European Journal of Operational Research 210 (2011) 287–300

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

A hybrid hypercube – Genetic algorithm approach for deploying many emergency response mobile units in an urban network

Nikolas Geroliminis [a,*], Konstantinos Kepaptsoglou [b], Matthew G. Karlaftis [b]

[a] Urban Transport Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[b] School of Civil Engineering, National Technical University of Athens, Greece

ARTICLE INFO

Article history:
Received 21 September 2009
Accepted 30 August 2010
Available online 9 September 2010

Keywords:
Emergency response
Hypercube
Spatial queues
Genetic algorithms

ABSTRACT

Emergency response services are critical for modern societies. This paper presents a model and a heuristic solution for the optimal deployment of many emergency response units in an urban transportation network and an application for transit mobile repair units (TMRU) in the city of Athens, Greece. The model considers the stochastic nature of such services, suggesting that a unit may be already engaged, when an incident occurs. The proposed model integrates a queuing model (the hypercube model), a location model and a metaheuristic optimization algorithm (genetic algorithm) for obtaining appropriate unit locations in a two-step approach. In the first step, the service area is partitioned into sub-areas (called superdistricts) while, in parallel, necessary number of units is determined for each superdistrict. An approximate solution to the symmetric hypercube model with spatially homogeneous demand is developed. A Genetic Algorithm is combined with the approximate hypercube model for obtaining best superdistricts and associated unit numbers. With both of the above requirements defined in step one, the second step proceeds in the optimal deployment of units within each superdistrict.

1. Introduction

Emergency response services are critical for modern societies; they provide assistance to incidents, protect and ensure public health and safety and preserve proper operation of lifelines. Transportation networks are among those important lifelines that are highly prone to emergencies (Nicholson and Du, 1997); because of the frequent occurrence of incidents such as car malfunctions and accidents and infrastructure failures, these networks are in need such services, capable of undertaking emergency response and network restoration activities. Transit systems in particular are considerably affected by incidents; their operations are disrupted, performance and quality of services degrade and their credibility is diminished. Moreover, the effects of such incidents to the road network are the same to those involving private vehicles, implying traffic congestion and delays.

Transit authorities establish special services for responding to incidents; these include tow-away vehicles and mobile repair units, capable of accessing the site of an incident and provide appropriate aid; units are strategically scattered around the transportation network and respond to incidents in their designated vicinity of responsibility. Design of such services includes the optimal deployment of units responsible for undertaking incidents; these units are positioned in such a way that some service level objective is optimized (Araz et al., 2007); typical objectives include the minimization of the average or maximum time to serve an emergency and the maximization of the area served by each unit (Goldberg, 2004).

In this context, this paper presents a model and a heuristic for the optimal deployment of many emergency response units in an urban transportation network, an application and a decision support system (DSS) for transit mobile repair units (TMRU) in the city of Athens, Greece. The model considers the stochastic nature of such services, suggesting that a unit may be already engaged when an incident occurs (such a phenomenon is characterized as *congestion* (Boffey et al., 2007)). The number of available units is most frequently limited, the transit networks extensive, the operating conditions strenuous, and the repair times frequently long; these conditions suggest the strong possibility of congestion and, therefore, an approach is adopted that considers repair unit availability. The proposed model integrates a queuing model (the hypercube model), a location model and a metaheuristic optimization algorithm (genetic algorithm) for obtaining appropriate unit locations.

The remainder of the paper is organized as follows: in the next section, a brief review of location models is provided; the review focuses on location models used in emergency response services and especially those that incorporate congestion aspects. Following, the model developed along with its elements is described

* Corresponding author. Tel.: +41 21 69 32481.
E-mail addresses: nikolas.geroliminis@epfl.ch (N. Geroliminis), kkepap@central.ntua.gr (K. Kepaptsoglou), mgk@central.ntua.gr (M.G. Karlaftis).

0377-2217/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.08.031

and analyzed. An application of the model for the Athens surface transit network follows and its results are discussed and a DSS incorporating the proposed method is presented. The final section contains the conclusions of the paper.

2. Background

Facility location models have been widely applied in real life problems with examples that include the sitting of emergency medical services (EMS), police and fire stations, bus garages and airline hubs (Current et al., 2002). Comprehensive reviews of such models can be found in Drezner and Hamacher (2002), Goldberg (2004), ReVelle and Eiselt (2005) and Jia et al. (2007) while Brotcone et al. (2003) provide a focused review of their application in emergency response services. Location models are distinguished in *coverage* and *median* type models (Berman and Krass, 2002): *Coverage-type* models attempt to locate servers so that adequate coverage is provided to demand points, implying that there is at least one server that can undertake demand for service in a position within a preset maximum distance. *Median-type*models minimize average or total travel cost between servers and demand and locate them accordingly. This section provides a short overview of emergency response service related to location models and then focuses on hypercube and its applications. The later part acts as the basis for introducing the novelties of the proposed model.

2.1. Location models in emergency response services: A general overview

Early efforts on emergency response service planning focused on two basic coverage models: the set covering location problem (SCLP) by Toregas et al. (1971) and the maximal coverage location problem (MCLP) by Church and ReVelle (1974). Later efforts considered the case of several server types (TEAM and FLEET models by Schilling et al. (1979) and a MCLP improvement by Marianov and ReVelle (1992)) and multiple coverage of demand for service (BACOP1 and BACOP2 by Hogan and ReVelle (1986), DSM and DDSM models by Gendreau et al. (1997, 2001)). On the other hand, the *p*-median problem, originally proposed by Hakimi (1964) was used by Calvo and Marks (1973), Carbone (1974), Carson and Batta (1990) and Paluzzi (2004); for planning emergency response services.

Basic location models are deterministic and, in that sense, do not capture inherent uncertainties often encountered in emergency response services (Brotcone et al., 2003; Jia et al., 2007). As a result, probabilistic models have been developed in an effort to incorporate uncertainty in design parameters of such systems, for example, demand, travel times, and so on, and to explicitly consider congestion. Some models address variations in inputs; an extensive review of these models is provided by Snyder (2006). Congestion on the other hand can significantly affect the performance of an emergency response service and should therefore be accounted for in planning such services, an aspect widely investigated by researchers (Snyder, 2006). Examples of congestion location models include the maximum expected coverage location model (MEXCLP) by Daskin (1983), the maximal availability location problem (MALP I,II) by Hogan and ReVelle (1986). Extensions to the MEXCLP include travel speed variations of servers (TIMEXCLP by Repede and Bernardo (1994)) and stochastic travel times (Goldberg et al., 1990), while Marianov and ReVelle (1996) improved the MALP model by applying queuing theory for estimating busy fractions (QMALP) and Marianov and Serra (1998, 2003) considered congestion in coverage models by constraining queue lengths for servers. A detailed review of these models can be found in Galvão and Morabito (2008).

2.2. Hypercube and applications

Larson (1974) introduced queuing theory in facility location modeling by presenting the hypercube model, an analytical tool for evaluating the performance of a spatially distributed system of servers. Based on the configuration of such a system, hypercube can derive a set of performance measures, useful for planning and decision making. Later, Larson (1975) developed the *A*-hypercube, an approximation of the original model, which reduced computational difficulties encountered when implementing the original model. Extensions of hypercube have been proposed by some researchers (Halpern, 1977; Jarvis, 1985; Katehakis, 1985; Burwell et al., 1992; Swersey, 1994), while applications and extension of the model for planning emergency response services have been developed by Brandeau and Larson (1986), Burwell et al. (1993), Chelst and Barlach (1981), Sacks and Grief (1994), Mendonça and Morabito (2001), Atkinson et al. (2006, 2008), Iannoni and Morabito (2007), Takeda et al. (2007) and Galvão and Morabito (2008).

As noted by Galvão and Morabito (2008), "*the hypercube model is not an optimization model; it is only a descriptive model that permits the analysis of scenarios*". Goldberg (2004) and Takeda et al. (2007) also state that hypercube must be embedded to an optimization framework for obtaining optimal server locations. Indeed, in early studies, Berman et al. (1985) extended Hakimi's one-median problem by embedding it in a queuing context while Berman et al. (1987) developed heuristics for locating mobile service units on a network in the presence of queueing-like congestion by taking advantage of the hypercube model and a location model for a single service unit. Batta et al. (1989) combined the MEXCLP with hypercube into an iterative, local search algorithm; hypercube was used to estimate expected coverage of located servers. Saydam and Aytug (2002, 2003) replaced the local search approach by a genetic algorithm. Galvão et al. (2005) extended Hogan's maximum availability location problem (MALP) by incorporating hypercube, in an effort to relax the initial model's assumption on server independence.

Two recent studies by Iannoni et al. (2007, 2008), embedded hypercube in a hybrid genetic algorithm focusing on optimizing the configuration and operation of EMS along a highway. Their objective was to either minimize the mean user response time or to remediate EMS server workloads, by determining the optimal areas of responsibilities for a single dimension space (a highway). An important contribution of this work is the modeling of the multiple dispatch problem (more than one servers can intervene at the same time). The authors provided a solution to the problem using GA for a small number of servers ($\sim$5 servers). Higher dimensions (e.g. network level), or higher number of servers increase computational times, making the use of these algorithms prohibitive. These studies did not integrate location and districting decision in the same optimization approach, a step done at Geroliminis et al. (2009) for a small number of servers, as well.

Recently, Geroliminis et al. (2009) extended the hypercube model and developed the spatial queueing model (SQM) to optimally locate emergency response vehicles. This model explicitly considers that (i) service rates are not identical and vary between servers (non-homogeneous servers of this type are also analyzed in Morabito et al., 2008) and (ii) for a given server the service rates depend on the incident's characteristics (interdistrict or intradistrict response). All service rates for both types of responses depend on the location of servers. Thus, this model (iii) links districting and dispatching to the location problem and (iv) proposes a hybrid formulation for coverage and mean response time to optimally locate servers and (v) simultaneously identify their areas of responsibilities (rather than analyzing performance measures for a given system configuration). Despite its theoretical elegance and flexibility, the model is computationally difficult to solve for a large number

of servers (>10). A recent review on hypercube applications in emergency service systems is offered by Galvão and Morabito (2008).

To date, there exists a variety of models for designing emergency services, with congestion being among those elements that have attracted attention by the research community, as an improvement for obtaining more realistic results. Among recent advances in this area, is the combined use of optimization techniques and hypercube models, which that can achieve promising results for obtaining improved EMS design configurations. To the best of our knowledge, current literature has not developed a methodological hypercube framework to optimize the response time in large urban networks with many servers, which unavoidably leads to a more complex design problem. This step is taken in this paper, where we propose a two-step, hypercube – optimization based model for obtaining districts and locations of EMS for large urban networks with more than 15 servers, which is prohibitive to be solved with an exact hypercube model. For this reason, at first step we do not directly seek to obtain the response area per server, but rather define a set of servers as a "superserver", which is responsible for providing service in a "superdistrict" and determine the required number of servers per superdistrict.

However, it is not straightforward to near-optimally partition a large network to superdistrics and estimate optimal numbers of servers in such a way that the heuristic solution – which assumes independence between superdistircts – will be close to the solution without partitioning. Another issue is how one can estimate the performance characteristics of a superdistrict, without solving an exact hypercube. To this end, we formulate an approximation of the hypercube model for a homogeneous demand network with symmetric servers. This approximate formulation decreases the number of state probabilities of a hypercube model from 2^N to N, where N is the number of servers and it is much faster to solve. Subsequently at step 2, we optimally deploy required servers within each superdistrict by solving a spatial queueing model (SQM), which carries the novelties earlier described in this section. This two-step approach is proved to be efficient for large scale networks and when compared with an exact solution of the SQM model for smaller network instances, it produces near-optimal locations with errors less than 3%. If an SQM model is applied in a random partitioning of the large network the solutions are far from optimal, highlighting the necessity for careful partitioning.

3. A note on the hypercube and spatial queueing model (SQM)

Consider a network $\boldsymbol{J}$ consisting of a set of regions/cells each generating requests for service (we use notation similar to Larson (1974)). A number of servers are to be located at points $x \in \boldsymbol{J}$. A server's primary response area (*district*) consists of those regions/cells to which the server would be dispatched if all other servers are available. Each server can be busy (one) or free (zero), generating 2^N possible states for the system (where N is the number of servers); these are the vertices of a hypercube named B_j ($j = 0, 1, \ldots, 2^{N-1}$) of dimension N. Demands occur solely at the center of each service region by time homogeneous Poisson requests for service (input) and exponential service rates ignoring any past system history. When a request for service arrives, if the responsible server is available, it is dispatched immediately to serve the incident and then returns to its base location before responding to the next request. If the responsible server is busy when a request arrives, another server will serve the request.

We assume that only one step transitions occur while multistep transitions are not allowed (i.e. two servers cannot be simultaneously assigned). This implies that transitions are allowed between states with Hamming distance equal to 1 (the number of digits by which the two vertices differ). Thus, the model is a finite-state continuous time Markov process (Larson, 1974), and the model's steady-state probabilities are determined from the equations of detailed balance that express a conservation of flow between consequent states. Geroliminis et al. (2009) extended the hypercube model to explicitly consider that service rates are not identical and vary between servers, while for a given server may depend on the incident's characteristics (interdistrict or intradistrict response). They also provided a heuristic to minimize average response time as this is estimated by the extended hypercube. The location of servers and dispatch preferences is estimated through the model which uses the closest available server policy implying that, each time an incoming call is received, the nearest available server is assigned.

The objective of the model is to minimize mean system response time, $\overline{T}$, subject to "hypercube" constraints; the formulation of the problem is (for more details see Geroliminis et al. (2009))

$$\text{Minimize } \overline{T} = \sum_{n=1}^{N} \sum_{j=1}^{J} (\rho_{nj} \cdot t_{nj}), \tag{1}$$

subject to:

$$\sum_{j=1}^{J} x_j = N, \tag{2}$$

$$\rho_{nj} = f_j \frac{\sum_{B_i \in E_{nj}} P\{B_i\}}{\left(1 - P\{B_{2^N-1}\}\right)}, \quad n = 1, \ldots, N,\ j = 1, \ldots, J, \tag{3}$$

$$P\{B_j\} \left[\sum_{\left\{ \substack{i \\ B_i \in C_N : d_{ij}^- = 1} \right\}} \lambda_{ij} + \sum_{\left\{ \substack{i \\ B_i \in C_N : d_{ij}^+ = 1} \right\}} \mu_{ij} \right] = \sum_{\left\{ \substack{i \\ B_i \in C_N : d_{ij}^- = 1} \right\}} \mu_{ij} P\{B_i\} + \sum_{\left\{ \substack{i \\ B_i \in C_N : d_{ij}^+ = 1} \right\}} \lambda_{ij} P\{B_i\}, \quad j = 0, 1, \ldots, 2^N - 1, \tag{4}$$

$$\sum_{i=0}^{2^N-1} P\{B_i\} = 1, \tag{5}$$

where J is the total number of regions/cells, N is the total number of response units/servers, x_j is 1 if a server is located in region j and 0 otherwise, t_{nj} is the mean travel time for server n to reach region j, f_j is the fraction of network-wide workload generated from region j, ρ_{nj} is the fraction of dispatches sending unit n to region j, $P\{B_k\}$ is the steady state probability of state corresponding to vertex B_k, E_{nj} is the set of states, where server n is the nearest available for region j, C_N are the vertices of N-dimensional unit hypercube, d_{ij}^-, d_{ij}^+ are the "downward" and "upward" Hamming distance between vertices B_i and B_j, (the number of binary digits switching from 0 to 1 and 1 to 0), while λ_{ij} and μ_{ij} are the upward and downward mean rates at which transitions are made from state i to state j corresponding to vertices B_i and B_j, given the system is in state i. For given servers' location and states i and j, there is a corresponding server, which changes its condition from idle to busy or the opposite. λ_{ij} is the total demand for requested service, which requires the intervention of the aforementioned server, while μ_{ij} is the service rate associated with the demand. Qualitatively speaking, this demand is the sum of interdistrict and intradistrict potential responses for which this

server is the nearest among the idle ones. For a detailed formulation of the transition rates and the partitioning of the network to areas of responsibilities, the reader should refer to Geroliminis et al. (2009).

Also, we should note that the objective function (1) and the sum of state probabilities (5) assumes a zero-line capacity system, which effectively means that in the case that all servers are busy, new calls (expressed as overflows) are either lost or treated by special reserve units. Alternatively one could model an infinite-line capacity system, in which case overflows are handled in a first-come first-served manner by the regular units within the region. The modifications for this formulation can be found in Larson (1974).

Once the number of servers becomes larger the minimization problem described in (1)–(5) is computationally very difficult to solve since the spatial queueing model is an NP-hard problem. A special case, the *p*-median problem, belongs to the class of NP-hard problems in the strong sense (Mirchandani and Francis, 1990). As Geroliminis et al. (2009) discussed, if all servers had infinite travelling speed and were spending zero service time at the incident, then all would be readily available to intervene and the optimal location of servers for the SQM would be the same with that of the *p*-median. In the case of the *p*-median, the number of dispatches requested by a server in a specific region (Eq. (4)) is the region's demand if the server is the nearest, and zero otherwise. The SQM formulation (or any other hypercube formulation) is more complex, as it requires to solve a $2^N \times 2^N$ linear system in each instance of the model. For example, the Coppersmith–Winograd algorithm in linear algebra, is one of the asymptotically fastest algorithms for matrix inversion with computational complexity of $0(k^{2.37})$ (Coppersmith and Winograd, 1990); given that the linear system in (4) is $2^N \times 2^N$ the optimal solution of the problem for $N+1$ servers will require $2^{2.37} = 5.2$ times higher computational time when compared with the problem for N servers, even if the number of iterations needed to reach the optimal solution is the same.

As the spatial queueing model (SQM) is difficult to solve for large N, an elegant heuristic could be developed if we were able to partition the service region in I sub-regions with N_i servers each $(\sum_i N_i = N)$ and solve an SQM in each sub-region. However, it is not straightforward to near optimally partition and estimate optimal N_i's in such a way that the heuristic solution - which would assume independence between subregions - would be close to the solution without partitioning. Although there are cases in real life where physical boundaries (e.g. mountains, rivers) provide some form of partitioning, a generally applicable computational methodology is required; to this end, we provide an approximate solution to the symmetric Hypercube model with spatially homogeneous demand, where all servers are treated equally. In this manner, instead of 2^N possible states for the queueing system, there are only N states (the number of servers).

4. Model

4.1. Overview and novelties

Design of emergency response services often requires the implementation of inherently NP-Hard facility location models which are usually intractable in cases of sizeable urban systems requiring the positioning of a large number of servers (Dimopoulou and Giannikos, 2007). An alternative strategy for solving such problems involves partitioning the service area into smaller ones and then developing individual location models for each sub-area. While such an approach could potentially lead to a "system-wide level" sub-optimal design, from a practical perspective it provides a set of tractable models along with an efficient design in each sub-area. Similar approaches have been followed by researchers for emergency response systems (Zografos et al., 2002), facility location (Novaes et al., 2009), and arc routing (Mourao et al., 2009).

Similarly, the problem of positioning transit mobile repair units (TMRU) in a large surface transit network involves the consideration of numerous bus lines with individual buses spread along them; this implies a considerable number of potential "demand generating" points and deployed TMRUs, in an extended service area. Therefore, a stepwise approach is proposed, consisting of two steps, as shown in Fig. 1:

1. Districting of the overall service area and determination of necessary TMRUs (Step A),
2. Optimal location of TMRUs within each district (Step B).

In step A, the service area is partitioned into sub-areas (called superdistricts) while, in parallel, the necessary number of TMRUs is determined for each district. A Genetic Algorithm is combined with an approximation of hypercube model for obtaining best superdistricts and associated TMRU numbers – that approximation is described in detail in the next section. With both the superdistricts and their TMRU requirements defined, step B proceeds in the optimal deployment of TMRUs within the district. Efficient locations for the units are determined through a hybrid-metaheuristic algorithm which again exploits genetic algorithms along with the spatial queueing model (Geroliminis et al., 2009).

4.2. Approximate hypercube with homogeneous demand

Consider an infinite area with homogeneous demand λ_h accidents/hour/km^2. If all servers are located symmetrically in the area, average workloads and service rates will be the same along all servers, because boundary conditions are omitted. Let's now focus on a subset of the area, A_i, where N servers are located and demand for intervention is $\lambda = \lambda_h A_i$. As each server may be idle or busy, there are 2^N different states for this system, which create a hypercube of dimension N. Then, we can describe this system as a finite state continuous time Markov process with N servers. As all servers are identical all the vertices of the same hyperplane (vertices with the same number of busy servers) have the same steady-state probabilities. This suggests that instead of a system with 2^N linear equations, only N linear equations describe the system and determine steady-state probabilities.

The notation needed for this type of formulation is simplified (when compared with the general hypercube formulation). Denote P_j the probability of a vertex in the jth hyperplane (j busy servers), λ_j, μ_j the upward and downward transitions for j busy servers.

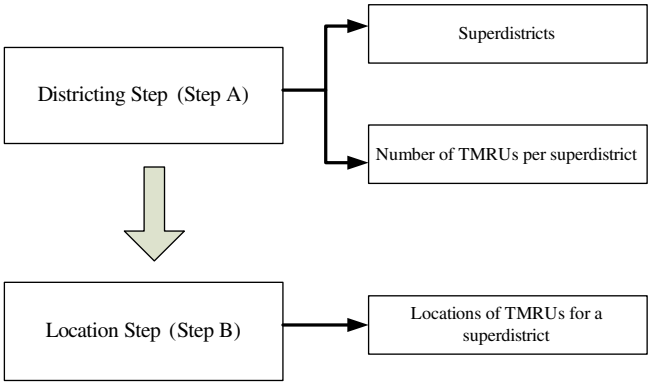


Fig. 1. TMRU deployment approach.

Notice that a vertex in the jth hyperplane is connected with vertices in the $(j-1)$th hyperplane and with $N-j$ vertices in the $(j+1)$th hyperplane.

$$\sum_{j=0}^{N}\binom{N}{j}P_j = 1, \tag{6}$$

$$P_j((N-j)\lambda_j + j\mu_{j-1}) = jP_{j-1}\lambda_{j-1} + (N-j)P_{j+1}\mu_j, \quad j = 1,2,\ldots,n. \tag{7}$$

Eq. (6) states that the sum of the steady-state probabilities should be equal to 1, where $\binom{n}{j}P_j$ is the probability that exactly j servers are busy. Upward and downward transition rates are estimated as follows. Rate λ_j is the demand for requested service that will create a transition from one vertex of the jth hyperplane to a vertex of the $(j+1)$th hyperplane and equals the sum of interdistrict (served by the nearest server) and intradistrict (served by another server as the nearest is not available) demand. If we assume up to 2nd order dispatching policy (service in a region is provided by the responsible (nearest) server and the 2nd nearest if the first one is busy at a different location) and symmetry among all servers, then

$$\lambda_j = \frac{\lambda}{N} + \frac{\lambda j}{N(N-1)}. \tag{8}$$

The first component in (8) is the demand for the available servers (1st order dispatching policy) while the second component is the 2nd order dispatching policy. Fig. 2a illustrates how an area A_i is partitioned into N homogeneous circles (one area of responsibility for each of the N servers) and how upward and downward transition rates are estimated for an example with 7 servers and transition from 2 to 3 busy servers. Fig. 2b shows the intervention area of the server that changes state in Fig. 2c for 1st and 2nd order dispatching; that is, R_1 is the primary area of responsibility (nearest server), while R_2 shows the area that, if a request occurs, this server should intervene as the nearest servers are busy in other locations (colored circles in Fig. 2).

Variable μ_j, is the service rate corresponding to the aforementioned demand λ_j, and is estimated as the *harmonic mean* of service rates for all interdistrict and intradistrict requests of demand λ_j (the reciprocal of the arithmetic mean of the reciprocals),

$$\mu_j = \frac{\left(1+\frac{j}{N-1}\right)}{\left(\frac{1}{\mu_{int}} + \frac{j}{(N-1)\mu_{ext}}\right)}, \tag{9}$$

where μ_{int} is the average service rate for interdistrict responses, while μ_{ext} is the average service rate for intradistrict responses. The average response time in this case, rather than Eq. (1) which represents the general case, becomes

$$\tau = \frac{1}{\sum_{j=0}^{N}\binom{N}{j}P_j\mu_j}. \tag{10}$$

Note: An interesting observation is that the steady-state probabilities of an approximate hypercube when $\mu_{int} = \mu_{ext}$ estimated through Eqs. (5) and (6), are very close to the steady-state probabilities of an M/M/n queueing system, when probability of a lost call is small (for example, for $N = 8$ servers, $\lambda = 3$ accidents/hour, $\mu_{int} = \mu_{ext} = 1$ we get an average error of the steady-state probabilities equal to 1.7%).

The average service rate for interdistrict and intradistrict responses, μ_{int} and μ_{ext} respectively, are estimated as follows:

Each server's area of responsibility is A_i/N. If we approximate this area with a circle of radius $r = \sqrt{\frac{A_i}{N\pi}}$, the average distance between a random point in the circle and a point distance from the center of the circle is

$$d(a,r) = \frac{\int_{-r}^{r}\int_{-\sqrt{r^2-y^2}}^{\sqrt{r^2-y^2}}\sqrt{(x-a)^2+y^2}\,dx\,dy}{\int_{-r}^{r}\int_{-\sqrt{r^2-y^2}}^{\sqrt{r^2-y^2}}1\,dx\,dy}. \tag{11}$$

The average distance traveled for an interdistrict response, d_{int}, is approximated with the average distance between a random point in a circle and its center, $d(0,r) = 0.67r$, while the average distance traveled for an intradistrict response, d_{ext}, is approximated with the average distance between a random point in a circle and the center of a circle tangent to the first one, $d(2r,r) = 2.06r$. After some calculations we get

$$d_{int} \cong d(0,r) = 0.376\sqrt{A_i/N}, \tag{12a}$$

$$d_{ext} \cong d(2r,r) = 1.164\sqrt{A_i/N}. \tag{12b}$$

The average time needed per intervention is the sum of the travel time from the server location to the incident plus the time spent at the incident, τ_{in}, and can be given as

$$\frac{1}{\mu_x} = \frac{2d_x}{v} + \tau_{in}, \tag{13}$$

where $x \in \{interdisric, intradistrict\}$ and v is the travel speed of a server.

4.3. Districting algorithm

The problem: Given area A with demand $\lambda(x)$, number of servers, N, and number of sub-regions (call superdistricts from now

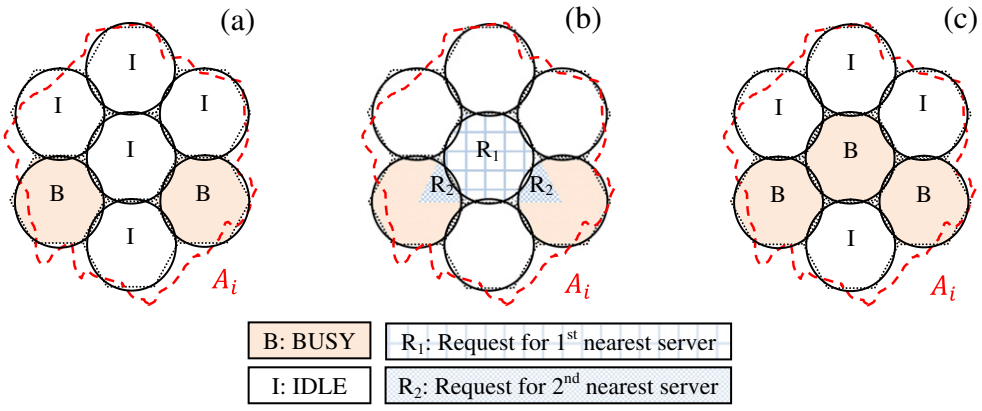


Fig. 2. Area partitioning and transition description for an area with servers: (a) system with 2 busy servers and the responsibility area of each of the servers; (b) request for intervention for the central server showing 1st and 2nd order dispatching; (c) system with 3 busy servers.

on), k, divide A in such a way to minimize mean response time $\bar{\tau}$, where $\bar{\tau} = \frac{1}{\lambda}\sum_i^k \tau_i \lambda_i$ and $N = \sum_{i=1}^k N_i$ (i is a superdistrict with area A_i, N_i, servers and λ_i demand). We will focus only on convex shape superdistricts and we require that the plane has no overlaps or gaps (in art history or mathematics this is frequently referred to as tessellation).

Area A is divided in cells, where each cell has demand d_j, and coordinates (x_j, y_j). To accurately define a possible partition, one should describe the cells belonging to each superdistrict. There is a vast literature and reviews for data classification and partitioning algorithms in many disciplines, e.g. computer science (Jain et al., 1999), health sciences (Clatworthy et al., 2005), geography (Openshaw, 1995), social networks etc. (e.g. using triangles, squares or other polygons of different shapes and sizes). We propose a simple partitioning algorithm not restricted by a specific shape, based on the Voronoi diagrams (Voronoi, 1907). As it is computationally prohibitive to consider all possible partitions, Voronoi diagrams may omit some; given k random points in the surface of A (called superdistrict centroids) "almost" each cell is assigned to the nearest centroid, yielding one superdistrict division. The word "almost" is used to indicate exceptions where a cell may be equidistant to two or more points of A. These centroids have nothing to do with the servers' location and dispatching policy; it is essentially a trick to divide the area in superdistricts and requires minimal computational effort. This procedure can generate the majority of possible divisions of the area in convex sub-areas to avoid restricting our search (an example for 5 superdistricts is given in Fig. 3). The segments of the Voronoi diagram are all the points in A that are equidistant to two centroids; the Voronoi nodes are the points equidistant to three centroids. Voronoi diagrams, in contrast with other partitioning algorithms, require the minimum information/number of variables to de defined, as a partitioning of N clusters can be perfectly described by the location of the N centroids/centers of mass.

For a given set of centroids, there is a unique Voronoi diagram (partitioning). If the number of servers in each partition is defined, such as $N = \sum_{i=1}^k N_i$, then the mean response time can be approximated as follows:

1. Choose k random points (centroids) in area A and assign each cell to the nearest centroid.
2. For each of the k superdistricts compute the total demand λ_i and area A_i.
3. Choose integer N_i's such that $N = \sum_{i=1}^k N_i$.
4. Compute μ_{int}^i, μ_{ext}^i for each superdistrict.
5. Estimate $\tau_i = f(n_i.A_i, \lambda_i, \mu_{int}^i, \mu_{ext}^i)$, where $f(\cdot)$ is the "approximate hypercube" operator that uses (6)–(10) and then $\bar{\tau}$.

In total there are $2k$ decision variables (k for the centers of superdistricts and k for the N_i's). To estimate the partitioning and the number of servers for each superdistrict that minimizes the average response time a GA approach is employed. The role of the k centroids in the partitioning algorithm is *only* to define the superdistricts. In the general case, to define superdistricts for an area with I cells (e.g. I = 500), one needs to introduce $I \times N$ binary variables, which is a very expensive procedure to be embedded in the algorithm and would slow down the optimization. Instead, by using Voronoi diagrams, only N variables are needed, the centers of mass or centroids.

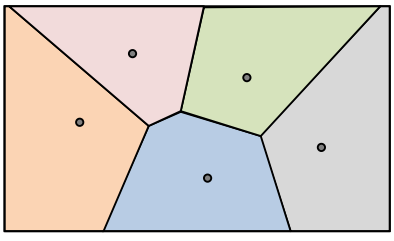


Fig. 3. A possible partition for superdistricts, based on Voronoi diagrams.

The analysis presented here considers cooperation of servers and secondary dispatches (when the nearest server is not available) only within each superdistrict and omits inter superdistrict responses. From a practical perspective upper level intervention is not desirable in emergency response systems, as system efficiency and reliability decreases dramatically. We should note that for systems where intervention times are small comparable to the time spent at the incident site, higher order dispatches might improve overall performance. One could approximate inter superdistrict responses with Eqs. (3)–(5) from the exact hypercube model, by considering a "superserver" for each superdistrict.

4.4. Optimization framework

As mentioned in previous sections, a genetic algorithm is combined with the hypercube model, in an effort to determine superdistrics and locations of TMRUs in steps A and B. Genetic algorithms (GA) were first introduced by Holland (1975) and are described as search techniques based on the process of natural evolution (Goldberg, 1989). Over the past two decades, GAs have been widely implemented in solving difficult optimization problems, with a rich literature of relevant papers and textbooks describing relevant applications in location theory, network design, scheduling problems, optimization of structures etc. Indeed, GAs exhibit inherent advantages such as their robust performance when solving combinatorial problems (Gen and Cheng, 2000) as well as their ability to incorporate external declarations/procedures and logical conditions to the optimization procedure and to handle discrete variables and non-linear constraints in a straightforward manner (Chakroborty, 2003). These qualities make GAs particularly attractive for potential combination with other methods and external procedures such as hypercube, with the background section revealing relevant past approaches.

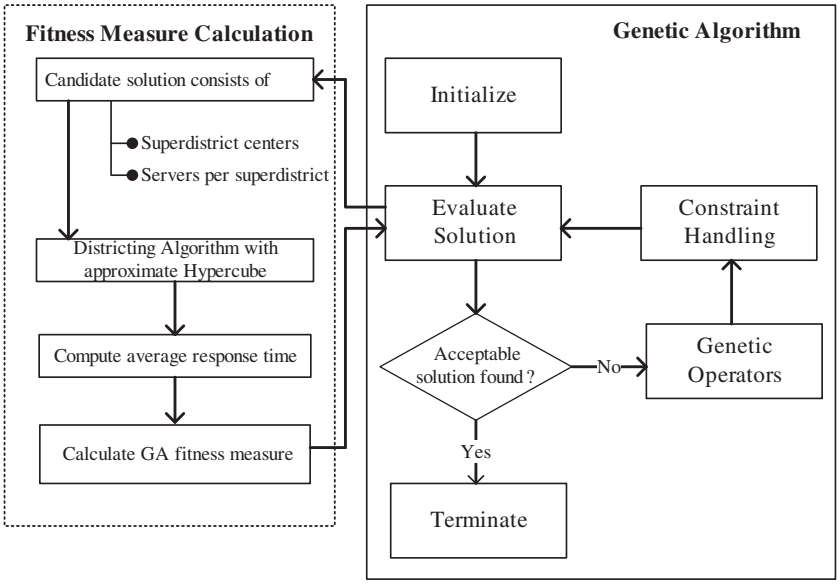
In the same context, we use GAs for obtaining optimal superdistricts and number of servers per superdistrict (Step A) and server locations (Step B). An external procedure (incorporating Hypercube) acts in both steps as the fitness measure and guides the GA into improved solutions. The associated framework for embedding the GA with hypercube for Steps A and B is presented in Fig. 4a and b respectively.

4.4.1. Optimization framework and GA Fitness measures

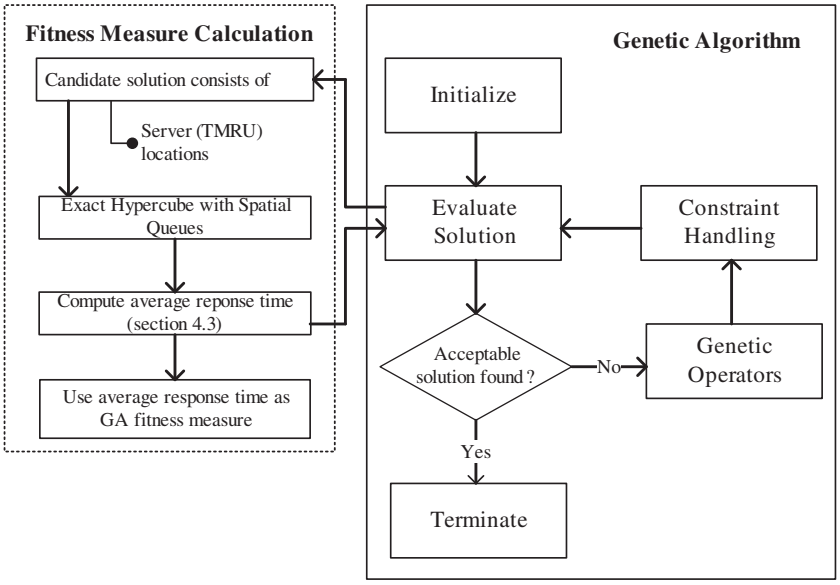
For Step A (Fig. 4a), the GA generates candidate solutions, which include superdistrict centers and number of servers per superdistrict. The districting algorithm of Section 4.3 is the used for determining superdistricts i and their average response time τ_i. The average response time for the overall area is then calculated as:

$$\bar{\tau} = \frac{1}{\lambda}\sum_{i=1}^{k} \tau_i \lambda_i. \tag{14}$$

Eq. (14) is used as a fitness measure, which guides the GA into superdistricts with a minimized average response time. However $\tau_i = f(n_i \cdot A_i, \lambda_i, \mu_{int}^i, \mu_{ext}^i)$ estimates the average response time for a superdistrict with homogeneous demand. As a result, if $\bar{\tau}$ is used as a fitness measure, the GA converges to a solution with the maximum number of servers concentrated in a single, large superdistrict and minimum few servers to the rest of the superdistricts (a fact also indicated by preliminary tests of the algorithm). This large superdistrict includes the part of the network with the highest demand and integrates some areas with lower demand and thus, the



(a) Optimization Framework for districting (Step A)



(b) Optimization Framework for location (Step B)

Fig. 4. Framework of hypercube incorporation in a GA.

average demand applied in τ_i for the large superdistrict is smaller than it should be. To avoid that algorithmic ineffectiveness, we added one additional component to the average response time of the approximate hypercube (representing the inhomogeneity of superdistrict i). The metric for this inhomogeneity is the standard deviation of spatial variation of demand across all cells that belong to superdistrict i, V_i. Thus, instead of τ_i, the average response time is approximated as $\tau_i + \delta \cdot V_i$, where δ is a weighting factor. The final fitness measure is then calculated by Eq. (14).

Similarly, in Step B, the exact hypercube with spatial queues (SQM) is used for evaluating performance of TRMUs for candidate locations again generated by a GA. In that case, the sum of average response times for all TRMUs (as obtained by SQM) acts as the GAs fitness measure; that measure is expected to be minimized through the GA optimization procedure.

4.4.2. Representation scheme

Both the districting and location steps require the determination of points along an area (superdistrict centers and exact server positions respectively). Furthermore, in the districting Step A, the number of servers needed per superdistrict should be derived. Assume that the analysis area consists of a set $Q = \{q\} = \{1, \ldots m\}$ of elementary quadrants (implying that each quadrant has a unique identification number between 1 and m). Each superdistrict $s \in S$ (S is the set of superdistricts) is expected to include some of these quadrants $Q_s \subseteq Q$and have a central point $q_s \in Q$ The number of TMRUs per superdistrict is denoted as g_s. Each candidate solution for districting Step A includes the superdistrict centers q_s and TMRU number per superdistrict g_s; these are represented by two integer-value strings

$$[q_1 \quad \ldots \quad q_s], \tag{15}$$

$$[g_1 \quad \cdots \quad g_s]. \tag{16}$$

For example, a candidate set of centers for an area of 60 quadrants divided in three superdistricts would be represented as [3 10 21], meaning that quadrants 3, 10 and 21 would be the centroids of the superdistricts. Futhermore, assuming a total of 10 servers, a potential solution would be represented as [4 3 3] – the first superdistrict would have four servers, while the second and third would each have three.

As for the location Step B, TMRUs per superdistric s will be located in quadrants $q_{l,s} \subseteq Q_s$. Since the number of TMRUs for each superdistrict s is fixed (obtained through Step A), the following integer representation is used for the location step:

$$[q_{1,s} \quad \cdots \quad q_{l,s}]. \tag{17}$$

Again, if a superdistrict of 20 quadrants was assigned to 4 TRMUs, a candidate solution for Step B would be [3 8 12 15], meaning that TMRUs would be located at quadrants 3, 8, 12 and 15. Steps A and B exploit a districting algorithm and SQM respectively, as fitness measures for evaluating candidate solutions.

4.4.3. Genetic operators and constraint handling

Selection and variation operators used are the same for both steps and are implemented independently for each representation scheme (for the Step A case). A Binary Tournament Parent Selection method, as described by Goldberg (1989) and proposed by Syswerda (1989). The method is straightforward to implement and according to Beasley and Chu (1996) gives results comparable in quality to those of other methods. According to this approach, a pair of strings is randomly selected from the population and the best individual of the pair is selected for reproduction. The process is repeated until a new population is set. A uniform crossover method is used (Gen and Cheng, 2000); that method randomly selects genes from each parent string to perform crossover according to a probability value and the probability value defines the percentage of genes for each string to be crossed over. Regarding mutation, the mutated gene is replaced by a randomly generated number within its valid range. Optimal mutation rates range, according to a rule-of-thumb between $1/P$ and $1/l$, where P is the population size and l is the string length (Eiben and Smith, 2003). The population replacement method used replaces worse strings from the old population by best strings created by the recombination, crossover and mutation operators. In both cases, resource constraints do exist (available number of TMRUs); we consider these as hard constraints and discard all violating solutions. The GA is in both steps terminated when the solution could not improve more than 1% after a number of iterations.

5. Application and results

The model is demonstrated for the case of the Athens (Greece) surface public transportation network; the network consists of over 390 bus and electric bus lines, with a daily passenger demand of 1.7 million passengers, spread over an area of about 650 km^2, and served by over 3000 buses of different sizes. The Athens Public Transport Organization (OASA) is responsible for planning and managing the bus system, while daily operations are handled by the city's bus company (ETHEL). In an effort to provide high level services, OASA uses approximately 25 TMRUs, ready to respond and provide rapid repair services in cases of bus accidents and/or malfunctions, tow-away of illegally parked vehicles (along bus lanes), and so on. Currently, all transit repair vehicles are stationed in the bus depots of OASA (located in the city's outskirts) and need on average 30 minutes to approach an incident site. Step A and B are applied to the Athens network. The implementation of the developed algorithm in this paper considers up to 3rd order districting when solving the exact hypercube model of Step B. This is a realistic assumption, given that the probability of having more than 3 accidents in the aggregated area of responsibility of 3 adjacent servers in a 15 minutes interval is less than 1%.

In order to apply the model to the examined network, we follow the approach of Karlaftis et al. (2004) which divides the network according to a grid of 1 km $\times$ 1 km cells (with each cell corresponding to a hypercube atom and a GA quadrant; a map of the city and the grid is shown in Fig. 5). Incident rates per cell are then derived as a function of the total length of bus lines within the cell and actual 10-year statistics on the average number of incidents per line type and vehicle size within the network. We assumed each server to be located in the center of each cell, while a Manhattan distance is considered between cells. The advantage of such a representation is related to the relatively frequent changes in bus routes of the Athens network; incident rates per cell can be easily recalculated as a result of obtaining new cell service 'density'.

5.1. Comparison of methods for small problem instances

To evaluate the performance of the developed two-step heuristic, we provide an application of the algorithm to two small regions (approximately 10 $\times$ 10 cells) of the Athens network. Region A is a medium demand region South-East of the Athens center, while region B is one of the highest demand region in the city center. We solved two instances of the exact hypercube optimization model (HC), as described by (1)–(5) for 7 servers (region A) and 10 servers (region B) using GA. Our goal was to obtain solutions as close to the optimum as possible and we let the GA to run for long times. We also estimated the approximate solutions with (i) the developed two-step heuristic for the same total number of servers divided in 2 superdistricts (M); (ii) an approximation of this heuristic (M-appr) without optimization for the size of superdistricts (only the number of servers per superdistrict was a variable; each region was horizontally divided in 2 equally sized superdistricts) and (iii) a p-median solution. After obtaining the location of servers, for each of the 3 problems, we applied these locations to an exact hypercube instance to estimate the mean response time (columns 3 and 5 of Table 1).

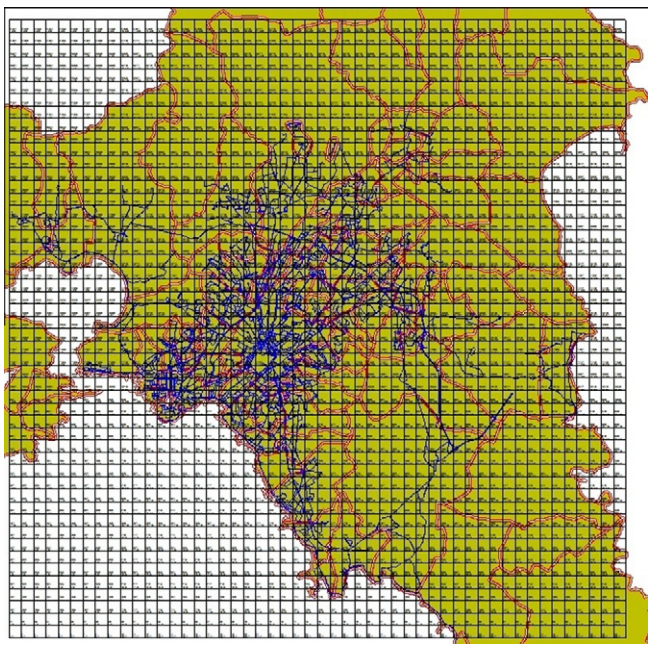


Fig. 5. Maps and model grid for the Athens Surface Public Transportation Network.

Table 1
Comparison of different approaches for a small network.

Mean response time (hour)	Network A (7 servers)		Network B (10 servers)	
	GA solution	Locations applied to HC	GA solution	Locations applied to HC
Exact (HC)	0.162	0.162	0.197	0.197
2-step (M)	0.175	0.165 (+1.8%)	0.221	0.204 (+3.5%)
2-step (M-appr)	0.192	0.185 (+14.2%)	0.246	0.232 (+17.7%)
p-median		0.173 (+6.8%)		0.233 (+18.3%)

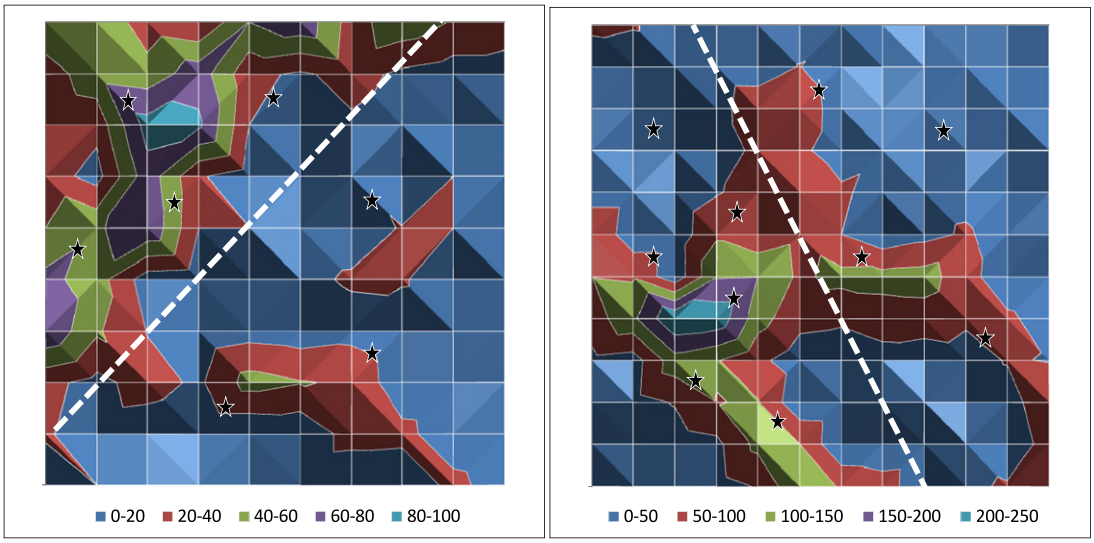


Fig. 6. Contour plot of demand for regions A and B and location of servers according to the 2-step model (M).

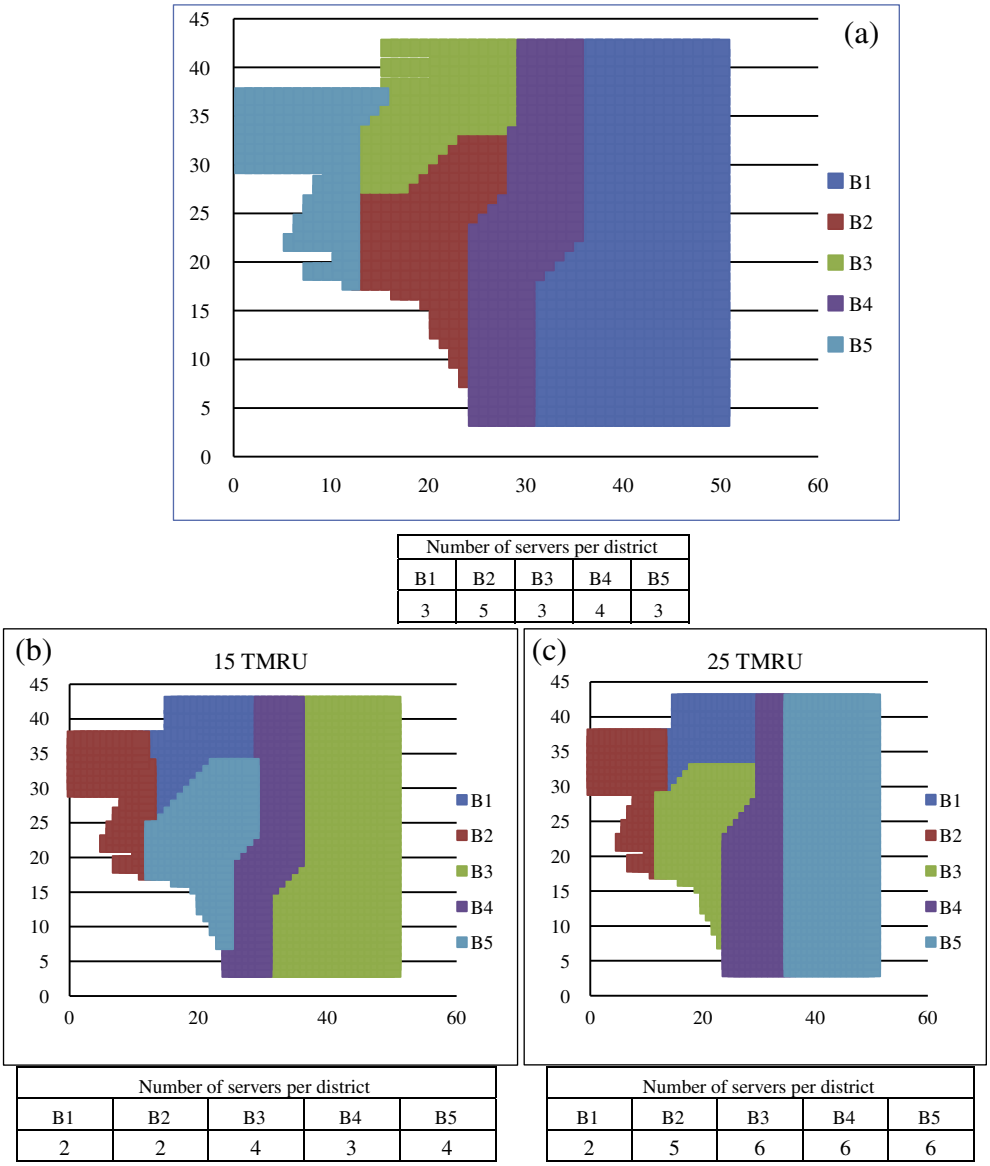
The results are summarized in Table 1. The 2-step heuristic provides near-optimal solutions (errors 1.8% for region A and 3.5% for region B), while both the *p*-median and the 2-step approach without optimizing the size of superdistricts (M-appr), provide much higher errors. Fig. 6 provides a contour plot of the demand for region A and B, the location of servers for the two-step heuristic (M) and the separation line of the two superdistricts.

5.2. Districting application and results

Firstly, the two-step heuristic presented in Section 4 is solved for different demand levels and different number of TMRUs. Demand scenarios vary between the average daily demand (ADD) for interventions and the peak-hour demand which was assumed 10 times higher than ADD, as reported by OASA organization.

Table 2
Districting results.

ID	Population size	Crossover rate	Mutation rate	Tot-t	Variance	Objective function
1	20	0.2000	0.0500	735.6529	158.9520	2325.1725
2	20	0.2000	0.1000	731.3319	160.5516	2336.8480
3	20	0.2000	0.2000	736.3738	160.6393	2342.7670
4	20	0.4000	0.0500	731.0644	169.2480	2423.5445
5	20	0.4000	0.1000	736.2546	158.7998	2324.2530
6	20	0.4000	0.2000	737.7657	159.0544	2328.3097
7	20	0.6000	0.0500	736.9229	160.1418	2338.3412
8	20	0.6000	0.1000	728.9308	166.7685	2396.6161
9	20	0.6000	0.2000	736.2505	161.0616	2346.8669
10	40	0.2000	0.0500	731.1240	159.6966	2328.0899
11	40	0.2000	0.1000	735.8488	155.6935	2292.7840
12	40	0.2000	0.2000	736.0291	159.8463	2334.4916
13	40	0.4000	0.0500	737.0534	165.0621	2387.6746
14	**40**	**0.4000**	**0.1000**	**734.2163**	**153.2082**	**2266.2980**
15	40	0.4000	0.2000	736.7877	159.2933	2329.7210
...	...	...	...	...	...	...
25	60	0.6000	0.0500	735.9129	156.4422	2300.3353
26	60	0.6000	0.1000	736.1473	163.5624	2371.7715
27	60	0.6000	0.2000	735.7490	153.3565	2269.3140
Average						2334.0391
Standard deviation						39.69624
Coefficient of variation						1.7%



Number of servers per district				
B1	B2	B3	B4	B5
3	5	3	4	3

Number of servers per district				
B1	B2	B3	B4	B5
2	2	4	3	4

Number of servers per district				
B1	B2	B3	B4	B5
2	5	6	6	6

Fig. 7. (a) Best districting solution for 20 servers (ID 14); (b) districting solutions for 15 and (c) 25 servers (for the best districting solution GA parameters corresponding to ID 14).

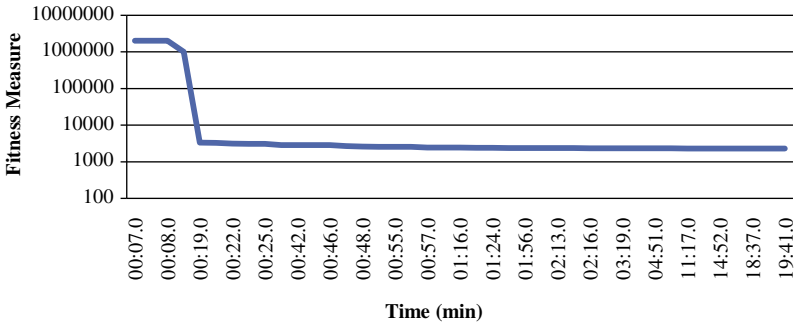


Fig. 8. Fitness function evolution for Step A with respect to time (ID 14).

Three values of TMRUs are applied, 15, 20 and 25. As explained in Section 4.4.1, the average response time per superdistrict is estimated by (15) plus a heterogeneity component equal to the standard deviation, V_i, of demand among all cells in a superdistrict, i, multiplied by a weighting factor δ. Factor δ was estimated as the best fit value that minimizes the error between different instances of the exact hypercube (Eqs. (1)–(5)) and the approximate hypercube $\tau_i + \delta \cdot V_i$ for different areas of the study network and for different server locations. The value of δ applied in the model is 8.2 with units hour/$\frac{\text{accident}}{\text{hour}}$. Results were obtained for different GA parameters and for the highest demand levels. They are summarized in Table 2. Results differ by 1.7% at most, providing a good indication of algorithmic performance and robustness.

Table 3
Optimal location results for various GA parameters.

ID	Population size	Crossover rate	Mutation rate	Objective function (h)
1	20	0.2	0.05	0.2307
2	20	0.2	0.1	0.2096
3	20	0.2	0.2	0.1934
4	20	0.4	0.05	0.2006
5	20	0.4	0.1	0.2013
6	20	0.4	0.2	0.1867
7	20	0.6	0.05	0.2025
...	...	...	...	...
18	**40**	**0.6**	**0.2**	**0.1869**
19	60	0.2	0.05	0.2066
20	60	0.2	0.1	0.2021
21	60	0.2	0.2	0.1969
22	60	0.4	0.05	0.1934
23	60	0.4	0.1	0.1972
24	60	0.4	0.2	0.1991
25	60	0.6	0.05	0.2097
26	60	0.6	0.1	0.2027
27	60	0.6	0.2	0.1904
Average				0.2005
Standard deviation				0.009197
Measure of coefficient				4.59%

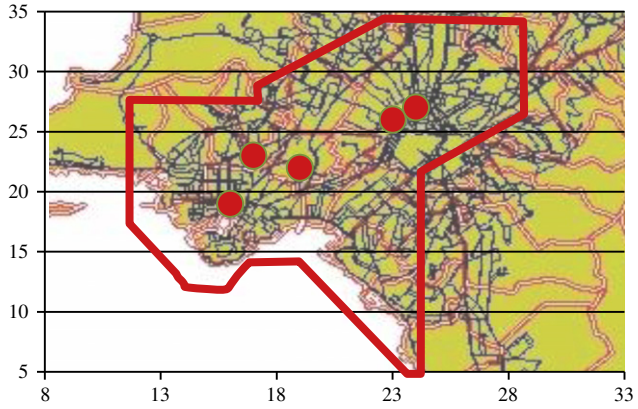


Fig. 9. TMRU locations in the Athens CBD for best solution (ID 18).

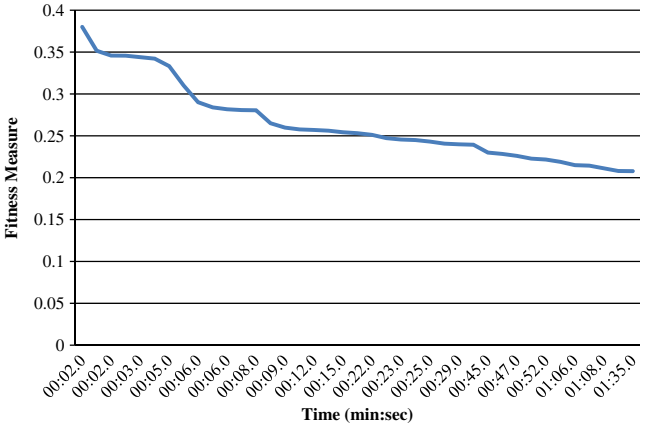


Fig. 10. Fitness function evolution for Step B with respect to time (ID 18).

Fig. 7a presents the results for the best derived solution (ID 14), while Fig. 7b and c show districting results for 15, 20 and 25 TMRU respectively. According to Fig. 7a, district B2 includes the Athens central business district (CBD) and densely populated suburbs; since the Athens bus network is relatively radial, most of that metropolitan area's bus lines cross and/or serve that district. This explains the finding that, compared to the district's size, there is an increased need for TMRUs. Fig. 8 results indicate again that the CBD is contained in a central superdistrict, with an increased number of TMRUs allocated in that super district. Also, the eastern district has a large number of TMRUs, which is attributed to the size and considerably lower density of bus lines in that part of the city.

Further to the above results, Fig. 8 depicts evolution of the GA fitness function value with respect to time for a total of about 20 minutes, on a Pentium Core Duo processor with 1 GB of RAM. As can be seen from Fig. 8, improvement of the solution after about two minutes of running time is very low, indicating that the algorithm can provides good results in a relatively low amount of time.

5.3. Location of TMRUs

We apply Step B (the exact SQM as developed by Geroliminis et al., 2009) for all districts using the GA approach of Section 4.4. We have to notice that in the aforementioned paper, the authors used a different heuristic to approximate the optimal locations (a random search followed by a steepest decent method). The results of the methods are similar, but the GA approach has been proved faster. We show here detailed results of the GA approach for district B2 since that district contains most of the system's bus lines, it is the most demanding among districts, in terms of TMRU services. Results for Step 2, district B2 and various GA parameters are presented in Table 3.

Again, results indicate low differences of the objective function for various GA parameters, while solution with ID 18 provides the best results depicted in Fig. 9. Fitness function evolution for ID 18 with respect to time, is presented in Fig. 10; it takes less than two minutes (on a Pentium Core Duo with 1 GB of RAM) to reach a good solution.

Furthermore, by comparing obtained results with the current TMRU response time (for the existing configuration where TMRUs are stationed in the transit system depots), the improvement is significant. Indeed, the new configuration has an average response time of 12 minutes (0.2 hour), which is lower than the current 30 minutes average response time.

6. Decision support system

The need for a tool that can aid planners in obtaining alternatives regarding possible areas of responsibility and locations of TMRU's, led to the development of a Decision Support System (DSS), tailor made for that purpose. The DSS incorporates the districting and location models of Steps A and B, a genetic algorithm solving module, a database, and an export procedure to a geographic information system (GIS), so that maps depicting the DSS results can be produced. A graphical user interface (GUI) is used to manage the database, model and GA components. In particular, the DSS consists of a set of modules, which are used for:

- Formulating and solving the models of Steps A and B.
- Setting the genetic algorithm parameters (population size, mutation and crossover rates, convergence time, etc.),
- Setting the service parameters (number of servers, travel speed, time at the incident, etc.),
- Performing sensitivity analysis of model and GA parameters,
- Producing reports, and
- Producing GIS maps with server locations and areas of responsibility.

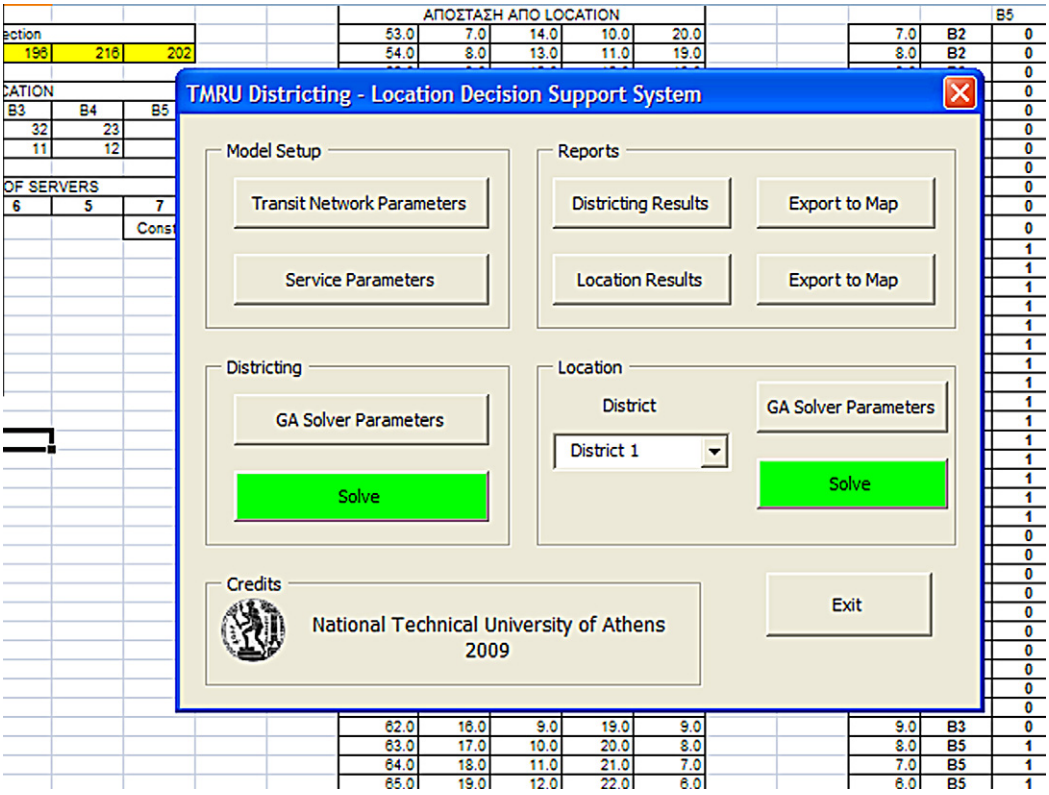


Fig. 11. Main GUI screen.

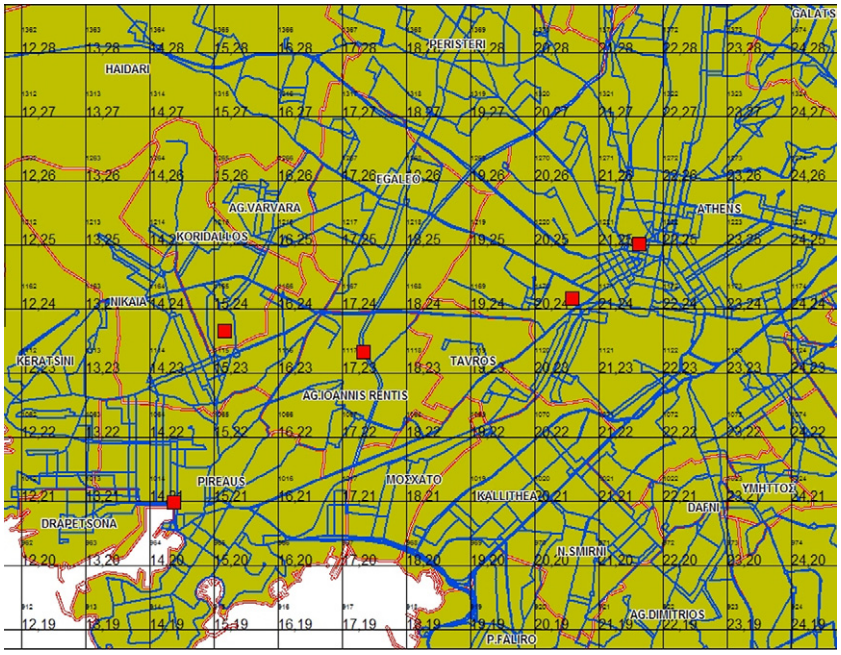


Fig. 12. Location outputs in GIS.

The DSS is built using MS-EXCEL™ spreadsheets and VBA™ environment; the rationale behind the selection of such an environment was its popularity among organizations worldwide, along with its user-friendliness and ability for efficiently setting up the models. The EXCEL™ environment is used for storing and manipulating network and service data, as well as the for model setup and any intermediate (worksheet based) calculations, while the GUI interface and solvers are built with the use of VBA™ and external libraries (Fig. 11 presents the main GUI screen for the DSS).

Through the GUI, the user can modify transit network parameters (number and type of lines for each quadrant) and service parameters. Further, GA parameters for each model step can also

be altered to perform sensitivity analyses and obtain improved results if possible. The GA solving component is triggered through the GUI and results are extracted in the forms of reports; the latter can also be exported to GIS maps (such a map is presented in Fig. 12 for the case of Fig. 9 locations).

7. Conclusions

In this paper we formulated an analytical approach for optimally deploying *many* emergency response units in an urban transportation network, and presented a heuristic solution methodology for that purpose. We also presented an application for TMRUs (transit mobile repair units) in the city of Athens, Greece. The developed model integrates a hypercube queuing model, a location model, and a genetic algorithm for obtaining appropriate unit locations in a two-step approach as a one-step approach is computationally infeasible. The results from the model application indicate that the proposed model is an important optimization tool, particularly in cases of high demand where the responsible server for an incident is not available to intervene although needed, and when many servers need to be located. Ongoing and future research includes extensions of the proposed model to large scale transportation networks where more than one servers may be needed to intervene during the same incident. Extensions of the model to capture demand scenarios during different times of the day is also an area of some research priority.

Acknowledgements

The authors thank the two anonymous referees for the useful comments and suggestions and Athens Public Transport Organization (OASA) for providing the data for the application of the model.

References

Araz, C., Selim, H., Ozkarahan, I., 2007. A fuzzy multi-objective covering-based vehicle location model for emergency services. Computers and Operations Research 34, 705–726.

Atkinson, J.B., Kovalenko, I.N., Kuznetsov, N., Mykhalevych, K.V., 2006. Heuristic solution methods for a hypercube queueing model of the deployment of emergency systems. Cybernetics and Systems Analysis 42 (3), 379–391.

Atkinson, J.B., Kovalenko, I.N., Kuznetsov, N., Mykhalevych, K.V., 2008. A hypercube queueing loss model with customer-dependent service rates. European Journal of Operational Research 191 (1), 223–239.

Batta, R., Dolan, J.M., Krishnamurthy, N.N., 1989. The maximal expected covering location problem: Revisited. Transportation Science 23 (3), 277–287.

Beasley, J.E., Chu, P.C., 1996. A genetic algorithm for the set covering problem. European Journal of Operational Research 94, 392–404.

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

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

Berman, O., Krass, D., 2002. Facility Location problems with Stochastic Demand and Congestion. In: Drezner, Z., Hamacher, H.W. (Eds.), Facility Location: Applications and Theory. Springer-Verlag, NY.

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

Brandeau, M., Larson, R.C., 1986. Extending and applying the hypercube queuing model to deploy ambulances in Boston. In: Swersey, A.J., Ingnall, E.J. (Eds.), Delivery of Urban Services. TIMS Studies in the Management Sciences, vol. 2. Elsevier, pp. 121–153.

Brotcone, L., Laporte, G., Semet, F., 2003. Ambulance location and relocation models. European Journal of Operational Research 147, 451–463.

Burwell, T.H., Jarvis, J.P., McKnew, M.A., 1993. Modeling co-located servers and dispatch ties in the hypercube model. Computers and Operations Research 20, 113–119.

Burwell, T.H., McKnew, M.A., Jarvis, J., 1992. An application of a spatially distributed queuing model to an ambulance system. Socio-Economic Planning Sciences 26, 289–300.

Calvo, A., Marks, H., 1973. Location of health care facilities: An analytical approach. Socio-economic Planning Sciences 7, 407–422.

Carbone, R., 1974. Public facility location under stochastic demand. INFOR 12, 261–270.

Carson, Y., Batta, R., 1990. Locating an ambulance on the Amherst Campus of the State University of New York at Buffalo. Interfaces 20, 43–49.

Chakroborty, P., 2003. Genetic algorithms for optimal urban transit network design. Computer Aided Civil and Infrastructure Engineering 18 (3), 184–200.

Chelst, K., Barlach, Z., 1981. Multiple unit dispatches in emergency services: models to estimate system performance. Management Science 27 (12), 1390–1409.

Church, R.L., ReVelle, C., 1974. The maximal covering location problem. Papers of the Regional Science Association 32, 101–118.

Clatworthy, J., Buick, D., Hankins, M., Weinman, J., Horne, R., 2005. The use and reporting of cluster analysis in health psychology. British Journal of Health Psychology 10 (3), 329–358.

Coppersmith, D., Winograd, S., 1990. Matrix multiplication via arithmetic progressions. Journal of Symbolic Computation 9, 251–280.

Current, J., Daskin, M., Schilling, D., 2002. Discrete network location models. In: Drezner, Z., Hamacher, H.W. (Eds.), Facility Location: Applications and Theory. Springer-Verlag, NY.

Daskin, M.S., 1983. A maximum expected covering location model: Formulation, properties and heuristic solution. Transportation Science 17, 48–70.

Dimopoulou, M., Giannikos, I., 2007. Advances in Location Analysis. European Journal of Operational Research 179 (3), 923–926.

Drezner, Z., Hamacher, H.W., 2002. Facility Location: Applications and Theory. Springer-Verlag, NY.

Eiben, A.E., Smith, J.E., 2003. Introduction to Evolutionary Computing. Springer-Verlag, Berlin, Germany.

Galvão, R.D., Chiyoshi, F.Y., Morabito, R., 2005. Towards unified formulations and extensions of two classical probabilistic location problems. Computers and Operations Research 32, 15–33.

Galvão, R.D., Morabito, R., 2008. Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems. International Transactions in Operational Research 15 (5), 525–549.

Gen, M., Cheng, R., 2000. Genetic Algorithms and Engineering Optimization, Interscience, U.S.A.

Gendreau, M., Laporte, G., Semet, F., 1997. Solving an ambulance location model by Tabu search. Location Science 5, 75–88.

Gendreau, M., Laporte, G., Semet, F., 2001. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. Parallel Computing 27 (12), 1641–1653.

Geroliminis, N., Karlaftis, M.G., Skabardonis, A., 2009. A spatial queuing model for the emergency vehicle districting and location problem. Transportation Research Part B 43 (7), 798–811.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and machine Learning. Addison-Wesley, New York.

Goldberg, J., Dietrich, R., Chen, J.M., Mitwasi, M.G., 1990. A simulation model for evaluating a set of emergency vehicle base locations: development, validation and usage. Socio-Economic Planning Sciences 24, 124–141.

Goldberg, J.B., 2004. Operations research models for the deployment of emergency services vehicles. EMS Management Journal 1 (1), 20–39.

Hakimi, S.L., 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research 12, 450–459.

Halpern, J., 1977. Accuracy of estimates for the performance criteria in certain emergency service queuing systems. Transport Science 11, 223–242.

Hogan, K., ReVelle, C.S., 1986. Concepts and applications of backup coverage. Management Science 34, 1434–1444.

Holland, J.H., 1975. Adaptation in natural and artificial systems, Ann Arbor, University of Michigan Press.

Iannoni, A.P., Morabito, R., 2007. A multiple dispatch and partial backup hypercube queuing model to analyze emergency medical systems on highways. Transportation Research E 43 (6), 755–771.

Iannoni, A.P., Morabito, R., Saydam, C., 2008a. A hypercube queueing model embedded into a genetic algorithm for ambulance deployment on highways. Annals of Operations Research 157 (1), 207–224.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. ACM Computing Surveys 31 (3), 264–323.

Jarvis, J.P., 1985. Approximating the equilibrium behavior of multi-server loss systems. Management Science 31, 235–239.

Jia, H., Ordonez, F., Dessouky, M., 2007. A modeling framework for facility location of medical services for large-scale emergencies. IIE Transactions 39, 41–55.

Karlaftis, M., Kepaptsoglou, K., Stathopoulos, A., 2004. "A genetic-algorithm based approach for the optimal location of transit repair vehicles on a complex network", transportation research record 1879, Journal of the Transportation Research Board, pp. 41–50.

Katehakis, M.N., 1985. A note on the hypercube model. Operations Research Letters 3 (6), 319–322.

Larson, R.C., 1974. A hypercube queuing model for facility location and redistricting in urban emergency services. Computers and Operations Research 1, 67–75.

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

Marianov, V., ReVelle, C.S., 1992. The capacitated standard response fire protection sitting problem: deterministic and probabilistic models. Annals of Operations Research 40, 303–322.

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

Marianov, V., Serra, D., 1998. Probabilistic maximal covering location–allocation for congested system. Journal of Regional Science 38, 401–424.

Marianov, V., Serra, D., 2003. Location-allocation of multiple-server service centers with constrained queues or waiting times. Annals of Operations Research 111, 35–50.

Mendonça, F.C., Morabito, R., 2001. Analyzing emergency service ambulance deployment on a Brazilian highway using the hypercube model. Journal of the Operation Research Society 52, 261–268.

Mirchandani, P., Francis, R. (Eds.), 1990. Discrete Location Theory. Wiley-Intersience.

Morabito, M., Chiyoshi, F., Galvão, R., 2008. Non-homogeneous servers in emergency medical systems: Practical applications using the hypercube queueing model. Socio-Economic Planning Sciences 42 (4), 255–270.

Mourao, M.C., Nunes, A.C., Prins, C., 2009. Heuristic methods for the sectoring arc routing problem. European Journal of Operational Research 196 (3), 856–868.

Nicholson, A.J., Du, Z.P., 1997. Degradable transportation systems: an integrated equilibrium model. Transportation Research B 31 (3), 209–223.

Novaes, A.G.N., Souza de Cursi, J.E., da Silva, A.C.L., Souza, J.C., 2009. Solving continuous location-districting problems with Voronoi diagrams. Computers and Operations Research 36 (1), 40–59.

Openshaw, S., 1995. Developing automated and smart spatial pattern exploration tools for geographical information systems applications. The Statistician 44 (1), 3–16.

Paluzzi, M., 2004. "Testing a heuristic *P*-median location allocation model for sitting emergency service facilities". in: The Annual Meeting of the Association of American Geographers, Philadelphia, PA.

Repede, J.F., Bernardo, J.J., 1994. Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. European Journal of Operational Research 75, 567–581.

ReVelle, C.S., Eiselt, H.A., 2005. Location analysis: A synthesis and survey. European Journal of Operational Research 165, 1–19.

Sacks, S.R., Grief, S., 1994. Orlando Police Department uses OR/MS methodology, new software to design patrol districts. OR/MS Today, Baltimore, pp. 30–32.

Saydam, C., Aytug, H., 2002. Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study. European Journal of Operational Research 141 (3), 480–495.

Saydam, C., Aytug, H., 2003. Accurate estimation of expected coverage: Revisited. Socio-Economic Planning Sciences 37, 69–80.

Schilling, D.A., Elzinga, D.K., Cohon, J., Church, R.L., ReVelle, C.S., 1979. The TEAM/FLEET models for simultaneous facility and equipment sitting. Transportation Science 13, 163–175.

Snyder, L.V., 2006. Facility Location under Uncertainty: A review. IIE Transactions 38, 537–554.

Swersey, A.J., 1994. The deployment of police, fire and emergency medical units. In: Pollock, S.M. et al. (Eds.), Handbooks in OR and MS, Vol. 6. Elsevier Science, Amsterdam, pp. 151–200.

Syswerda, G., 1989. Uniform Crossover in Genetic Algorithms. In: David Schaffer, J. (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. Morgan–Kaufmann, MA, pp. 2–9.

Takeda, R.A., Widmer, J.A., Morabito, R., 2007. Analysis of ambulance decentralization in urban emergency medical service using the hypercube queueing model. Computers and Operations Research 34 (3), 727–741.

Toregas, C., Swain, R., Revelle, C., Bergman, L., 1971. The location of emergency service facilities. Operations Research 19 (1), 1363–1373.

Voronoi, G., 1907. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Journal für die Reine und Angewandte Mathematik 133, 97–178.

Zografos, K., Androutsopoulos, K.N., Vasilakis, G.M., 2002. A real-time decision support system for roadway network incident response logistics. Transportation Research Part C: Emerging Technologies 10 (1), 1–18.