Location models for airline hubs behaving as M/D/c queues

Vladimir Marianov\textsuperscript{a,*,1}, Daniel Serra\textsuperscript{b}

\textsuperscript{a}Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile
\textsuperscript{b}Department of Economics and Business, IET and GREL, Universitat Pompeu Fabra, Barcelona, Spain

Abstract

Models are presented for the optimal location of hubs in airline networks, which take into consideration the congestion effects. Hubs, which are typically the most congested airports, are modeled as M/D/c queuing systems. A formula is derived for the probability of a number of customers in the system, which is later used to propose a capacity constraint. This constraint limits the probability of more than \( b \) airplanes in queue, to be smaller than or equal to a given value. Due to the computational complexity of the formulation, the model is solved using a heuristic based on tabu search. Computational experience is presented together with an example using a data set available in the literature.

Scope and purpose

In order to take advantage of economies of scale, most major airlines have selected some airports as transshipment points (hubs). As hubs become congested, the usual hub siting techniques fail in finding their best locations. We propose new methods for locating congested hubs and allocating runways to hub airports. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Hub location; Congestion; Tabu search

1. Introduction

Networks involving hubs are important in transportation and telecommunications. In both settings, when there is traffic between several origins and several destinations, there are economic benefits if this traffic is concentrated through some nodes and/or arcs of the network. A hub is a node of the network that concentrates traffic from several origins and distributes it to the final destinations. In multi-hub networks, the traffic is concentrated at a hub and sent from there to a second hub,
which distributes it to the final destinations. The transportation between hubs is less expensive per unit flow than the transportation between a hub and a non-hub, because of the economies of scale. The problem is to find the least expensive hub network, given the traffic volumes between each origin–destination node pair (optimal selection of certain nodes as hubs). We focus on airline hubs.

As traffic increases, hub airports are more congested than non-hub airports, because they receive higher traffic levels. We propose models and solution methods for the hub location problem, when there is congestion at hub airports, for either designing new hub networks or optimally assigning new runways to the existing hub networks. Congestion at airports is hard to deal with, because of several issues. First of all, the arrival rate of the planes is highly variable throughout the day. Although flights follow a schedule, they are subject to delays at their origin airports and during the flight itself, which make their arrival non-deterministic. Secondly, although the service rate can be considered constant during short periods of time, it is also variable over longer periods, depending on weather conditions and the type of planes that are serviced. Also, since there are passengers transferring between flights, the service time for a flight may depend on the arrival time of other flights. Thus, service times are not independent and identically distributed. Finally, at their arrival to the airport, airplanes have to go through three stages of service: landing at a landing runway, service at a gate and departure through a take-off runway. Thus, the probabilistic distributions of these service times are very difficult to determine. Due to these issues, it is useless to develop detailed models of congestion for planning purposes, and approximation models are much more useful.

We analyze the queue formed by airplanes waiting for landing. Under certain assumptions, the analysis is applicable to take-off runways or combined landing–departure runways. We use a peak hour analysis, assuming that during the peak hour the average arrival rate and the service rate are both constant. This allows us to model an airport as an M/D/c queueing system, i.e. Poisson arrivals, deterministic service time and c servers. This election of probabilistic modeling is justified in the body of the paper. We state an analytic formula for the steady-state probabilities of different numbers of customers in an M/D/c system, found using an approach similar to the one used by Gross and Harris [1] for single server queues and Prabhu [2] for waiting time distributions. Later, we use this formula for the development of a deterministic equivalent of a probabilistic constraint in an integer optimization formulation for the location of congested hubs.

In the next section, we review briefly the related literature. We then develop the probabilistic analysis and the probabilistic constraint. Finally, we present the full model and show some computational experience obtained through the use of exact and heuristic algorithms together with an example.

2. Literature review

2.1. Hubs

As opposed to what happens in other location problems, facilities (hubs) interact with each other. This interaction results in non-linear models (O’Kelly [3,4], Aykin [5]), which can be linearized through the replacement of non-linear products in the objective by new variables (Campbell [6]) or through flow models, as in Ernst and Krishnamoorthy [7,8]. Many models and algorithms have been proposed for the hub location problem, both on a plane and on a network. A good classification of
the models can be found in Campbell [6] and [9], and a complete review of models and algorithms in Bryan and O’Kelly [10]. Not covered in this review are the Hub Arc Location Problem by Campbell et al. [11] and the Maximum-Capture Hub Location Model for competitive environments by Marianov et al. [12]. Also, Kara and Tansel [13], study the p-hub center problem, which minimizes the maximum dissatisfaction of passengers in air travel (or maximum travel time). Most of these models can be written in single-allocation versions (a demand node is allocated to exactly one hub) or multiple-allocation versions (a demand node is allocated to several hubs, depending on the destination of the traffic).

With the exception of O’Kelly [14], who minimizes variability of hub usage, none of the hub models consider congestion. Some of the models, as Ernst and Krishnamoorthy [8] and Ebery et al. [15] include capacity limits, in terms of the traffic each hub receives from the nodes allocated to it.

2.2. Congestion and airports

Location models that explicitly include spatial queuing are mostly found in the context of emergency systems. A review of these models can be found in Marianov and Serra [16]. A few researchers have studied queues at airports, mostly for obtaining a model of their behavior, as opposed to locational purposes. Peterson et al. [17] propose a queuing model of the set of landing strips or runways of an airport, the latest being considered as a single server. This model is intended for schedule policy making. Their model is not suitable for locational purposes, because a location algorithm using it should have as a starting point the location of the hubs and a proposed schedule plan for all flights, for each proposed hub location set. Thus, the use of their model imbedded in an optimal location algorithm can be discarded. Newell [18] presents a complete analysis of the airport operations.

2.3. M/D/c queuing systems

The single server M/D/1 queue has been completely characterized. However, few results are available in the literature on the computation of the steady-state probabilities of different numbers of customers in a multiple server M/D/c queue. Syski [19], Gross and Harris [1], Prabhu [2] and Saaty [20] present a method for finding the generating function of these probabilities, based on a seminal development by Crommelin [21]. Altinkemer et al. [22] propose an approximation to the average waiting time in an M/D/c queue with non-preemptive priorities. Prabhu [2] presents a formula for the limiting waiting time cumulative distribution. Knessl et al. [23] review the results for multiple server queues and present an integral approach for the M/G/2 queue, being a generalization of the M/D/2 queue. Chaudhry et al. [24], provide an algorithm for numerically finding the limiting distribution of the number in the system for a bulk arrival M/D/c queue. Many approximations and bounds for the average waiting time are available; see Altinkemer et al. [22] for a survey.

3. Location of congested hubs, with a fixed number of runways at the hubs

In order to construct the optimal location model, we use a peak-hour analysis. This is a worst-case analysis, in the hope that improving the performance at peak hours will reduce also the queuing
effects during off-peak time. We concentrate on the landing runways. Take-off runways could be assumed to behave in the same way as landing runways, only delayed by the service time at the gate. For the sake of clarity, we assume that there are different runways for landing and take-off, so the demand for service is composed only by the landing flights.

Peterson et al. [17] model, in detail, the transient behavior of a queuing system at an airport. They conclude that there is always a variation around scheduled arrival times, so a Poisson process is a good representation of the arrival rates. Furthermore, they confirm this conclusion using real data, so we safely assume that arrivals of flights to a hub located at node \(k\) follow a Poisson distribution during peak hours. We also assume a deterministic service time at the same node. As opposed to other authors as Gerla and Kleinrock [25] and Peterson et al. [17], we do not approximate the set of indistinguishable servers (landing runways) as one server. Instead, we use a more exact M/D/c model for the system.

Our goal is to construct a model for strategic planning purposes (hub location), so we focus on a steady-state analysis of the delays at the airports. We use an approximation of the behavior of the airport from the point of view of congestion, which allows us to develop a tractable hub location model.

The first hub location model we propose is based on the plant location model by Balinski [26], O’Kelly [27] and Campbell [6] presented different uncapacitated versions of the hub location model. Note that there are formulations with fewer variables, such as that by Ernst and Krishnamoorthy [8]. However, we use Campbell’s [6] formulation for the sake of clarity. Since we need to maintain congestion within acceptable limits, we add an extra probabilistic “capacity” constraint to this formulation:

\[
\text{Min} \sum_i \sum_j \sum_k \sum_m c_{ijkm}x_{ijkm} + \sum_k f_k y_k,
\]

s.t.

\[
\sum_k \sum_m x_{ijkm} = 1 \quad \forall i, j,
\]

\[
x_{ijkm} \leq y_k \quad \forall i, j, k, m,
\]

\[
x_{ijkm} \leq y_m \quad \forall i, j, k, m,
\]

\[
P[\text{queue length at node } k > b] \leq \theta_q \quad \forall k,
\]

\[
y_k \in \{0, 1\}, \quad 0 \leq x_{ijkm} \leq 1 \quad \forall i, j, k, m,
\]

where

\[
x_{ijkm} = \begin{cases} 
1 & \text{if traffic from node } i \text{ to node } j \text{ goes through hubs located at nodes } k \text{ and } m, \\
0 & \text{otherwise},
\end{cases}
\]

\[
y_k = \begin{cases} 
1 & \text{if there is a hub located at node } k, \\
0 & \text{otherwise},
\end{cases}
\]
\( c_{ijkm} \) is the transportation cost from node \( i \) to node \( j \) going through hubs located at nodes \( k \) and \( m \),

\( f_k \) the fixed cost of locating a hub (with a predetermined number of runways) at node \( k \), and

\( \theta_q \) the desired upper bound for the probability of an excessive queue length at a hub.

In this model, which allows multiple allocation, the objective minimizes the sum of transportation and fixed costs of locating the hubs. The first constraint forces the traffic to go through one or two hubs (note that if both hub subscripts \( k \) and \( m \) are the same, the traffic goes only through one hub).

The next two equations force the traffic through sites \( k \) and \( m \) to be zero unless there are hubs located at those sites. The new probabilistic constraint (5) forces the probability of more than \( b \) airplanes waiting on queue to be less than or equal to \( \theta_q \).

Probabilistic equation (5) must be rewritten in an analytic form for the model to be possible to solve. In order to write a deterministic and hopefully linear equivalent to this equation, let \( p_s \) be the steady-state probability of \( s \) customers being in the system with \( c \) servers. Then, Eq. (5) becomes

\[
\sum_{s=b+c+1}^{\infty} p_s \leq \theta_q \quad \text{or} \quad 1 - \sum_{s=0}^{b+c} p_s \leq \theta_q. \tag{7}
\]

In the first form, the left-hand side represents the probability of more than \( b \) airplanes on queue, while \( c \) are being served. The second form just uses the fact that the sum of all probabilities is one.

We need the expression for the probabilities \( p_s \). This expression (which we have not been able to find in the literature) can be derived as follows. Given an arrival rate of \( \lambda \) and a service time of \( T = 1/\mu \), where \( \mu \) is the service rate, we use the generating function for these probabilities (Prabhu [2])

\[
P(z) = \frac{(1 - z)\sum_{i=0}^{c-1} v_i z^i}{1 - z^c e^{\lambda T/(1-z)}}, \tag{8}
\]

where \( c \) is the number of servers, and \( v_i \) is calculated using the equations

\[
\sum_{i=0}^{c-1} v_i = c - \lambda T \tag{9}
\]

and

\[
\sum_{i=0}^{c-1} v_i z_j^i = 0, \quad j = 1, 2, \ldots, c - 1. \tag{10}
\]

In Eq. (10), \( z_j \) is the \( j \)th root of equation

\[
1 - z^c e^{\lambda T/(1-z)} = 0. \tag{11}
\]

Note that the limiting distribution for \( p_s \) exists only for \( c > \lambda T \). Otherwise, the queue length tends to infinity. In this case, Eq. (11) has exactly \( c \) distinct roots \( z_j \) (one of which is 1), within and on the unit circle \( |z| \leq 1 \) (Prabhu [2]). Once these roots are known, the set of \( c \) equations composed by Eqs. (9) and (10) have a determinant that does not vanish, and the \( c \) unknowns \( v_0, v_1, \ldots, v_{c-1} \) are uniquely determined. Thus, for a complete knowledge of the generating function, Eq. (11) is solved first for the roots \( z_j \) and next Eqs. (9) and (10) are solved for the parameters \( v_j \). Note that, because of the way in which the generating function is constructed, and the form in which the
parameters $v_s$ are defined, for $s \leq c - 1$,

$$p_s = v_s - v_{s-1} \quad \text{or} \quad v_s = \sum_{i=0}^{s} p_s.$$  

The roots $z_j$ are complex numbers. However, it can be shown that the values of the parameters $v_i$, as well as the values of the probabilities $p_s$ are real (see Appendix A.1). Once the generating function is known, the probabilities $p_s$ are the coefficients of $z^s$ in the generating function, when this function is written as

$$P(z) = \sum_{s=0}^{\infty} p_s z^s.$$  

(12)

In order to write Eq. (7) in the form of Eq. (12), we expand its denominator in series, in the region $|z^c| < |e^{-\lambda T(1-z)}|$, which coincides with the region of interest $|z| \leq 1$:

$$\frac{1}{1 - z^c e^{\lambda T(1-z)}} = \sum_{k=0}^{\infty} [z^c e^{\lambda T(1-z)}]^k$$

$$= \sum_{k=0}^{\infty} z^{kc} e^{-k\lambda T} e^{-k\lambda T z}$$

$$= \sum_{k=0}^{\infty} z^{kc} e^{k\lambda T} \sum_{m=0}^{\infty} \frac{(-k\lambda T z)^m}{m!}$$

$$= \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} e^{k\lambda T} (-1)^m \frac{(k\lambda T)^m}{m!} z^{kc+m}.$$  

(13)

defining a new index $n = kc + m$,

$$\frac{1}{1 - z^c e^{\lambda T(1-z)}} = \sum_{k=0}^{\infty} \sum_{n=kc}^{\infty} e^{k\lambda T} (-1)^{n-kc} \frac{(k\lambda T)^{n-kc}}{(n-kc)!} z^n.$$  

(14)

The interested reader can verify (graphing the region of summation on the $n - k$ plane) that Eq. (14) is equivalent to

$$\frac{1}{1 - z^c e^{\lambda T(1-z)}} = \sum_{n=0}^{\infty} \sum_{k=0}^{r} e^{k\lambda T} (-1)^{n-kc} \frac{(k\lambda T)^{n-kc}}{(n-kc)!} z^n,$$  

(15)

where $r = \lfloor n/c \rfloor$, that is, the largest integer smaller than or equal to $n/c$. Thus, using Eqs. (8) and (15),

$$P(z) = (1 - z) \sum_{j=0}^{c-1} v_j \sum_{n=0}^{\infty} \sum_{k=0}^{r} e^{k\lambda T} (-1)^{n-kc} \frac{(k\lambda T)^{n-kc}}{(n-kc)!} z^{n+j}.$$  

(16)
We define again a new summation index \( s = n + j \), rewrite \( r = [(s - j)/c] \), and perform a reordering of terms analogous to that done to Eq. (14). Then, the generating function can be rewritten as

\[
P(z) = (1 - z) \sum_{s=0}^{c-1} \sum_{j=0}^{r} \sum_{k=0}^{r} v_j e^{k \lambda T} (s-j-kc) (s-j-kc)! z^s \]

\[
+ (1 - z) \sum_{s=c}^{\infty} \sum_{j=0}^{r} \sum_{k=0}^{r} v_j e^{k \lambda T} (s-j-kc) (s-j-kc)! z^s.
\]

After some short algebraic operations of the same nature as those already done, we can find the coefficients \( p_s \) of \( z^s \) in the generating function

\[ p_0 = v_0 \]

and, for \( s \geq 1 \),

\[ p_s = \sum_{j=0}^{p} \sum_{k=0}^{r} v_j e^{k \lambda T} (s-j-kc) (s-j-kc)! \]

\[ - \sum_{j=0}^{q} \sum_{k=0}^{u} v_j e^{k \lambda T} (s-j-kc-1) (s-j-kc-1)! , \]

where \( p = \min(s, c-1) \), \( r = [(s - j)/c] \), \( q = \min(s - 1, c - 1) \) and \( u = [(s - j - 1)/c] \). Now, we use the fact that, for \( s \leq c - 1 \), \( p_s = v_s - v_{s-1} \), to simplify the formula to

\[ p_s = v_s - v_{s-1} \quad \text{if} \quad s \leq c - 1 \]

and, if \( s \geq c \)

\[ p_s = \sum_{j=0}^{c-1} \left[ \sum_{k=0}^{r} v_j e^{k \lambda T} (s-j-kc) (s-j-kc)! \right] - \sum_{k=0}^{u} v_j e^{k \lambda T} (s-j-kc-1) (s-j-kc-1)! \]

\[
= \sum_{j=0}^{c-1} \sum_{k=0}^{r} v_j e^{k \lambda T} (s-j-kc) (s-j-kc)! \]

\[ - \sum_{k=0}^{u} v_j e^{k \lambda T} (s-j-kc-1) (s-j-kc-1)! , \]

If the values of \( c \), \( \lambda \) and \( T \) were known, it would be possible to compute the values of \( p_s \) for all \( s \), using Eq. (18), and probabilistic constraint (5) could be written as

\[
1 - \sum_{s=0}^{b+c} p_s \leq \theta_q \quad \text{or} \quad \sum_{s=0}^{b+c} p_s \geq 1 - \theta_q
\]

for all nodes where the hubs are located. In this case, \( \lambda \) is the airplane arrival rate to the hub, \( c \) the number of runways at the same hub, and \( T \) the service time at each runway. Thus, this constraint would be

\[
v_{c-1} + \sum_{s=c}^{c+b} \left\{ \sum_{j=0}^{c-1} \sum_{k=0}^{r} v_j e^{k \lambda T} (s-j-kc) (s-j-kc)! \right. \]

\[ - \sum_{k=0}^{u} v_j e^{k \lambda T} (s-j-kc-1) (s-j-kc-1)! \right\} \geq 1 - \theta_q.
\]
However, neither the location of hubs and runways nor the arrival rates to hubs are known before solving the model. The locations of the hubs are given by the values of the variables \( y_k \), and the arrival rate to a hub located at node \( k \) is given by

\[
\lambda_k = \sum_i \sum_j \sum_m a_{ij}x_{ijkm} + \sum_i \sum_j \sum_m a_{ij}x_{ijmk},
\]

where \( a_{ij} \) is the known average rate of airplanes travelling from node \( i \) to node \( j \) through hub \( k \), computed for the peak hour. Note that the airplanes arriving directly from an origin node \( i \) (the first term of the expression), plus the airplanes arriving from another hub \( m \) (the second term of the expression) compose the total arrival rate to a hub \( k \).

If the expression for \( \lambda_k \) given in Eq. (21) could be inserted in Eq. (20), we would obtain a nonlinear, deterministic equivalent to constraint (5). Unfortunately, this is not possible, because the values of the parameters \( v_s \) are computed numerically, starting from a known value of the arrival rate. However, for \( \lambda > \lambda T \) (i.e. an equilibrium exists), the left-hand side of Eq. (20) must be decreasing with increasing values of \( \lambda \). This is so, because physically, when the arrival rate increases, the queue length increases and, consequently, the probabilities \( p_s \) must increase for higher values of \( s \) and decrease for lower values of \( s \). As an example, suppose there is one server. If the arrival rate is zero, evidently, \( p_0 = 1 \), and Eq. (19) is satisfied. As the arrival rate increases, \( p_0 \) decreases and the probabilities \( p_s \), for \( s \geq 1 \), increase. If the arrival rate keeps increasing, probabilities \( p_s \) with small values of \( s \) decrease more and more, until Eq. (19) is no longer satisfied. What this means is, since the left-hand side of Eq. (20) is equivalent to the left-hand side of Eq. (19), it decreases with increasing values \( \lambda \), and consequently, there must exist a continuous range of values of \( \lambda \), defined by \( \lambda \leq \lambda_{\text{max}} \), that satisfy the equation. Furthermore, \( \lambda_{\text{max}} \) is the value of \( \lambda \) for which Eq. (20) holds as equality. Thus, we can numerically solve Eq. (20) for the variable \( \lambda \) and find the value \( \lambda_{\text{max}} \). Once this value is found, any smaller value of \( \lambda \) will satisfy Eq. (20). What this means is that Eq. (20) is equivalent to equation

\[
\lambda \leq \lambda_{\text{max}}
\]

or, using Eq. (21), if a value of \( \lambda_{\text{max}} \) is computed for each node \( k \) (assuming that there are differences between nodes in terms of service time or number of servers),

\[
\sum_i \sum_j \sum_m a_{ij}x_{ijkm} + \sum_i \sum_j \sum_m a_{ij}x_{ijmk} \leq \lambda_{\text{max},k}.
\]

Eq. (22) is the deterministic, linear equivalent to Eq. (5).

In the models by Ernst and Krishnamoorthy [8] and Ebery et al. [15], the capacity constraint for each hub limits only the flow coming directly from the nodes allocated to it. In our model, constraint (22) considers both the flow coming directly from the origin nodes, as well as the flow coming through other hubs. The interested reader can verify that the equivalent capacity constraint in the case of Ebery et al. [15] would be

\[
\sum_i \left( Z_{ik} + \sum_l Y_{ik}^l \right) \leq \lambda_{\text{max},k}H_k \quad \forall i, k,
\]

where \( H_k \) is the hub location variable, \( Z_{ik} \) the flow from node \( i \) to hub \( k \), and \( Y_{ik}^l \) the flow from node \( i \) through hubs \( l \) and \( k \).
In synthesis, in order to write the model, the following steps must be taken:

1. Find the candidate nodes. For each candidate node, estimate the service time, \( T \) and the number \( c \) of runways that can be built.

2. For each candidate node with particular values of \( T \) and \( c \), find \( \lambda_{\text{max}} \). This is done iteratively, by giving the arrival rate \( \lambda \) a starting value, solving Eq. (11) for roots \( z_j \) (Appendix A.2), then the set of Eqs. (9) and (10) for parameters \( v_j \) (Appendix A.1), and finally, checking if Eq. (20) holds as an equality. If it does, stop. If it does not, increase or decrease the value of the arrival rate, depending on whether the left-hand side of Eq. (20) is greater than or less than the right-hand side, respectively, and solve the equations again.

3. Once the values of \( \lambda_{\text{max}} \) have been found, use the model

\[
\begin{align*}
\text{Min} & \quad \sum_i \sum_j \sum_k \sum_m c_{ijkm}x_{ijkm} + \sum_k f_k y_k, \\
\text{s.t.} & \quad \sum_k \sum_m x_{ijkm} = 1 \quad \forall i, j, \\
& \quad x_{ijkm} \leq y_k \quad \forall i, j, k, m, \\
& \quad x_{ijkm} \leq y_m \quad \forall i, j, k, m, \\
& \quad \sum_i \sum_j \sum_m a_{ij}x_{ijkm} + \sum_i \sum_j \sum_m a_{ij}x_{ijmk} \leq \lambda_{\text{max},k}, \\
& \quad y_k \in \{0, 1\}, \quad 0 \leq x_{ijkm} \leq 1 \quad \forall i, j, k, m.
\end{align*}
\]  

4. Model for allocation of runways

If the number of runways at each airport must be optimized, new variables and parameters must be defined. The model is HLRA2:

\[
\begin{align*}
\text{Min} & \quad \sum_i \sum_j \sum_k \sum_m c_{ijkm}x_{ijkm} + \sum_k f_k y_k^c, \\
\text{s.t.} & \quad \sum_k \sum_m x_{ijkm} = 1 \quad \forall i, j, \\
& \quad x_{ijkm} \leq \sum_c y_k^c \quad \forall i, j, k, m, \\
& \quad x_{ijkm} \leq \sum_c y_m^c \quad \forall i, j, k, m.
\end{align*}
\]
\[
\sum_{c} y^{c}_k \leq 1 \quad \forall k, \quad (26)
\]
\[
\sum_{i} \sum_{j} \sum_{m} a_{ij} x_{ijkm} + \sum_{i} \sum_{j} \sum_{m} a_{ij} x_{ijmk} \leq \sum_{c} \lambda^{c}_{\text{max},k} y^{c}_k \quad \forall k, \quad (27)
\]
\[
\sum_{c} y^{c}_k \geq 1 \quad \forall k \text{ where an airport must be located}, \quad (28)
\]
\[
y^{c}_k \in \{0, 1\}, \quad 0 \leq x_{ijkm} \leq 1 \quad \forall i, j, k, m. \quad (29)
\]

In this model, the new variable \( y^{c}_k \) is one if a hub with \( c \) runways is located at node \( k \). Constraints (24) and (25) have the same meaning as (3) and (4). Constraint (26) states that, for node \( k \), only one variable \( y^{c}_k \) is one, thus an exact number of runways is located. Parameter \( \lambda^{c}_{\text{max}} \) in constraint (27) has to be computed for each \( c \). Constraint (28) is used when some airport locations \( k \) are preset, but the optimal number of runways is yet to be allocated. We propose but do not solve this model.

5. A heuristic procedure to solve the model

The models presented in the previous section, although linear, have thousands of variables and constraints for relatively small networks. Thus, traditional optimal solution methods such as linear programming plus branch and bound can become useless in terms of computing times. Furthermore, when using these methods, the number of branches is likely to increase dramatically because of the capacity constraints; see ReVelle [28]. Therefore, we offer a heuristic for the HLRA1 model.

For a fixed number of servers \( p \), the procedure has a construction phase and an improvement phase. In the first phase, a greedy adding heuristic is used to find the initial set of \( p \) locations. However, in the procedure we use, instead of adding the node with the best objective to the set of locations, we chose at random one of the three best nodes.

In the second phase, a one-opt exchange heuristic based on the Teitz and Bart [29] vertex insertion procedure is used. Each hub in the initial solution is moved to a non-hub location and, if the value of the objective is better than before the trade, the new set of locations is stored as the new solution. The procedure is executed for all facilities and potential locations, until no improvement is obtained. In order to avoid being trapped in a local optimum, a tabu search procedure is developed, similar to the one presented by Benati and Laporte [30]. In essence, this tabu search explores a part of the solution space by repeatedly examining all neighbors of the current solution, and moving a facility to the best neighbor even if this causes the objective function to deteriorate. To avoid cycling, recently examined solutions are inserted in a constantly updated tabu list. At each iteration, a facility (hub) is selected and relocated to each of the \( m \) vertices that are closest to it. The objective is computed for each relocation, and the one with the highest objective is chosen, provided it is not on the tabu list. If the value of the new solution improves the objective, the new solution is stored as the best one, and the vertex where the facility has been moved to is declared tabu for \( t \) iterations. Otherwise, the new solution is still saved but it is not considered as the best solution so far. If all neighbor vertices are declared tabu, then the one with the lowest tabu tag is chosen as the new solution. The number of one-opt trades needs to be fixed a priori.
Once the number of one-opt trades is reached, the tabu procedure is re-started using as initial solution the $p_t$ nodes that were least visited in the previous tabu phase. This is known as the diversification step.

Observe that in each one-opt trade a new solution is found. This solution may not be feasible due the nature of the capacity constraint. If this happens, a penalty is added to the objective. The penalty value is proportional to the extent of the total violation of the capacity constraint and is added to the final objective.

Since the number of hub airports is not known a priori, the procedure starts with 2 hubs and the best solution (if any is feasible) is stored. A new hub is added and the new solution is stored if it is better than the best one so far and feasible. In the same fashion, hubs are added one by one until a feasible solution is found. Once the feasible solution is found, we keep on adding hubs until the objective value is greater than the best (minimum) solution so far, by a given percentage. The reason for this stopping criterion is to explore the neighborhood of the best solution in search for better minima.

6. Computational experience

In order to test the heuristic, we randomly generated 900 instances of a 30-node network, with traffic between nodes uniformly distributed in $[0, 5]$. The right-hand side of the capacity-like constraint was set to 1200, 1400 and 1600 to see how the tightness of the constraint affects the solution of the algorithm. The transportation costs between vertices were obtained by computing their Euclidean distances. Savings in hub-to-hub transportation were set to 50%. Fixed costs were set to 10,000, 25,000 and 50,000.

The number of one-opt trades in the tabu phase of the heuristic was set to $200(30 - p)$, where $p$ is the number of hubs to locate. The diversification phase was executed three times. The size of the neighborhood (the $m$ nodes around the current solution) was between 4 and 8. The size of the tabu list was set to 5.

For each generated network, the model was also solved to optimality using complete enumeration with $p = 2, 3, 4, \ldots$, until no improvement of the objective was found. Therefore, we were able to examine the performance of the heuristic in terms of accuracy. Note that enumeration can be used only up to 50 nodes, 4 or 5 facilities. For higher figures, enumeration becomes unmanageable.

Results are shown in Table 1, where the first column indicates the fixed cost of each hub. The fixed cost was the same for all candidate nodes. The second column indicates the value of the right-hand side used for all the capacity constraints. In the third column, the average left-hand side of the capacity-like constraint is shown for each fixed cost and the right-hand side. The average value of the total transportation cost is indicated in the fourth column. The fifth column shows the average number of hubs that are located. The sixth column shows the number of times that the one-opt heuristic found the optimal solution, and the next column indicates the number of times that the tabu phase improved the solution of the one-opt phase and obtained the optimal solution. Finally, the eighth column shows the number of times that the algorithm did not find the solution and the ninth column shows the percentage of the average deviation of the objective value when its optimal value was not found.
As expected, as the fixed cost increases, the average cost also increases, since fewer hubs are located.

The performance of the heuristic in computing terms was quite satisfactory. As shown in Table 1, the worst case corresponds to a fixed cost of 10,000 and a right-hand side value of 1200, where 6% of the runs (6 out of 100) did not obtain the optimal solution, with an average deviation from optimality of 6%. In general, as the number of hubs decreases, the performance of the heuristic tends to improve, as it is usual when one-opt heuristics are used in location models.

Computing time was very similar across the different instances. On average, the heuristic took 25 s on a Pentium II 366 PC with 256 Mb of RAM.

### 7. Examples using the CAB data set

The model was also applied to the standard hub location CAB data set, which corresponds to 25 US cities and can be downloaded from [http://www.mscmga.ms.ic.ac.uk](http://www.mscmga.ms.ic.ac.uk) (Fig. 1). References to the source and prior results for these data can be found in O’Kelly et al. [31]. In this set, flows represent intercity passengers. Despite their size (not representative of airplane flows), note that by scaling the parameters of the problem, these flows can be used in our models without loss of generality. In order to present the results of these runs in a standard form, we use these figures.

Several instances of the first model were obtained using different savings in hub-to-hub transportation costs (saving percentage \( z = 0.25, 0.50 \) and 0.75), together with different values of the right-hand side of the capacity constraints and different opening fixed costs. The flow capacity of the hubs was set to \( \text{totflow}/2 \) and \( \text{totflow}/3 \), where \( \text{totflow} \) is the total number of inter-city flows (\( \text{totflow} = 8,540,006 \)). Fixed costs were set to 40,000 and 60,000. Results are presented in Table 2, which shows objective function values and hub locations for the 25 city CAB data set. As \( z \) increases (lower hub-to-hub transportation costs), the number of hubs tends to decrease.
Table 2
Results, CAB data

<table>
<thead>
<tr>
<th>Fixed costs</th>
<th>Maximum flow capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4,270,003</td>
</tr>
<tr>
<td>x = 0.25</td>
<td>40,000</td>
</tr>
<tr>
<td></td>
<td>654,041</td>
</tr>
<tr>
<td></td>
<td>744,346</td>
</tr>
<tr>
<td></td>
<td>2,10,12,21,23,24</td>
</tr>
<tr>
<td></td>
<td>2,12,21,24</td>
</tr>
<tr>
<td>x = 0.50</td>
<td>759,119</td>
</tr>
<tr>
<td></td>
<td>838,824</td>
</tr>
<tr>
<td></td>
<td>18,19,21</td>
</tr>
<tr>
<td></td>
<td>18,19,21,24</td>
</tr>
<tr>
<td>x = 0.75</td>
<td>824,507</td>
</tr>
<tr>
<td></td>
<td>897,555</td>
</tr>
<tr>
<td></td>
<td>2,19,21,24</td>
</tr>
<tr>
<td></td>
<td>18,19,21</td>
</tr>
</tbody>
</table>

Fig. 1. CAB Data set.

1 ATLANTA
2 BALTIMORE
3 BOSTON
4 CHICAGO
5 CINCINNATI
6 CLEVELAND
7 DALLAS-FW
8 DENVER
9 DETROIT
10 HOUSTON
11 KANSAS CITY
12 LOS ANGELES
13 MEMPHIS
14 MIAMI
15 MINNEAPOLIS
16 NEW ORLEANS
17 NEW YORK
18 PHILADELPHIA
19 PHOENIX
20 PITTSBURGH
21 ST. LOUIS
22 SAN FRANCISCO
23 SEATTLE
24 TAMPA
25 WASHINGTON
Table 3
Example with the CAB data

<table>
<thead>
<tr>
<th></th>
<th>13</th>
<th>17</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1,184,776</td>
<td>164,588</td>
<td>125,878</td>
<td>752,606</td>
</tr>
<tr>
<td>17</td>
<td>164,588</td>
<td>791,520</td>
<td>288,155</td>
<td>291,521</td>
</tr>
<tr>
<td>19</td>
<td>125,878</td>
<td>288,155</td>
<td>1,368,538</td>
<td>186,335</td>
</tr>
<tr>
<td>20</td>
<td>752,606</td>
<td>291,521</td>
<td>186,335</td>
<td>1,577,006</td>
</tr>
<tr>
<td>Capacity</td>
<td>2,227,848</td>
<td>1,535,784</td>
<td>1,968,906</td>
<td>2,807,468</td>
</tr>
<tr>
<td>% over max</td>
<td>78.3</td>
<td>53.9</td>
<td>69.2</td>
<td>98.6</td>
</tr>
</tbody>
</table>

Note: \( x = 0.5 \), fixed costs = 60,000, flow capacity = 2,846,669.

Results for a specific instance using the 25 city CAB data set are presented in Table 3. They correspond to \( x = 0.50 \), flow capacity = 2,846,669 and fixed costs = 60,000. In this table, hub-to-hub flows are presented. For example, there are 1,184,776 trips that go exclusively through the hub located at node 13 and 164,588 that go through hubs at node 13 and 17. The hub at node 20 is the busiest, since it reaches almost full flow capacity (98.6%).

Note that if the capacity constraint is eliminated, the basic model is very similar to the standard uncapacitated multiple assignment hub median problem (UMAHMP), for which results are available in O’Kelly et al. [31] in the multiple assignment hub location model. In order to compare those results (for non-congested systems) to our congested hubs model’s results, we set the fixed costs to 0 and added a new constraint to our basic model that fixes the number of hubs. Again the CAB data were used, now with \( n = 20 \), as in O’Kelly et al. [31]. Flows were scaled to sum to one over the entire network. Solutions were obtained by complete enumeration for different values of \( x \), i.e. \( x = 0.0, 0.3, 0.6, 0.9 \), and 3 hubs. The right-hand side of the capacity-like constraint was set to 0.4 (which is approximately the value corresponding to one runway per hub, a maximum of 3 airplanes waiting for service with a 95% probability and a service time of 2 h). Results are presented in Tables 4 and 5. In Table 4, for both models and for each value of \( x \), the optimal objective value is presented together with the locations. Table 5 shows the left-hand side of the capacity-like constraint for both models. In other words, these figures represent the workload of each hub, where a low number corresponds to a non-congested hub. In these tables we can see a clear trade-off between the objective value (transportation costs) and the workload balance. While, as expected, the UMAHMP obtained a better objective value (between 5% and 10%), the HLRA ensures a more balanced workload among hubs, and there is a 95% probability that no more than three airplanes will be waiting on line for landing. For example, in the UMAHMP the second hub is clearly underused as compared to the other two hubs.

Figs. 2–5 show multiple assignment networks for the 20-node problem for various values of \( x \) and 3 hubs using both models. The dark lines represent interhub links. As O’Kelly et al. [31] observed, as the value of \( x \) increases, the number of multiple assignments also increases in the UMAHMP. This is also true in the HLRAI. Their behavior in terms of location allocation is very similar.
Fig. 2. Solutions for the multiple assignment problem with $\alpha = 0.0$.

Table 4
Comparison between UMAHMP and HLRAI models

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>UMAHMP</th>
<th>HLRAI</th>
<th>% diff. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>597.53</td>
<td>639.12</td>
<td>6.96</td>
</tr>
<tr>
<td></td>
<td>4,12,17</td>
<td>4,10,17</td>
<td></td>
</tr>
<tr>
<td>0.30</td>
<td>759.85</td>
<td>830.00</td>
<td>9.23</td>
</tr>
<tr>
<td></td>
<td>4,12,17</td>
<td>7,9,17</td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>884.64</td>
<td>936.34</td>
<td>5.84</td>
</tr>
<tr>
<td></td>
<td>4,12,17</td>
<td>6,7,17</td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td>964.81</td>
<td>997.40</td>
<td>3.38</td>
</tr>
<tr>
<td></td>
<td>4,7,17</td>
<td>6,7,17</td>
<td></td>
</tr>
</tbody>
</table>
Table 5
Comparison of hub capacities (in %)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>UMAHMP</th>
<th></th>
<th>HLRAI</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hub 1</td>
<td>Hub 2</td>
<td>Hub 3</td>
<td>Hub 1</td>
</tr>
<tr>
<td>0.0</td>
<td>0.43</td>
<td>0.13</td>
<td>0.44</td>
<td>0.32</td>
</tr>
<tr>
<td>0.30</td>
<td>0.46</td>
<td>0.11</td>
<td>0.43</td>
<td>0.25</td>
</tr>
<tr>
<td>0.60</td>
<td>0.42</td>
<td>0.12</td>
<td>0.46</td>
<td>0.40</td>
</tr>
<tr>
<td>0.90</td>
<td>0.42</td>
<td>0.12</td>
<td>0.46</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Fig. 3. Solutions for the multiple assignment problem with $\alpha = 0.30$. 
8. Conclusions

In this paper, a new hub location model has been formulated. This model locates hubs so as to minimize total (fixed and transportation) costs while taking into account congestion. The model also considers the number of runways to open in each hub. Two versions of the model are offered: in the first, the number of runways is fixed a priori, while in the second, the number of runways to open at each hub is determined by the model itself. The key feature of the model is the transformation of the probabilistic constraint that states that the amount of congestion in a hub cannot exceed a given threshold with a given probability, into a deterministic linear constraint. Hubs are modeled as M/D/c queuing systems, and a novel procedure is developed for solving exactly such systems.

A one-opt meta-heuristic was used to obtain solutions. In its first phase, this heuristic finds an initial solution to the problem, while in the second phase, the solution is improved by using first
a one-opt heuristic, followed by a tabu search. The heuristic was successfully tested in 900 different 30 nodes-networks. Finally, the solved examples are compared to other uncapacitated multiple assignment hub location models.

Further research on congested hub systems could include more exact congestion models, although it may be expected that this will lead to intractable formulations.

Acknowledgements

The authors gratefully acknowledge the careful work and valuable contributions made by the editor and three anonymous referees.
Appendix A

A.1. The roots of equation

\[ 1 - z^e e^{\lambda T(1-z)} = 0 \]  

(11)

are, in general, complex numbers. Some of them may have real values. If \( z_j \) is a complex root of the equation, so is \( z_j^* \), the conjugate complex. In fact, if

\[ z_j = a + ib = me^{i\phi} \]

is a root of Eq. (11), then

\[ 1 - m^e e^{ie\phi} e^{\lambda T(1-a-ib)} = 0, \]

\[ 1 - m^e e^{i(c\phi-b\lambda T)} e^{\lambda T(1-a)} = 0, \]

\[ 1 - m^e [\cos(c\phi-b\lambda T) + i \sin(c\phi-b\lambda T)] e^{\lambda T(1-a)} = 0, \]

\[ 1 - m^e e^{i\lambda T(1-a)} \cos(c\phi-b\lambda T) + ie^{i\lambda T(1-a)} \sin(c\phi-b\lambda T) = 0 \]  

(A.1)

both the real and the imaginary part of the left-hand side of the equation must vanish, for the equation to hold. Since both vanish, the same happens for \( z_j^* \), the conjugate of \( z_j \), since for this conjugate, the left-hand side of Eq. (11) is just the conjugate of Eq. (A.1).

In order to compute the parameters and \( v_i \), we use Eqs. (9) and (10). Since \( z_j \) and \( z_j^* \) are roots of Eq. (11), the following two equations are part of set (10):

\[ v_0 + v_1 z_j + v_2 z_j^2 + \cdots + v_{c-1} z_j^{c-1} = 0 \]

and

\[ v_0 + v_1 z_j^* + v_2 z_j^{*2} + \cdots + v_{c-1} z_j^{*c-1} = 0. \]

Note that \( z_j^{*s} \) is the conjugate of \( z_j^s \). We can replace these equations for the sum of them and the difference of them. The sum of both equations is an equation whose coefficients are real (the imaginary part vanishes). The difference is an equation whose coefficients are purely imaginary. If we divide the equation by \( i \), the root of \(-1\), the equation is entirely real. Thus, all the equations are real. Since the right-hand side of these equations, as well as Eq. (9) are real, so are the values of the parameters \( v_i \) and, consequently, the values of the probabilities \( p_s \).

A.2.

In order to solve Eq. (11), write \( z = \beta/\gamma \), with \( \gamma = \lambda T/c \), and plug in Eq. (11), which can then be put in the following form (Syski [19]):

\[ \beta e^{\beta} = \sqrt{1 \gamma e^{2\gamma}}. \]
The $c$th root of 1 is

<table>
<thead>
<tr>
<th>$c$</th>
<th>Roots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$1, -1$</td>
</tr>
<tr>
<td>3</td>
<td>$1, e^{i120\degree}, e^{-i120\degree}$</td>
</tr>
<tr>
<td>4</td>
<td>$1, -1, e^{i90\degree}, e^{-i90\degree}$</td>
</tr>
</tbody>
</table>

and so on.

References


Vladimir Marianov is a Professor at the Department of Electrical Engineering at the Pontificia Universidad Católica de Chile. He earned his Ph.D. degree in 1989, in the Department of Geography and Environmental Engineering at the Johns Hopkins University (Baltimore). His interests include formulation and applications of models and algorithms for facility location, as well as the design and planning of telecommunications networks.

Daniel Serra is professor of Management in the Department of Economics and Business, Universitat Pompeu Fabra (Barcelona, Spain). He earned his Ph.D. in 1989, in the Department of Geography and Environmental Engineering at the Johns Hopkins University (Baltimore). He has been working in the development and application of operations research models to logistic problems and to health care, with special emphasis on location and transportation modeling. He is also vice-chancellor of the Universitat Pompeu Fabra.