IIE Transactions (2009) **41**, 467–481
Copyright © "IIE"
ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170802382758

Ensuring feasibility in location problems with stochastic demands and congestion

OPHER BARON[1], ODED BERMAN[1], SEOKJIN KIM[2] and DMITRY KRASS[1,*]

[1] *Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, Ontario, Canada M5S 3E6*
E-mail: krass@rotman.utoronto.ca
[2] *Department of Information Systems and Operations Management, Sawyer Business School, Suffolk University, 8 Ashburton Place, Boston, MA 02108, USA*

Received May 2007 and accepted June 2008

A location problem with stochastic demand and congestion where mobile servers respond to service calls originating from nodes is considered. The problem is of the set-covering type: only servers within the coverage radius of the demand-generating node may respond to a call. The service level constraint requires that at least one server must be available to respond to an arriving call, with some prespecified probability. The objective is to minimize the total number of servers. It is shown that earlier models quite often overestimate servers' availability and thus may lead to infeasible solutions (i.e., solutions that fail to satisfy the service level constraint). System stability conditions and lower bounds on system availability are developed by analyzing the underlying partially accessible queueing system. These lead to the development of two new models for which feasibility is guaranteed. Simulation-based computational experiments show that the proposed models achieve feasibility without significantly increasing the total number of servers.

[Supplementary materials are available for this article. Go to the publisher's online edition of *IIE Transactions* for the following free supplemental resource: Appendix of Tables of Computational Results for Section 7.]

Keywords: Location, network, queueing, congestion

1. Introduction

The fundamental problem facing emergency system planners can be described as follows. Customers residing at the nodes of a transportation network generate stochastic streams of service calls. These calls are served by mobile servers housed at a number of facilities. In emergency settings, calls must be handled promptly—which means that only a server located within a certain *coverage radius* of the customer may provide service. Moreover, since servers may be unavailable due to congestion, it is important to ensure that an incoming call finds a free server with a sufficiently high probability. Given the high cost of maintaining each additional server (a single police car staffed around the clock costs over one million dollar per year—see, e.g., Larson (1975), inflation adjusted), a service system planner must balance the service levels achieved with the cost incurred. This paper deals with two of the most critical strategic decisions for such systems: location of facilities and allocation of server capacity.

This problem belongs to the general class of Location Problems with Stochastic Demands and Congestion (LPSDC) reviewed in Berman and Krass (2002). One of the most important applications for mobile-server LPSDC models is the location of emergency service facilities such as ambulance, police and fire stations. The ability to quickly respond to a service call is particularly important in such problems. Potential applications also exist in other areas, including the location of service centers, sales offices, and cars by car sharing services (e.g., Zipcar.com).

Most mobile-server LPSDC problems combine aspects of classical location problems with the dynamics of spatially distributed queueing systems (Larson and Odoni, 1981) and are analytically intractable. Thus, a number of simplifying assumptions are usually made to obtain tractable models. Standard assumptions are that call arrival processes are Poisson and call service times are exponentially distributed. Even under these conditions, no analytical expressions exist for expected system performance measures such as server availability at a given customer node ("node availability"). This has led to the development of several models that employ various estimates of node availability in order to enforce the service level constraints. Our paper includes an examination of several best-known models in this area. We show that the availability estimates used in these models may result in systems that significantly underachieve the

[*] Corresponding author

desired availability levels—i.e., these models fail to obtain feasible solutions. Through a series of computational experiments we demonstrate that this phenomenon is not uncommon—the vast majority of instances we generated had at least one demand-generating node where the desired availability was not met.

The LPSDC models we consider belong to the class of location set-covering problems, where the objective is to minimize the total number of servers required to provide adequate service (see Berman and Krass (2002) for an overview of the other main direction in LPSDC research—the median-type problems where the objective is to minimize the total response time).

The stochastic elements in set-covering models were introduced by Daskin (1983). Daskin makes the following three simplifying assumptions.

1. Servers operate independently of one another—thus the busy fraction of a server (the probability of finding a particular server busy at any time) is independent of that of other servers.
2. The busy fraction is identical for all servers.
3. The busy fraction is irrespective of allocation decisions.

The servers' busy fraction, p, is treated as an external parameter. Indeed, the congestion aspect of the underlying system is not explicitly captured in Daskin's model, however, this model has led to several models that integrate congestion and seek to relax some of the basic assumptions made listed above.

Batta *et al.* (1989) attempt to relax the assumption that the busy fraction of different servers are independent. Following Larson (1975), they introduce an adjustment factor to account for interdependence between servers. However, they still treat p as an external parameter. The first model to make the busy fraction p an endogenous parameter was developed by ReVelle and Hogan (1989a, 1989b), who use region-based estimates of p, but still retain Daskin's assumption that servers within each region are independent. The next significant step in set-covering LPSDC model development was taken by Marianov and ReVelle (1994, 1996) who represent each region as a multi-server Markovian queue and use queueing-based formulas to estimate node availability. While they use an $M/M/k/k$ loss system representation, assuming that any call that cannot find a free server is routed to a back-up system, their approach easily extends to $M/M/k$ systems (where incoming calls are queued) or other standard queueing models. Their model inherits ReVelle and Hogan's assumption that the probability of a server being busy is region-specific and is identical for all servers in each region. As discussed in Section 3 below, neither the Revelle–Hogan nor Marianov–Revelle models guarantee the availability requirements in the underlying system.

Ball and Lin (1993) developed the first model that ensures system feasibility, but under fairly stringent assumptions. In particular, they assume that service times are deterministic and equal to $T > 0$ and derive facility-specific lower bounds for server availability. Their approach also extends to the case where service times are stochastic and bounded above by T, however, in this case the availability estimate becomes loose, which may lead to solutions that require an unrealistic number of servers. The approach does not easily extend to the case where service times are stochastic and unbounded (e.g., exponential). Further discussion of this model is provided in Section 3.

Borras and Pastor (2002) provide the ex-post evaluation of the availability level of several known models (including the ones listed above) by simulation, observing that desired availability is often not met. We note that their simulation operates as a loss system—i.e., any incoming call that cannot find a free server is dropped; in our simulation experiments, all calls are queued. They also suggest a new model formulated similar to the Ball–Lin model, but incorporating the estimate for the busy fraction of the ReVelle–Hogan model. While their model behaves well in computational experiments, it still does not guarantee a feasible solution. As mentioned earlier, our approach focuses on deriving provably feasible models via the analysis of the underlying queuing network.

The key contribution of this paper is the development of lower bounds for the node availability of the underlying systems. These bounds allow us to develop location models that guarantee feasibility with an increase of at most 20–30% in the total number of servers in comparison to the ReVelle–Hogan and Marianov–Revelle models, which as shown in Section 6 are often infeasible.

The plan for the remainder of the paper is as follows. In the next section we formulate the LPSDC model considered in the paper. Section 3 reviews three main models from the literature and demonstrates that their solutions may be infeasible. In Section 4 we analyze the underlying queuing network, representing it as a partially-accessible queueing system and deriving easily-verifiable stability conditions and lower bounds on system availability. In Section 5 these bounds are used to derive two new location models. In Section 6 we briefly discuss the issues involved in extending the two models to more general queueing systems. Section 7 presents the results of a series of computational experiments that use simulation to evaluate the performance of the existing and new models. Section 8 contains concluding remarks and directions for future research.

2. The set-covering LPSDC and mobile servers

2.1. *Basic definitions*

We consider an undirected network $G = (N, L)$, where N is the node set and L is the link set. For $i, j \in N$, $d(i, j)$ denotes the shortest distance between i and j on G. The set-covering-type LPSDC is defined by the following elements: demand points (nodes), facilities, servers, buffers and service discipline. Each of these elements are discussed below.

2.1.1. *Demand points*

We assume that customers are located at the nodes of the network, the demand process for each node $i \in N$ is an independent Poisson process with rate λ_i and a call at node i can be served only by servers within a prespecified distance $\delta > 0$ from node i; δ is the *coverage radius* of the system.

The node subset $N_i = \{j \in N | d(i, j) \leq \delta\}$ is called *region* N_i. It represents the set of nodes that can be covered by a service facility at i. We also refer to node i as the *center* of region N_i and all the other nodes in N_i as the *peripheral nodes* of N_i.

For a subset $V \subset N$, we denote the set of nodes *accessible* from V to be $\bigcup_{j\in V} N_j$ (this is the set of potential facility locations that may serve at least one call from V). Region N_i is said to be "isolated" if $\bigcup_{j\in N_i} N_j = N_i$ and "overlapping" otherwise. Calls from an isolated region have to be served within the region, while those from an overlapping region may be served from outside the region.

For a subset $V \subset N$ of nodes, we denote the total rate at which demand calls originate from V by $\lambda(V) = \sum_{n\in V} \lambda_n$. We illustrate the definitions above with the following example, which will be used throughout the paper:

Example 1. (A three-node path example.)

Consider a three-node path: $N = \{1, 2, 3\}$, $d(1, 2) = 1.9$ and $d(2, 3) = 2$. Suppose that $\lambda_1 = 2$, $\lambda_2 = 1$, $\lambda_3 = 2$ and $\delta = 2$. The network is illustrated in Fig. 1.

There are three regions: $N_1 = \{1, 2\}$, $N_2 = \{1, 2, 3\}$ and $N_3 = \{2, 3\}$. Observe that node 2 is the center of N_2 and the two peripheral nodes (1 and 3) in N_2 are not within the coverage radius from each other. Note that regions N_1 and N_3 are overlapping and that region N_2 is isolated.

The demand rates for the three regions are $\lambda(N_1) = \lambda_1 + \lambda_2 = 3$, $\lambda(N_2) = \lambda_1 + \lambda_2 + \lambda_3 = 5$ and $\lambda(N_3) = \lambda_2 + \lambda_3 = 3$.

2.1.2. *Facilities*

We assume that the set of potential facility locations (or sites) consists of the node set N. We note that this assumption can be made without loss of generality. This is so because N can be extended to be the finite dominating set (see Berman *et al.* (1985) for details). We also note that all the results in the paper continue to hold if the set of potential locations consists of some subset of N (i.e., if some nodes cannot be used as facility locations).

The number of servers $x(j) \geq 0$ to be placed at site $j \in N$ is a decision variable. We denote an *allocation vector* by $\mathbf{x} = (x(1), \ldots, x(|N|))$. If a facility is located at site j, $x(j) > 0$, otherwise $x(j) = 0$. Let $X = \{j \in N | x(j) > 0\}$ be the set of facility nodes, $|X|$ be the number of facility nodes and $X_i = \{j : x(j) > 0,\ j \in N_i\} = X \cap N_i$ be the set of facility nodes within region N_i for $i \in N$. For a subset of nodes $V \subset N$, let $X_V = \bigcup_{i\in V} X_i$ be the set of facilities accessible to the nodes in V.

Node $i \in N$ is said to be "covered" if and only if there is at least one facility within the coverage radius of i, i.e., $X_i \neq \emptyset$. We require that a feasible allocation vector cover all nodes on the network, i.e., $|X_i| > 0$ for each $i \in N$ (this requirement can be modified to include only nodes with positive demand without affecting any results in the paper).

2.1.3. *Servers, buffers and service discipline*

If located, a facility at $j \in N$ houses $x(j) > 0$ identical *mobile* servers that provide service to nodes in region N_j. One server is required to serve each call. Once assigned to a service call, a server is routed to the node from which the call originated, performs on-scene service and then returns to its home facility. Thus, the total service time is the summation of the times required for these three activities, as well as other relevant components such as the time required to dispatch the call, provide the necessary information and supplies to the service unit, etc. Unless stated otherwise, we assume that the total service time is exponentially distributed with identical service rate μ. This assumption has been made in the location literature in the context of emergency services (Larson, 1975; Ball and Lin, 1993; Marianov and ReVelle, 1994; Marianov and ReVelle, 1996) even though the travel time component is not likely to have an exponential distribution. However, approximating the distribution of the overall service time as exponential is reasonable when the coverage radius is not too large so that travel times are a minor component of the overall service time. We discuss the issues arising from relaxing this assumption for our models in Section 6.

We assume that an infinite-capacity buffer is positioned at each node; the service calls originating from the node that cannot find an available server are added to the buffer. We also assume the following dispatch policy: assign calls from node i to a closest available server in N_i (ties are broken randomly). If no servers in N_i are available, the call joins the queue at node i. A facility serves calls within its coverage radius in a First-Come First-Serve (FCFS) manner (ties are broken arbitrarily). We refer to this service discipline as *Dynamic Discipline* since the determination of which service facility will handle a particular call depends on the state of the system at the time when the call is dispatched. Observe

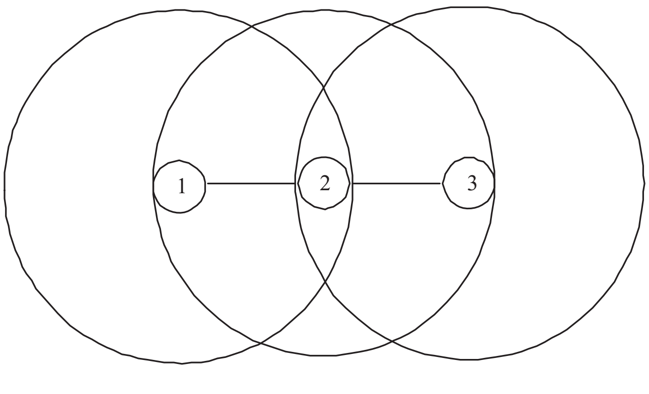

Fig. 1. Three regions of the three-node path in example 1.

that for each facility j our service discipline satisfies the following work-conserving property within N_j.

Definition 1. (Bertsimas, 2007) A work-conserving service discipline does not allow a server to be available whenever there are customers in the system and does not cause customers to leave before completing their service requirement.

2.2. *Model formulation and node availabilities*

Assuming that an allocation vector $\mathbf{x}$ is given and the underlying queueing system specified by $\mathbf{x}$ is stable, we define the availability $A_i(\mathbf{x})$ of node i to be the steady-state probability that a new call from node i finds an available server in X_i (i.e., within N_i). We assume that quantity $\alpha \in (0, 1)$, representing the minimum required availability for all customer nodes, has been specified. In emergency system applications, the required minimum availability is usually quite high, with α values of 80% or higher. For example, Advanced Life Support in the Toronto Emergency Medical Services responds to 95.5% of the most important calls and 70.7% of the second most important calls (http://www.toronto.ca/ems/overview/statistics.htm, accessed September 8, 2007). Under the Dynamic Discipline, we attempt to minimize the total number of servers on a network while satisfying the required availability α at all nodes. This leads to the following mathematical programming formulation, which we call problem (P):

$$(\mathrm{P}): \qquad \min \sum_{j \in N} x(j),$$

subject to

$$\begin{aligned} &A_i(\mathbf{x}) \geq \alpha \quad \forall i \in N, \\ &x(j) = 0, 1, \ \dots \quad \forall j \in N. \end{aligned} \tag{1}$$

Note that constraints (1) imply that the coverage conditions $|X_i| > 0$ hold for all i. The main difficulty in model (P) is the estimation of availabilities $A_i(\mathbf{x})$.

It is tempting to focus on each region N_i, $i \in N$ and treat it as an $M/M/k$ queue with arrival rate λ (N_i) and number of servers $k = |X_i|$, because the availability formulas are readily available for such systems. In fact, this is the approach followed by many previous papers. Indeed, when region i is isolated and all servers are located in the center of the region, it behaves as an $M/M/k$ system. However, when N_i overlaps with some other regions, this approach may lead to incorrect estimates of availabilities (over- or underestimated). For example, consider an overlapping region N_i where servers are concentrated in the center node i. Since some of the peripheral nodes have access to servers outside of N_i, treating N_i as an isolated region will lead to underestimates of node availabilities (because the actual load faced by servers at i is less then $\lambda(N_i)$.) On the other hand, if the servers in N_i are located at a peripheral node j that also belongs to other regions, then these servers will only be able to provide service to calls from N_i part of the time (because they are also serving calls outside of N_i), and thus the effective service rate will be less than $k\mu$, which may lead to an overestimate of node availabilities for N_i.

To summarize, even when the interarrival and service times are assumed to be Markovian and all servers are identical, the resulting system does not behave like a set of separable $M/M/k$ queues. The dispatching policy used in model (P) creates overlapping service regions, causing different servers to face different loads that cannot be analytically calculated.

To illustrate these issues consider:

Example 2. This is a continuation of example 1.

Suppose $\mu = 3$ and $\alpha = 0.65$. Recall that $\lambda_1 = \lambda_3 = 2$ and $\lambda_2 = 1$.

Consider the allocation vector $\mathbf{x}^{\mathrm{c}} = (0, 3, 0)$ (i.e., three servers located at node 2). This creates an $M/M/3$ system since each node is accessible to each server. Using the standard availability formula for this system, (e.g., Gross and Harris (1985)), we compute: $A_1(\mathbf{x}^{\mathrm{c}}) = A_2(\mathbf{x}^{\mathrm{c}}) = A_3(\mathbf{x}^{\mathrm{c}}) = 0.7 > \alpha$. Hence, $\mathbf{x}^{\mathrm{c}}$ is feasible for model (P). Note, however, that treating N_1 as an isolated region consisting of nodes 1, 2 and having three servers, we would estimate the availabilities as $\widehat{A}_1(\mathbf{x}^{\mathrm{c}}) = \widehat{A}_2(\mathbf{x}^{\mathrm{c}}) = 0.91$, a serious overestimate of the actual availabilities given above.

Now consider an allocation vector $\mathbf{x}^{\mathrm{d}} = (1, 1, 1)$, i.e., one server located at each of the three nodes. The resulting system is much harder to analyze since calls from node 1 can only access servers at nodes 1 and 2, while calls from node 2 can access all three servers. Similarly, servers at node 1 and 3 have a different load than the server at node 2. We are not aware of any close-form analytical expressions for estimating the availability of such a system. We use Monte Carlo simulation to estimate availabilities, yielding $A_1(\mathbf{x}^{\mathrm{d}}) = 0.61$, $A_2(\mathbf{x}^{\mathrm{d}}) = 0.74$, $A_3(\mathbf{x}^{\mathrm{d}}) = 0.61$. It follows that this allocation vector is not feasible since availabilities at nodes 1 and 3 are less than $\alpha = 0.65$. Note, however, that this infeasibility would be impossible to detect if each region was approximated as an $M/M/k$ queue. For example, region $N_1 = \{1, 2\}$ has two servers and a demand rate of three. The corresponding $M/M/2$ queueing system has an availability of 0.67 leading to the wrong conclusion that the availability of node 1 is adequate. Note that locating two servers at node 2, i.e., using location vector $\mathbf{x}^{\mathrm{l}} = (0, 2, 0)$, leads to (via simulation) $A_1(\mathbf{x}^{\mathrm{l}}) = A_2(\mathbf{x}^{\mathrm{l}}) = A_3(\mathbf{x}^{\mathrm{l}}) = 0.24 < 0.65$. Thus, three servers is the minimal number that can achieve feasibility in this example.

As discussed in the following section, the effects described above are present in many previous models for problem (P), leading to possibly infeasible solutions.

3. Representative models in the literature

Because analytical expressions for node availability $A_i(\mathbf{x})$ in problem (P) are not available, it is necessary to use an approximation $\widehat{A}_i(\mathbf{x})$ in place of $A_i(\mathbf{x})$ for $i \in N$ in constraints

(1). We say that such an approximation is *valid* if for any allocation vector $\mathbf{x}$ satisfying the coverage constraint $|X_i| > 0$ for all $i \in N$, we have $\widehat{A}_i(\mathbf{x}) \leq A_i(\mathbf{x})$. In the current section we briefly discuss several well-known models for problem (P) and show that the approximations of node availability used in these models are often not valid, possibly leading to infeasible solutions.

We denote the availability of an $M/M/k$ queueing system with demand rate d and service rate μ by $A(d, k)$. We note that $A(d, k)$ is decreasing in d and increasing in k. Furthermore, we denote the minimum number of servers to achieve the required availability α in this system by $m(d) = \min\{k \geq 1 \mid A(d, k) \geq \alpha\}$ that is a non-decreasing step function of d. Thus, $A(d, m(d)) \geq \alpha$.

3.1. *The ReVelle–Hogan model*

ReVelle and Hogan (1989a, 1989b) approximate $A_i(\mathbf{x})$ assuming that: (i) all demand calls from N_i are served by servers in N_i; (ii) servers in N_i only serve demand from N_i; and (iii) availability of each server is independent of that of other servers. We note that an alternative interpretation of assumptions (i) and (ii) is that the number of demand calls originating outside of N_i that are served by servers in N_i is roughly the same as the number of demand calls originating from N_i that are served by servers outside of N_i. We will refer to this model as (RH).

From the assumptions above it follows that for $i \in N$, the fraction of time that each server in N_i is busy is $\rho_i = \lambda(N_i)/(|X_i|\mu)$ and (using the server independence assumption) the estimate availability of node i is

$$\alpha \leq A_i(\mathbf{x}) \approx \widehat{A}_i(\mathbf{x}) = 1 - \rho_i^{k_i} = 1 - \left(\frac{\lambda(N_i)}{k_i\mu}\right)^{k_i}.$$

While this constraint is non-linear, since $\widehat{A}_i(\mathbf{x})$ is increasing in $|X_i|$, it can be converted to the linear constraints:

$$\sum_{j \in N_i} x(j) \geq m_i \qquad \forall i \in N,$$

where m_i is the smallest $k_i \geq 0$ satisfying $1 - [\lambda(N_i)/(k_i\mu)]^{k_i} \geq \alpha$. Thus, replacing constraints (1) in model (P) with the constraints above, we get a linear integer program. The solution of this model may lead to infeasible solutions, as illustrated in the following example.

Example 3. Consider the same system as in example 2 above.

It can be verified that $m_1 = m_3 = 2$ and $m_2 = 3$. An optimal solution for the linear integer program formulation of model (RH) for this example is $\mathbf{x}^{\mathrm{r}} = (1, 1, 1)$, which was shown to be infeasible in the previous example. We note that this is not the only optimal solution to the integer program above—the other alternative optima include $\mathbf{x} = (0, 3, 0)$, which is feasible, and $\mathbf{x} = (1, 2, 0)$ or $(0, 2, 1)$ not feasible—simulation results show that the availability for the node that does not contain a facility is only 0.5475. We conclude that an optimal solution for model (RH) may be infeasible for model (P).

3.2. *The Marianov–ReVelle model*

Marianov and ReVelle (1994) extended the ReVelle–Hogan model by introducing a more explicit, queueing-based representation of region-specific availability estimates. They use assumptions (i) and (ii) of the (RH) model, but replace the binomial-based estimate with a queuing-based one. We will refer to this model as (MR).

Marianov and ReVelle estimate availability based on the $M/M/k/k$ loss system, i.e., they allow no buffers in their model assuming that any calls that cannot be handled immediately are lost (or transferred to backup service providers). However, in the current model queues are allowed; thus we use the analytical formula for $M/M/k$ systems instead. We thus estimate $\widehat{A}_i(\mathbf{x}) = A(\lambda(N_i), m(\lambda(N_i)))$. We note that the infeasibility effects demonstrated below do not disappear even if the original assumptions are retained—see Borras and Pastor (2002).

The constraints in model (MR) are $A_i(\mathbf{x}) \approx \widehat{A}_i(\mathbf{x}) = A(\lambda(N_i), m(\lambda(N_i))) \geq \alpha$. As before these constraints are linearized into

$$\sum_{j \in N_i} x(j) \geq m(\lambda(N_i)) \qquad \forall i \in N, \tag{2}$$

These constraints are used in place of constraints (1) in integer program (P).

For example 2, the value of $m(\lambda(N_i)) = m_i$, as defined for model (RH) in example 3 above. Thus, the model (MR) for this particular example is identical to model (RH). As discussed above, in both models (RH) and (MR) there is an optimal solution that is feasible, and three other alternative optima that are not feasible. Neither model contains guidance of how the feasible solution should be chosen over the infeasible ones. Moreover, it is not hard to construct larger examples where all optimal solutions are infeasible for both models.

3.3. *The Ball–Lin model*

Ball and Lin (1993) took a reliability-based approach to estimate the node availability, and we will refer to their model as (BL). They start by estimating the availability of a facility. Suppose a facility at $j \in N$ has $x(j)$ servers. The maximum demand rate this facility faces is $\lambda(N_j)$—this assumes that *all* calls from region N_j must be handled by this facility. Model (BL) assumes that the service times are constant and equal to T, while the arrivals are Markovian. Let $D(j)$ be the Poisson random variable representing the total number of calls generated from region N_j during time period T. Then the probability that facility j has no available servers during time period $(t, t + T)$ is given by $P[D(j) \geq x(j)]$. Note that for an overlapping region this is an overestimate, because some of the calls from N_j may be handled by servers from other facilities.

Now consider the availability of a demand-generating node $i \in N$. A call from i during time $(t, t+T)$ will find no available servers if none of the facilities in X_i have available servers, implying

$$\widehat{A}_i(\mathbf{x}) = 1 - \prod_{j \in X_i} P[D(j) \geq x(j)].$$

Ball and Lin prove that $\widehat{A}_i(\mathbf{x}) \leq A_i(\mathbf{x})$. Clearly, this remains valid when the service times are random variables bounded above by T.

The constraints $\widehat{A}_i(\mathbf{x}) \geq \alpha$, can be linearized through introducing the binary variables $x(j)_k = 1$, if k servers are placed at facility j; 0 otherwise. After taking logarithms, we get:

$$\sum_{j \in N_i} \sum_{k=1}^{K_j} -\log\{\Pr[D(j) \geq k]\}\, x(j)_k \geq -\log(1-\alpha) \quad \forall i \in N. \tag{3}$$

The remainder of the formulation is similar to problem (P) with the variable substitution $x(j) = \sum_{k=1}^{K_j} k x(j)_k$, where K_j is the maximum number of servers that can be located at node j.

Compared with the models discussed above, this formulation involves a much larger number of decision variables and a highly dense constraint matrix, making solvability an issue (Ball and Lin partially address this by developing some families of valid inequalities for the formulation above).

However, a more fundamental issue is the choice of parameter T. As noted earlier, the service times in emergency systems usually exhibit a high level of variability. Even if an upper bound on the range of possible values exists, using its value for parameter T might result in an unrealistic number of servers in the optimal solution. For example, consider a police station where service times range from 1 to 60 minutes. Setting $T = 60$ minutes in the model requires the station to have enough units to be able to dispatch a new unit to every call that is expected to arrive during an hour. This is likely to be an order of magnitude more than the number of units stationed at any police station we are aware of (Metropolitan Toronto Police Simulation Project, 1997, background data). Thus, T has to be set to some percentile, rather than the upper bound, of the service time distribution. Not only is the choice of the percentile not clear, but the guarantee that the resulting solution is feasible is lost in the process. This issue is illustrated in the following example.

Continuing with the network from example 2, we tested the integer program for model (BL) with different values of T (we used $K_j = 10$ for all j). Since service times are assumed to be exponential in example 1, we tested two values for T: the 50th percentile, leading to $T = 0.231$, and the 75th percentile, with $T = 0.462$. As before, we use simulation to estimate actual node availabilities.

With $T = 0.231$ the optimal solution is $\mathbf{x}^1 = (0, 2, 0)$; as discussed in example 2 above this yields node availabilities of 0.24—far below the required $\alpha = 0.65$. The choice of $T = 0.462$ leads to $\mathbf{x}^2 = (0, 4, 0)$ with the estimated node availability of 0.9. While this solution is feasible, it locates too many servers—the allocation vector (0, 3, 0) is sufficient to achieve feasibility. In our experience, this is a typical outcome for model (BL): an "aggressive" choice of T leads to infeasibility, while the "conservative" choice leads to an overly large number of servers.

4. Queueing analysis of the LPSDC

This section develops our main results, including stability conditions and bounds for availability for a partially accessible queueing system. It is important to emphasize that even though this paper deals mainly with Markovian interarrival and service times, the results in this section apply to significantly more general settings. For example, our main results in Section 4.3 are valid for any continuous service time distribution with finite variance.

4.1. *Multi-class multi-server queueing systems and their stability*

After a location–allocation decision is made, the resulting system specified on a network can be viewed as a Multi-Class Multi–Server Queueing (MCMSQ) system with restricted customer–server matching (Caldentey and Kaplan, 2007). For example, in Fig. 2 we consider two matchings between customer classes 1, 2, 3 and servers s, t, u. The resulting MCMSQ systems can be represented as a bipartite graph with node set $B = \{N, X\}$, where N is the set of customer classes and X is the set of servers. An incomplete bipartite graph, such as the one depicted in Fig. 2(a), indicates that some servers are not accessible to some of the customer classes; we call such a system a Partially Accessible Queueing (PAQ) system. A completely connected graph is a special case where each server is accessible to all customer classes as in Fig. 2(b); we call this a Fully Accessible Queueing (FAQ) system.

A vast literature is available on MCMSQ systems (also known as systems with lane selection), see e.g., Schwartz (1974). Dynamic disciplines in an MCMSQ system have received much attention in the literature. Dai (1995) provides a unified approach to the stability conditions for open queueing networks. Positioning an infinite-capacity buffer at each server, Foley and McDonald (2001) derive a stability condition for the MCMSQ system and some asymptotic performance measures for the "join the shortest queue" system with two servers. Caldentey and Kaplan (2007) provide a stability condition for the queueing system with an infinite-capacity buffer at each node by showing that its

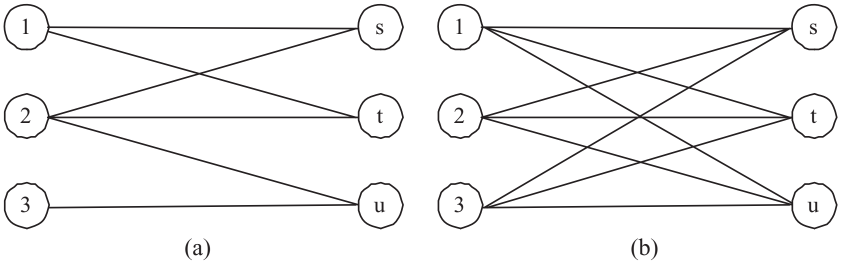

Fig. 2. (a) A partially accessible queueing system; and (b) a fully accessible queueing system.

corresponding fluid limit version is stable. Similarly to the current paper, they assume that class-i customers arrive according to a homogeneous Poisson process with rate λ_i, server k service in an exponential time with rate μ_k, and a newly available server serves customers at different queues in an FCFS manner. Let S be the set of servers and define the set of servers accessible from the subset $V \subset N$ of customer classes to be $S(V)$. Then, the following necessary and sufficient condition for stability holds.

Proposition 1. (*Caldentey and Kaplan, 2007*) *Assume Markovian arrivals and service and that infinite-capacity buffers are available for each customer class. Then, the resulting MCMSQ system is stable, if and only if*

$$\sum_{i \in V} \lambda_i < \sum_{s \in S(V)} \mu_s \quad \forall\, V \subset N. \tag{4}$$

In fact, this stability condition holds under any work-conserving service discipline (Caldentey and Kaplan, 2007). If an MCMSQ system is stable, by the PASTA (Poisson Arrivals Sees Time Averages) property (Wolff, 1989), typical queueing performance measures for each node customer class exist.

4.2. *PAQ systems in LPSDC*

The concepts described above are directly applicable to the model we consider. Each node can be viewed as a customer class. An allocation vector $\mathbf{x}$ with facility location set X creates an MCMSQ system which can be represented by a bipartite graph G^{b} with node set $B(\mathbf{x}) = \{N, X\}$ where node $i \in N$ is connected to facility $f \in X$ iff $f \in X_i$.

For example, the two allocation vectors $\mathbf{x}^{\mathrm{d}} = (1, 1, 1)$ and $\mathbf{x}^{\mathrm{c}} = (0, 3, 0)$ in our example 2 above create two different systems depicted on Fig. 3: the vector $\mathbf{x}^{\mathrm{d}}$ creates a PAQ system on Fig. 3(a), while the centered allocation vector $\mathbf{x}^{\mathrm{c}}$ creates a FAQ system on Fig. 3(b). We note that the graph G^{b} need not be connected. For example, if the coverage radius was set to 0.5 in example 2, the vector $\mathbf{x}^{\mathrm{d}}$ above would lead to the graph on Fig. 3(c), which consists of three separate $M/M/1$ FAQ systems.

In general, each allocation vector on a network creates one or more PAQ systems (the multiple-system case occurs when the graph G^{b} is disconnected). Certain allocation vectors may also lead to FAQ systems, whose performance measures are available from the results for the $M/M/k$ queues. Because we assume that $\mu_{\mathrm{k}} = \mu \;\; \forall k$, the stability condition (4) simplifies to

$$\lambda(V) = \sum_{i \in V} \lambda_i < \mu \sum_{j \in X_V} x(j) \quad \forall\, V \subset N \tag{5}$$

(recall that X_V is the set of facility nodes accessible to nodes in V). This condition can be considered an extension of the stability condition for the $M/M/k$ system in the sense that the average load of a server accessible to any subset of call classes is less than one. Since the condition above needs to be checked for $2^{|N|}$ possible subsets, it is not easy to verify when N is large. In contrast, the following sufficient condition for the stability of the PAQ system induced by an allocation vector $\mathbf{x}$ is easily verifiable.

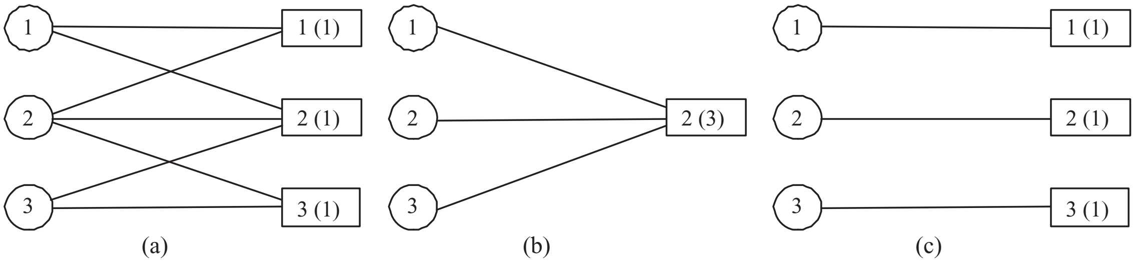

Fig. 3. (a) A PAQ system created by $\mathbf{x}^{\mathrm{d}} = (1, 1, 1)$; (b) a FAQ system created by $\mathbf{x}^{\mathrm{c}} = (0, 3, 0)$; and (c) a system created by $\mathbf{x}^{\mathrm{d}} = (1, 1, 1)$ with reduced coverage radius consisting of three FAQ systems.

Proposition 2. *Any allocation vector* $\mathbf{x}$ *such that*:

$$|X_i| \geq 1 \quad \forall\, i \in N,$$
$$x(j) > \lambda(N_j)/\mu \quad \forall\, j \in X,$$

results in a stable PAQ system.

Proof. For an arbitrary subset $V \subset N$ let $V_j = V \cap N_j$ for each $j \in X$. Since for each $i \in N$, $|X_i| > 0$, it follows that $i \in N_j$ for some $j \in X$. Thus, $\bigcup_{j\in X} N_j = N$, and therefore $V \subseteq \bigcup_{j\in X} V_j$. Moreover, if $j \notin X_V$, then clearly $V_j = \emptyset$. Thus

$$V \subseteq \bigcup_{j\in X_V} V_j.$$

Note, however, that sets $V_j,\ j \in X$ may not be disjoint. Thus, $\lambda(V) \leq \sum_{j\in X_V} \lambda(V_j)$. Moreover, for each $j \in X$, we have $\lambda(V_j) \leq \lambda(N_j) < \mu x(j)$, where the last inequality holds by hypothesis. It follows that:

$$\lambda(V) \leq \sum_{j\in X_V} \lambda(V_j) < \sum_{j\in X_V} \mu x(j),$$

which completes the proof. ■

Observe that Proposition 2 holds for any service discipline under which the average load faced by facility j is at most $\lambda(N_j)$ for each $j \in X$. Note that the average load is exactly $\lambda(N_j)$ only when each node in N_j is assigned only to facility j and to no other facilities on the network.

We also note that condition of Proposition 2 is sufficient but not necessary. For example, the two allocation vectors $\mathbf{x}^{c} = (0, 3, 0)$ and $\mathbf{x}^{d} = (1, 1, 1)$ in example 2 above both satisfy condition (5) and thus lead to stable systems, but only $\mathbf{x}^{c}$ satisfies the condition of Proposition 2.

4.3. *Stochastic orders for waiting times in general PAQ systems*

We are interested in lower bounds for availabilities that can be translated into tractable forms of constraints in mathematical programming models similar to the ones discussed in Section 3. Moreover, we want these bounds to be sufficiently tight so that the optimal solution of the model does not place an excessive number of servers. To that end, we construct a modified system whose waiting times of nodes are valid (i.e., lower bound) approximations for those in the underlying system. We note that the results of this section do not require the Markovian assumption.

Consider an allocation $\mathbf{x}^{U}$ that induces a PAQ system U. Assume that U satisfies the following conditions.

1. Interarrival times at node i follow a renewal process with rate λ_i.
2. Demand processes at different nodes are independent.
3. Continuous service times of a server are drawn from some general distribution with mean $1/\mu$ and finite variance.
4. The allocation satisfies the condition in Proposition 2.

We now construct a modified system M by decoupling the underlying system U based on facilities into $|X|$ FAQ systems.

As before, we represent U by a bipartite graph G^{b} with node set (N, X). Consider some node $i \in N$. For every facility $j \in X_i$ there is a link (i, j) in G^{b}. We construct the modified system M as follows. First define a new node set N' by copying each node $i \in N$ into $|X_i|$ nodes $ij,\ j \in X_i$, each with the same arrival process as node i. The node set for the bipartite graph representing M is (N', X)—i.e., the facility set is the same as in U. Create a link (ij, j) for each $ij \in N'$ and $j \in X$ such that $i \in N_j$ in U (that is, node i belongs to the coverage area of the facility j). Since node ij in M is copied from node i in U, the interarrival times for each node ij in $N_j^{m} = \{ij,\ i \in N_j\}$ are drawn from some general distribution with mean $1/\lambda_i$. Since U and M have identical server sets, the service distribution in M is identical to U—with mean $1/\mu$. As in U, the service discipline in M is work-conserving. An available server in M serves calls from different queues at accessible demand nodes in an FCFS manner. Ties are broken randomly.

Note that for each $j \in X$, the region N_j in U is mapped into region N_j^{m} in M, moreover for $k, j \in X$ with $k \neq j$, the sets N_j^{m} and N_k^{m} are disjoint. Thus, each facility node j together with the set N_j^{m} in M forms an $G/G/x(j)$ FAQ system. Note also that $\lambda(N_j^{m}) = \lambda(N_j) < \mu x(j)$ where the last inequality follows since the allocation $\mathbf{x}^{U}$ is assumed to satisfy Proposition 2. This shows that the same condition holds for system M. Thus, M consists of $|X|$ stable separate $G/G/k$ FAQ systems. Moreover, when interarrival and service times are Markovian for all $j \in X$ the $G/G/x(j)$ system reduces to an $M/M/x(j)$ system with demand rate $\lambda(N_j^{m})$ and service rate μ.

The construction of the modified system M is illustrated in Fig. 4 for system U induced by allocation $\mathbf{x}^{U} = (2, 0, 1)$ in example 2. The original PAQ system has been decoupled into separate $M/M/1$ and $M/M/2$ queues. Note that the overall arrival rate in M is $\lambda_1 + 2\lambda_2 + \lambda_3$ that is higher than in the original system.

Intuitively, facilities in M are more "loaded" than in U. Moreover, since M consists of separate FAQ systems, the node availability in M can be obtained from

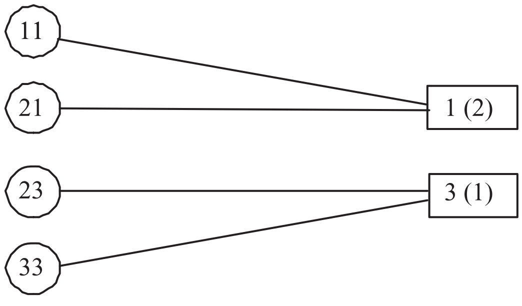

Fig. 4. The modified system M corresponding to the PAQ system created by $\mathbf{x}^{d} = (2, 0, 1)$.

standard queueing formulas (particularly for $M/M/k$ systems). These availabilities will be lower bounds for the corresponding node availabilities in U. To substantiate this intuition we establish a stochastic order relationship between the waiting times in systems U and M. We use the following definition (Shaked and Shanthikumar, 1994):

Definition 2. Let Y and Z be two random variables such that $\Pr\{Y > u\} \leq \Pr\{Z > u\}$ for all real u. Then Y is said to be smaller than Z in the usual stochastic order (denoted by $Y \leq_{st} Z$).

Without loss of generality, we assume that the service time of calls is known at their arrivals. Thus, for any assignment rule of calls to facilities, the actual assignment is known at the time of arrivals. With this assumption, the waiting time for calls is known at their arrival and thus, the virtual waiting times at each facility are also known. We let $W_j(t)$ and W_j^{∞} be the virtual waiting time at time t and the steady-state waiting time for facility j in U, respectively. Let $W_j^m(t)$ and $W_j^{m,\infty}$ denote these quantities in M. Thus, $W_j(t)$ is the waiting time of a node-i call arriving at time t and assigned to facility j. If the call is assigned to facility $f \neq j$, the waiting time of this call is $W_f(t)$.

Suppose node i is covered by facility j in U. While node-ij calls (i.e., all calls from N_j^m) are assigned to facility j in M, a node-i call might not be assigned to facility j in U (when this node is covered by more than one facility). Thus, facility j in M faces heavier demand than facility j in U and we expect its waiting time to be longer and its availability to be lower. Theorem 1 and Corollary 1 below establish that this is indeed the case.

Theorem 1. *Assume that calls are assigned to servers according to the Dynamic Discipline. Let the underlying system U and its modified system M be empty at* $t = 0$. *Then*

$$W_j(t) \leq_{st} W_j^m(t) \quad \forall j \in X, \ \forall t \geq 0. \tag{6}$$

Moreover, if $W_j^{m,\infty}$ *is well defined, so is* W_j^{∞} *and*

$$W_j^{\infty} \leq_{st} W_j^{m,\infty} \quad \forall j \in X. \tag{7}$$

Proof. Denoting the arrival time distributions of U and M by $F_A(a)$ and $G_{A^m}(a^m)$, respectively, we observe that $A^m \leq_{st} A$. Thus, Equations (6) and (7) follow from the discussion of the Kiefer–Wolfowitz recursion in Section 11.6 of Wolff (1989), despite the fact that this recursion is not a Markov Chain in the underlying system. Specifically, Equation (99) on page 495 of Wolff (1989) holds and it also holds as a sample-path result. Also, Equation (99) together with the remarks following it implies our Equation (6). Then, Equation (97) together with its discussion on page 494 of Wolff (1989) implies our Equation (7). ■

Observe that $W_j^{\infty} = 0$ is equivalent to the event that at least one server in facility j is available. We use $P(W_j^{\infty} = 0)$ and $P(W_j^{m,\infty} = 0)$ to bound the steady-state availability, $A_i(\mathbf{x})$.

Corollary 1. *For any node* $i \in N$, *the steady-state availability* $A_i(\mathbf{x})$ *in U is bounded as follows*:

$$A_i(\mathbf{x}) \geq \max_{j \in X_i} P\left(W_j^{\infty} = 0\right) \geq \max_{j \in X_i} P\left(W_j^{m,\infty} = 0\right) \quad \forall i \in N. \tag{8}$$

Moreover, when node-i calls follow a Poisson process we have:

$$A_i(\mathbf{x}) \geq 1 - \prod_{j \in X_i} \left(1 - P\left(W_j^{m,\infty} = 0\right)\right) \quad \forall i \in N. \tag{9}$$

Proof. Equation (8) is straightforward from Theorem 1. Because Poisson random variables have the discrete "new better than used" property (e.g., Proposition 3 and Theorem 1 in Ball and Lin (1993)), Equation (9) follows from Ball and Shanthikumar (1994). ■

When the arrival and service times are Markovian, each facility j in M operates as an $M/M/x(j)$ queue, and analytical expressions for availability of an $M/M/x(j)$ system can replace $P\left(W_j^{m,\infty} = 0\right)$ in (8) and (9) above.

We also note that Theorem 1 can be used to derive lower bounds on other performance measures related to the waiting times in the original system U, including limits on the average wait in queue, maximal wait in queue, etc. The results will be similar to the expressions in Corollary 1 above.

5. New stochastic location models

In this section we develop two new location models for the LPSDC problem (P) formulated in Section 2 above. The models are motivated by the results in the previous section and are guaranteed to achieve the required node availability.

5.1. *Model (BBKK1)*

This model is motivated by the bound (8) above. Recall that for an $M/M/k$ queue with arrival rate λ and specified availability level $\alpha \in (0, 1)$, the quantity $m(\lambda)$ is defined as the minimum number of servers needed to achieve the required level of availability. Note also that for each $i \in N$, the value of $m(\lambda(N_i))$ can be computed during the preprocessing stage, and that an allocation placing more than the minimal required number of servers at a facility cannot be optimal. This leads to the following integer program:

$$\text{(BBKK1)} \quad \min \sum_{j \in N} m(\lambda(N_j)) y_j,$$

$$\text{subject to} \quad \sum_{j \in N_i} y_j \geq 1 \quad \forall i \in N, \quad y_j = 0, 1 \quad \forall j \in N.$$

The binary decision variable y_j indicates whether a facility is located at site j (when $y_j = 1$) or not. The constraint specifies that each node $i \in N$ must belong to at least one region N_j centered at one of the open facilities. The objective function specifies that if a facility is open at $j \in N$, it must house exactly $m(\lambda(N_j))$ servers. The following corollary of Theorem 1 now follows.

Corollary 2. *Let* $\mathbf{y}$ *be an optimal solution to model* (*BBKK1*), *and define the allocation vector* $\mathbf{x}^1$ *as follows*:

$$x^1(j) = \begin{cases} 0 & \text{if } y_j = 0, \\ m(\lambda(N_j)) & \text{if } y_j = 1, \end{cases}$$

for $j \in N$. *Then* $A_i(\mathbf{x}^1) \geq \alpha$ *for all* $i \in N$.

Proof. First note that by the feasibility of $\mathbf{y}$ in model (BBKK1) and the definition of $\mathbf{x}$, the resulting system satisfies the stability condition in Proposition 2. The result now follows from Equation (8) in Corollary 1. ■

Note that the integer program (BBKK1) is a special case of the classic Set-Covering Location Problem (SCLP), and thus is relatively easy to solve even for large-scale instances. Observe also that the model can be easily extended to other queueing performance measures (such as mean waiting time, mean queue length, etc.) simply by redefining $m(\lambda(N_j))$. An additional advantage of (BBKK1) is that it tends to centralize servers at the open facilities, thus limiting their number. The obvious disadvantage of (BBKK1) is that by ignoring the fact that a node in N_j may get service from a facility other than j it may overestimate the required number of servers. Nevertheless, as the following example shows, the estimated number of servers can be tight (i.e., smaller number of servers cannot maintain feasibility).

Example 4. (BBKK1) for the three-node path of example 2.

As noted in the discussion of model (MR) earlier, the m values for this example are: $m(\lambda(N_1)) = 2, m(\lambda(N_2)) = 3, m(\lambda(N_3)) = 2$. The unique optimal solution of model (BBKK1) for example 1 is $\mathbf{y} = (y_1 = 0, y_2 = 1, y_3 = 0)$ and the corresponding allocation vector $\mathbf{x}^c = (0, 3, 0)$ is feasible and, as discussed in example 2, is also tight.

5.2. *Model (BBKK2)*

This model is similar in structure to model (BL), but without the assumption that the service times are constant. It is motivated by bound (9) above that we rewrite as

$$\prod_{j \in X_i} [1 - A(\lambda(N_j), x(j))] \leq 1 - \alpha. \tag{10}$$

As long as Equation (10) is satisfied, the required level of availability is ensured in M, and thus in U as well. Taking the logarithm of Equation (10) leads to

$$\sum_{j \in N_i} \sum_{k=1}^{K_j} - \log(1 - A(\lambda(N_j), k)) y_{jk} \geq - \log(1 - \alpha) \quad \forall i \in N. \tag{11}$$

Replacing Equation (3) in (BL) by Equation (11) leads to model (BBKK2). Note that as will be shown below, we can always set $K_j = m(\lambda(N_j))$ (in applications capacity restrictions may force the value of K_j even lower). We note that the ability to easily incorporate capacity restrictions on individual facilities is an attractive feature of (BBKK2); in contrast, the solution produced by model (BBKK1) may lose feasibility if the number of servers at a facility is forced below $m(\lambda(N_j))$.

The following result establishes the feasibility of (BBKK2) with respect to the original model (P) and shows that an optimal solution to (BBKK2) will never use more servers than an optimal solution to (BBKK1).

Corollary 3. *Let* $\mathbf{y}$ *be an optimal solution to model* (*BBKK2*), *and define the allocation vector* $\mathbf{x}^2$ *as follows*:

$$x^2(j) = k \text{ iff } y_{jk} = 1, \quad j \in N.$$

Then

(a) $A_i(\mathbf{x}^2) \geq \alpha$ *for all* $i \in N$;
(b) *If* $K_j \geq m(\lambda(N_j))$ *for all* $j \in N$, *then* $\sum_{j \in N} x^2(j) \leq \sum_{j \in N} x^1(j)$, *where* $\mathbf{x}^1$ *is the allocation vector derived from an optimal solution to* (*BBKK1*) *as in Corollary* 2 *above.*

Proof. First note that since $\alpha > 0$, constraint (11) implies that $|X_i| > 0$ for any feasibility solution $\mathbf{y}$. Moreover, since all the terms on the left-hand side of (11) for which $k \leq \lambda(N_j)/\mu$ are equal to zero, in any optimal solution $x^2(j) > \lambda(N_j)/\mu$ must hold if $y_{jk} > 0$. Thus, under the allocation vector $\mathbf{x}^2$ the resulting system satisfies the stability condition of Proposition 2. Part (a) now follows immediately from Equation (9) in Corollary 1.

Part (b) follows by observing that, by Corollary 2, the allocation vector $\mathbf{x}^1$ satisfies Equation (10) and thus is a feasible solution to (BBKK2). ■

We note that the integer program (BBKK2) does not have the attractive SCLP structure of (BBKK1), and is significantly harder to solve. Moreover, as the computational results in Section 6 show, the expected improvement of (BBKK2) over (BBKK1) was minimal in our simulation experiments.

Example 5. (BBKK2) for the three-node path of example 2.

As discussed earlier, we have $\lambda(N_1) = \lambda(N_3) = 3, \lambda(N_2) = 5$, $\alpha = 0.65$ and $\mu = 3$. Thus, the minimal number of servers k for a facility located at any of the nodes is two (availabilities in constraint (11) are zero for $k = 1$). On the other hand, as computed earlier, $m(\lambda(N_1)) = m(\lambda(N_3)) = 2$ and

$m(\lambda(N_2)) = 3$. Thus, $K_1 = K_3 = 2$, $K_2 = 3$ and the only decision variables we need to consider are $y_{12}, y_{22}, y_{23}, y_{32}$. The objective function is $\min(y_{12} + y_{22} + y_{23} + y_{32})$.

It can also be verified that $\log(1 - A(2, 2)) = \log(1 - 0.667) = -0.477$, $\log(1 - A(5, 2)) = \log(1 - 0.242) = -0.120$, $\log(1 - A(5, 3)) = \log(1 - 0.702) = -0.523$, and $\log(1 - \alpha) = -0.456$. Thus, constraint (11) for $i = 1$ is

$$0.477y_{12} + 0.12y_{22} + 0.523y_{23} \geq 0.456,$$

(with standard integer programming pre-processing techniques, this would be converted to $y_{12} + y_{23} \geq 1$). The other constraints in (BBKK2) are derived similarly.

The unique optimal solution to (BBKK2) is $y_{23} = 1$, with all other decision variables set to zero. This results in an allocation vector $\mathbf{x}^c = (0, 3, 0)$—the same feasible and tight solution obtained for (BBKK1).

6. Extending the new models to more general queueing systems

In this section we briefly discuss the issues involved in extending models (BBKK1) and (BBKK2) to more general queueing systems. In particular, we discuss the possibility of non-Markovian service, non-Markovian arrivals and finite-buffer systems.

First, we observe that both models developed above can be used as heuristics for any queueing system, as long as the formulas for $m(\lambda(N_j))$ and $A(\lambda(N_j), x(j))$ are adjusted appropriately for $j \in N$. We are, however, more interested in whether the models remain provably feasible under the modified conditions.

For all three extensions discussed above, gaps remain in the current methodology in order for the feasibility to hold. The most problematic appears to be the extension to a general call arrival process. The arrivals to each facility may no longer form a renewal process, impacting both Caldentey and Kaplan's (2007) stability condition (4) and our waiting time bounds (8, 9).

The relaxation of the exponential service assumption (as mentioned in the Introduction, this is—arguably—the weakest assumption in the model) appears to be somewhat less problematic. As long as the stability condition (4) can be established, both models (BBKK1) and (BBKK2) will produce feasible solutions since our waiting time bounds (8, 9) do hold for $M/G/N$ systems.

Finally, the extension to systems with finite buffer capacity appears to be possible. Such systems are automatically stable, so a stability condition is not required. However, Theorem 1 does need to be re-established; the current proof relies on the results in Wolff (1989), which may not hold for finite buffer systems. A new proof, perhaps following the arguments of the proof of similar bound in Kim (2005), is required for this case.

7. Computational experiments

In previous sections we claimed that many of the previously available models for the LPSDC problem may lead to infeasible solutions. We also developed two models where feasibility is guaranteed. However, several important questions remain.

1. How prevalent is the feasibility issue, i.e., how likely are infeasible solutions to arise in the previously available models and what level of infeasibility can be expected?
2. How practical is the "fix" suggested by the new (BBKK1) and (BBKK2) models—does it achieve feasibility at the cost of placing an excessive number of servers and facilities (much more than really needed to achieve feasibility)?
3. How large an advantage does (BBKK2) hold over (BBKK1)? From Corollary 3 (b) we know that an optimal solution to (BBKK2) will not require more servers than (BBKK1), but does this translate into practical gains?

In order to answer these questions we conducted a series of computational experiments.

For each experiment we generated a random network with $N = 20, 30$ or 50 nodes and link lengths $l(i, j) \sim \text{Uniform}(1, 50)$. We computed the shortest distance matrix for this network, as well as the average shortest distance $\bar{d}$. The coverage radius was set to $\delta = (0.5)\bar{d}$, $(0.75)\bar{d}$ and $(1.0)\bar{d}$. The arrival rates λ_i, $i \in N$ were drawn from a Uniform (1, 10) distribution. The values used for the service rate μ were 20, 35 and 50. Finally, the required availability level α was set to 0.65, 0.75, 0.85 and 0.95. Altogether, 108 random instances were generated. For each instance we obtained optimal solutions to models (RH), (MR), (BL), (BBKK1) and (BBKK2) using the CPLEX integer programming Solver. Since model (BL) assumes a constant service time T, while the service is exponential in our experiments, we set T to the 50th and 75th percentiles of the service time distribution—the corresponding results are referred to as (BL(50%)) and (BL(75%)). Once the allocation vector $\mathbf{x}$ was determined for each model, we estimated the actual expected availability by running a discrete-event simulator for the network; the number of simulation iterations is set to $|N|(500\,000)$; one iteration is an event of either a demand arrival at a node or a service completion by a server. For each instance we computed three availability measures: the number of "infeasible" nodes—i.e., nodes with $A_i(\mathbf{x}) < \alpha$, the "minimum deviation" defined as $\min_{i \in N}(A_i(\mathbf{x}) - \alpha)$ and the "maximum deviation" defined as $\max_{i \in N}(A_i(\mathbf{x}) - \alpha)$ representing the minimum and maximum differences between the achieved and required server availability at all nodes. While we only report a summary of our experiments here, the detailed results for $|N| = 50$ are available in the online Appendix.

The average and maximal proportions of nodes with infeasibilities for all models, the average and minimum values

Table 1. Fraction of nodes with infeasibility, minimum and maximum deviations from α for the six models

	(RH) (%)	*(MR) (%)*	*(BL)* *(50%) (%)*	*(BL)* *(75%) (%)*	*(BBKK1) (%)*	*(BBKK2) (%)*
Ave. % infeas.	28.28	7.57	27.18	0	0	0
Max. infeas.	100.00	50.00	80.00	0	0	0
Ave. min. deviation	−15.00	−7.89	−6.64	10.94	7.72	7.57
Min. min. deviation	−56.86	−40.29	−36.49	2.13	0	0
Ave. max. deviation	14.31	17.50	17.76	19.69	19.56	19.55
Max. max. deviation	34.90	34.95	34.40	35.00	35.00	35.00

of the minimum deviations from α, and the average and maximum values of the maximum deviations from α are shown in Table 1. From this table we can see that models (RH), (MR) and (BL(50%)) can lead to infeasible solutions. Moreover, infeasibility is not an infrequent phenomena: on average 28.28, 7.57 and 27.18% of nodes were infeasible for these models, respectively, with maximum proportions of up to 100% for some instances. In fact, detailed results show that the optimal solutions for these models had at least one infeasible node for 87, 75 and 77.8% of all instances, respectively. Model (BL(75%)) and, obviously, models (BBKK1) and (BBKK2) did not result in any infeasibilities.

The deviation results in Table 1 show that for models (RH), (MR) and (BL(50%)) the average availability is 15, 7.89 and 6.64% below the required level, respectively—indicating fairly significant departures from feasibility. We note that in some instances these deviations are very severe—ranging up to 56.86, 40.29 and 36.49%, respectively. We note that the departures from feasibility for (RH) and (MR) models in our experiments were much larger (both in terms of frequency and minimum deviation from α) than for the experiments reported by Borras and Pastor (2002), likely to be because their simulation model dropped any incoming call that could not find an available server, thus reducing the load on the system.

It is also interesting to observe minimum deviations for the three "feasible" models: (BL(75%)), (BBKK1) and (BBKK2) (of course, only the latter two guarantee feasibility). Ideally, we want the minimum deviation to be 0%—since larger numbers may indicate that excessive number of servers have been located. The average values for these three models are 10.94, 7.72 and 7.57%, respectively—indicating that the availability approximation used in (BL(75%)) is not as tight as in the other two models. Positive values in all three cases indicate that it may be possible to construct even tighter approximations (i.e., all three models are likely to overallocate server capacity). Still, the bounds used for (BBKK1) and (BBKK2) are tight for some facilities in several instances, as the minimum min. deviation for both models is zero.

While average values of minimum deviations range from −15 to 10.94% for all models, we find relatively small differences in the range of maximum deviations varying from 14.31 to 19.69% and even smaller differences in the maximum of the maximum deviations ranging from 34.40 to 35%. Thus, by the last measure the price paid for the conservativeness of the lower bound is not significant.

We investigated the sensitivity of the models with respect to the service level α and the coverage radius δ. We observed that the average of the average deviations for all six models is decreasing with α and for the three feasible models is also increasing with δ. This suggests that the bounds developed are tighter for higher service levels and smaller coverage radii. In terms of feasibility we observed that (RH) infeasibility is increasing with α but there is no clear relation between infeasibility and α for the (MR) and (BL (50%)) models. Moreover, we observed that the infeasibility of both (RH) and (MR) is increasing with δ and there is no clear relation between the latter and the infeasibility for the (BL (50%)) model.

We next turn our attention to the server and facility allocations for the different models. The primary measures we use here are the ratios of the number of servers (facilities) to the number of nodes and the relative differences in the number of servers (facilities) required by each model versus the minimum number of servers (facilities) for all models with respect to the current instance. The results are summarized in Table 2. We note that the relative number of servers is the primary measure of solution quality in our models, since none of the models included fixed costs for opening new facilities (even though such costs would be easy to introduce). However, some models tend to "naturally" centralize servers at the facilities, while others do not—hence it is interesting to look at the relative number of facilities as a secondary measure.

The average ratio of the number of servers to the number of nodes of both (BBKK1) and (BBKK2) is 31.4% (=(64%−48.7%)/48.7%), 21.2 and 25.7% higher than those of (RH), (MR) and (BL(50%)), respectively. The three "infeasible" models–(RH), (MR) and (BL(50%))—tend to use fewer servers. With respect to the differences in the average number of servers, the more conservative (BL(75%)) model uses 66.3% more servers (on average) then the minimum, while (BBKK1)/(BBKK2) solutions use 44.4% more than the minimum, on average. This indicates that the two new models developed in the current paper hold clear advantages

Table 2. Ratio of the number of servers (facilities) to the number of nodes for the six models

Models	*(RH) (%)*	*(MR) (%)*	*(BL (50%)) (%)*	*BL (75%) (%)*	*(BBKK1) (%)*	*(BBKK2) (%)*
Ave. number servers/number nodes	48.7	52.8	50.9	73.4	64.0	64.0
Max. number servers/number nodes	106.7	120.0	116.7	160.0	136.7	136.7
Ave. number facilities/number nodes	33.6	35.2	23.9	23.9	22.4	22.7
Max. number facilities/number nodes	65.0	75.0	60.0	55.0	55.0	55.0

over the (BL(75%)) model since they achieve guaranteed feasibility with a smaller number of servers.

It is interesting to note that (BBKK1) tends to use the fewest number of facilities among all models developed—this is because of the natural server centralization mechanism built into this model. This holds advantages in real-life applications where one generally wants to limit the number of facility locations.

It is important to note that models (BBKK1) and (BBKK2) use exactly the same number of servers in all 108 instances. Recall that since any solution to (BBKK1) is always feasible for (BBKK2), it means that (BBKK2) picked an alternate optima.

However, as the following example shows, for some instances model (BBKK2) can result in fewer servers than model (BBKK1).

Example 6. Separation of (BBKK1) and (BBKK2) solutions.

Consider a four-node cycle with all link lengths set to one and the coverage radius $\delta = 1$. Thus, for each node i, the region N_i consists of itself and two adjoining nodes. Let $\lambda_1 = \lambda_3 = 1.5$, $\lambda_2 = \lambda_4 = 0.5$ (i.e., there are two heavy-demand nodes located opposite to each other, and two light-demand nodes). Let $\mu = 4$ and $\alpha = 0.4$.

Note that $\lambda(N_1) = \lambda(N_3) = 2.5$ and $\lambda(N_2) = \lambda(N_4) = 3.5$. Therefore, with one server at each node the availability estimates are: $A(2.5, 1) = 0.375$ for nodes 1 and 3 and $A(3.5, 1) = 0.125$ for nodes 2 and 4. On the other hand the availabilities with two servers are much higher (0.851 and 0.733, respectively). It can be verified that locating two servers at any two nodes yields an optimal solution to (BBKK1) denoted by $\mathbf{x}^1$ (we omit the formulation details). On the other hand, the optimal solution to (BBKK2) denoted by $\mathbf{x}^2$ uses only three servers located at any three facilities (again, details are omitted). We note that models (RH) and (MR) yield solutions that are identical to (BBKK2) in this case.

Simulation results show the following node availabilities for the two allocation vectors above:

$$A(\mathbf{x}^1) = (0.896, 0.912, 0.985, 0.985);$$
$$A(\mathbf{x}^2) = (0.854, 0.859, 0.859, 0.926).$$

Note that in both cases, the actual node availabilities are much higher than the required $\alpha = 0.4$. It is interesting to note that with $\alpha = 0.5$, both (BBKK1) and (BBKK2) have $\mathbf{x}^1$ as an optimal solution. This example also shows that the availability bounds used to derive both models are quite conservative for some instances.

To investigate the separation between models (BBKK1) and (BBKK2) further, and observing that the cycle example has small differences in arrival rates and distances, we regenerated the 108 instances used in the computational experiment above changing only the values of $\lambda \backsim \text{Uniform}(1, 2)$, instead of the original range of $(1, 10)$ and link length $l(i, j) \backsim \text{Uniform}(1, 10)$, instead of the original range of $(1, 50)$. We found only three instances (out of 108) where (BBKK2) outperforms (BBKK1) in terms of the number of servers. To summarize, even though model (BBKK2) can produce solutions with smaller number of servers than (BBKK1) for some instances, our simulation results indicate that (BBKK2) may not hold practical advantages over (BBKK1) except in some special circumstances.

8. Conclusions

In this paper we addressed a location model with stochastic demand, congestion and mobile servers. Service quality is ensured through a server availability constraint imposing the minimum probability that a new call from a given demand node will find an available server.

The underlying queueing network operates as a PAQ system, for which performance measures, such as server availability, cannot be computed analytically. This leads to the need for easily-computable lower bounds on the availability—we derive two such bounds, which allows us to develop two new location models (BBKK1) and (BBKK2), which guarantee that the availability constraints are met.

We also examine in detail three previously developed models for this problem—the (RH) (MR) and (BL) models. We demonstrate that none of these models can guarantee that the minimum availability constraints are met. Our simulation study shows that infeasibility is a frequent phenomena in (RH) and (MR) models, with the level of infeasibility quite severe in some instances.

It is interesting to note that in our experiments infeasibility occurs much more frequently in the (RH) than in the (MR) model. This points to the importance of capturing server interdependence in queueing systems (the (RH) model's estimate of availability is based on the binomial approximation which treats the busy periods for each server as independent events, while (MR) uses queueing-based approximation). We note that the well-known (MAXECLP) model of Daskin (1983) employs the same binomial-based approximation as the (RH) model.

The (BL) model achieves theoretical feasibility only when there exists a deterministic upper bound T on service times. Where such an upper bound does not exist (e.g., when service times are exponential) or where the variability of service times is large, T must be set to a certain percentile of the service time distribution, which may lead to loss of feasibility. Our results demonstrate the difficulty of picking the "right" percentile. Setting T to the 50th and 75th percentiles we observe that the former tends to lead to infeasible solutions, while the latter appears to be overly conservative, requiring too many servers.

The (BBKK1) and (BBKK2) models developed in this paper appear to strike a sensible balance—achieving feasibility at a modest increase in the required server capacity, at least compared to (BL(75%)). Of these two models, (BBKK1) appears to be more successful—it is easier to solve, the solutions tend to require smaller number of facilities and in the vast majority of instances generated in our simulation experiments it used the same number of servers as (BBKK2) (even though the latter can result in better solutions).

The most obvious direction for future research is the need to develop tighter estimates of node availability in partially-accessible queues. The estimate underlying (BBKK1) is very conservative. The tighter estimate underlying (BBKK2) appears not to provide significant advantages in practical problems. We note that the "price" of feasibility—about 20–30% increase in the required number of servers over the "infeasible" models—is not insignificant; better availability estimates may be able to substantially reduce this cost.

Other directions include: relaxing some of the modeling assumptions, as discussed in Section 6, and incorporating multiple priority levels for calls (perhaps with different dispatching rules).

References

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

Ball, M. and Shanthikumar, J. (1994) Bounding a probability measure over a polymatroid with an application to transportation problems. *Mathematics of Operations Research*, **19**(1), 112–120.

Batta, R., Dolan, J. and Krishnamurthy, N. (1989) The maximal expected covering location problem: revisited, *Transportation Science*, **23**, 277–287.

Berman, O. and Krass, D. (2002) Facility location problems with stochastic demands and congestion, in *Facility Location: Applications and Theory*, Z. Drezner and H. Hamacher (eds.), Springer, Berlin, pp. 329–371.

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

Bertsimas, D. (2007) *Introduction to Queueing Systems*, monograph in preparation.

Borras F. and Pastor, J. (2002) The ex-post evaluation of the minimum local reliability level: an enhanced probabilistic location set covering model. *Annals of Operations Research*, **111**(1), 51–74.

Caldentey, R. and Kaplan, E. (2007) A heavy traffic approximation for queues with restricted customer-server matchings. Working Paper. OM-2007-4, Stern School of Business, New York University, New York.

Dai, J. (1995) On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability*, **5**(1), 49–77.

Daskin, M. (1983) A maximum expected covering location model: formulation, properties and heuristic solution. *Transportation Science*, **17**, 48–70.

Foley, R. and McDonald, D. (2001) Join the shortest queue: stability and exact asymptotics. *Annals of Applied Probability*, **11**, 567–607.

Gross, D. and Harris, C. (1985) *Fundamentals of Queueing Theory*, Wiley.

Kim, S. (2005) Service level commitment in location models with stochastic demands and congestion. Ph.D. thesis. Rotman School of Management, University of Toronto, Toronto, Canada.

Larson, R. (1975) Approximating the performance of urban emergency service systems. *Operations Research*, **23**, 845–868.

Larson, R. and Odoni, A. (1981) *Urban Operations Research*, Prentice Hall, Englewood Cliffs, NJ.

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

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

ReVelle, C. and Hogan, K. (1989a) The maximum availability location problem and α-reliable p-center problem: derivatives of the probabilistic location set covering problem. *Annals of Operations Research*, **18**, 155–174.

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

Schwarz, B. (1974) Queueing model with lane selection. *Operations Research*, **22**, 331–339.

Shaked, M. and Shanthikumar, J. (1994) *Stochastic Orders and Their Applications*. Academic Press, San Diego, CA.

Wolff, R. (1989) *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, Englewood Cliffs, NJ.

Biographies

Opher Baron is an Assistant Professor of operations management at the Rotman School of Management, the University of Toronto. He has a Ph.D. in Operations Management from the MIT Sloan School of Management along with an MBA and a B.Sc. in Industrial Engineering and Management from the Technion, Israel Institute of Technology. His research interests include applied probability and its application to facility location, service operations, inventory planning and revenue management.

Oded Berman is the endowed Sydney Cooper Chair in Business and former Associate Dean at the Rotman School of Management, University of Toronto. He received his Ph.D. (1978) in Operations Research from MIT. He had been with the Electronic Systems Lab at MIT, the University of Calgary, and UMASS Boston, where he was also the Chairman of the Department of Management Sciences. He has published over 180 articles and has contributed to several books. His main research interests include operations management in the service industry, location theory, network models and stochastic inventory control. He is an Associate Editor for *Management Science* and *Transportation Science*, and an editorial board member for *Computers and Operations Research*.

Seokjin Kim is an Assistant Professor of Information Systems and Operations Management at the Suffolk University's Sawyer Business School.

He received his Ph.D. degree in Operations Management from the University of Toronto's Rotman School of Management and an M.S. degree in Engineering-Economic Systems (currently, Management Science and Engineering) from Stanford University. He also received M.B.A. and B.B.A. degrees in Business Administration at the Yonsei University, South Korea. His research interests include network design in service operations and supply chains, workforce management in retail chains and the dynamics of quality costs.

Dmitry Krass is a Professor of Operations Management and Statistics at the Rotman School of Management, University of Toronto, where he has been since 1989, after obtaining his Ph.D. degree in Operations Research from Johns Hopkins University. His research interests include service operations, facility location and management, the study of distributive service systems, stochastic models and analytical modeling in marketing. He also has wide consulting experience in the areas of marketing analytics and decision support systems.