z_{tk}, $z_{t'k}$ $(\forall k \in \mathcal{K})$ be the resource provisioning levels before the swap at data centers t and t', respectively. Suppose t' is not over-provisioned after the swap. Then $\delta(s,t,t')$ satisfies:*

(a) If $t' \in \mathcal{B}$, then the upper bound $\delta(s,t,t')$ is given by

$$\delta(s,t,t') = d_s \left(\tau_s \sum_k u_{sk} \left(\frac{1}{z_{t'k} - \sum_i d_i u_{ik} y_{it'}} - \frac{1}{z_{tk} - \sum_i d_i u_{ik} y_{it}} \right) + \alpha(c_{t'} - c_t) \sum_k u_{sk} w_k + t_{st'} - t_{st} \right).$$

(b) If $t' \in \mathcal{C}$ and t' has a moderate size such that $p_{t'} \geq \sum_k w_k (z_{tk} - \sum_i d_i u_{ik} y_{it})$, the upper bound $\delta(s,t,t')$ is given by

$$\delta(s,t,t') = f_{t'} + \alpha c_{t'} \sum_k w_k \left(z_{tk} - \sum_i d_i u_{ik} y_{it} \right) - d_s \left(\alpha c_t \sum_k u_{sk} w_k - t_{st'} + t_{st} \right).$$

In light of the supermodularity result of Proposition 5, the heuristic, while selecting a new host, first employs a simple rule as described in lines 8 and 17 to check approximately whether the candidate host has sufficient unused power and loose resource ratio constraints. After each swapping step of the heuristic, line 24 of Algorithm 1 re-optimizes the levels of resource provisioning for the two data centers associated with the swap, and then calculates the corresponding total endhost latency costs. The **(Re-optimize)** procedure takes the assigned footprints as inputs to determine the optimal resource provisioning under power capacity and resource ratio constraints. Specifically, this procedure solves the following subproblem:

$$\begin{aligned}
(\textbf{\textit{Re-optimize}}) \quad \min_{\mathbf{z}_{j\cdot}, \mathbf{v}_j} \quad & \sum_k \alpha c_j w_k z_{jk} + v_{jk} \\
\text{s.t.} \quad & \left\| \begin{pmatrix} 2\Lambda_k \mathbf{y}_{\cdot j} \\ v_{jk} - z_{jk} + \sum_{i \in \mathcal{I}} d_i u_{ik} y_{ij} \end{pmatrix} \right\|_2 \leq v_{jk} + z_{jk} - \sum_{i \in \mathcal{I}} d_i u_{ik} y_{ij}, \ \forall k \in \mathcal{K} \\
& z_{jk} \leq \bar{r}_{kl} z_{jl}, \quad \forall k, l \in \mathcal{K} \\
& \sum_{k \in \mathcal{K}} w_k z_{jk} \leq p_j.
\end{aligned}$$

This subproblem is a second-order cone program with only continuous decision variables and can be solved efficiently by using commercial software packages.

We provide another heuristic, as illustrated in Algorithm 2, that accelerates the update of $\mathbf{z}$ and $\mathbf{v}$. This heuristic first iterates myopically over all resources to calculate the optimal provisioning level of resources with neither power capacity nor resource ratio constraints and then truncates (line 3) and scales (line 7) the solutions to ensure that they satisfy the constraints. By combining Algorithms 1 and 2 (or subproblem **(Re-optimize)**), we are able to generate feasible solutions to the original problem **(P)** that we use to update the upper bounds $\bar{Z}$. In extremely tight instances, the heuristic may not return a feasible solution in some iterations. It is worth noting that the generalization of these heuristics and Proposition 6 for model extensions is straightforward yet cumbersome. It also leads to weaker bounds due to the nonlinear power consumption and network congestion functions.

Algorithm 1 Heuristic algorithm for finding a feasible solution to problem **(P)**.

Initialize $\mathcal{A}=\{j: x_j=1, \sum_k w_k z_{jk} > p_j\}$, $\mathcal{B}=\{j: x_j=1, \sum_k w_k z_{jk} \le p_j\}$, $\mathcal{C}=\{j: x_j=0\}$
while $\mathcal{A}\neq\emptyset$ **do**
  $t=\arg\max_{j\in\mathcal{J}}\{\sum_k(w_k z_{jk})-p_j\}$
  $s=\arg\min_{i\in\{i|y_{it}=1\}} \sum_k d_i u_{ik} w_k$
  $y_{st}\leftarrow 0$
  while $\mathcal{B}\neq\emptyset$ **do**
    $t'\leftarrow \arg\min_{j\in\mathcal{B}}\left\{\tau_s \sum_k \frac{u_{sk}}{z_{jk}-\sum_i d_i u_{ik} y_{ij}} + \sum_k \alpha c_j u_{sk} w_k + t_{sj}\right\}$
    if $\sum_k(w_k z_{t'k} + d_s u_{sk} w_k) \le p_{t'}$, and $(d_s u_{sk} + z_{t'k}) \le \bar{r}_{kl}(d_s u_{sl} + z_{t'l}), \forall k,l \in \mathcal{K}$ **then**
      $y_{st'}\leftarrow 1$
      break
    else
      delete t' from $\mathcal{B}$
    end if
  end while
  while $\mathcal{C}\neq\emptyset$ **do**
    $t'\leftarrow \arg\min_{j\in\mathcal{C}}\{\alpha c_j \sum_k w_k(z_{tk}-\sum_i d_i u_{ik} y_{ij}) + d_s t_{sj} + f_j\}$
    if $\sum_k(w_k(z_{tk}-\sum_i d_i u_{ik} y_{it})) \le p_{t'}$ **then**
      $y_{st'}\leftarrow 1$ and $x_{t'}\leftarrow 1$
      break
    else
      delete t' from $\mathcal{C}$
    end if
  end while
  Update $\mathcal{A},\mathcal{B},\mathcal{C}$, $(\mathbf{z}_{t\cdot}, \mathbf{v}_{t\cdot}) \leftarrow$ ***Re-optimize***$(\mathbf{y}_{\cdot t})$, and $(\mathbf{z}_{t'\cdot}, \mathbf{v}_{t'}) \leftarrow$ ***Re-optimize***$(\mathbf{y}_{\cdot t'})$
end while

3.3. Lagrangian Relaxation with Extended Polymatroid Cuts

The Lagrangian relaxation algorithm presented in the previous subsection can be applied to the general model with various extensions. In this subsection, we further exploit the structural property of the endhost latency terms introduced in Section 2.2 to develop a highly efficient Lagrangian relaxation algorithm for the base model. In addition to the power capacity constraint, we also relax the resource ratio constraints of problem **(P)** to achieve separability. Afterward, the resource provisioning decisions can be solved explicitly, and the supermodular cost function can be transformed into a submodular function. While the resulting subproblem can again be formulated as an MISOCP and solved directly by using commercial solver packages, its structural property (i.e., submodularity) allows us to add strengthening cuts to improve the computing efficiency for the

Algorithm 2 Heuristic algorithm for updating $\mathbf{z}_{j\cdot}$.

[**Function**] ***Re-optimize***$(\mathbf{y}_{\cdot j})$
for $k = 1, ..., |\mathcal{K}|$ **do**
  $z_{jk} = \max\{\min\{\sum_i d_i u_{ik} y_{ij} + \sqrt{\frac{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}{\alpha c_j w_k}}, \bar{r}_{kl} z_{jl}\}, \frac{z_{jl}}{\bar{r}_{lk}}\}$, for all $l \le k$
end for
if $\sum_k w_k z_{jk} > p_j$ **then**
  for $k = 1, ..., |\mathcal{K}|$ **do**
    $z_{jk} \leftarrow \frac{p_j}{\sum_k w_k z_{jk}} z_{jk}$
  end for
end if
$v_{jk} \leftarrow \frac{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}{z_{jk} - \sum_i d_i u_{ik} y_{ij}}$

Lagrangian subproblems. We note that submodularity may not hold for the various model extensions presented in Section 2, and the original Lagrangian relaxation algorithm is still required to solve the extensions.

Specifically, for problem **(P)**, relaxing the resource ratio constraints (1f) and the power capacity constraints (1e) yields

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z}} \ (1a) + \sum_{j\in\mathcal{J}} \lambda_j \left(\sum_{k\in\mathcal{K}} w_k z_{jk} - p_j \right) + \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} \sum_{l\in\mathcal{K}} \zeta_{jkl} \left(z_{jk} - \bar{r}_{kl} z_{jl} \right)$$
$$\text{s.t. Constraints (1b)–(1d), (1j), (1k),}$$

where $(\boldsymbol{\lambda}, \boldsymbol{\zeta})$ are Lagrangian multipliers associated with constraints (1e) and (1f), respectively. Note that, for any given $\mathbf{y}$ and for data center j and resource k, optimizing z_{jk} in the Lagrangian subproblem creates the following problem:

$$\min_{z_{jk}\ge 0} \phi_{jk} z_{jk} + \frac{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}{z_{jk} - \sum_i d_i u_{ik} y_{ij}},$$

where $\phi_{jk} = (\alpha c_j + \lambda_j) w_k + \sum_{l\in\mathcal{K}} (\zeta_{jkl} - \bar{r}_{lk} \zeta_{jlk})$. Given that $\phi_{jk} > 0$, as is required for the Lagrangian dual, the optimal resource provisioning solution is given by

$$z^*_{jk} = \sum_i d_i u_{ik} y_{ij} + \sqrt{\sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij} / \phi_{jk}}.$$

The optimal cost associated with resource provisioning is given by

$$\phi_{jk} \sum_i d_i u_{ik} y_{ij} + 2 \sqrt{\phi_{jk} \sum_{i\in\mathcal{I}} \tau_i d_i u_{ik} y_{ij}}. \tag{17}$$

Let $g_{jk}(\mathbf{y}_{\cdot j}) = \sqrt{\phi_{jk}\sum_{i\in\mathcal{I}}\tau_i d_i u_{ik} y_{ij}}$. With the optimal z_{jk}, the Lagrangian subproblem becomes

$$\begin{aligned}\textbf{(P-LC)}\quad \min_{\mathbf{x},\mathbf{y}}\ & \sum_j f_j x_j + \sum_i\sum_j \left(d_i t_{ij} + \sum_k \phi_{jk} d_i u_{ik}\right) y_{ij} + 2\sum_j\sum_k v_{jk} - \sum_j \lambda_j p_j \\ \text{s.t.}\ & g_{jk}(\mathbf{y}_{\cdot j}) \le v_{jk}, \quad \forall j\in\mathcal{J}, k\in\mathcal{K} \qquad (18)\\ & \text{Constraints (1b), (1c), (1j)}\end{aligned}$$

Constraint (18) can be reformulated as a second-order cone constraint. Therefore, subproblem **(P-LC)** can again be solved directly by using commercial solver packages. However, the standard branch-and-bound procedures of solver packages can be improved by adding strengthening cuts, illustrated as follows. This idea stems from Atamtürk and Narayanan (2008). We define polyhedron $\mathcal{Q}_{jk}$ to denote the lower convex envelope of $g_{jk}(\cdot)$, that is,

$$\mathcal{Q}_{jk} = \text{conv}\{(\boldsymbol{\eta}, t)\in\{0,1\}^{|\mathcal{I}|}\times\mathbb{R} : g_{jk}(\boldsymbol{\eta}) \le t\}.$$

In addition, let

$$\text{EP}_{jk} = \left\{\boldsymbol{\pi}\in\mathbb{R}^{|\mathcal{I}|} : \boldsymbol{\pi}^\mathsf{T}\boldsymbol{\eta} \le g_{jk}(\boldsymbol{\eta}),\ \forall \boldsymbol{\eta}\in\{0,1\}^{|\mathcal{I}|}\right\}.$$

Atamtürk and Narayanan (2008) show that the linear inequality

$$\boldsymbol{\pi}^\mathsf{T}\boldsymbol{\eta} \le t \qquad (19)$$

is valid for $\mathcal{Q}_{jk}$ if and only if $\boldsymbol{\pi}\in\text{EP}_{jk}$. Consequently, we can add the inequalities (19) that are violated as cuts during the branch-and-bound process to strengthen the formulation. In particular, for a given fractional solution $\hat{\mathbf{y}}_{\cdot j}\in[0,1]^{|\mathcal{I}|}$ and $\hat{v}_{jk}\ge 0$, define

$$\hat{\boldsymbol{\pi}}_{jk} = \arg\max\{\boldsymbol{\pi}^\mathsf{T}\hat{\mathbf{y}}_j,\ \boldsymbol{\pi}\in\text{EP}_{jk}\}. \qquad (20)$$

If $(\hat{\boldsymbol{\pi}}_{jk})^\mathsf{T}\hat{\mathbf{y}}_j > \hat{v}_{jk}$, then we add the following cut to the conic relaxation of problem **(P-LC)**:

$$(\hat{\boldsymbol{\pi}}_{jk})^\mathsf{T}\mathbf{y}_{\cdot j} \le v_{jk}. \qquad (21)$$

Problem (20) is generally difficult to solve. Nonetheless, note that $g_{jk}(\cdot)$ is a *polymatroid function* on $\{0,1\}^{|\mathcal{I}|}$, and hence EP_{jk} defines an *extended polymatroid* associated with function $g_{jk}(\cdot)$. Consequently, problem (20) becomes a maximum weighted-sum problem over polymatroids. The seminal work of Edmonds (1970) establishes that a *greedy algorithm* solves the max-weight problem over polymatroids and that the optimal solutions of (20) are extreme points of the extended polymatroid EP_{jk}. As a result, the induced cuts are termed as *extremal extended polymatroid inequalities* (Atamtürk and Narayanan 2008). To generate cuts (21) by using Edmonds (1970)'s algorithm, we sort $\hat{y}_{ij}$ in descending order such that $\hat{y}_{(1)j}\ge\hat{y}_{(2)j}\ge\cdots\ge\hat{y}_{(|\mathcal{I}|)j}$, where $\{(1),(2),\ldots,(|\mathcal{I}|)\}$ is a permutation of footprint indices. Afterward, letting $S_i = \{(1),(2),\ldots,(i)\}$ and setting

$$\hat{\pi}_{(i)j} = \sqrt{\phi_{jk}\sum_{i\in S_i}\tau_i d_i u_{ik}} - \sqrt{\phi_{jk}\sum_{i\in S_{i-1}}\tau_i d_i u_{ik}},$$

we obtain the $(\hat{\boldsymbol{\pi}}_{jk})^\mathsf{T}$ of the strengthening cuts (21).

4. Numerical Results

We present numerical results to demonstrate the efficacy of the proposed data center network design model and to draw managerial insights. First, we conduct experiments on instances constructed using real-world datasets to analyze how important factors, such as latency cost, affect the optimal design, and to address several practical design questions. Second, we benchmark the computational performance of the tailored solution approaches against that of the commercial software Gurobi with randomly generated instances to demonstrate their efficiency.

4.1. Case Study

Our first set of experiments uses the proposed model and solution approaches with real-world datasets. We analyze the impact of important contributory factors such as the latency cost, and study how the colocation service and traffic between data centers shape the optimal data center network design.

The experiment settings are as follows. We adopt *the facility location test dataset* of Daskin (1995), which contains both a 49-node case and an 88-node case, as our footprint locations. We use the 49-node setting in the current set of experiments. In Online Appendix EC.3.4, we repeat the test with the 88-node setting and obtain similar results. We assume that the demand rate is proportional to the number of households that own computers (U.S. Census Bureau 2015b), and that fulfilling each job requires only two types of resources, namely, computing and storage. In addition, we assume that the required computing of each job is proportional to the average education level attained at the location where the job originates (U.S. Census Bureau 2015a). Similarly, we assume that the required storage of each job is proportional to the number of subscriptions to high-speed Internet at the location where the job originates (U.S. Census Bureau 2015b).

We select 36 candidate data center locations based on several sources. Of these locations, 14 come from *DPR construction* (`https://www.dpr.com/projects`), which is a third-party technical builder that has constructed data centers for many firms, including Motorola, eBay, HBO, etc. The remainder of the candidate locations includes the locations of *existing data centers* that are owned by major Internet companies, including eight from Google, five from Microsoft, five from Facebook, and four from Amazon. The unit power costs at the candidate locations are obtained from *the US Energy Information Administration* (`https://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_5_6_a`).

Other important parameters include the unit fabric latency and unit endhost latency costs. To embed user heterogeneity, we assume that footprints with a higher GDP are willing to pay more to receive better service. In other words, these footprints are more sensitive to latency costs. For example, users in the Silicon Valley or financial districts are arguably more sensitive to service

delays, and these regions (California, New York, and Illinois) also have higher GDPs. We omit colocation for the moment. The specific demand and candidate data center settings are summarized in Tables EC.1 and EC.2 in Online Appendix EC.3.2.

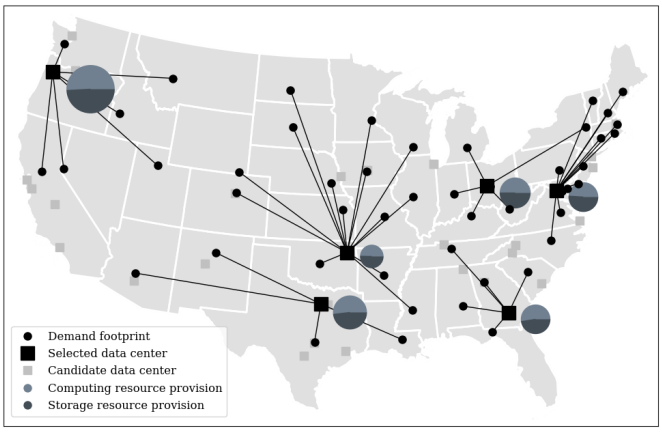


Figure 3 Optimal data center network for a case constructed with real-world datasets.

Following the aforementioned assumptions, we generate a data center network design for the 49-node setting, as illustrated in Figure 3, where the dots represent the footprints, the blue squares are the selected data centers, the gray squares are the unselected candidate data centers, and the line segments connecting the dots and blue squares represent the assignment of footprints to the data centers. We use pie charts to illustrate the resource mix, and the sizes of these pie charts denote the volumes of total resource provisioning.

The six data centers selected by our model are located in the states of Georgia, Ohio, Oklahoma, Oregon, Texas, and Virginia, respectively. Interestingly, our selection echoes those of three major cloud computing infrastructure providers, namely, Amazon (Georgia, Ohio, Oregon, Virginia, https://aws.amazon.com/about-aws/global-infrastructure/), Google (Georgia, Oklahoma, Oregon, https://www.google.com/about/datacenters/inside/locations/), and Microsoft (Texas, Virginia, https://azure.microsoft.com/en-us/regions/?v=17.23h). In particular, all four locations used by Amazon coincide with our solution. This result supports the validity of our network design model.

Finally, the numerical results justify the necessity of jointly optimizing long-term strategic decisions, such as location, and short-term tactical decisions such as resource provisioning, because failing to include tactical decisions can result in highly sub-optimal strategic decisions. However, the integration of decisions with different time scales may also raise some concerns over the fidelity of the model. Certain parameters, such as demand, have a considerable impact on both strategic and tactical decisions but may be hard to predict. While tactical decisions can easily adapt to

changing demand, strategic decisions generally cannot be adjusted easily. Therefore, it is important for the proposed model to generate strategic decisions that are robust to changing parameters. To this end, we experiment on the robustness of the network design under demand and electricity price perturbations (see Online Appendix EC.3.3). The results indicate that while tactical and operational decisions can be significantly affected by parameter uncertainties, strategic location and allocation decisions are relatively consistent.

4.2. Structural Properties and Sensitivity Analyses

In light of Proposition 6, we first provide the following result, which indicates that assigning a new demand to an existing data center does not require overprovisioning for this new demand to enjoy the same quality of service as existing demand.

COROLLARY 1. *Assigning a new demand d_i to data center j, and supplementing resource provisioning at j with $d_i u_{ik}$ for all k, then for all i' such that $y_{i'j} = 1$, the sojourn time of i' at data center j remains the same.*

Corollary 1 suggests that, when assigning new demand to an existing data center, if we augment the resource provisioning level by the same amount as the new demand, then the endhost latency experienced by the existing demand remains the same, while the new demand enjoys an endhost latency proportional to the average endhost latency of other allocated demand at the focal data center. In other words, there exist opportunities for *free-riding* current resource-provisioning slacks at data centers. In turn, this result provides guidelines for practitioners who may need to determine the allocation of emerging demand continuously.

Next, we present several structural properties of the problem based on a simplified model by fixing integer decision variables and decomposing the resulting problem by data center locations. Specifically, we fix the location and allocation decisions, and study the impacts of demand volume, power cost, and unit end-host latency cost on optimal solutions.

PROPOSITION 7. *Suppose the capacity constraints (1e) and the resources ratio constraints (1f) are not binding. The optimal solutions at each data center possess the following sensitivity properties:*

(i) both the optimal resource provisioning levels and the optimal objective value concavely increase in the allocated demand, suggesting economies of scale in allocated demand;

(ii) the optimal resource provisioning levels convexly decrease in unit power cost and power consumption per unit of resource, while the optimal objective value concavely increases in unit power cost and power consumption per unit of resource;

(iii) both the optimal resource provisioning levels and the optimal objective value concavely increase in the unit endhost latency cost.

Proposition 7 yields several insights. First, item *(i)* indicates the existence of economies of scale. In other words, data centers with larger capacity, which can host more demand, are more cost-efficient. Consequently, network designers are motivated to choose large-capacity data centers, which serve more demand nodes on average, to achieve economies of scale. This implication is in line with data center network designs in practice. Indeed, Internet giants such as Google and Amazon are building large data centers that can serve broad regions. From the derivation of the optimal resource provisioning levels, we also see that a data center's resource provisioning decision only affects its own performance. Therefore, given location and allocation decisions, resource provisioning decisions can be decentralized to local data centers; that is, each data center will choose a resource provisioning level that is optimal for a centralized system. Second, item *(ii)* suggests that, should there be investments in technological advancements, it would be beneficial to invest in power efficiency; however, the marginal benefit is diminishing, or one can say that a little saving in power efficiency goes a long way. Third, item *(iii)* implies that when consumers are more sensitive to service quality, data centers should provide more resources. Similarly, the impact of increased service quality sensitivity on the optimal resource provisioning level and the optimal objective value is diminishing.

In the following substion, we substitute synthetic data for real-world data to conduct further analyses on different parameters to provide additional managerial insights.

4.2.1. Latency Cost. We first study the impact of different levels of unit fabric and endhost latency costs. The experiment results are summarized in Table 2, where columns "Fabric" and "Endhost" show the scaling factors for the unit fabric and endhost latency costs compared with the base case. Column "DC Built" shows the number of selected data centers.

Table 2 Impact of fabric and endhost latency cost on the number of data centers.

No.	Fabric	Endhost	DC Built	No.	Fabric	Endhost	DC Built	No.	Fabric	Endhost	DC Built
1	1	1	10	5	5	1	16	9	10	1	20
2	1	10	9	6	5	10	15	10	10	10	20
3	1	80	11	7	5	80	16	11	10	80	19
4	1	100	12	8	5	100	17	12	10	100	19

The experiment results suggest that more data centers are built when the unit fabric latency cost is large. This result is rather intuitive, as more data centers offer closer access for users to save on fabric latency costs. By contrast, Table 2 shows that the impact of higher unit endhost latency is not straightforward. On the one hand, as the unit endhost latency increases, the optimal design may open fewer data centers (e.g., compare cases 1 and 2, or 5 and 6). Specifically, when the power capacity and resource ratio constraints are loose, the optimal endhost latency cost takes the form of (17), which is submodular in the set of assigned footprints. In other words, the endhost latency

cost exhibits economies of scale; that is, when assigning more footprints to a sufficiently large data center, the optimal average endhost latency cost per unit of demand decreases. As a result, higher unit endhost latency leads to the pooling of footprints. On the other hand, higher unit endhost latency may lead to *counter-pooling* (e.g., compare cases 2, 3, and 4, or 6, 7 and 8), because the power capacities and resource ratio constraints may change the property of the optimal endhost latency costs. Specifically, when resource provisioning at a data center is limited by a tight power capacity, assigning more demand to this data center will increase the marginal endhost latency cost according to Proposition 5. We summarize these insights into the following two observations that can be used as practical design guidelines:

Observation 1 ***(Impact of Fabric Latency)*** *Higher unit fabric latency costs lead to* counter-pooling*, that is, more data centers will be built to provide closer access distances.*

Observation 2 ***(Impact of Endhost Latency)*** *Higher unit endhost latency costs lead to pooling in data centers with more extra power capacity, yet lead to counter-pooling in those data centers with tighter power capacity constraints or more stringent resource ratio constraints.*

4.2.2. Colocation Data Centers. While keeping other parameters unchanged, we enable the colocation option and experiment on the impact of different levels of fixed and unit power costs at candidate colocation sites. We assume that the power capacities at the colocation sites are sufficient to serve the local demand.

We first vary the fixed and unit power costs. Given that colocations are usually contracted with third parties that offer space and server rental, the associated fixed costs are typically lower than the costs of privately owned centers. Nonetheless, the unit power costs at colocations are typically higher due to their proximity to large metropolitan areas. In addition, we account for rent and maintenance fees in the "power cost" term. The experiment results are summarized in Table 3, where columns "Fixed" and "Power" are the scaling factors for the colocations' fixed and unit power costs (plus rent and maintenance per unit power consumed), respectively. Columns "Colo" and "DC" are the number of selected colocations and the number of regular data centers built, respectively.

The results indicate that there are fewer colocations when either the fixed cost or the unit power costs associated with colocation is higher. Moreover, as the costs associated with colocation increase, there could be more regular data centers due to the key trade-off between savings in the fabric latency costs via closer access to colocations and savings in fixed costs, power costs and the economies of scale from resource sharing via regular data centers.

Table 3 Impact of fixed and variable costs on the number of colocations and private data centers.

No.	Fixed	Power	Colo	DC	No.	Fixed	Power	Colo	DC	No.	Fixed	Power	Colo	DC
1	0.01	1	39	4	4	0.01	1.5	31	5	7	0.01	2	17	6
2	0.05	1	20	5	5	0.05	1.5	7	6	8	0.05	2	0	7
3	0.1	1	8	6	6	0.1	1.5	0	7	9	0.1	2	0	7

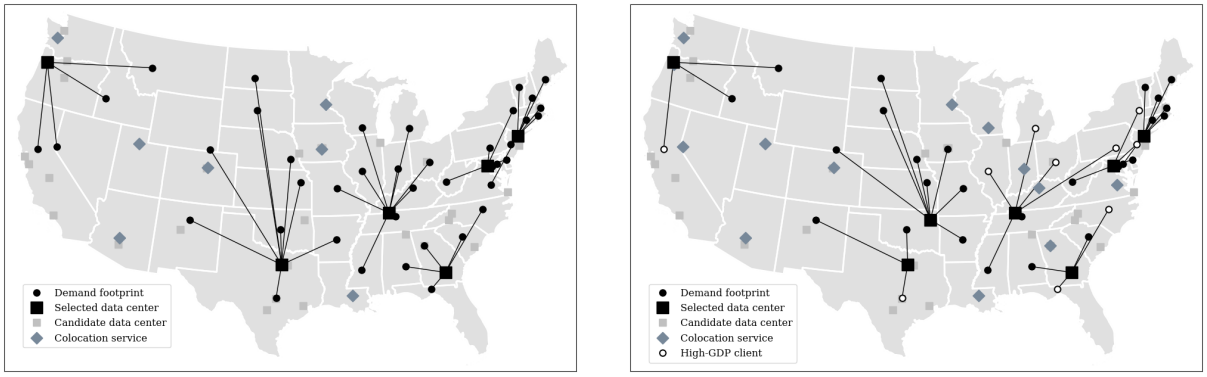
In the same vein as in the study with only regular data centers, the unit fabric and endhost latency costs also affect the network design with colocations. We vary the unit latency cost levels to test their influences. The results are summarized in Table 4. On the one hand, the results verify the intuition that a higher unit fabric latency cost leads to both more regular data centers and more colocations. On the other hand, a higher unit endhost cost brings more colocations, but possibly fewer regular data centers (e.g., compare cases 4, 5 and 6). The latter result (i.e., fewer regular data centers) echoes Observation 2 because of the pooling effect, whereas the former result (i.e., more colocations) seems to contradict previous arguments. There are two potential reasons for such discrepancies. First, pooling may either be infeasible or necessitate extra fabric latency cost. Second, as more footprints are pooled together, resource provisioning at the data center has diminishing marginal returns (Proposition 5).

Table 4 Impact of latency costs on the number of regular and colocation data centers.

No.	Fabric	Endhost	Colo	DC	No.	Fabric	Endhost	Colo	DC	No.	Fabric	Endhost	Colo	DC
1	1	1	7	6	4	5	1	30	7	7	20	1	39	7
2	1	5	15	7	5	5	5	36	5	8	20	5	41	6
3	1	20	20	7	6	5	20	39	5	9	20	20	41	6

Next, we conduct another set of experiments to identify the key criteria that render a footprint suitable for the colocation option. Figure 4a shows a design with seven colocations denoted by diamonds. The colocations are located in the states of Washington, Minnesota, Louisiana, Arizona, Colorado, Iowa, and Utah. By comparing these colocations and analyzing the differences between them and other candidate colocations, we find that footprints that are located far from any regular candidate site (CO, IA, UT, LA, MN, AZ), and/or those with low electricity costs (WA, LA), and/or those with a moderate demand volume (WA, AZ, CO, MN) are more likely to be served by colocations.

Intuitively, colocation should be used for footprints that are highly sensitive to latency. Surprisingly, we find that this is not necessarily the case. To test this hypothesis, we increase the unit latency costs for the 10 states with the highest GDP by five times (https://www.bea.gov/newsreleases/regional/gdp_state/2017/pdf/qgsp0517.pdf). The new optimal network design is depicted in Figure 4b. The updated design has six additional colocations. However, the newly added colocations are not necessarily for the footprints with high endhost latency costs. Instead,



(a) Optimal data center network with colocations and moderate latency-sensitivity

(b) Optimal data center network with colocations and high latency-sensitivity

Figure 4 Impact of latency-sensitivity on the optimal data center network with colocations.

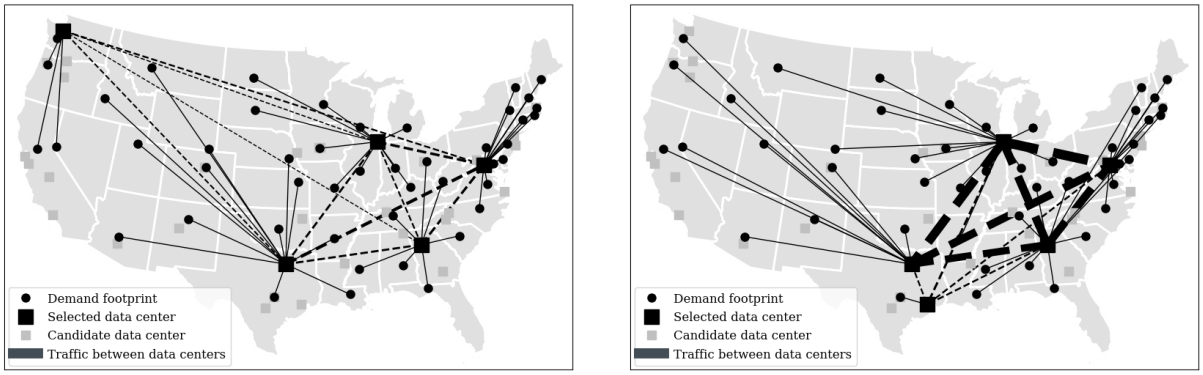
these colocations are built for footprints with smaller demand volume to reduce the endhost latency of the delay-sensitive large footprints at regular data centers. For example, as shown in Figure 4b, the optimal design opens colocations in Indiana and Kentucky to better serve Ohio and Illinois, and a colocation in Oregon to better serve California. While multiple competing factors contribute to this observation, we discuss two potential explanations. First, Proposition 1 shows that better serving large footprints requires a larger resource margin, which is defined as provisioned resources minus demand. Assigning large footprints to a data center creates positive spillovers as it allows other pooled demands at the data center to have a low endhost latency by enjoying the margin. Second, utilizing colocations incurs fixed costs that may dominate the benefits for small footprints. Consequently, footprints with moderate demand and a long distance from regular sites may greatly benefit from colocation. We summarize these results in the following observation:

Observation 3 ***(Guidelines for Colocation Selection)*** *Colocation data centers are more suitable for locations with low electricity rates (as well as low rent and maintenance fees), moderate demand volume, and a long distance from regular sites. In addition, smaller and less sensitive footprints may be forced to open colocations to allow regular data centers to better serve larger and more sensitive footprints.*

4.2.3. Interdependent Footprints. As discussed in Section 2.3, sometimes jobs need to be rerouted from their designated data centers for further service, and the total latency cost includes additional fabric latency costs incurred by routing jobs between data centers. We then investigate how the intensity of the interdependence between footprints affects the network design. Recall that in the presence of traffic between data centers, the total demand at each data center consists of jobs arriving directly from the users and jobs rerouted from other data centers. A higher total

demand will obviously affect the design. To examine the impact of interdependence, we scale the rerouting probability P_{ij} and external arrival d_i in Eq. (3) simultaneously to adjust the intensity of traffic between data centers, while keeping the total demand λ_{ij} unchanged.

The optimal network design for the contiguous U.S. is shown in Figure 5. The width of the dashed line connecting the data centers represents the intensity of traffic between data centers. Figures 5a and 5b plot the optimal designs with light and heavy traffic intensity, respectively. Note that the case with heavy traffic intensity has approximately three times heavier traffic between data centers, but ten percent lighter traffic between footprints and data centers. By comparing these two cases, we find that when the intensity of traffic between data centers increases, it is better to select data centers that are close to one another to save on fabric latency costs. We summarize these results in the following observation:



(a) Light traffic between data centers (b) Heavy traffic between data centers

Figure 5 Optimal data center networks under different traffic intensities between data centers.

Observation 4 ***(Impact of Interdependence)*** *A high level of interdependence between footprints (i.e., high traffic intensity between data centers) leads to the selection of data centers that are close to one another in terms of unit fabric latency cost, or even to the pooling of multiple data centers when possible.*

4.3. Capacity Expansion with Average Shadow Price

Thus far, we have focused on data center network design with fixed capacities. With the rapid growth in the demand for cloud computing and Internet-related services, service providers may often need to determine where to build more capacity. The Lagrangian approaches developed in the previous section can be leveraged to provide guidance for capacity expansion decisions. However, unlike continuous optimization for which the Lagrangian multipliers provide the shadow prices of the capacity, the Lagrangian multipliers in mixed-integer programming have limited economic

interpretations (Kim and Cho 1988). In this subsection, we discuss how to guide capacity expansion decisions using the *average shadow price* (Kim and Cho 1988, Crema 1995). Denote **P** to represent our original problem, V the objective value, and $V_j(w)$ the corresponding objective value if we increase the power capacity for the j-th candidate data center by w. Then, the average shadow price for the power constraint of data center j is defined as

$$p_j = \inf\{p \geq 0 : V - V_j(w) - pw \leq 0, \quad \forall\, w \geq 0\}.$$

Intuitively, the average shadow price can be interpreted as the maximum price that the network designer is willing to pay for one unit of additional capacity. In other words, it is not worthwhile to expand the data center's capacity whenever the unit expansion cost exceeds the average shadow price.

We provide a numerical example to illustrate how the average shadow prices can guide capacity expansion decisions. We use the same parameter settings as those used for Figure 1, which includes 32 demand nodes and 15 candidate data centers. We adopt the algorithm from Crema (1995) to calculate the average shadow prices, as shown in Online Appendix EC.3.5. The optimal Lagrangian multipliers and the average shadow prices for the candidate data centers are summarized in the following table.

Table 5 Average Shadow Prices and Lagrangian Multipliers

# DC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ASP	1.02	7.74	4.41	2.92	6.20	0.00	16.79	0.00	22.55	4.75	0.00	3.24	5.47	0.00	1.49
LM	0.00	0.55	0.00	0.00	1.52	0.00	0.00	0.00	0.34	0.00	0.00	0.00	0.00	0.00	0.07

Note. "# DC" denotes the indices of data centers; "ASP" denotes the average shadow price; and "LM" denotes the Lagrangian multipliers.

From the table we observe that there are only four positive Lagrangian multipliers, but eleven positive average shadow prices. Data center 5 has the largest Lagrangian multiplier and also has a large average shadow price, implying high cost-saving potential if were to expand its power capacity. Nonetheless, the average shadow prices and the Lagrangian multipliers deliver mixed messages for other data centers.

To elucidate the insights of the average shadow prices, we conduct additional numerical studies, by supplementing the power capacities of the data centers one by one and with five different magnitudes, ranging from 10 to 5000 (the average power capacity in the example is around 8500). The results are summarized in Table 6. Specifically, the table records the decrements in the optimal objective values in respective rows for each data center whose power capacity is augmented. The Lagrangian multipliers loosely capture the marginal benefits of capacity expansion. In comparison,

Table 6 Saved cost with extra capacity

Saved cost		Expanded capacity				
		10	100	1000	2000	5000
# DC	1	0.00	0.00	0.00	0.00	4944.29
	2	0.00	0.00	0.00	0.00	36021.29
	3	0.00	0.00	759.11	7724.34	18126.42
	4	0.00	0.00	0.00	3481.12	9952.03
	5	53.07	461.18	2184.29	11205.29	27904.51
	6	0.00	0.00	0.00	0.00	0.00
	7	170.34	1484.67	11316.64	11912.81	12561.41
	8	0.00	0.00	0.00	0.00	0.00
	9	223.24	216.33	8006.98	8595.65	8595.65
	10	0.00	0.00	0.00	7476.43	18885.93
	11	0.00	0.00	0.00	0.00	0.00
	12	0.00	0.00	0.00	5784.18	10798.03
	13	43.24	376.56	4234.69	6819.44	13143.15
	14	0.00	0.00	0.00	0.00	0.00
	15	0.00	0.00	2986.79	5157.86	5198.38

Note. "# DC" denotes the indices of data centers.

the average shadow prices reported in Table 5 better delineate the cost-saving potentials, and thus excel in guiding capacity expansion decisions.

4.4. Computational Performance

Our next set of experiments tests the computational performance of the two Lagrangian relaxation solution approaches as described in Subsections 3.2 and 3.3. Specifically, we randomly generate instances with different sizes, solve the data center network design model by using the two proposed approaches and by using Gurobi directly, and compare the computational performance of the three methods.

The size of an instance consists of the number of candidate data centers and the number of footprints. We uniformly sample the locations of both footprints and candidate data centers within a square. The fabric latency cost between footprint i and candidate data center location j is assumed to be linear in the Euclidean distance between i and j. We scale the size of the square to equate the average unit fabric latency cost between data centers for instances of different sizes. Then, for each instance, the rest of the parameters are sampled with distributions selected such that the instances can be drastic to test the proposed solution approaches. The settings are explained in detail in Table EC.3 in the Online Appendix. We generate 100 instances for each size and calculate the average performance metrics.

The experiments are conducted by using Gurobi 7.5.2 with Python 2.7 on a 64-bit Windows 10 operating system equipped with an Intel Core i7-4770 3.4GHz quad-core processor and 16 GB RAM. When solving each instance, we set a limit of 1800 seconds on the computing time

and a limit of 2000 on the number of subgradient iterations (for the two Lagrangian relaxation approaches). Some instances cannot be solved within the time or iteration limit. We calculate the "solved ratio" for each solution approach, which is defined as the percentage of instances that are successfully solved within the computing time and subgradient iteration limits. We calculate the average computing time as well as the average number of iterations only for the solved instances. We also record the remaining optimality gaps for the unsolved instances.

The main results of this set of experiments are summarized in Table 7, where the first two columns specify the sizes of the instances. The "Ratio" column reports the solved ratio as defined above, the "Time" and "Iteration" columns record the average computing time and the average number of subgradient iterations for the successfully solved instances, respectively, and the "Gap" column records the average optimality gaps of the unsolved instances when the algorithm is terminated due to the time or iteration limit.

The results demonstrate that both of the two Lagrangian relaxation solution approaches outperform the use of Gurobi in solving the MISOCP reformulation, especially when the instances are large and the power capacity constraints are tight. Specifically, the solved ratio of using Gurobi directly drops below 15% when 30 footprints and 15 candidate data centers are present under tight power capacity constraints. Moreover, for the successfully solved instances, the computing time of using Gurobi directly grows much faster than those of implementing the Lagrangian relaxation approaches. This can be ascribed to the fact that the average computing time for the relaxed problem in Lagrangian relaxation scales mildly with the size of instances. Nonetheless, for instances with total demand very close to total capacity, many subgradient iterations may be required in order for the Lagrangian relaxation approaches to converge. Furthermore, although the Lagrangian relaxation approaches have "unsolved" instances when the instance is large, the remaining optimality gaps are below 12%.

Between the two Lagrangian relaxation approaches, the one with extended polymatroid cuts as described in subsection 3.3 relaxes an additional set of resource provisioning ratio constraints, thereby resulting in an increased number of subgradient iterations. However, the total computing time of this approach is actually shorter than that of the standard Lagrangian approach as described in subsection 3.2, thanks to the strengthening cuts. Intuitively, the performance of the Lagrangian relaxation approach with extended polymatroid cuts is subject to the tightness of the resource ratio constraints and the shape distribution of the footprints. However, our numerical results show that the performance of this approach is robust under non-extreme practical settings.

Finally, we compare the three linearization methods discussed in subsection 2.3 for the model extension with interdependent footprints. We find that the compact linearization method in equation (12) outperforms the other two methods, in general. The results are presented in detail in Online Appendix EC.3.6.

Table 7 Computational performance of the proposed solution approaches benchmarked with Gurobi.

# F	# S	Gurobi			Lagrangian				Lagrangian with cut			
		Ratio	Time	Gap	Ratio	Time	Iterations	Gap	Ratio	Time	Iterations	Gap
Loose Power Capacities												
10	5	100%	0.23	–	100%	0.29	3.26	–	100%	0.27	6.95	–
10	10	100%	0.45	–	100%	0.46	1.47	–	100%	0.36	2.78	–
20	10	100%	16.09	–	100%	1.48	4.00	–	100%	1.06	4.57	–
20	15	100%	11.26	–	100%	3.24	5.13	–	100%	1.27	3.40	–
30	15	100%	135.13	–	100%	5.52	7.32	–	100%	3.14	7.82	–
30	20	99%	93.31	5.98%	100%	28.36	20.00	–	100%	3.15	6.02	–
40	20	85%	260.12	7.78%	93%	131.15	79.12	8.39%	100%	50.83	81.02	–
50	25	82%	366.02	7.14%	93%	226.20	102.52	6.05%	100%	150.42	105.14	–
Medium Power Capacities												
10	5	100%	0.21	–	100%	0.32	3.93	–	100%	0.31	9.16	–
10	10	100%	0.33	–	100%	1.50	9.40	–	100%	0.47	4.72	–
20	10	100%	40.37	–	100%	5.23	22.07	–	100%	1.98	12.24	–
20	15	100%	43.73	–	100%	23.19	40.27	–	100%	2.52	9.35	–
30	15	90%	400.63	6.38%	97%	28.57	47.83	6.98%	100%	8.11	26.68	–
30	20	78%	394.39	7.55%	97%	65.11	54.61	8.28%	100%	9.16	23.67	–
40	20	17%	799.16	15.29%	88%	82.21	71.38	8.10%	100%	30.87	64.90	–
50	25	0%	–	20.44%	82%	141.42	84.27	8.73%	95%	211.39	147.61	7.92%
Tight Power Capacities												
10	5	100%	0.86	–	100%	0.39	4.89	–	100%	0.54	16.95	–
10	10	100%	2.79	–	100%	3.68	27.00	–	100%	1.12	17.49	–
20	10	88%	377.72	7.86%	100%	5.95	20.19	–	100%	3.49	17.02	–
20	15	73%	454.90	7.81%	100%	48.90	68.95	–	100%	4.05	18.34	–
30	15	12%	1099.78	11.59%	97%	51.59	106.61	7.21%	100%	15.62	58.07	–
30	20	3%	1475.89	11.07%	83%	93.81	80.72	8.53%	100%	19.16	58.02	–
40	20	0%	–	12.79%	78%	149.08	147.85	10.39%	95%	54.60	120.74	10.16%
50	25	0%	–	14.64%	63%	287.89	201.82	11.22%	75%	295.62	211.42	9.53%

Notes. "# F": number of footnotes; "# S": number of sites; 100 instances are solved for each problem size combination; "Ratio": the percentage of instances that are solved within the time and iteration limit; "Time": the average computing time for the *solved* instances; "Iterations": the average number of Lagrangian iterations for the *solved* instances; "Gap": the average optimality gaps of the *unsolved* instances when the algorithm is terminated due to the time or iteration limit.

5. Conclusion

In this paper, we propose a location-allocation and resource provisioning model for data center network design. Our model aims to find the optimal trade-off between cost efficiency and quality of service by considering a variety of cost terms and service level metrics affected by decisions spanning from strategic location to tactical resource provisioning. We explicitly model the service delay experienced by users consisting of fabric and endhost latency, and study how these two latency costs affect the design of data center networks. Further, we model important practical extensions with interdependent footprints, convex power consumption in utilization, and network

congestions. We show that the proposed model, as well as the three extensions, can be reformulated as MISOCPs that are potentially solvable with off-the-shelf software packages. We then develop two Lagrangian relaxation solution approaches by exploiting the structural properties of our model to solve practical problems of larger scale.

We conduct numerical experiments to validate the effectiveness and efficiency of the proposed model and the solution approaches, and to draw managerial insights. We first shed light on the importance of employing an integrated model by comparing the output of the proposed model and that of a hierarchical approach. Moreover, by using several real-world datasets from sources such as the U.S. Census Bureau and the U.S. Energy Information Administration, we generate the optimal network design for the contiguous U.S., and find that our design largely echoes the current data center locations of Amazon Web Services (AWS) from Amazon, Compute Engine from Google, and Azure from Microsoft, thereby confirming the high validity of our model. By varying important parameters, we find that when the unit fabric latency cost is high, the optimal network design comprises more data centers. However, when the unit endhost latency cost is high, the optimal network may either pool more demands in fewer data centers with sufficient power capacity or build more data centers (or colocations), a phenomenon which we refer to as "counter-pooling". When colocation is available, footprints that are remote and have a moderate demand volume should be considered for colocation first, whereas large latency-sensitive footprints may not be suitable for colocation. Finally, when excessive traffic is present between data centers, the optimal design will open data centers that are close to one another in terms of unit fabric latency cost.

In this paper, we did not consider the *dynamic* network design problem in which the data center network expands over time to cope with demand growth or power rate changes. Similarly, incorporating dynamic routing policy decisions into the network design may present an interesting direction for future research. Also, when modeling resource provisioning decisions, we omitted machine-level details, except for machine configurations represented by the resource ratio constraints. Additional machine level decisions include *sharding* and *packing* the footprints, and *shelfing* the machines into racks. Incorporating these decisions into the long-term strategical planning model may achieve further cost savings and service improvements. Finally, developing alternative models that enable analytical characterizations of the optimal solutions will help uncover more structural properties of the optimal network design.

References

Aboolian R, Berman O, Drezner Z (2008) Location and allocation of service units on a congested network. *IIE Transactions* 40(4):422–433.

Aboolian R, Berman O, Krass D (2007) Competitive facility location model with concave demand. *European Journal of Operational Research* 181(2):598–619.

Abouee-Mehrizi H, Babri S, Berman O, Shavandi H (2011) Optimizing capacity, pricing and location decisions on a congested network with balking. *Mathematical Methods of Operations Research* 74(2):233–255.

Altman E, Ayesta U, Prabhu BJ (2011) Load balancing in processor sharing systems. *Telecommunication Systems* 47(1-2):35–48.

Atamtürk A, Berenguer G, Shen ZJM (2012) A conic integer programming approach to stochastic joint location-inventory problems. *Operations Research* 60(2):366–381.

Atamtürk A, Narayanan V (2008) Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters* 36(5):618–622.

Balas E, Mazzola JB (1984) Nonlinear 0–1 programming: I. linearization techniques. *Mathematical Programming* 30(1):1–21.

Baron O, Milner J, Naseraldin H (2011) Facility location: A robust optimization approach. *Production and Operations Management* 20(5):772–785.

Barroso LA, Clidaras J, Hölzle U (2013) The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis lectures on computer architecture* 8(3):1–154.

Baskett F, Chandy KM, Muntz RR, Palacios FG (1975) Open, closed, and mixed networks of queues with different classes of customers. *Journal of the Association for Computing Machinery* 22(2):248–260.

Beasley JE (1993) Lagrangean heuristics for location problems. *European Journal of Operational Research* 65(3):383–399.

Ben-Tal A, Nemirovski A (2001) *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2 (Siam).

Berman O, Krass D (2015) Stochastic location models with congestion. Laporte G, Nickel S, Gama FSd, eds., *Location Science*, 443–486 (Springer International Publishing).

Castillo I, Ingolfsson A, Sim T (2009) Social optimal location of facilities with fixed servers, stochastic demand, and congestion. *Production and Operations Management* 18(6):721–736.

CBRE (2018) U.s. data center trends report h2 2017, retrieved April 10, 2018, https://www.cbre.us/research-and-reports/US-Data-Center-Trends-H2-2017.

Chaovalitwongse W, Pardalos PM, Prokopyev OA (2004) A new linearization technique for multi-quadratic 0–1 programming problems. *Operations Research Letters* 32(6):517–522.

Chen Ky, Xu Y, Xi K, Chao HJ (2013) Intelligent Virtual Machine Placement for Cost Efficiency in Geo-Distributed Cloud Systems. *Communications (ICC), 2013 IEEE International Conference on. IEEE* 3498–3503.

Chudak FA, Williamson DP (2005) Improved approximation algorithms for capacitated facility location problems. *Mathematical programming* 102(2):207–222.

Cornuéjols G, Sridharan R, Thizy JM (1991) A comparison of heuristics and relaxations for the capacitated plant location problem. *European journal of operational research* 50(3):280–297.

Cox DR (1955) A use of complex probabilities in the theory of stochastic processes. *Mathematical Proceedings of the Cambridge Philosophical Society* 51(2):313–319, URL http://dx.doi.org/10.1017/S0305004100030231.

Crema A (1995) Average shadow price in a mixed integer linear programming problem. *European Journal of Operational Research* 85(3):625–635.

Daskin MS (1995) *Network and Discrete Location: Models, Algorithms, and Applications* (John Wiley & Sons).

Daskin MS, Coullard CR, Shen ZJM (2002) An inventory-location model: Formulation, solution algorithm and computational results. *Annals of operations research* 110(1-4):83–106.

Edmonds J (1970) Submodular functions, matroids, and certain polyhedra. *Edited by G. Goos, J. Hartmanis, and J. van Leeuwen* 11.

Elhedhli S (2006a) Service system design with immobile servers, stochastic demand, and congestion. *Manufacturing & Service Operations Management* 8(1):92–97.

Elhedhli S (2006b) Service System Design with Immobile Servers, Stochastic Demand, and Congestion. *Manufacturing & Service Operations Management* 8(1):92–97.

Fisher ML (1981) The Lagrangian relaxation method for solving integer programming problems. *Management Science* 27(1):1–18.

Geunes J, Levi R, Romeijn HE, Shmoys DB (2011) Approximation algorithms for supply chain planning and logistics problems with market choice. *Mathematical Programming* 130(1):85–106.

Geunes J, Pardalos PM (2003) Network optimization in supply chain management and financial engineering: An annotated bibliography. *Networks: An International Journal* 42(2):66–84.

Goudarzi H, Ghasemazar M, Pedram M (2012) Sla-based optimization of power and migration cost in cloud computing. *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, 172–179, URL http://dx.doi.org/10.1109/CCGrid.2012.112.

Greenberg A, Hamilton J, Maltz DA, Patel P (2008) The cost of a cloud: Research problems in data center networks. *SIGCOMM Comput. Commun. Rev.* 39(1):68–73.

Harchol-Balter M (2013) *Performance Modeling and Design of Computer Systems: Queueing Theory in Action* (Cambridge University Press).

He L, Hu Z, Zhang M (2018) Robust repositioning for vehicle sharing. Working Paper, National University of Singapore.

He L, Mak HY, Rong Y, Shen ZJM (2017) Service region design for urban electric vehicle sharing systems. *Manufacturing & Service Operations Management* 19(2):309–327.

Iyoob I, Zarifoglu E, Dieker AB (2013) Cloud computing operations research. *Service Science* 5(2):88–101.

Kim S, Cho Sc (1988) A shadow price in integer programming for management decision. *European journal of operational research* 37(3):328–335.

Kong Q, Lee CY, Teo CP, Zheng Z (2013) Scheduling arrivals to a stochastic service delivery system using copositive cones. *Operations Research* 61(3):711–726.

Kong Q, Li S, Liu N, Teo CP, Yan Z (2017) Appointment scheduling under schedule-dependent patient no-show behavior. Working Paper, National University of Singapore.

Larumbe F, Sansò B (2012) Optimal location of data centers and software components in cloud computing network design. *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 841–844, CCGRID '12 (Washington, DC, USA: IEEE Computer Society).

Larumbe F, Sansò B (2013) A tabu search algorithm for the location of data centers and software components in green cloud computing networks. *IEEE Transactions on Cloud Computing* 1(1):22–35.

Luna HPL, Mahey P (2000) Bounds for global optimization of capacity expansion and flow assignment problems. *Operations Research Letters* 26(5):211–216.

Mak HY, Rong Y, Shen ZJM (2013) Infrastructure planning for electric vehicles with battery swapping. *Management Science* 59(7):1557–1575.

Mak HY, Rong Y, Zhang J (2015) Appointment scheduling with limited distributional information. *Management Science* 61(2):316–334.

Miller R (2018) Facebook accelerates its data center expansion, retrieved July 15, 2018, https://datacenterfrontier.com/facebooks-accelerates-data-center-expansion/.

Paraskevopoulos DC, Gürel S, Bektaş T (2016) The congested multicommodity network design problem. *Transportation Research Part E: Logistics and Transportation Review* 85:166–187.

Pinedo ML (2016) *Scheduling: Theory, Algorithms, and Systems* (Springer).

Qi W, Liang Y, Shen ZJM (2015) Joint planning of energy storage and transmission for wind energy generation. *Operations Research* 63(6):1280–1293.

Sen A, Atamturk A, Kaminsky P (2017) A conic integer programming approach to constrained assortment optimization under the mixed multinomial logit model. *arXiv preprint arXiv:1705.09040* .

Shen ZJM, Coullard C, Daskin MS (2003) A joint location-inventory model. *Transportation science* 37(1):40–55.

Sherali HD, Smith JC (2007) An improved linearization strategy for zero-one quadratic programming problems. *Optimization Letters* 1(1):33–47.

Silberschatz A, Galvin PB, Gagne G, Silberschatz A (1998) *Operating System Concepts*, volume 4 (Addison-wesley Reading).

Spiess H (1990) Conical volume-delay functions. *Transportation Science* 24(2):153–158.

Sridharan R (1995) The capacitated plant location problem. *European Journal of Operational Research* 87(2):203–213.

Sverdlik Y (2018) Oracle to launch 12 cloud data centers around the world, retrieved July 15, 2018, http://www.datacenterknowledge.com/oracle/oracle-launch-12-cloud-data-centers-around-world.

Taaffe K, Geunes J, Romeijn HE (2008) Target market selection and marketing effort under uncertainty: The selective newsvendor. *European Journal of Operational Research* 189(3):987–1003.

Topkis DM (1998) *Supermodularity and Complementarity* (Princeton University Press).

US Census Bureau (2015a) Education attainment 1-year estimate. URL https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_15_1YR_S1501&prodType=table.

US Census Bureau (2015b) Types of computers and Internet subscription 1-year estimate. URL https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_15_1YR_S2801&prodType=table.

Vaughan A (2016) The Guardian: Google uses ai to cut data centre energy use by 15%, retrieved April 11, 2018, https://www.theguardian.com/environment/2016/jul/20/google-ai-cut-data-centre-energy-use-15-per-cent.

Vazirani VV (2013) *Approximation algorithms* (Springer Science & Business Media).

Verma A, Ahuja P, Neogi A (2008) pMapper: Power and migration cost aware application placement in virtualized systems. Issarny V, Schantz R, eds., *Middleware 2008*, 243–264 (Berlin, Heidelberg: Springer Berlin Heidelberg), ISBN 978-3-540-89856-6.

Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J (2015) Large-scale cluster management at Google with Borg. *Proceedings of the Tenth European Conference on Computer Systems*, 18 (ACM).

Wan G, Leung JYT, Pinedo ML (2007) Scheduling imprecise computation tasks on uniform processors. *Information processing letters* 104(2):45–52.

Wang Q, Batta R, Rump CM (2004) Facility location models for immobile servers with stochastic demand. *Naval Research Logistics* 51(1):137–152.

Williamson DP, Shmoys DB (2011) *The design of approximation algorithms* (Cambridge university press).

Zhang Y, Berman O, Marcotte P, Verter V (2010) A bilevel model for preventive healthcare facility network design with congestion. *IIE Transactions* 42(12):865–880.

Zhang Y, Berman O, Verter V (2009) Incorporating congestion in preventive healthcare facility network design. *European Journal of Operational Research* 198(3):922–935.