Optimally Locating Application Virtualization Resources on a Network

Kelly M. Sullivan[1]
Department of Industrial Engineering
University of Arkansas
4207 Bell Engineering Center
Fayetteville, AR 72701 USA

David T. Abdul-Malak[2] and Jeffrey P. Kharoufeh[3]
Department of Industrial Engineering
University of Pittsburgh
1048 Benedum Hall
3700 O'Hara Street
Pittsburgh, PA 15261 USA

Rusty O. Baldwin[4]
Cyber Research Laboratory
Riverside Research
2640 Hibiscus Way
Beavercreek, OH 45431 USA

Final version appears in
Military Operations Research, 20 (1), 5–20, 2015

Abstract

We consider the problem of optimally locating application virtualization resources among a network of military treatment facilities within the Military Health System (MHS). Application virtualization consolidates computer application processing and administration in a centralized data center and delivers a virtual image of the application over the network, thereby enabling MHS medical providers to work with the virtual image as if the application were hosted locally on their own machines. However, remotely-located facilities may experience response times that are too high to host application virtualization, or have bandwidth capacities that cannot support the high traffic intensities that are likely to ensue. To address this problem, we first formulate an optimization model that seeks to maximize the probability that the response times at all facilities are bounded above by a fixed time threshold by locating virtualization resources (or virtual hubs) and allocating demand to them. Because this problem is difficult to solve for realistically-sized networks, we propose an alternative max-min optimization model that leads to an exact mixed integer linear program that can be solved using commercial software. Computational results highlight the effectiveness of the latter formulation in solving problems with over 200 medical treatment facilities.

[1]Email: ksulliv@uark.edu; Ph: (479) 575-2563.
[2]Email: dta10@pitt.edu.
[3]Corresponding author. Email: jkharouf@pitt.edu; Ph: (412) 624-9832.
[4]Email:rbaldwin@RiversideResearch.org; Ph: (937) 427-7118.

INTRODUCTION

The Military Health System (MHS) is the U.S. Department of Defense (DoD) organization responsible for providing health care services to active and retired U.S. military personnel and their dependents. Recent estimates show that the MHS network of providers serves approximately 9.5 million current or former military members and their dependents, and the MHS employs about 137,000 staff members in 65 hospitals and over 800 clinics (including medical and dental clinics) in the U.S. and abroad. The U.S. government has called for all health and medical records to be converted to electronic health records (EHRs) and electronic medical records (EMRs) within the next few years. Presumably, by consolidating digital medical records for military personnel and their dependents, MHS providers can improve the speed and accuracy of their medical decision making and the quality of care provided to active duty and retired military personnel. Ideally, providers in the U.S. and abroad will gain seamless access to service members' full medical records, with authentication, using only a few key strokes from any networked computer.

Currently, the DoD provides digitized information via the Armed Forces Health Longitudinal Technology Application (AHLTA), the world's largest EHR database. But providing time-efficient and secure access to AHLTA has proven to be challenging due to geographical, mobility, and connectivity issues across the network. In an attempt to overcome these challenges, the DoD has recommended the use of an Application Virtualization Hosting Environment (AVHE) at remote access sites across the MHS network. Application virtualization consolidates application processing and administration in a centralized data center and delivers a virtual image of the application over the network. MHS providers (and other application users) work with the virtual image as if the application were hosted locally on their own machines. However, there is a concern that some remotely-located sites may experience latencies (or response times) that are too high to sustain a deployment of AVHE, or have network bandwidth capacities that cannot support the high traffic intensities that are likely to ensue.

It is anticipated that, subsequent to the deployment of AVHEs, the network's configuration will be modified so that a greater proportion of network traffic is routed through the Defense Information Systems Network (DISN) wide area network (WAN) rather than the currently heavily-used local area networks. The increased traffic on a centralized network may result in longer delays at some sites; however, these delays can be minimized by strategically deploying virtualization resources. The virtualization portion of this network communications system involves the connection between the virtualized AHLTA application and the Military Treatment Facility (MTF) End User Device (EUD). This virtualized AHLTA application will be installed on desktop machines, but functionality and the graphical user interface will remain unchanged. It is envisioned that, in its future state, the EUD will function solely as an interface device, displaying the output from the virtual instance of AHLTA and returning user input. Currently, the application and EUD do not communicate or require any data flow. After deploying AVHE, the EUDs of the military treatment facilities and the AHLTA application will be connected via the DISN WAN. However, this connection is limited by the bandwidth between the location of the application and the site of the instance of AVHE, in addition to the capacity of the DISN WAN through which it must run.

For these reasons, there is a need to determine the optimal locations of AVHE sites to deploy in the MHS network, and the optimal routing of requests to those locations, in order to achieve

critical performance objectives. One of the critical measures of performance for MHS providers is the time needed to retrieve a patient's medical record to support a patient encounter (we refer to this as the *response time*). While one can consider minimizing the expected aggregate delay experienced by users across the entire network by selecting the locations of virtual resources, it may be more appropriate to maximize the likelihood that the worst-case delay does not exceed a fixed time threshold. Therefore, a novel approach for allocating virtualization resources is needed to ensure that quality-of-service guarantees can be met at all MHS facilities, irrespective of their locations and local resource capacities. This paper proposes a virtualization location/assignment model for this purpose.

The problem of locating virtualized resources in resource-constrained MTFs (and assigning users to them) is closely related to the problem of locating facilities that are subject to congestion. Research on that subject can be traced to a sequence of papers in the 1980's that sought to locate mobile facilities (e.g., emergency service vehicles) to service randomly occurring demand requests dispersed across a geographical region and represented as a set of nodes. Specifically, Berman et al. (Berman, Larson, & Chiu, 1985) considered the problem of locating a single facility in a network under the assumption that demand is Poisson distributed and service times are generally distributed, but travel times for the mobile facility are deterministic. Extensions of that work included consideration of random travel times (Berman, 1985), a basic location-allocation model in which two mobile facilities are located and demand nodes are allocated to one of the two facilities (Berman & Larson, 1985), and a more general p-facility location-allocation model (Berman & Mandowsky, 1986). However, none of these models allow for cooperation between facilities (i.e., each demand node is always serviced by the same facility), motivating the stochastic queue p-median (Berman, Larson, & Parkan, 1987). While the work in (Berman & Larson, 1985), (Berman et al., 1987) and (Berman & Mandowsky, 1986) utilized heuristic optimization procedures, more recent research examines exact optimization procedures for a similar class of problems (Filippi & Romanin-Jacur, 1996). A comprehensive review of a number other publications related to locating mobile facilities subject to congestion was published in 2004 (Berman & Krass, 2004).

Relevant to our work here, the problem of locating *immobile* facilities under congestion has received a significant amount of attention since the late 1980's, with applications including the design of airline systems (Marianov & Serra, 2003 ; Mohammadi, Jolai, & Rostami, 2011) and healthcare systems (Marianov & Serra, 2001). A survey of models for the location of immobile facilities under congestion is provided by Boffey et al. (Boffey, Galvão, & Espejo, 2007). Most of the congestion-related research focus on probabilistic generalizations of classical, deterministic location problems. For context, we define those here. Demands for service arising at a set of demand nodes $\mathcal{N}$ are to be satisfied (in full or in part) by locating facilities at a subset of the nodes $\mathcal{N}$. Travel times $\ell_{ii'}$ are given for each node pair $i, i' \in \mathcal{N}$, and demand λ_i exists at node $i \in \mathcal{N}$. The *location set covering problem* (LSCP) (Toregas, Swain, ReVelle, & Bergman, 1971) seeks to locate a minimum number of facilities such that each node $i \in \mathcal{N}$ is within a constant T time units of some facility. The *maximal covering location problem* (MCLP) of (Church & ReVelle, 1974) relaxes the strict quality-of-service requirements of LSCP, aiming instead to maximize the proportion of demand located within T time units of a facility by locating a fixed number of facilities.

The first probabilistic generalizations of the deterministic location models involve busy fractions, i.e., *a priori* estimates of the proportion of time a facility is unavailable to service demand

requests. ReVelle and Hogan (ReVelle & Hogan, 1989a, 1989b) use busy fractions to pose the *maximum availability location problem* (MALP) and the *probabilistic location set covering problem* (PLSCP) – the probabilistic analogs of MCLP and LSCP, respectively. Marianov and ReVelle contributed methods for estimating busy fractions associated with requests at each demand node by considering all other demand nodes and servers within a fixed distance as an autonomous $M/G/s$ loss system (Marianov & ReVelle, 1994, 1996). Each of the papers summarized here presents a solution approach based on solving an integer-linear program. None of the aforementioned immobile facility location models explicitly account for how demand is assigned to a facility when there are multiple possibilities. Ball and Lin proposed an integer-linear program for a problem similar in nature to PLSCP (Ball & Lin, 1993). Their model, which assumes each node generates demand independently according to Poisson process, minimizes the cost of installing facilities while constraining a conservative bound on service reliability that is valid regardless of demand-to-facility assignment policy.

More recently, models have accounted for demand-to-facility assignment by explicitly including allocation decisions as a part of the model. Marianov and Serra considered a location-allocation version of MCLP in which demands occur at each node according to a Poisson process, and each facility acts as an $M/M/m$ queue in processing the demands allocated to that facility (Marianov & Serra, 1998). The authors demonstrated how to formulate the problem as a deterministic integer-linear program when probabilistic constraints are given for the queue length or waiting time for a particular facility. The number of servers at each facility (m) is fixed in this model; however, Marianov and Serra considered a related location-allocation problem in which the number of servers at each facility is variable (Marianov & Serra, 2002). Related work demonstrates that similar linear models can be obtained when service times are deterministic (Marianov & Serra, 2003) or r-Erlang distributed (Marianov, Boffey, & Galvão, 2009) instead of exponentially distributed.

Other objectives have been considered within queueing-based location-allocation models as well. For example, Elhedhli considered minimizing the total expected delay experienced by a demand request (Elhedhli, 2005, 2006), as do Vidyarthi and Jayaswal (Vidyarthi & Jayaswal, 2014). Elhedhli assumes that facilities behave as standard $M/M/1$ queues, whereas Vidyarthi and Jayaswal extend that approach by solving a sequence of integer programs for facilities with generally-distributed service times. Aboolian et al. considered the maximization of profit when locating facilities that behave as $M/M/m$ queueing systems and determined their capacities (Aboolian, Berman, & Krass, 2012). For all of the location-allocation models discussed thus far, the allocation decisions are assumed to be controlled at the system level, i.e., the decision maker can make both the location and allocation decisions. However, Wang et al. (Wang, Batta, & Rump, 2002) pioneered a new line of research in which the allocation decisions are exogenously controlled by customers generating demand. This idea spawned a significant line of new research (Aboolian, Berman, & Drezner, 2008), (Aboolian, Berman, & Drezner, 2009), (Berman & Drezner, 2007), (Berman, Huang, Kim, & Menezes, 2007), (Berman, Krass, & Wang, 2006), (L. Zhang & Rushton, 2008), (Y. Zhang, Berman, Marcotte, & Verter, 2010), (Y. Zhang, Berman, & Verter, 2009).

The primary objective of our work is to formulate, analyze, and solve a mathematical programming model of (a subset of) the MHS system that includes 243 medical treatment facilities (medical clinics, hospitals, and large medical centers). The aim is to optimally locate virtual hubs on a finite subset of existing candidate sites and allocate each site's traffic to a hub so as to maximize the like-

lihood that the worst-case response time experienced by an arbitrary user is no more than a fixed time threshold. The problem is similar to the location-allocation problem considered by Marianov and Serra (Marianov & Serra, 1998) in that response time includes delays due to congestion, which we model by viewing each hub as an $M/M/1$ queueing system. However, whereas other researchers consider random response time as a chance constraint (Marianov & Serra, 1998, 2002), we consider it within the objective. The approach in (Marianov & Serra, 1998, 2002) is to convert the chance constraint into a linear, deterministic constraint by defining a threshold on the volume of traffic that can be assigned to any hub. Introducing the random response time into the objective complicates this approach. Specifically, the traffic threshold defined in (Marianov & Serra, 1998, 2002) depends on a fixed probability associated with the chance constraint. No such transformation is apparent in our problem because this probability (the likelihood that response time does not exceed the time threshold) is variable. Moreover, unlike the problems studied in (Marianov & Serra, 1998, 2002), our problem also considers delays due to geographical distances between nodes.

The remainder of the paper is organized as follows. The next section presents notation and provides a natural formulation for the problem of optimally locating virtual hubs and assigning facilities to them. Due to difficulties in solving this problem, we subsequently present an alternative, exact formulation that is linearized and facilitates the solution of larger problem instances. Next, we provide the results of an extensive computational experiment designed to illustrate the best attainable performance of virtualization under a wide range of scenarios. Finally, we provide some concluding remarks and important directions for future research.

PROBLEM FORMULATION

Consider N nodes (i.e., large medical centers, hospitals, and medical clinics) whose geographical locations are given by coordinates in $\mathbb{R}^3$ along the surface of a sphere. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the node (or vertex) set and suppose that any of the N nodes can be selected to host an Application Virtualization Hosting Environment (AVHE). For simplicity, henceforth we will refer to AVHE host sites as "virtual hubs." While all of the nodes are candidate sites, the number of nodes that can be selected as hubs (H) is limited (e.g., due to budgetary constraints). The nodes generate demand independently of one another according to a Poisson process. That is, node $i \in \mathcal{N}$ generates requests for information according to a Poisson process with average rate λ_i ($\lambda_i > 0$) requests/sec. We refer to λ_i as the *demand rate* of node i. Likewise, associated with each node $j \in \mathcal{N}$ is a capacity for processing data (i.e, for receiving and/or transmitting data). We refer to this average rate as the *transmission rate* of node j and denote it by μ_j ($\mu_j > 0$). Define d_{ij} as the distance between nodes $i, j \in \mathcal{N}$, and let $D = [d_{ij}]$ be the distance matrix. We approximate D by calculating the orthodromic distance between node pairs, but the model is valid under other definitions of D as well (e.g., the shortest-path distances through a network that contains nodes in $\mathcal{N}$). The *latency* experienced between nodes i and j is denoted by ℓ_{ij} and represents the time needed for data to travel (via some transmission medium) from i to j and vice versa (i.e., the time needed for a request sent from i to be received at j). It is assumed that $\ell_{ij} = \ell_{ji}$ for each $i, j \in \mathcal{N}$, and that these values are deterministic. Later, we discuss in greater detail a method for computing ℓ_{ij}.

In addition to the latency incurred by transmitting data over geographical distances, requests

will also experience delays due to congestion at a particular site. This delay stems from the fact that each virtual hub has a limited request processing capacity (i.e., the transmission rates μ_j), and requests for information will arrive from multiple, independent external sources, thereby leading to queueing effects. Because the arrival stream of requests is the superposition of multiple, independent Poisson streams, the aggregate arrival process of requests to each virtual hub is a Poisson process whose intensity is the sum of the intensities of the individual streams. However, even if the sources are not Poisson, the aggregate arrival process can still be adequately approximated by a Poisson process if the number of sources is large and the traffic intensity at the node is moderate (Albin, 1982). Therefore, we assume that each virtual hub operates as an $M/M/1$ queueing system wherein the single server represents the facility's device for transmitting and/or receiving requests. Denote by T_{ij} the total response time for a request sent from node i and processed at virtual hub j. This total time can be decomposed as

$$T_{ij} = R_j + \ell_{ij},$$

where R_j denotes the total time (in steady state) that an arbitrary request spends at virtual hub j (in queue and being transmitted), and ℓ_{ij} is the latency. It is well known that, in steady state, the sojourn time R_j is exponentially distributed with rate parameter $\mu_j - \Lambda_j$, where Λ_j represents the aggregate demand rate assigned to virtual hub j (Gross, Shortle, Thompson, & Harris, 2008). Therefore, the cumulative distribution function (c.d.f.) of T_{ij} is

$$\mathbb{P}(T_{ij} \leq \alpha) = 1 - e^{-(\alpha - \ell_{ij})\left(\mu_j - \sum_{k \in \mathcal{N}} \lambda_k x_{kj}\right)}, \quad \alpha \geq 0, \tag{1}$$

where α is the maximum tolerable response time and x_{kj} is a binary variable with $x_{kj} = 1$ if requests from node k are satisfied by virtual hub j, and $x_{kj} = 0$ otherwise. More precisely, $\mathbb{P}(T_{ij} \leq \alpha)$ is the likelihood that the total time needed to answer a request from node i at virtual hub j does not exceed the maximum threshold α. Note that we may have $\alpha \leq \ell_{ij}$ for some $i, j \in \mathcal{N}$. However, we require $\alpha > \ell_{ij}$ if node i is assigned to virtual hub j. Before providing a preliminary mathematical programming formulation, we first summarize the problem data and decision variables.

Problem Data:

$\mathcal{N}$: The set of all nodes with $\mathcal{N} = \{1, \ldots, N\}$;

ℓ_{ij}: The transmission latency from MTF i to MTF j (in seconds);

R_j: The probabilistic sum of queueing and transmission time at virtual hub j (in seconds);

λ_i: The average demand rate of node i (requests/sec);

μ_j: The average transmission rate of node j (requests/sec);

α: A threshold for the total response time (in seconds);

H: The maximum allowable number of AVHE sites ($H \leq N$).

Decision Variables:

y_j: A binary variable for virtual hub assignment;

$$y_j = \begin{cases} 1, & \text{if a virtual hub is placed at site } j, \\ 0, & \text{otherwise;} \end{cases}$$

x_{ij}: A binary variable for facility assignment to a virtual hub;

$$x_{ij} = \begin{cases} 1, & \text{if facility } i \text{ is assigned to virtual hub } j, \\ 0, & \text{otherwise.} \end{cases}$$

Next, we discuss a natural optimization model that one might consider for locating locate virtual hubs and assigning the demand of each military treatment facility to exactly one hub. In this preliminary formulation, the objective is to maximize the likelihood that the (steady state) maximum response time at any node in the network does not exceed a fixed threshold duration α. The optimization problem is

$$\max \quad \mathbb{P}\left(\max_{i,j\in\mathcal{N}}\{T_{ij} : x_{ij} = 1\} \le \alpha\right) \tag{2a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}} x_{ij} = 1 \quad \forall\, i \in \mathcal{N} \tag{2b}$$

$$\sum_{j\in\mathcal{N}} y_j \le H \tag{2c}$$

$$x_{ij} \le y_j \quad \forall\, i, j \in \mathcal{N} \tag{2d}$$

$$x_{ij} \in \{0,1\} \quad \forall\, i, j \in \mathcal{N} \tag{2e}$$

$$y_j \in \{0,1\} \quad \forall\, j \in \mathcal{N} \tag{2f}$$

The objective function (2a) can be interpreted as follows. Suppose that N requests are generated, one from each of the nodes in $\mathcal{N}$. The probability in (2a) represents the likelihood that, in steady state, the maximum response time among these N does not exceed the threshold α. Constraint (2b) ensures that each node is assigned to exactly one virtual hub, while constraint (2c) limits the number of virtual hubs to H. Constraint (2d) forbids the assignment of a facility to a node that is not a virtual hub. Finally, constraints (2e) and (2f) force each x_{ij} and y_j to be binary.

Next, basic probability results must be applied to express the objective function (2a). Suppose $X_1, \ldots, X_n$ is an independent sequence of non-negative random variables such that X_i has c.d.f. $F_i(x) = \mathbb{P}(X_i \le x)$, $x \ge 0$, $i = 1, \ldots, n$. It is well known that the random variable $Y \equiv \max\{X_1, \ldots, X_n\}$ has c.d.f.

$$G(y) \equiv \mathbb{P}(Y \le y) = \prod_{i=1}^{n} F_i(y), \quad y \ge 0. \tag{3}$$

Using (1) and (3), we can express (2a) in the multiplicative form

$$\mathbb{P}\left(\max_{i,j\in\mathcal{N}}\{T_{ij} : x_{ij} = 1\} \le \alpha\right) = \prod_{i,j\in\mathcal{N}} \left[1 - e^{-(\alpha-\ell_{ij})(\mu_j - \sum_{k\in\mathcal{N}} \lambda_k\, x_{kj})}\right]^{x_{ij}};$$

hence, the objective function can be expressed as

$$\max \prod_{i,j \in \mathcal{N}} \left[1 - e^{-(\alpha - \ell_{ij})(\mu_j - \sum_{k \in \mathcal{N}} \lambda_k x_{kj})}\right]^{x_{ij}}. \tag{4}$$

This objective equally weights each node's probability of receiving a response within α time units, but it can be easily modified to place greater emphasis on response times for high-volume facilities. This might be accomplished, for example, by replacing the exponent x_{ij} with $\lambda_i x_{ij}$.

The objective (4) contains a product of nonlinear functions of x-variables, which creates significant challenges in developing an efficient solution procedure. Moreover, this objective tends to take very small values, especially in the case of large N or small H, making results somewhat difficult to interpret. For these reasons, the next section presents an alternative optimization model that, like the previous model, favors solutions in which each node's quality-of-service requirement is reliably met. The alternative model's objective corresponds to the quality-of-service reliability of only the least reliable node, thereby avoiding the complexities associated with the multiplicative form (4). We next present the alternative model and demonstrate how to reformulate it as a mixed integer linear program that can be readily solved using commercial software.

AN ALTERNATIVE MODEL

Each term in the product of (4) represents the probability that a request from a given node experiences a response time that does not exceed α. Hence, each term in the product must be close to unity or else the objective is very small. Moreover, because the product magnifies this effect for large N, the objective value is somewhat difficult to interpret. In lieu of objective (4), we consider the problem of maximizing the minimum (across all request-generating nodes) probability that a response time does not exceed α. This model is more practical in that it circumvents the scaling issue associated with the multiplicative form (4), and it appears to be more tractable. While no exact approach for solving (4) is apparent, we demonstrate in this section that the max-min model can be solved as a mixed integer linear program.

Defining a variable δ to represent the objective function value, the max-min problem can be formulated as follows, where X denotes the set of feasible pairs (x, y) satisfying constraints (2b)–(2f):

$$\begin{aligned}
\max \quad & \delta, && \text{(5a)}\\
\text{s.t.} \quad & \delta \le 1 - e^{-(\alpha - \ell_{ij})(\mu_j - \sum_{k \in \mathcal{N}} \lambda_k x_{kj})} + (1 - x_{ij}),\ \forall i, j \in \mathcal{N}, && \text{(5b)}\\
& (x, y) \in X.
\end{aligned}$$

From (5b), it follows that $\delta \le 1 - e^{-(\alpha - \ell_{ij})\mu_j}$ whenever $x_{ij} = 1$. Noting that some x_{ij} must equal one, we have that

$$\delta \le \max\left\{1 - e^{-(\alpha - \ell_{ij})\mu_j} :\ i, j \in \mathcal{N}\right\}. \tag{6}$$

Letting $Q_{ij} = (\alpha - \ell_{ij})\mu_j$, observe that the maximum in (6) is $1 - e^{-Q}$, where $Q \equiv \max\{Q_{ij} :\ i, j \in$

$\mathcal{N}\}$. Thus, for fixed binary $(\bar{x}, \bar{y}) \in X$, an equivalent formulation to (5) is

$$\begin{aligned} \max \quad & \delta, && \text{(7a)} \\ \text{s.t.} \quad & \delta \leq 1 - e^{-(\alpha-\ell_{ij})(\mu_j - \sum_{k\in\mathcal{N}} \lambda_k \bar{x}_{kj})}, \ \forall i, j \in \mathcal{N}, \ \bar{x}_{ij} = 1, && \text{(7b)} \\ & \delta \leq 1 - e^{-Q}, && \text{(7c)} \end{aligned}$$

which (by rearranging and substituting Q_{ij}) is equivalent to

$$\begin{aligned} \min \quad & 1 - \delta, && \text{(8a)} \\ \text{s.t.} \quad & 1 - \delta \geq e^{-Q_{ij} + (\alpha-\ell_{ij}) \sum_{k\in\mathcal{N}} \lambda_k \bar{x}_{kj}}, \ \forall i, j \in \mathcal{N}, \ \bar{x}_{ij} = 1, && \text{(8b)} \\ & 1 - \delta \geq e^{-Q}, && \text{(8c)} \end{aligned}$$

Because $\ln(a)$ is nondecreasing in a, we may replace the objective $1 - \delta$ with $\ln(1 - \delta)$ and (for the same reason), take the natural logarithm of both sides of (8b) and (8c). Thus, the following formulation is also equivalent using the substitution $z = \ln(1 - \delta)$:

$$\begin{aligned} \min \quad & z, && \text{(9a)} \\ \text{s.t.} \quad & z \geq -Q_{ij} + (\alpha - \ell_{ij}) \sum_{k\in\mathcal{N}} \lambda_k \bar{x}_{kj}, \ \forall i, j \in \mathcal{N}, \ \bar{x}_{ij} = 1, && \text{(9b)} \\ & z \geq -Q. && \text{(9c)} \end{aligned}$$

However, this formulation is equivalent to

$$\begin{aligned} \min \quad & z, && \text{(10a)} \\ \text{s.t.} \quad & z \geq -Q(1 - \bar{x}_{ij}) - Q_{ij}\bar{y}_j + (\alpha - \ell_{ij}) \sum_{k\in\mathcal{N}} \lambda_k \bar{x}_{kj}, \ \forall i, j \in \mathcal{N}, && \text{(10b)} \end{aligned}$$

because (a) when $\bar{x}_{ij} = 1$, we must have $\bar{y}_j = 1$, and the right-hand side (r.h.s.) of (10b) equals the r.h.s. of (9b); (b) when $\bar{x}_{ij} = \bar{y}_j = 0$, the r.h.s. of (10b) equals $-Q$; and (c) when $\bar{x}_{ij} = 0$ and $\bar{y}_j = 1$, the r.h.s. of (10b) is no more than $-Q$ because $\mu_j \geq \sum_{k\in\mathcal{N}} \lambda_k \bar{x}_{kj}$ implies $-Q_{ij}\bar{y}_j + (\alpha - \ell_{ij}) \sum_{k\in\mathcal{N}} \lambda_k \bar{x}_{kj} \leq 0$. Freeing x and y, we observe that the formulation

$$\begin{aligned} \min \quad & z, && \text{(11a)} \\ \text{s.t.} \quad & z \geq -Q(1 - x_{ij}) - Q_{ij}y_j + (\alpha - \ell_{ij}) \sum_{k\in\mathcal{N}} \lambda_k x_{kj}, \ \forall i, j \in \mathcal{N}, && \text{(11b)} \\ & (x, y) \in X. \end{aligned}$$

is equivalent to formulation (5). A solution (x, y) is optimal in (11) if, and only if, it is optimal in (5), and the value z is mapped back to δ by noting that $z = \ln(1 - \delta)$.

Formulation (11) is preferable to formulation (5) for a few reasons. First, not only is the new formulation linear, but it is also exact. Second, the linearization has not added a significant number of constraints to the problem. In the next section, we provide a set of computational results illustrating the usefulness of this formulation in solving realistically-sized problems.

COMPUTATIONAL RESULTS

This section presents computational results for 150 randomly-generated problem instances. The parameter values for these instances were estimated using real demand and facility location data. Before presenting the numerical results, detailed descriptions of the data and problem generation procedures are provided.

Description of the Data

Although the MHS is comprised of more than 800 facilities worldwide, we restrict our attention to a subset of 243 facilities located primarily in the United States, Europe and Japan (with some exceptions). Working collaboratively with a private firm – Morgan Borszcz Consulting, LLC, via the U.S. Air Force Institute of Technology – we obtained location data (latitudinal and longitudinal coordinates) for individual MTFs identified at the Tricare website (http://www.tricare.mil/mtf/). The breakdown of facilities in the study is as follows: 20 large medical centers (8.2%); 43 hospitals (17.7%); and 180 clinics (74.1%). These proportions were used to randomly generate MTF networks in five distinct batches partitioned according to problem size: $N = 50, 100, 150, 200$, and 243 nodes. Within each batch, the number of nodes, demand rates, transmission rates, and latency values were fixed. As an example, for all of the cases in which $N = 50$, we generated one set of nodes and the corresponding demand rates, transmission rates, and latency values for that network. We then varied the parameters H and α to assess their effect on solution quality.

The locations and latencies associated with the 243 facilities in the MHS network are known and calculated in advance. For problems of size N, each of the N nodes is first randomly assigned a facility type (clinic, hospital, or medical center). While the facility assignments were randomized within all batches, the ratios of facility types were consistent with the ratios observed in the data: 74.1% clinics, 17.7% hospitals, and 8.2% medical centers. Next, each node was randomly assigned a location from amongst the true 243 locations. These assignments were made by randomly selecting integers (without replacement) from the set $\{1, \ldots, 243\}$ to ensure that each node uniquely corresponds to the location of an actual MTF. This procedure guarantees that the latencies between facilities are consistent with one another, and that the locations represent reasonable locations for treatment facilities.

Next, the average demand rates of the facilities (λ_i, $i \in \mathcal{N}$) were generated. Data for this purpose were available from 152 of the 243 facilities. For each facility, the number of transactions generated by its servers over several hours is known, and this information was used to estimate the average rate at which the facility generates demand (transactions per second). The sample averages of the demand rates for clinics, hospitals, and medical centers are 0.196, 0.313, and 0.856, respectively. For each type of facility, we constructed empirical distribution functions with the aim of fitting the observed data to known parametric distributions. It was hypothesized that, for clinics and hospitals, the data originated from exponential populations with respective means equal to the observed sample averages. In the case of medical centers, sufficient data were available for only 16 of the 20 facilities. Based on these 16 facilities, it was hypothesized that the demand rate follows a continuous uniform distribution on the interval $[0, 1.527]$. To assess goodness-of-fit for all three facility types, we performed two-sided Kolmogorov-Smirnov (K-S) one-sample tests. For the K-S

test, the null and alternative hypotheses are, respectively,

$$H_0 : \widehat{F}(x) = F(x), \text{ for all real values } x$$
$$H_1 : \widehat{F}(x) \neq F(x), \text{ for at least one } x,$$

where $\widehat{F}(x)$ is the empirical distribution function evaluated at x and $F(x)$ is a completely specified, hypothesized distribution (either exponential or uniform here). The K-S statistic is given by (Conover, 1999)

$$T = \sup_{x \in X} \left| \widehat{F}(x) - F(x) \right|,$$

where X is the set of empirical observations with $|X| = n$. The null hypothesis H_0 is rejected at the 0.05 level of significance if $T > T^*$, where T^* is the 95th quantile of the K-S statistic. For the cases in which an exponential distribution is hypothesized (clinics and hospitals), we used the modified K-S critical values (Lilliefors, 1969). Specifically, for $n > 30$, the critical value is $T^* = 1.06/\sqrt{n}$. In the case of medical centers, for which a uniform distribution is hypothesized and $n = 16$, the value T^* is obtained via standard K-S tables (Conover, 1999). The K-S statistics, and their associated critical values, are summarized in Table 1 for each facility type. Table 1 indicates that, at the 0.05 level of significance, we fail to reject the null hypothesis that the demand rates of clinics and hospitals originate from exponential populations (with means equal to their respective sample averages). This table also indicates that we fail to reject the null hypothesis that the demand rate of medical centers originates from a continuous uniform distribution on $[0, 1.527]$.

Table 1: Kolmogorov-Smirnov (T) test statistics for demand generated by facilities.

Facility Type	No. of Observations	Assumed Population	T^*	T
Clinic	101	Exponential	0.105	0.0976
Hospital	35	Exponential	0.179	0.0689
Medical Center	16	Uniform	0.327	0.1208

Next, we randomly generated the average transmission rates (μ_j, $j \in \mathcal{N}$) for each facility in the MTF network. Unfortunately, the actual rates were not available in the data, as they typically are limited by the capacity of the military installation itself. Therefore, we generated values that ensure the existence of feasible solutions of the problem, even for reasonably small hub limits. It was assumed that the transmission rate of each facility type is proportional to its size. Specifically, the values were generated so that, in expectation, the rate of clinics is half that of hospitals, which is half that of medical centers. The rates for clinics, hospitals, and medical centers were sampled from continuous uniform distributions on the intervals $[0.6, 0.73]$, $[1.2, 1.46]$, and $[2.4, 2.93]$, respectively.

Finally, we describe the estimation of latency values using the latitudinal and longitudinal coordinates of 243 military treatment facilities around the world. The distance between two facilities along the earth's surface was approximated by the orthodromic distance, assuming that the earth is spherical. Mathematically, the distance, d, between two points on the sphere can be expressed as $d = R\sigma$, where R is the earth's radius and σ is the central angle between the two points. However,

because this formula can be numerically ill-conditioned, we instead employed the mathematically equivalent haversine formula given by

$$d = 2R \arcsin\left[\left(\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\omega_1 - \omega_2}{2}\right)\right)^{1/2}\right], \tag{12}$$

where ϕ_i and ω_i are the latitude and longitude of coordinate i, respectively. Now, for any two $i, j \in \mathcal{N}$, $j \neq i$, let d_{ij} be the distance between these sites. The latency of a request generated at site i and processed at site j is denoted by ℓ_{ij}. It is important to note that latency will refer only to the time needed for data to travel via some transmission medium. Any delays due to congestion at a particular node are captured in the sojourn time R_j. The latency is approximated by

$$\ell_{ij} = \frac{2d_{ij}}{kc}, \tag{13}$$

where k is a constant determined by the transmission medium and c is the speed of light (approximately 186,282 miles/sec). Under the assumption that the transmission media are fiber optic cables, the denominator of (13) is approximately 121,083 miles/sec. The value d_{ij}/kc is doubled to account for the time needed for a request to be sent and its response returned (i.e., $\ell_{ij} = \ell_{ji}$ for each i, j pair in $\mathcal{N}$).

Results and Discussion

Using the generated data, we considered 150 problem instances by varying the number of nodes (N), the allowable number of hubs (H) and the delay tolerance (α). Table 2 summarizes the parameter values used in the problem instances.

Table 2: Summary of parameter values for the experiments.

Network Size (N)	Allowable Hubs (H)	Tolerable Delay (α)
50	3, 5, 50	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0, 10
100	5, 10, 100	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0, 10
150	8, 15, 150	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0, 10
200	10, 20, 200	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0, 10
243	13, 25, 243	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0, 10

Tables 3 – 6 summarize computational results from these 150 problem instances, which were solved using IBM ILOG CPLEX 12.4 using compute nodes on the University of Arkansas Razor Cluster, each of which contains four 16-core, Opteron Interlagos 6276 2.3GHz CPUs and 32×16 GB DDR3 DIMMs. Each problem was solved to an optimality gap of 4% (0.04), or until a time limit of 5400 seconds was reached. The optimality gap of 4% was chosen on the basis of preliminary runs. Specifically, it was observed that, for many of the large problem instances, CPLEX was able to reach an optimality gap of 4% in a moderate amount of time (e.g., within one hour). However, further reductions to the gap were unattainable, even after allowing the program to run for several

days. The 5400-second time limit applies to the CPLEX run time and does not include the time needed to load the problem data. In the numerical results (see Tables 3 – 6), the reported run time is the sum of the time to load problem data and the CPLEX run time. For all problem instances, the variable z in formulation (11) was restricted to take non-positive values. This modification allowed us to restrict our attention to feasible solutions that induce stable queueing systems. For this reason, the solver was unable to determine feasible solutions in two problem instances, and the run times of these cases are marked as "Timed Out" (see Tables 4 and 6).

Results from instances 1–30 demonstrate that CPLEX is consistently able to obtain optimal, or near optimal, solutions within minutes for the smallest instances considered ($N = 50$). For larger values of N (instances 31–150), the results suggest that the problem is very difficult to solve under either (a) strict response-time thresholds (smaller values of α), or (b) limited resources for installing virtual hubs (smaller values of H). Decreasing either α or H restricts the model, resulting in lower objective values and fewer feasible solutions. Computational results illustrate both of these effects: For instances with small H and small α (e.g., instances 31–39, 61–69, 91–99, 121–129), the objective values tend to be small (no more than 0.7 in most cases) suggesting more than H hubs are needed to guarantee a high likelihood of response time within α seconds. In some cases (e.g., instances 61 and 131), the solver is unable to identify a feasible solution within 5400 seconds, perhaps suggesting that one of the hubs must be assigned more demand than it can process, thereby leading to queueing instability.

The results also indicate that, when H or α are larger (e.g., instances 40–60, 70–90, 110–130, 140–150), the problems are easier to solve. In these cases, CPLEX is consistently able to close the optimality gap within 20% under 5400 seconds and frequently able to close it within 10%. A subset of these instances (51–60, 81–90, 111–120, 141–150) correspond to the special case $H = N$ in which as many as all the MTFs may be selected as virtual hub locations. It is important to note that this special case is non-trivial. Because the candidate hub locations each have different transmission rates (i.e., request-processing capacities), it may be sub-optimal for a virtual hub at node j to service its own requests if μ_j is insufficient. That is, the impact of congestion at this hub would be significant, resulting in a low probability that requests can be processed within α seconds. Computational results verify this point based on the observation that the instances in which $H = N$ require a significant amount of computation time to obtain a reasonably small optimality gap. However, it is interesting to note that the performance difference between cases $H = N$ and $H = 0.1\,N$ is typically very small – in a majority of the instances, the hub assignments are identical between these cases. For example, in comparing the objective values of instances 45 and 55, it is noted that they agree to three significant figures. In a typical solution where $H = N$, the hubs are assigned to medical centers and, occasionally, a few hubs that only assign to themselves. As an example, in the instances with $H = N = 150$, no more than 16 hubs are ever utilized. Results for the largest problem instances ($N = 243$), which were constructed to represent a realistic application of the model, mirror the properties described earlier. As long as H is not too small relative to α, the instances are consistently solved to within 10% of optimality.

Further insights are gained by examining the geographical attributes of the solutions obtained. To this end, let us define the residual capacity of virtual hub j as

$$C_j = \mu_j - \sum_{k \in \mathcal{N}} \lambda_k \, x_{kj}. \tag{14}$$

Refer to the hub with the least residual capacity as the "bottleneck" hub and the hub with greatest residual capacity as the "flexible" hub. The computational results reveal a few interesting properties of the bottleneck hub. First, in terms of geographical distances, it is typically the most remote hub, i.e., it is the hub that is furthest from the greatest number of MTFs in the network. For example, in instances 31–60, the bottleneck hub is a medical center located on Hawaii, which is the furthest hub from 88 of the remaining 99 MTFs. Second, the assignments made to the bottleneck hub are typically from the MTFs which are nearest in proximity. By contrast, the flexible hub is typically the least remote hub (i.e., furthest from the fewest MTFs) and is assigned demand from the MTFs that are furthest away. For example, consider instances 16 ($N = 50$, $H = 5$) and 46 ($N = 100$, $H = 10$) – two of the smaller instances for which the optimality gap is less than 1%. For problem instance 16, the bottleneck hub has eight of the 15 nearest MTFs assigned to it, and only two of the 15 furthest. Similarly, in instance 46, the bottleneck hub has seven of the 15 nearest MTFs assigned to it, and only four of the 50 furthest. By contrast, in instance 16, the flexible hub is assigned eight of the 15 MTFs furthest from it, but only two of the 30 nearest to it. Similarly, in instance 46, the flexible hub is assigned seven of the 10 furthest from it, and only two of the 60 nearest to it.

These observations may have an intuitive explanation. For the max-min formulation, only one of the constraints (11b) needs to be binding at optimality, and this constraint typically corresponds to the bottleneck hub. As such, optimal solutions prioritize making the best assignments to this hub, and the objective value is invariant to changes in the assignments at the other hubs (as long as they do not become the bottleneck hub). Additionally, the large residual capacity at the flexible hub helps to ensure that the response times of requests sent to this hub remain small, even if the latencies are large. Naturally, it will be important to examine the quality of the solution's assignments to hubs whose residual capacities fall between those of the bottleneck and flexible hubs. It is clear that this formulation does not guarantee Pareto optimal solutions with respect to the assignments to these intermediate hubs. While this may appear to be a drawback of our model, solutions that perform better across all hubs might be attained by iteratively solving a sequence of optimization problems of the form of model (11). Each successive iteration can re-optimize by fixing the node and hub assignments to the previous iteration's bottleneck hub. This would result in a smaller problem to be solved at each iteration, thereby reducing the computational time as compared to solving H problems at the original size. Such a re-optimization scheme will be explored in future work.

Table 3: Summary of results for the computational experiments (instances 1–40).

Problem instance	N	H	α	Run time (sec)	Objective value	Solution gap
1	50	3	0.2	74.51	0.1160	2%
2	50	3	0.3	1677.46	0.1741	2%
3	50	3	0.4	1169.72	0.2322	2%
4	50	3	0.5	244.29	0.2861	2%
5	50	3	0.6	77.44	0.3362	2%
6	50	3	0.7	173.21	0.3824	1%
7	50	3	0.8	222.34	0.4258	1%
8	50	3	0.9	119.85	0.4660	1%
9	50	3	1	59.10	0.5019	1%
10	50	3	10	111.04	0.9992	0%
11	50	5	0.2	24.67	0.1964	0%
12	50	5	0.3	15.47	0.3491	2%
13	50	5	0.4	19.38	0.4423	1%
14	50	5	0.5	15.76	0.5180	1%
15	50	5	0.6	16.20	0.5835	1%
16	50	5	0.7	20.64	0.6400	1%
17	50	5	0.8	17.35	0.6889	1%
18	50	5	0.9	16.64	0.7312	1%
19	50	5	1	16.95	0.7677	1%
20	50	5	10	30.06	1.0000	0%
21	50	50	0.2	13.74	0.1943	1%
22	50	50	0.3	15.64	0.3494	1%
23	50	50	0.4	15.55	0.4423	1%
24	50	50	0.5	20.94	0.5180	1%
25	50	50	0.6	15.95	0.5835	1%
26	50	50	0.7	14.83	0.6400	1%
27	50	50	0.8	13.56	0.6889	1%
28	50	50	0.9	11.30	0.7312	1%
29	50	50	1	12.61	0.7677	1%
30	50	50	10	12.61	1.0000	0%
31	100	5	0.2	5426.04	0.2019	14%
32	100	5	0.3	5408.79	0.3036	17%
33	100	5	0.4	5436.43	0.3886	23%
34	100	5	0.5	5406.13	0.4624	18%
35	100	5	0.6	5429.78	0.5270	18%
36	100	5	0.7	5415.19	0.5863	16%
37	100	5	0.8	5453.22	0.6357	15%
38	100	5	0.9	5438.71	0.6804	13%
39	100	5	1	5427.93	0.7204	12%
40	100	5	10	2537.78	1.0000	0%

Table 4: Summary of results for the computational experiments (instances 41–80).

Problem instance	N	H	α	Run time (sec)	Objective value	Solution gap
41	100	10	0.2	797.45	0.2891	0%
42	100	10	0.3	838.87	0.4436	0%
43	100	10	0.4	1959.08	0.5666	4%
44	100	10	0.5	1920.71	0.6560	2%
45	100	10	0.6	1935.88	0.7242	2%
46	100	10	0.7	1934.42	0.7775	1%
47	100	10	0.8	5415.72	0.7727	7%
48	100	10	0.9	1993.00	0.8507	2%
49	100	10	1	1947.87	0.8863	1%
50	100	10	10	1915.28	1.0000	0%
51	100	100	0.2	252.40	0.2891	0%
52	100	100	0.3	302.23	0.4439	0%
53	100	100	0.4	1870.32	0.5690	3%
54	100	100	0.5	1866.44	0.6579	2%
55	100	100	0.6	1865.96	0.7249	1%
56	100	100	0.7	1888.33	0.7764	1%
57	100	100	0.8	1879.12	0.8230	1%
58	100	100	0.9	1862.87	0.8582	1%
59	100	100	1	1877.17	0.8861	1%
60	100	100	10	1891.83	1.0000	0%
61	150	8	0.2	Timed Out	–	–
62	150	8	0.3	5423.65	0.3232	50%
63	150	8	0.4	5404.07	0.4100	43%
64	150	8	0.5	5470.20	0.4667	45%
65	150	8	0.6	5406.88	0.5430	35%
66	150	8	0.7	5403.96	0.5998	31%
67	150	8	0.8	5401.70	0.6467	28%
68	150	8	0.9	5403.19	0.6916	24%
69	150	8	1	5424.14	0.7383	20%
70	150	8	10	3647.22	1.0000	0%
71	150	15	0.2	5443.51	0.3279	7%
72	150	15	0.3	5459.22	0.4650	6%
73	150	15	0.4	2558.04	0.5767	3%
74	150	15	0.5	2549.93	0.6492	4%
75	150	15	0.6	2603.26	0.7195	3%
76	150	15	0.7	2618.73	0.7881	1%
77	150	15	0.8	2549.74	0.8237	2%
78	150	15	0.9	2548.10	0.8652	1%
79	150	15	1	2569.20	0.8882	1%
80	150	15	10	2608.11	1.0000	0%

Table 5: Summary of results for the computational experiments (instances 81–120).

Problem instance	N	H	α	Run time (sec)	Objective value	Solution gap
81	150	150	0.2	1328.85	0.3371	2%
82	150	150	0.3	2615.38	0.4833	2%
83	150	150	0.4	2616.65	0.5902	1%
84	150	150	0.5	2641.55	0.6722	1%
85	150	150	0.6	2608.14	0.7399	0%
86	150	150	0.7	2632.67	0.7922	0%
87	150	150	0.8	2536.49	0.8344	0%
88	150	150	0.9	2621.66	0.8674	0%
89	150	150	1	2992.78	0.8941	0%
90	150	150	10	2617.62	1.0000	0%
91	200	10	0.2	5404.78	0.2215	43%
92	200	10	0.3	5406.67	0.3187	37%
93	200	10	0.4	5407.98	0.3372	58%
94	200	10	0.5	5408.01	0.4073	51%
95	200	10	0.6	5404.58	0.4683	45%
96	200	10	0.7	5411.29	0.5226	41%
97	200	10	0.8	5404.39	0.5722	37%
98	200	10	0.9	5405.99	0.6133	34%
99	200	10	1	5407.07	0.6534	30%
100	200	10	10	5418.59	1.0000	0%
101	200	20	0.2	5412.64	0.2581	23%
102	200	20	0.3	5415.99	0.3899	12%
103	200	20	0.4	5406.88	0.4859	10%
104	200	20	0.5	5411.91	0.5291	16%
105	200	20	0.6	5412.37	0.6279	8%
106	200	20	0.7	5414.06	0.6803	8%
107	200	20	0.8	5409.08	0.6953	12%
108	200	20	0.9	5413.72	0.7680	7%
109	200	20	1	5408.78	0.8030	6%
110	200	20	10	5413.55	1.0000	0%
111	200	200	0.2	5415.15	0.3144	17%
112	200	200	0.3	5412.65	0.4493	10%
113	200	200	0.4	5421.25	0.5495	9%
114	200	200	0.5	5403.15	0.6310	8%
115	200	200	0.6	5410.65	0.7003	6%
116	200	200	0.7	5428.23	0.7497	6%
117	200	200	0.8	5403.12	0.7966	5%
118	200	200	0.9	5410.81	0.8294	5%
119	200	200	1	5411.44	0.8616	4%
120	200	200	10	5418.92	1.0000	0%

Table 6: Summary of results for the computational experiments (instances 121–150).

Problem instance	N	H	α	Run time (sec)	Objective value	Solution gap
121	243	13	0.2	5412.92	0.1181	132%
122	243	13	0.3	5410.04	0.1742	119%
123	243	13	0.4	5417.26	0.2260	110%
124	243	13	0.5	5430.27	0.2749	101%
125	243	13	0.6	4715.79	0.3206	93%
126	243	13	0.7	5427.56	0.3608	87%
127	243	13	0.8	5418.04	0.4038	79%
128	243	13	0.9	5406.54	0.4395	74%
129	243	13	1	5431.89	0.4736	69%
130	243	13	10	5428.03	0.9978	0%
131	243	25	0.2	Timed Out	–	–
132	243	25	0.3	5428.54	0.3549	8%
133	243	25	0.4	5421.01	0.4311	10%
134	243	25	0.5	5410.35	0.5252	5%
135	243	25	0.6	5412.49	0.5840	6%
136	243	25	0.7	5408.71	0.6500	4%
137	243	25	0.8	5415.13	0.7006	3%
138	243	25	0.9	5413.62	0.7372	4%
139	243	25	1	5421.12	0.7695	4%
140	243	25	10	5418.96	1.0000	0%
141	243	243	0.2	5418.60	0.2898	11%
142	243	243	0.3	5415.18	0.4042	10%
143	243	243	0.4	5406.71	0.4974	9%
144	243	243	0.5	5411.38	0.5808	7%
145	243	243	0.6	5420.15	0.6420	7%
146	243	243	0.7	5421.78	0.6961	7%
147	243	243	0.8	5415.05	0.7435	6%
148	243	243	0.9	5426.63	0.7835	6%
149	243	243	1	5417.13	0.8152	5%
150	243	243	10	5414.18	1.0000	0%

CONCLUSIONS

In this article, we have considered the problem of locating AVHE resources in order to provide remote access to electronic health records at MTFs throughout the MHS system. For this problem, we have contributed two optimization models that seek to locate virtual hubs (and assign responsibility for each MTF's requests to hubs) in order to guarantee that each MTF's requests will be satisfied within a fixed time threshold with maximum likelihood. In our models, delays in responding to EHR requests occur due to (a) the geographical distance between the MTF generating an EHR request and its assigned virtual hub, and (b) congestion at the virtual hub. We demonstrated how to formulate one of the models, a max-min model, as a mixed integer linear program that can be solved directly using commercially available solvers. Extensive computational results that utilized representative demand and location data illustrate the effectiveness of this formulation in solving realistically-sized problems to optimality gaps of less than 10%. The computational time needed, and the quality of solutions, were shown to be strongly influenced by the problem's parameter values. A close examination of the solutions revealed that the bottleneck hub, which is typically the most geographically remote hub, affects the solution most significantly. A method for determining solutions that may perform better across the entire network was briefly discussed.

More generally, the optimization problem modeled in this paper appears to be quite difficult to solve, even in the case when facilities can be installed at all candidate locations. In this case, all of the location variables are fixed and the problem reduces from a location-allocation problem to a simple allocation problem. An efficient algorithm for this version of the problem might lead to improved decomposition-based approaches for solving the full location-allocation problem. However, the nonlinear congestion effects combined with binary allocation variables pose significant challenges to the development of an efficient algorithm for the allocation problem. This is a promising area for future developments. Additional follow-on investigations may explore other important factors in the design of a network of virtual hubs. For instance, if connectivity is lost between MTFs and data centers as a result of a natural disaster or cyber-attack, the consequences could be devastating. It will be important to develop a more refined model to optimally locate virtualization resources and allocate health records in a way that is resilient to failures.

ACKNOWLEDGEMENTS

The authors are grateful to two anonymous referees and Professor Richard Deckro for helpful comments that have improved the content and presentation of this work. The research was sponsored, in part, by the Air Force Research Laboratory via Oak Ridge Association of Universities (ORAU). Computational results were obtained using computing resources supported in part by the National Science Foundation under grants ARI 0963249, MRI 0959124 (Razor), EPS 0918970 (CI TRAIN), and a grant from the Arkansas Science and Technology Authority, managed by the Arkansas High Performance Computing Center.

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., & Drezner, Z. (2009). The multiple server center location problem. *Annals of Operations Research*, *167*(1), 337–352.

Aboolian, R., Berman, O., & Krass, D. (2012). Profit maximizing distributed service system design with congestion and elastic demand. *Transportation Science*, *46*(2), 247–261.

Albin, S. (1982). On Poisson approximations for superposition arrival processes in queues. *Management Science*, *28*(2), 126–137.

Ball, M. O., & Lin, F. L. (1993). A reliability model applied to emergency service vehicle location. *Operations Research*, *41*(1), 18–36.

Berman, O. (1985). Locating a facility on a congested network with random lengths. *Networks*, *15*(3), 275–293.

Berman, O., & Drezner, Z. (2007). The multiple server location problem. *Journal of the Operational Research Society*, *58*(1), 91–99.

Berman, O., Huang, R., Kim, S., & Menezes, M. (2007). Locating capacitated facilities to maximize captured demand. *IIE Transactions*, *39*(11), 1015–1029.

Berman, O., & Krass, D. (2004). Facility location problems with stochastic demands and congestion. In Z. Drezner & H. Hamacher (Eds.), *Facility location: Applications and theory.* New York, NY : Springer-Verlag.

Berman, O., Krass, D., & Wang, J. (2006). Locating service facilities to reduce lost demand. *IIE Transactions*, *38*(11), 933–946.

Berman, O., & Larson, R. C. (1985). Optimal 2-facility network districting in the presence of queuing. *Transporation Science*, *19*(3), 261–267.

Berman, O., Larson, R. C., & Chiu, S. S. (1985). Optimal server location on a network operating as an $M/G/1$ queue. *Operations Research*, *33*(4), 746–771.

Berman, O., Larson, R. C., & Parkan, C. (1987). The stochastic queue p-median problem. *Transporation Science*, *21*(3), 207–216.

Berman, O., & Mandowsky, R. R. (1986). Location-allocation on congested networks. *European Journal of Operational Research*, *26*(2), 238–250.

Boffey, B., Galvão, R., & Espejo, L. (2007). A review of congestion models in the location of facilities with immobile servers. *European Journal of Operational Research*, *178*(3), 643–662.

Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association*, *32*(1), 101–118.

Conover, W. (1999). *Practical nonparametric statistics, third edition.* New York, NY : John Wiley & Sons, Inc.

Elhedhli, S. (2005). Exact solution of a class of nonlinear knapsack problems. *Operations Research Letters*, *33*(6), 614–624.

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

Filippi, C., & Romanin-Jacur, G. (1996). Optimal allocation of two fixed service units acting as $M/G/1$ queues. *Transportation Science*, *30*(1), 60–74.

Gross, D., Shortle, J., Thompson, J., & Harris, C. (2008). *Fundamentals of queueing theory, 4th edition.* New York, NY : John Wiley & Sons, Inc.

Lilliefors, H. W. (1969). On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown. *Journal of the American Statistical Association*, *64*(325), 387–389.

Marianov, V., Boffey, T. B., & Galvão, R. D. (2009). Optimal location of multi-server congestible facilities operating as $M/E_r/m/N$ queues. *Journal of the Operational Research Society*, *60*(5), 674–684.

Marianov, V., & ReVelle, C. (1994). The queuing probabilistic location set covering problem and some extensions. *Socio-Economic Planning Sciences*, *28*(3), 167–178.

Marianov, V., & ReVelle, C. (1996). The queueing maximal availability location problem: A model for the siting of emergency vehicles. *European Journal of Operational Research*, *93*(1), 110–120.

Marianov, V., & Serra, D. (1998). Probabilistic, maximal covering location-allocation models for congested systems. *Journal of Regional Science*, *38*(3), 401–424.

Marianov, V., & Serra, D. (2001). Hierarchical location-allocation models for congested systems. *European Journal of Operational Research*, *135*(1), 195–208.

Marianov, V., & Serra, D. (2002). Location-allocation of multiple-server service centers with constrained queues or waiting times. *Annals of Operations Research*, *111*(1–4), 35–50.

Marianov, V., & Serra, D. (2003). Location models for airline hubs behaving as $M/D/c$ queues. *Computers and Operations Research*, *30*(7), 983–1003.

Mohammadi, M., Jolai, F., & Rostami, H. (2011). An $M/M/c$ queue model for hub covering location problem. *Mathematical and Computer Modelling*, *54*(11–12), 2623–2638.

ReVelle, C., & Hogan, K. (1989a). The maximum availability location problem. *Transportation Science*, *23*(3), 192–200.

ReVelle, C., & Hogan, K. (1989b). The maximum reliability location problem and α-reliable p-center problem: Derivatives of the probabilistic location set covering problem. *Annals of Operations Research*, *18*(1), 155–173.

Toregas, C., Swain, R., ReVelle, C., & Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, *19*(6), 1363–1373.

Vidyarthi, N., & Jayaswal, S. (2014). Efficient solution of a class of location-allocation problems with stochastic demand and congestion. *Computers and Operations Research*, *48*(1), 20–30.

Wang, Q., Batta, R., & Rump, C. M. (2002). Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of Operations Research*, *111*(1–4), 17–34.

Zhang, L., & Rushton, G. (2008). Optimizing the size and locations of facilities in competitive multi-site service systems. *Computers and Operations Research*, *35*(2), 327–338.

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.