An optimization approach for ambulance location and the districting of the response segments on highways

Ana Paula Iannoni
Reinaldo Morabito[1]
Department of Production Engineering
Federal University of Sao Carlos, SP, Brazil
iannoni93@hotmail.com; morabito@power.ufscar.br

Cem Saydam
Department of Business Information Systems and Operations Management
University of North Carolina at Charlotte, NC 28269, USA
saydam@uncc.edu

Abstract: In this paper we present a method to optimize the configuration and operation of emergency medical systems on highways. Different from the approaches studied in the previous papers, the present method can support two combined configuration decisions: the location of ambulance bases along the highway and the districting of the response segments. For example, this method can be used to make decisions regarding the optimal location and coverage areas of ambulances in order to minimize mean user response time or remedy an imbalance in ambulance workloads within the system. The approach is based on embedding a well-known spatially distributed queueing model (hypercube model) into a hybrid genetic algorithm to optimize the decisions involved. To illustrate the application of the proposed method, we utilize two case studies on Brazilian highways and validate the findings via a discrete event simulation model.

Keywords: location and dispatching of ambulances**,** hypercube model, genetic algorithm, spatially distributed queues, highways.

1. Introduction

The operation of many emergency medical systems (EMS) on Brazilian highways is under the management of private organizations as part of privatization contracts with the

[1] Corresponding author: fax 55-16-33518240.

Brazilian government. During the last years, new EMS have been installed along Brazilian highways, and the configuration and operation of the existing EMS have been revisited. In these highway EMS, an ambulance provides the first medical treatment to the individuals involved in an accident, transports them to the nearest hospital (if necessary), and then goes back to its home base on the highway. These systems are typically zero-line capacity, and they operate within particular ambulance dispatching policies, which stipulate that only specific vehicles can be dispatched to a given region on the highway (partial backup), mainly due to the limitations of travel distance or time. In addition, some policies involve multiple dispatching, as in some cases (depending on the type of call), it is necessary to dispatch more than one ambulance to the same call location.

The mean user response time is considered the main performance measure. In general, the limitations for the response time specified in the privatization contract must be followed by the private organizations, which are responsible for managing the highway. Other performance measures to the EMS are: the balance of ambulance workloads, the fraction of calls not serviced by the EMS (loss probability), and the fraction of calls not serviced within a predetermined threshold (i.e., fraction of calls with response times exceeding T minutes). The former measure has especially been utilized by the EMS analysts. For example, the United States EMS Act of 1973 states that 95% of the emergency medical responses should be serviced within 10 minutes in urban areas and within 30 minutes in rural areas (Ball and Lin, 1993). In some EMS on Brazilian highways, this statistic is also used to evaluate the system, and these regulations are specified in the privatization contract.

Studies by Swersey (1994), Owen and Daskin (1998) and Brotcorne et al. (2003) present revisions of the classic location models to analyze the emergency systems developed during the last few decades. In particular, the hypercube model based on spatially distributed queueing theory and Markovian analysis approximations has been one of the most effective methods for analyzing these systems (Larson, 1974; Larson and Odoni, 1981). The model implies the solution of a linear system of O (2^N) equations (N is the number of ambulances in the system), where the variables involved are the equilibrium state probabilities of the system. With these probabilities, a number of interesting and critical performance measures for managing the system can be estimated. Examples of applications of the hypercube model in urban EMS in the United States can be found in studies by Larson and Odoni (1981), Chelst and Barlach (1981), Brandeau and Larson (1986), Burwell et al. (1993) and Sacks and Grief (1994). More recently, the hypercube has been considered as a deployment model for response to terrorism attacks and other major emergencies (Larson, 2004). In Brazil, the hypercube model has been applied to analyze urban EMS (Takeda et al., 2007) and EMS on highways (Mendonça and Morabito, 2001; Iannoni and Morabito, 2007).

Some studies have extended the original hypercube model to remove its limiting assumptions for application to EMS on highways. For example, Mendonça and Morabito (2001) modified the model to consider dispatching with partial backup, Iannoni and Morabito (2007) extended that model to consider multiple dispatching of identical and distinct servers, and Atkinson et al. (2006, 2007) proposed heuristic methods based on the model in the Mendonça and Morabito (2001) study to estimate the loss probability for large-scale systems. Other studies have been focused on combining the hypercube model with optimization procedures, such as those conducted by Batta et al. (1989), Saydam and Aytug (2003), Chiyoshi et al. (2003), Galvao et al. (2005) and Rajagopalan et al. (2007). These studies present successful implementations of hypercube embedded metaheuristic search methods applied to ambulance location problems.

Recently, Iannoni et al. (2008) integrated the hypercube model into a standard genetic algorithm in order to determine the optimal primary and secondary response areas for the ambulances (districting problem), considering their current location, while taking into account different conflicting objectives such as the mean user response time, the imbalance of ambulance workloads and the fraction of calls with response times exceeding a predetermined threshold. In that study it was shown that these different objectives could be better met by simply modifying the atom sizes of the system, without relocating ambulances and without requiring additional capacity investments.

In this study, we extend the study conducted by Iannoni et al. (2008). First, we modify the districting GA/hypercube algorithm to optimize the location of ambulance bases along the highway (location problem), which we call location GA/hypercube algorithm. We assume ambulance bases can be located anywhere along the stretch of highway under study, which is quite different than locating ambulances on a set of previously determined candidate posts (nodes, points). In addition, the location GA/hypercube algorithm includes a local search procedure to evaluate the local neighborhood of each solution generated by the GA operators (hybrid GA algorithm). We show that the location GA/hypercube algorithm provides better solutions for each objective than the districting GA/hypercube proposed in Iannoni et al. (2008). Then, we extend this algorithm to optimize the two combined decisions: the location of ambulance bases (the location problem) and the districting of ambulance response or coverage areas (the districting problem), which we call the location and districting hybrid GA/hypercube algorithm. This algorithm searches for the best ambulance locations and their coverage areas, in order to minimize region-wide response times and/or ambulance workload imbalances in the system. In addition, we discuss how the algorithms can be adapted to generate trade-off curves among the conflicting performance measures.

Computational results are analyzed by applying the algorithms to two case studies. The first case corresponds to an EMS operating on a portion of an interstate highway connecting the cities of Sao Paulo and Rio de Janeiro, which was initially studied by Mendonça and Morabito (2001). The second is an EMS operated by a private firm. It is based on two busy stretches of highway located in the state of Sao Paulo and recently studied by Iannoni et al. (2008) and Iannoni and Morabito (2007). To verify the quality of the solutions produced by the algorithms, we developed a simple procedure that incorporates the hypercube model into a simple enumerative algorithm and provides the optimal configuration for smaller problems (i.e., in terms of N number of ambulances). In order to validate the performance measures obtained by the hypercube model, we compare them with the results obtained via a discrete event simulation model of the system.

This article is organized as follows: Section 2 presents a brief description of the EMS case studies, while section 3 discusses how the hypercube model can be adapted to analyze these EMS systems. Section 4 presents the location hybrid GA/hypercube algorithm (*location problem*), and in section 5, this algorithm is extended to support the combined decisions of regarding ambulance location and coverage area (the *location and districting problem*). Section 6 analyzes the outcomes from the application of the algorithm to the case studies. Finally, section 7 presents concluding remarks and perspectives for future research.

2. EMS case studies on highways

2.1 The first case study

As described by Mendonça and Morabito (2001), this EMS provides emergency medical treatments on a portion of an interstate highway connecting the cities of Sao Paulo and Rio de Janeiro. It has six ambulances located in six fixed bases along the highway, and one ambulance is located in each base. The operations center, located in Rio de Janeiro, handles the calls, dispatches the ambulances and tracks the ambulances' movements. This is a zero-line capacity system. In accordance with the dispatching policy, when a call arrives in the system, the nearest ambulance is dispatched to the call location. If the nearest ambulance is busy, the second nearest ambulance (called backup) is sent. When the two nearest ambulances are busy, the call is considered lost to the system (even if there are other ambulances available) and transferred to another system such as a local hospital or a police station, which is usually unable to provide the same quality of service. Figure 1 illustrates the location of ambulance bases along the highway. The distance between two adjacent bases is divided into two atoms. An atom corresponds to a segment of highway (district), where the calls arrive with a specific dispatch preference list. According to this list, and except for ambulances 1 and 6, each ambulance is dispatched as preferential to two atoms (immediately to the left and right of its base), and as backup to the set of two atoms adjacent to the preferential ones (right and left of the adjacent ambulances in its left and right, respectively.) For ambulances stationed in bases 1 and 6, the primary response atoms (districts) are atoms 1 and 10, and the secondary (backup) response atoms are 2 and 9, respectively. For example, for ambulance 4, atoms 6 and 7 are the primary response areas whereas atoms 5 and 8 are its backup atoms. The reader can find additional details related to this system in Mendonça and Morabito (2001)´s study.

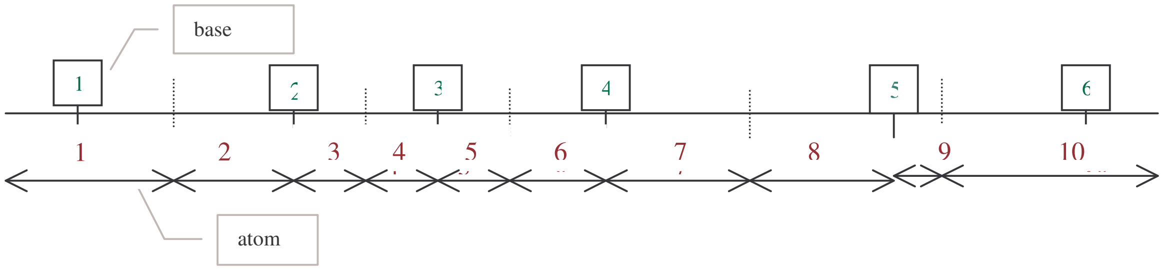

Figure 1 Ambulance bases and atoms along the highway

2.2 The second case study

This EMS has five fixed bases along the two busy highways which intersect, and each base has one ambulance. The ambulances are identical, and the operations center is located in one of the bases. When the operations center receives a call requiring only one ambulance (single dispatch), the nearest available ambulance is immediately dispatched to the call location. If the nearest ambulance is busy, then the next nearest ambulance (called backup) is dispatched. When the call requires double dispatch, the two nearest ambulances are dispatched. If one of them is busy, only the available ambulance is dispatched as a single dispatch. If the two closest servers are busy, the call (either a single or double dispatch) is transferred to another system (for example, the nearest local hospital) and the call is considered lost to the system. Figure 2 illustrates the configuration of this EMS on the stretches of highways in the state of Sao Paulo. More details related to this system can be found in Iannoni et al. (2008).

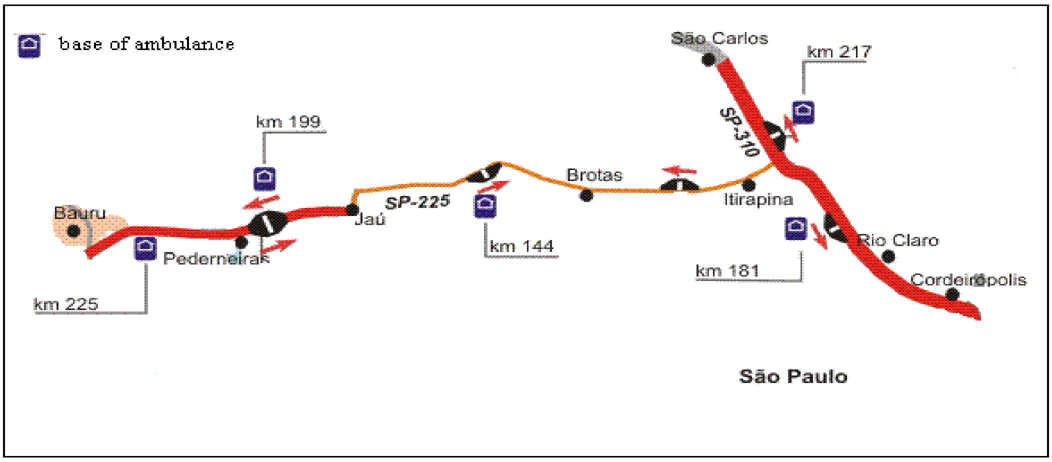

Figure 2. EMS on stretches of highways in Sao Paulo state.

3. Hypercube model to EMS on highway

As the case studies have some different characteristics, we describe how the hypercube model can be adapted to analyze EMS with single dispatch (model 1 – first case study) and single and double dispatch (model 2 – second case study). Model 2 corresponds to an extension of Chelst and Barlach (1981)´s multiple dispatch hypercube model that was developed for police deployment.

3.1 Assumptions of models 1 and 2

The main specific assumptions of the hypercube models 1 and 2 are:

- In model 1, the calls arrive in each atom j with arrival rate λ_j, and all calls are of the same type (single dispatch). In model 2, the calls can be of two types: type 1 calls (with arrival rate $\lambda_j^{[1]}$) require the dispatch of only one ambulance, whereas type 2 calls (with arrival rate $\lambda_j^{[2]}$) require the simultaneous response of two ambulances. The arrival processes have a Poisson distribution.
- For each atom, there is a dispatch preference list ranking the servers to be dispatched depending on the call type. According to this list, in the case of a type 1 call, the first ambulance (the nearest) is dispatched, and if it is busy, the second on the list is sent (backup). If the backup ambulance is also busy, the call is considered lost to the system, even if there are other ambulances available (partial backup). In model 2, in the case of a type 2 call, the two first ambulances of the list are dispatched, and if only one of them is available, it is assigned as a single dispatch (possibly with the help of other EMS). If both

ambulances are busy, then regardless of its type, the call is lost, since a third ambulance is never assigned. This is called partial backup system.

- The models assume that the service times are exponentially distributed. As discussed in the Chelst and Barlach (1981) study, in model 2, type 1 calls are serviced by a single ambulance i with mean service rate μ_i, and type 2 calls are serviced by two ambulances i and k, which operate independently with mean service rates μ_i and μ_k, respectively. Note that the service times of two ambulances servicing a single type 2 call are treated the same as two ambulances servicing two separate type 1 calls.

These particular dispatching policies for EMS on highways also require modifications to the equilibrium equations of the basic hypercube model (the solution of these equations result in the state equilibrium probabilities of the system). In that model, a transition can occur only when a single server changes its status, and a call is lost only when all servers are busy. However, in model 2, a transition can also occur when a type 2 call arrives in the system and consequently two servers become busy simultaneously. Moreover, in models 1 and 2, there can be calls lost by the system even when there are available servers in the system. Additional details about the equilibrium equations and the application of models 1 and 2 to analyze the two case studies can be found in Mendonça and Morabito (2001) and Iannoni et al. (2008).

3.2 Performance measures for models 1 and 2

After determining the state equilibrium probabilities of the system, a number of practical and important performance metrics can be estimated, such as mean user response times, loss probabilities, ambulance workloads, fraction of dispatches of each server to each atom and aggregated travel time measures, and the fraction of calls with response times exceeding a predetermined threshold. In model 2, we can obtain some additional measures considering the two types of calls, for example, loss probability to type 1 and 2 calls, mean region wide travel time (considering the two types of calls), mean travel times to type 1 and type 2 calls, mean travel time of the first and second ambulance arriving at a type 2 call location, mean fraction of dispatches of each server to each region according to the type of call and others.

The mean user response time is simply the set-up time plus the mean system travel time (note that it does not include the mean queue waiting time, since the system does not allow queuing calls). Thus, in model 1 the mean system travel time is defined by:

$\overline{T} = \sum_{i=1}^{N}\sum_{j=1}^{N_A} f_{ij}\, t_{ij}$, where f_{ij} is the fraction of dispatches of ambulance *i* to atom *j* (calculated as a function of the equilibrium state probabilities, see e.g., Mendonça and Morabito, 2001), t_{ij} is the mean travel time of server *i* to atom *j* (estimated from the input travel times between regions), N is the number of servers, and N_A corresponds to the number of atoms of the system. We calculated travel time matrix t_{ij} using the distances from each ambulance base to the atoms' centroids and the average speed of each ambulance.

In model 2, the mean system travel time is calculated as a function of the fraction of dispatches of type 1 ($f'^{[1]}_{ij}$) and type 2 ($f'^{[2]}_{ij}$ and $f'^{[2]}_{(i,k)j}$), considering all dispatches in the system. The mean system travel time is defined by:

$\overline{T} = \sum_{j=1}^{N_A}\left\lfloor \sum_{i=1}^{N} (f'^{[1]}_{ij} + f'^{[2]}_{ij})t_{ij} + \sum_{i=1}^{N-1}\sum_{k=i+1}^{N} f'^{[2]}_{(i,k)j}\min(t_{ij}, t_{kj}) \right\rfloor$, where $f'^{[1]}_{ij}$ corresponds to the fraction of dispatches that ambulance *i* is dispatched to service a type 1 call at atom *j* (single dispatch), $f'^{[2]}_{(i,k)j}$ is the fraction of dispatches that ambulances i and k are simultaneously dispatched to service a type 2 call at atom j, and $f'^{[2]}_{ij}$ is the fraction of dispatches that only ambulance i is dispatched to service type 2 calls at atom j (e.g., when ambulance *k* is busy). These fractions are also calculated as a function of the equilibrium probabilities (see e.g., Iannoni et al. 2008). Furthermore, using models 1 and 2 we can obtain the mean travel time for each ambulance *i* in the system. For example, in model 1 this measure can be defined by:

$$\overline{TU}_i = \frac{\sum_{j=1}^{N_A} f_{ij}\, t_{ij}}{\sum_{j=1}^{N_A} f_{ij}}$$

The workload imbalance can be estimated by the standard deviation of ambulances' workloads defined for the two models by:

$\sigma_\rho = \sqrt{\frac{\sum_{i=1}^{N}(\rho_i - \overline{\rho})^2}{N}}$, where ρ_i is the workload of server *i* (i.e., the sum of the equilibrium probabilities of the states for which server *i* is busy) and $\overline{\rho} = \sum_{i=1}^{N}\rho_i / N$ is the mean server workload.

As discussed previously, an interesting measure is the fraction of calls not serviced within a time limit (i.e., requiring more than 10 minutes of travel time). In model 1, this measure is calculated as follows: $P_{t>10} = \sum_{i=1}^{N}\sum_{j=1}^{N_A} f_{ij}\, p(t_{ij} > 10)$, where the term $f_{ij}\, p(t_{ij} > 10)$ is the fraction of all dispatches of ambulance *i* to atom *j* to which the travel time exceeds 10 minutes, and $p(t_{ij} > 10)$ is the probability that the travel time of server *i* to atom *j* is greater than 10 minutes. In this study, as the travel time data is not available, we estimate $p(t_{ij} > 10)$ by determining the portion of each atom *j* (call location) that ambulance *i* cannot reach under 10 minutes (using geometric probability concepts). In model 2, this measure is defined by: $P_{t>10} = \sum_{i=1}^{N}\sum_{j=1}^{N_A}(f'^{[1]}_{ij} + f'^{[2]}_{ij} + f'^{[2]}_{(i,k)j})p(t_{ij} > 10)$, where the term $(f'^{[1]}_{ij} + f'^{[2]}_{ij} + f'^{[2]}_{(i,k)j})p(t_{ij} > 10)$ is the fraction of all dispatches of ambulance *i* to atom *j* to which the travel time exceeds 10 minutes. More details related to the calculation of all of these measures and others can be found in Chelst and Barlach (1981) and Iannoni et al. (2008).

4. A hybrid location GA/hypercube algorithm

As mentioned previously, Iannoni et al. (2008) propose a standard genetic algorithm embedded with the hypercube model to search for near-optimal atom sizes for the system (districting problem) while considering different performance measures. Here we modify this districting GA/hypercube algorithm to determine the near-optimal ambulance locations along the highway segments (location problem) and also introduce a local search procedure. Therefore we call it the location hybrid GA/hypercube algorithm. Instead of considering only a discrete set of candidate locations for the ambulances along the highway, we assume that their location can be at any point on the highway deemed appropriate.

We begin with a standard genetic algorithm as described in the classic literature of genetic and population algorithms, such as Holland (1975), Goldberg (1989), Michalewicz (1996) and Beasley (2002). When we reallocate the ambulance bases, the algorithm re-divides the highway in atoms (or regions) and recalculates the arrival rates in each atom in order to preserve the demand distribution along the highway. For simplicity, the algorithm divides the distance between two adjacent bases in two equal atoms (except the extremities).

In systems similar to that in the second case study, which involves stretches of two different highways, it is necessary to consider some modifications in the algorithm. For example, the modified algorithm assumes that there are two linear stretches (or a set of linear stretches): a stretch with n_1 ambulances and another with n_2 ambulances. It also considers that the ambulances from a stretch can service calls in another stretch, as it is observed in the real system. As the second case study's dispatching policy involves multiple dispatching, and there are two different types of calls in the system (type 1 and type 2 calls), the algorithm re-calculates the arrival rates in each atom, considering these two types of calls. Moreover, in calculating the distances between ambulance bases and the centroid of each atom and the dispatching preference lists, the algorithm takes into account that the ambulances from one stretch can travel to the other. The main components considered in the implementation of the location GA/hypercube algorithm are briefly presented in the following sections.

4.1 Chromosome representation and generation

The algorithm employs a decimal representation for the chromosomes. Each chromosome is represented as a vector $y = (y_1, y_2, \ldots, y_N)$, where y_i corresponds to the fraction of the stretch being analyzed, and $y_1 < y_2, \ldots < y_N$. It utilizes a procedure to randomly generate the initial population, assuming that $0 \leq y_i \leq 1$. To generate possible system configurations (chromosomes) for initial and subsequent populations, this approach simply adds to each stretch, increments $k\Delta$, where Δ is fixed and k is an integer randomly sorted in the range $[0, M = 1/\Delta]$. Therefore, there are $M+1$ possible values for each y_i. The same discrete vector was applied in the mutation procedure in order to randomly replace the gene. For example, consider a system similar to the first case study as shown in Figure 1. With $N = 6$ ambulances, the total distance $D = 100$Km, and the chromosome $y = (y_1, y_2, y_3, y_4, y_5, y_6) = (0.10; 0.25; 0.40; 0.60; 0.75; 0.90)$, then this configuration can be illustrated by Figure 3.

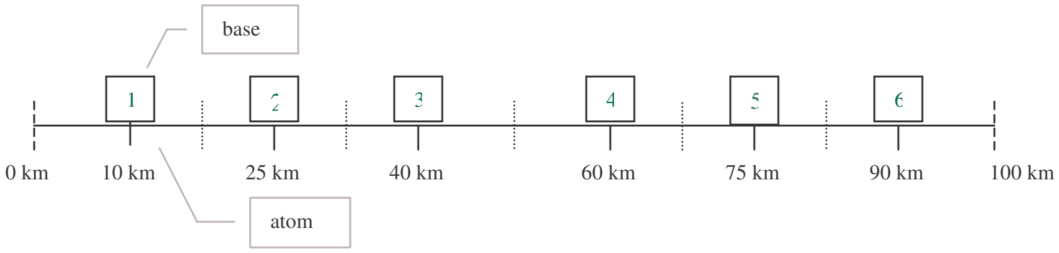

Figure 3. Ambulance locations along the highway for a given chromosome

We also introduced a restriction for the generation of new chromosomes, which is related to the minimum distance between two adjacent bases. Thus, the algorithm takes into account the additional restriction of the minimum distance $d_{\min}$ between two bases, depending on the operational condition of the system analyzed. If we consider the total length of the stretch of highway being analyzed as D, then: $y_i - y_{i-1} \leq d_{\min}/D$. For example, if

$d_{\min}$ = 20 km, in each generated configuration, the minimum distance between two adjacent bases $i-1$ and i must be at least 20 km. In the second case study, we need to consider the parameter $d_{\min}$ for each stretch, and the chromosome *y* is divided in pieces (each piece representing a stretch). For example, $y = (y_{1,}.....y_{n_1}, y_{n_1+1}....y_{n_1+n_2})$ where n_1 is the number of ambulances in stretch 1 and n_2 is the number of ambulances in stretch 2.

4.2 Evaluation and fitness function, selection of chromosomes, crossover and mutation

The evaluation procedure of the GA/hypercube utilizes the hypercube models outlined in section 3 to compute the performance measures for each configuration (represented by a chromosome). Similar to the study in Iannoni et al. (2008), we conducted different experiments to optimize three separate measures (fitness function $f(y)$ of chromosome *y*). The first objective is to minimize mean region wide travel time, that is, $\min f(y) = \overline{T}(y)$. In other experiments, the objective is to minimize the imbalance of ambulance workloads (evaluated by the standard deviation), that is, $\min f(y) = \sigma_\rho(y)$, or the fraction of calls not serviced within 10 minutes, that is, $\min f(y) = P_{t>10}(y)$. The expressions for calculating these measures are described in section 3.2.

Similar to the algorithm proposed in Iannoni et al. (2008), the selection of parent chromosomes follows the roulette wheel method with probabilities based on the fitness function value (Goldberg, 1989; Michalewicz, 1996). After selecting two parents, with probability p_c we applied the well-known single-point crossover (and with probability $1 - p_c$, the selected parents are preserved in the next generation). The mutation procedure is applied to each gene in the chromosome with a predefined probability p_m. In order to randomly replace the gene, we use the same discrete initialization procedure described in section 4.1. For example, for each gene y_i (with probability p_m) this value is mutated to $y_i = k\Delta$, where k is uniformly sorted in the interval $[0,...,M]$. Given the constraint of minimum distance between two adjacent bases, there can be invalid chromosomes generated by crossover and mutation. For that reason, we replace a parent chromosome with a child chromosome *y* only if this child *y* is feasible, i.e., it assures $y_i - y_{i-1} \leq {}^{d_{\min}}\!/_{D}$ in the same linear stretch.

Initially, we set the critical parameters of the standard GA/hypercube algorithm (without local search procedure) such as crossover and mutation probabilities (p_c and p_m), number of generations (*G*) and population size (Pop). Then, we ran extensive tests varying the values of these parameters within certain ranges. The set of values: $p_c = 0.7$, $p_m = 0.1$, $G = 1000$ generations and $Pop = 200$ chromosomes yielded the best results in most of the tests. The intervals $\Delta = 0.03$ and $\Delta = 0.01$ (i.e., $M = 33$ and $M = 100$, respectively) were tested in each of the three fitness functions. An additional parameter is the minimum distance between two adjacent bases $d_{\min}$. Therefore, we conducted different experiments for each of the fitness functions using $d_{\min} = 20$ km and $d_{\min} = 30$ km, respectively. For example, in the original configuration of the first case study, the smallest distance between two bases is 21 km.

4.3 Local search and GA/hypercube algorithm

Several studies, such as those by Hertz and Kobler (2000), Jaszkiewicz (2002), Beasley (2002) and Arroyo and Armentano (2005) have pointed out the superior performance of hybrid genetic algorithms which includes a local search procedure. In the present study, this alternative was also adopted by evaluating a local neighborhood for each solution generated by the GA/hypercube algorithm in the first population and after crossover and mutation operators.

For evaluating a local neighborhood to a given solution generated by the GA algorithm, we consider the following procedure. For all ambulances $i = 1, 2, \ldots, N$ of the system, we analyze two possible movements: to the left and the right of their current locations y_i, i.e., modifying each location to $y_i + \Delta y$ and $y_i - \Delta y$, where Δy is an input parameter (we tested different values for Δy; e.g., $\Delta y = 0.01$ and 0.03). In this marginal analysis of each movement of each ambulance i we preserve the original location of the other ambulances in the system and we apply the hypercube model (model 1 or 2) to evaluate the changes in the fitness function value. If there are improvements, we update the incumbent solution with the best improvement and we repeat the above procedure; otherwise the procedure is finished.

In order to avoid cycles during the iterations of the procedure, we keep in a simple tabu list of the two best movements achieved in the last two iterations. Before moving the next candidate, we not only verify that is does not appear on the tabu list but also check if the constraint of minimum distance ($y_i - y_{i-1} \leq {}^{d_{\min}}\!/_{D}$), and evaluate the movement (by the hypercube model) only if these two conditions are found.

We set new parameters for the hybrid GA/hypercube algorithm such as number of generations (G) and population size (Pop). As discussed later in section 6, the hybrid GA/hypercube requires a much smaller population size and number of generations than the location standard GA/hypercube algorithm (without local search) to find the same solutions: $G = 10$ generations and $Pop = 10$ or 20 chromosomes. The others parameters are the same as discussed above. Figure 4 presents the general scheme of the location hybrid GA/hypercube algorithm.

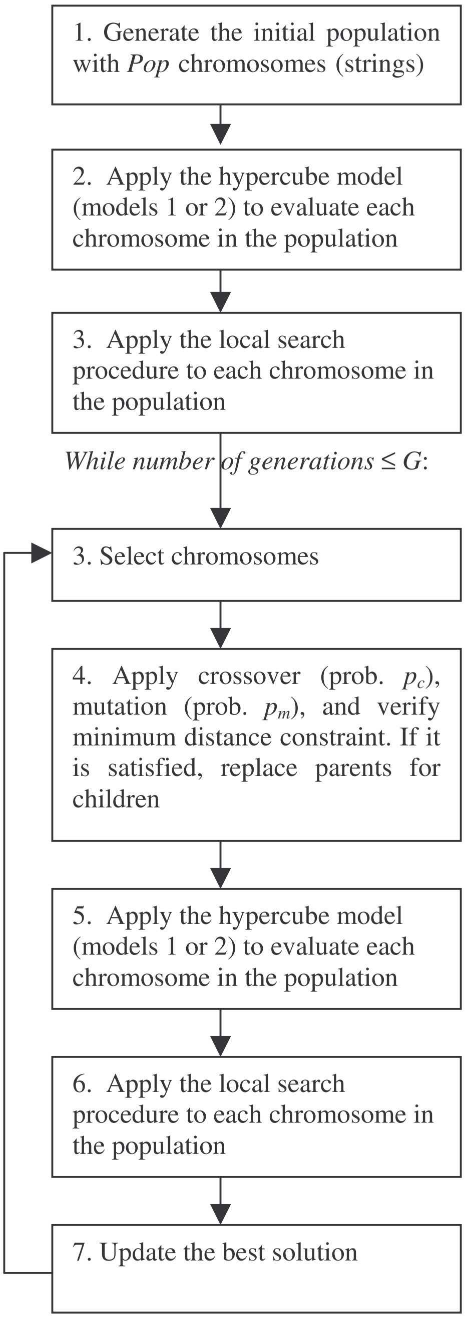
1. Generate the initial population with *Pop* chromosomes (strings)
2. Apply the hypercube model (models 1 or 2) to evaluate each chromosome in the population
3. Apply the local search procedure to each chromosome in the population
While number of generations ≤ G:
3. Select chromosomes
4. Apply crossover (prob. p_c), mutation (prob. p_m), and verify minimum distance constraint. If it is satisfied, replace parents for children
5. Apply the hypercube model (models 1 or 2) to evaluate each chromosome in the population
6. Apply the local search procedure to each chromosome in the population
7. Update the best solution

Figure 4 General structure of the hybrid GA/hypercube algorithm

4.4 Trade-off curve generation

Some performance measures may conflict due to the different interests of the parties involved in the EMS. For example, the mean user response time is an external performance measure of the system, which is important to the users of the system. On the other hand, the balancing of the server workloads is an internal performance measure, which is of particular interest to the system managers. In some cases, when we reduce mean response time, we may worsen the workload imbalance, and vice-versa. We use the concept of domination criterion (called the Pareto domination) to analyze the compromise among these conflicting objectives. One simple method for generating Pareto-optimal solutions is the ε-constraint method (Cohon, 1978). It operates by optimizing one objective, while all others are constrained to a limiting value ε. For example, taking the objectives: $\min f_1(y) = \overline{T}(y)$ and $\min f_2(y) = \sigma_\rho(y)$ as defined in section 3.2, we have the objective function: min Z = ($f_1(y)$, $f_2(y)$) = ($\overline{T}(y)$, $\sigma_\rho(y)$). Using the ε-constraint method to optimize the mean region wide travel time, whereas the standard deviation of server workloads specifies the constraint, the problem is formulated as follows:

Min $Z = f_1(y) = \overline{T}(y)$

s.a $f_2(y) = \sigma_\rho(y) \le \varepsilon$

$y \in Y^*$ where y corresponds to the solution vector, Z corresponds to the image of y (or objective space), Y^* corresponds to the set of problem's feasible solutions, and ε corresponds to the superior limiting value to σ_ρ. Note that in the problem above we could also choose to optimize σ_ρ (instead of $\overline{T}$) and limit $\overline{T}$ (instead of σ_ρ).

To solve this problem, the location hybrid GA/hypercube algorithm was modified as follows: During a simple run, we range the values of ε for each set of G = 10 generations. Then, the algorithm saves (keeping in memory) the non-dominated solutions found until the last G generations to the next G generations and updates the optimal value of objective function $\overline{T}$ for a correspondent value of σ_ρ. Furthermore, additional modifications are required during the process of generating the initial population, crossover and mutation, in order to allow the generation of only feasible solutions to the problem. For example, after the crossover and mutation procedure, we replace a parent chromosome with a child chromosome y if the child is feasible (satisfying $\sigma_\rho(y) \le \varepsilon$). Note that to verify if a chromosome y is feasible we apply the hypercube to evaluate $\overline{T}$ and σ_ρ. After wide-ranging the different values of ε, the algorithm describes the approximate trade-off curve between $\overline{T}$ and σ_ρ (e.g., non-dominated solutions found throughout the run considering all possible values of ε).

5. Location and districting GA/hypercube algorithm

We then modify the location GA/hypercube algorithm described in section 4 to consider two combined decisions: (i) locating the ambulances and (ii) determining their coverage areas (atom sizes for the system). The location and districting GA/hypercube algorithm basically has two steps. In step 1, the algorithm optimizes the location of ambulance bases y^* applying the location GA/algorithm described in section 4. In step 2, starting from the solution of step 1, the algorithm optimizes the atoms' sizes for this solution (configuration) applying the districting GA/hypercube algorithm proposed in Iannoni et al. (2008), modified in the present study to include the local search procedure.

In the districting hybrid GA/hypercube we consider the initial location of the ambulances determined by the location hybrid GA/hypercube (note that, in the final solution of the location algorithm, the distance between two adjacent bases is divided in two equal atoms). The sizes of the atoms are modified to produce different feasible configurations (chromosomes) for the system. Each chromosome is represented as a vector $x = (x_1, x_2, ..., x_{N-1})$, where each x_i is the fraction of the distance between bases *i* and *i*+1. If the distance between these two adjacent bases is d_i, then the size of the preferential atom for server *i* is given by $x_i d_i$. The remaining distance between the bases *i* and $i+1$ becomes the first preferred atom for base $i+1$ (i.e., $d_i - x_i d_i$). As suggested by the system managers, we impose that 0.2 ≤ x_i ≤ 0.8, limiting the preferential area of each server *i* to between 20 and 80 percent of the distance d_i.

For generating possible system configurations (chromosomes) in step 2 (districting GA/hypercube), we conducted computational experiments which favored a finite discrete approach over a continuous approach where each chromosome is populated by a sorted set of continuous random numbers between 0.2 and 0.8. The finite approach simply adds an increment $k\Delta$ to 0.2 (the lower limit of the interval), where Δ is fixed and *k* is an integer randomly sorted in the range $[0, M = (0.8 - 0.2)/\Delta]$. Therefore, there are $M+1$ possible values for each x_i. As with the location GA/ hypercube, while searching for the optimal atom sizes, we preserve the arrival rate distribution along the entire highway by proportionally re-distributing the initial arrival rates. The other components of the districting hybrid GA/hypercube algorithm such as the fitness function evaluation, selection of chromosomes, crossover and mutation operators and the local search procedure are similar to the same components for the location hybrid GA/hypercube described in section 4.

Note that the two genetic algorithms (location and districting) are applied sequentially, one for each step of this approach. Alternatively, we could also apply the districting genetic algorithm (step 2) to each solution (generated chromosome) of the location genetic algorithm (step 1). However, this approach would result in high computational costs, as it can be verified by the results in section 6.

In setting the parameters for the two hybrid genetic algorithms, such as crossover (pc_1 and pc_2), mutation probabilities (pm_1 and pm_2), the number of generations (G_1 and G_2), population size (Pop_1 and Pop_2), and the intervals Δ_1 and Δ_2, we ran extensive tests varying the values of these parameters within certain ranges. The best results were obtained using: $pc_1 = 0.7$ and $pc_2 = 0.5$; $pm_1 = 0.1$ and $pm_2 = 0.06$; $G_1 = 10$ and $G_2 = 10$. The combination of the two intervals $\Delta_1 = 0.01$ and $\Delta_1 = 0.03$, and $\Delta_2 = 0.01$ and $\Delta_2 = 0.03$ were tested for each case study.

Additionally, we also investigated a small modification in the algorithm described above. In this new version, during step 1, the location GA/hypercube algorithm saves (keep in memory) the *nc* best solutions, instead of saving only the best solution (in terms of location). Moreover, in step 2, the districting GA/hypercube algorithm is applied for each *nc* solution from step 1, resulting in the best solution (in terms of location and districting). Note in the previous version the algorithm can be considered a particular case of the present version where $nc = 1$.

6. Computational results:

The location hybrid GA/hypercube algorithm (section 4) and the location and districting hybrid GA/hypercube algorithm (section 5) proposed in this paper were coded in Pascal and run on a 1.66 GHz Centrino Duo T2300 microcomputer.

6.1 Results of the location hybrid GA/hypercube algorithm

Results of the first case study: According to data reported in Mendonça and Morabito (2001), the original configuration of this system based on the bases location is represented by the chromosome $y = (y_1, y_2, y_3, y_4, y_5, y_6) = (0.0, 0.2192, 0.3315, 0.4973, 0.7807, 1.0)$. The stretch of the highway in Figure 1 is 187 km in length. In this configuration, there are atoms that are not equal to the half of the distance between two adjacent bases. The main performance measures result in: $\overline{T}$ = 7.912 min, σ_ρ = 0.0551 and $P_{t>10}$ = 0.299.

To verify the quality of the solutions produced by the location hybrid GA/hypercube algorithm, we developed a simple procedure that incorporates the hypercube models of section 3 into a simple enumerative algorithm. This exhaustive procedure determines the optimal configuration in terms of the ambulance bases location (under a precision Δ), testing all possible configurations to the system (considering also the minimum distance $d_{\min}$ constraint). Note this enumerative algorithm is computationally treatable only for problems of moderate size, as for the EMS under consideration with only N = 6 ambulances and a precision of Δ = 0.03 or Δ = 0.01.

Initially, we conducted experiments individually optimizing each of the fitness functions discussed in section 4.2: $\min f(y) = \overline{T}(y)$, $\min f(y) = \sigma_\rho(y)$ or $\min f(y) = P_{t>10}(y)$. The corresponding optimal values of $\overline{T}(y)$, $\sigma_\rho(y)$ and $P_{t>10}(y)$ obtained by the hybrid GA/hypercube and enumerative algorithms in each experiment are presented in bold in Table 1 (where GA is the genetic algorithm and EA is the enumerative algorithm). In addition to the optimal values (in bold) for each objective, Table 1 also shows the values of the other two measures obtained by the algorithms in each experiment, as well as the relative deviation (percentage improved) to the original configuration. The additional parameters to obtain the results in Table 1 are: Δ = 0.01, Δy = 0.01 and $d_{\min}$ = 20km.

Table 1. Results of the location hybrid GA/hypercube algorithm for three fitness functions

Measu-re	Origin. Config.		Objec-tive $\min \overline{T}(y)$	% Impro-ved	Objective $\min \sigma_\rho(y)$	% Impro-ved	Objective $\min P_{t>10}(y)$	% Impro-ved
$\overline{T}(y)$	7.912	**GA** **EA**	**6.2311** **6.2311**	21.25%	7.7026 7.3445	2.65%	6.4024 6.4024	19.08%
$\sigma_\rho(y)$	0.0551	**GA** **EA**	0.0507 0.0507	7.98%	**0.0218** **0.0216**	60.44%	0.0480 0.0480	12.88%
$P_{t>10}(y)$	0.299	**GA** **EA**	0.166 0.166	44.48%	0.268 0.271	10.36%	**0.147** **0.147**	50.84%

Note in Table 1 the three experiments, $\overline{T}$, $P_{t>10}$ and σ_ρ are all improved when compared to the original configuration of the system. These measures are better than the results obtained by applying the districting standard GA/hypercube algorithm proposed in Iannoni et al. (2008), which optimizes only the coverage areas of the ambulances given their original locations (i.e., varying only the size of atoms). For example, the results for the objective function obtained by the districting GA/hypercube minimizing $\overline{T}$, σ_ρ and $P_{t>10}$ are: 7.778 min, 0.0245 and 0.255, respectively, whereas for the location GA/hypercube these results are: 6.2311 min, 0.0218 and 0.147, respectively (Table 1).

The location GA/hypercube finds the optimal solution obtained by the enumerative algorithm in the first and third experiments (Table 1). Note in the first experiment $\overline{T}$ (to be minimized) reduces 21.25%, and in the third experiment, $P_{t>10}$ (to be minimized) reduces 50.84%. In the second experiment, σ_ρ (to be minimized), the GA/hypercube algorithm finds a solution very close to the optimal solution of the enumerative algorithm (0.0218 and 0.0216, respectively). Regarding the computational time, a single run of the location standard GA/hypercube for these experiments took an average of 500 seconds (8.3 minutes) (using G = 1000 and Pop = 200), whereas the location hybrid GA/hypercube algorithm took an average of 11 seconds to find these results (using G = 10 and Pop = 10), and the exhaustive procedure took more than 11 hours in order to test all possible configurations (for all of these algorithms, we use Δ = 0.01 and $d_{\min}$ = 20km). The configuration (with 12 atoms) that minimizes the mean region wide travel time ($\min f(y) = \overline{T}(y)$) is represented by chromosome $y = (y_1, y_2, y_3, y_4, y_5, y_6)$ = (0.07, 0.23, 0.37, 0.56, 0.74, 0.88).

Simulation of the configuration that minimizes the mean region travel time: To validate the performance measures found in this optimal configuration, we developed a discrete event simulation model of the first case study using the software *Arena*. The procedures to calculate the transient period (warm up) and the simulation run length are described in detail in Iannoni and Morabito (2006). Table 2 presents the results for the mean travel time $\overline{TU}_i$ and the workload ρ_i of each ambulance of the system. These results are also validated via analysis of the confidence interval (confidence level $\alpha = 0.05$). Note that all average values obtained by the hypercube are within the simulation confidence intervals, validating the model outputs. In particular, the mean region-wide travel time obtained by simulation is $\overline{T}$ = 6.3058 min (Conf. interv. 6.1867 – 6.4248), whereas the result calculated by the hypercube model in Table 1 is $\overline{T}$ = 6.2311 min.

Table 2.Travel time (minutes) and workload of each ambulance

Ambulance i	Model	$\overline{TU}_i$	ρ_i
1	hypercube simulation (Conf. interv.)	4.680 4.687 4.517 – 4.857	0.148 0.152 0.145 – 0.166
2	hypercube simulation (Conf. interv.)	6.177 6.402 6.051 - 6.753	0.206 0.214 0.198 – 0.229
3	hypercube simulation (Conf. interv.)	7.064 7.145 6.789 – 7.501	0.164 0.162 0.148 – 0.176
4	hypercube simulation (Conf. interv.)	5.902 5.806 5.631 – 5.981	0.295 0.299 0.280 – 0.318
5	hypercube simulation (Conf. interv.)	7.413 7.653 7.245 – 8.060	0.147 0.144 0.134 – 0.154
6	hypercube simulation (Conf. interv.)	5.796 5.812 5.644 – 5.980	0.191 0.199 0.183 – 0.215

Trade-off curve: Figure 5 shows the graph of the trade-off frontier obtained by plotting the non-dominated (or efficient) solutions. These points are obtained by applying the location hybrid GA/hypercube algorithm which we modified to solve the bi-objective problem (objective $\min f_1(y) = \overline{T}(y)$ and constraint $\sigma_\rho(y)$), for different values of ε varying from 0.026 to 0.055 (using Δ = 0.01). According to the procedure described in section 4.4, we vary the values of restriction σ_ρ to update the values of $\overline{T}$ in each set of generations, obtaining the final values to non-dominated solutions showed in Figure 5. Note that the gap of optimality is relatively small: for example, for the interval σ_ρ = 0.050 – 0.051, the mean region wide travel time $\overline{T}$ is 6.231 (the best solution found by the enumerative algorithm).

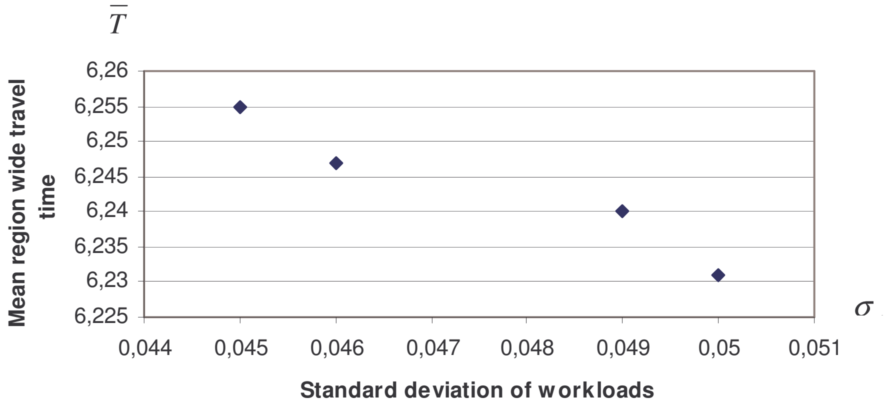

Figure 5: Approximate trade-off frontier between $\overline{T}$ and σ_ρ obtained by the location hybrid GA/hypercube algorithm

Results of the second case study: According to Iannoni et al. (2008) the original configuration of the system is represented by the chromosome y = (y_1, y_2, y_3, y_4, y_5) = (0.135, 0.622, 0.368, 0.750, 0.924), where n_1 = 2 and n_2 = 3 (total of 5 ambulances). The stretch 1 is 74 km in length and stretch 2 is 144 km in length. The location of the ambulance (km) along the two connected stretches according to y is: in stretch 1 (10 and 46km, bases 1 and 2, respectively), and in stretch 2 (53, 108 and 134km, bases 3, 4 and 5, respectively). In planning the configuration and operation of this EMS, the managers and operators consider two possible configurations for the system in terms of the ambulances coverage areas (i.e., the atoms size); see Figures 6 and 7. The first configuration was analyzed in Iannoni et al. (2008) by applying the districting standard GA/hypercube algorithm in order to determine the coverage areas of each ambulance. Note in Figure 6 that, in this configuration, there are 8 atoms in the system (2 atoms in stretch 1 and 6 atoms in stretch 2.) The second configuration comprises 10 atoms in the system (the distance between two adjacent bases are divided into two equal atoms, except for the extremities). In this case, there is only one atom between the connection point and ambulance 3, while stretch 1 has 4 atoms and stretch 2 has 6 atoms (see Figure 7). In the present study, we apply the algorithms described only for the second configuration (Figure 7) which is both simpler and similar to the first case study. Applying the multiple dispatch hypercube model to analyze the second original configuration, we find the following performance measures: $\overline{T}$ = 7.1576 min, σ_ρ = 0.0151 and $P_{t>10}$ = 0.228.

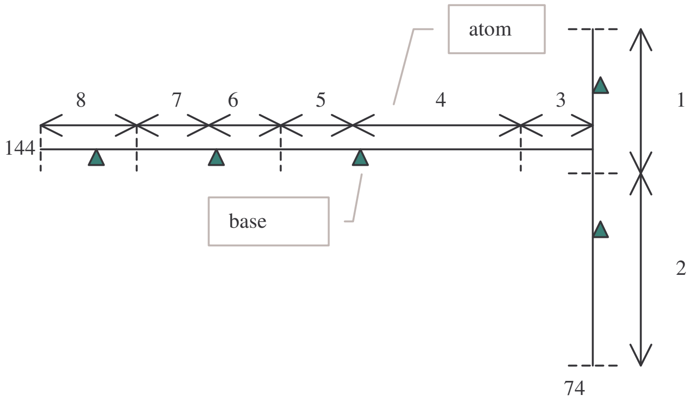

Figure 6 – First original configuration of the second case study

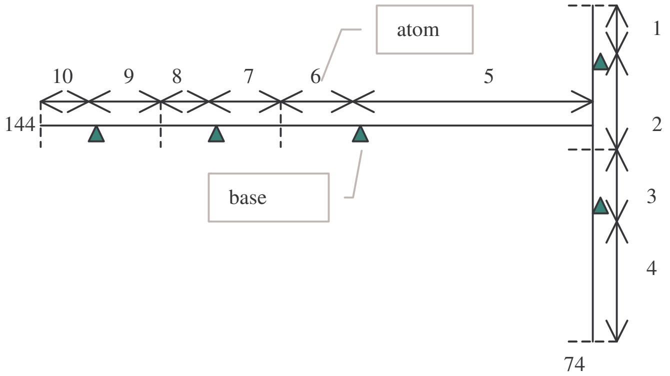

Figure 7 – Second original configuration of the second case study

To verify the quality of the solutions produced by the location GA/hypercube algorithm, we also applied the enumerative algorithm. However, using the exhaustive procedure, we were unable to test all possible configurations in fewer than 24 hours using the same previous parameters Δ = 0.01 and $d_{\min}$ = 20km. It should be mentioned that, for the second case study, the number of possible combinations is much greater than the first case study, since the total region is larger and there is a smaller number of ambulances. Only the results of $\overline{T}(y)$, $\sigma_{\rho}(y)$ and $P_{t>10}(y)$ obtained by the hybrid GA/hypercube algorithm in each experiment are presented in Table 3. The location standard GA/hypercube (using G = 1000 and Pop = 200) took an average of 148 seconds of computational time, whereas the location hybrid GA/hypercube (using G = 10 and Pop = 20) took an average of 19 seconds to find the results in Table 3. Applying the enumerative algorithm to a lesser precision Δ = 0.03, we obtain for each experiment $\overline{T}(y)$ = 4.5182 min, $\sigma_{\rho}(y)$ = 0.01019 and $P_{t>10}(y)$ = 0.1492, respectively. Note that these results are close to the values (in bold) in Table 3, confirming the quality of the solutions produced by the GA/hypercube algorithm. The additional parameters to obtain the results in Table 3 are: $\Delta = 0.01$, $\Delta y = 0.01$ and $d_{\min} = 20$km.

Table 3. Results of the location hybrid GA/hypercube algorithm for three fitness functions

Measure	Origin. Config.		Objective $\min \overline{T}(x)$	% Improved	Objective $\min \sigma_\rho(x)$	% Improved	Objective $\min P_{t>10}(x)$	% Improved
$\overline{T}(y)$	7.1576	**GA**	**4.5117**	36..97%	10.2697	-43.48%	6.4837	9.41%
$\sigma_\rho(y)$	0.0151	**GA**	0.0315	-108.6%	**0.0094**	37.75%	0.0176	-16.56%
$P_{t>10}(y)$	0.228	**GA**	0.205	10.09%	0.369	-61.84%	**0.143**	37.28%

Note in Table 3 the three experiments, $\overline{T}$, σ_ρ e $P_{t>10}$ are all improvements over the original configuration of the system. In the first experiment, $\overline{T}$ (to be minimized) improves by 36.97% while workload imbalance (σ_ρ) degrades by 108.6%. Similarly, in the second experiment, σ_ρ (to be minimized) reduces by 37.75% while $\overline{T}$ and $P_{t>10}$ increases by 43.48% and 61.84%, respectively. In the third experiment $P_{t>10}$ (to be minimized) improves by 37.28%, whereas σ_ρ increases by 16.56% and $\overline{T}$ reduces by 9.41%. Note that the results of these analyses (applying each objective separately) show that these measures may be in conflict, since when we improve one of the measures, we may worsen other. For example, the best solution in terms of $\overline{T}$ results in the worst solution in terms of σ_ρ.

As in the first case study, the results for the objective performance measures in Table 3 are better than the results obtained by applying the districting standard GA/hypercube algorithm proposed in Iannoni et al. (2008) to this second configuration (Figure 7). For example, the results for the objective function obtained by the districting GA/hypercube minimizing $\overline{T}$, σ_ρ and $P_{t>10}$ are: 7.1353 min, 0.0144 and 0.221, respectively, whereas for the location GA/hypercube these results are: 4.5117 min, 0.0094 and 0.143, respectively (Table 3). Figure 8 illustrates the ambulance bases' locations in the configuration that minimizes the mean region wide travel time ($\min f(y) = \overline{T}(y)$) in Table 3, represented by chromosome y = (y_1, y_2, y_3, y_4, y_5) = (0.59, 0.87, 0.0, 0.55, 0.87). Note that in this configuration y_3 = 0.0 (the ambulance 3 is located in the border of stretch 2). However the distance between ambulance 1 and 3 is not divided into two equal atoms, and the division takes place between bases 1 (km 43.7) and 2 (km 64.4).

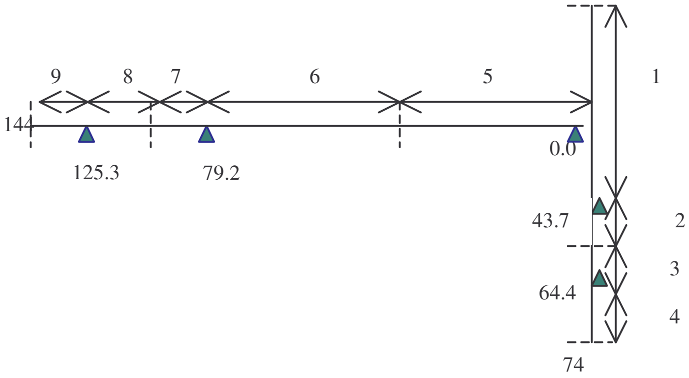

Figure 8. Configuration of the second case study that minimizes the mean region wide travel time.

Simulation of the configuration that minimizes the mean region wide travel time: Similar to the previous case study, we validate the optimal solution in Figure 8 using a discrete event simulation model (implemented via the *Arena* software). As discussed in Iannoni et al. (2008), the assumption of exponential distribution for the service times was rejected for all servers in the system. Therefore, we performed statistical analysis using the *Best-Fit* software in order to obtain the best-adjusted statistical distributions representing the data (using goodness-of-fit tests) to the simulation model. The following distributions for the service time (in minutes), not including the travel time from the server's base to the call location, were found for each ambulance in the system: ambulance 1 - Lognormal (41.10, 57.24); ambulance 2 - Erlang (16.51, 3); ambulance 3 – Lognormal (69.46, 66.59); ambulance 4 - Erlang (9.59, 5) and ambulance 5 – Lognormal (49.51, 39.13). Table 4 compares the results of type 1 call travel times and the workload for each ambulance obtained by the hypercube and the simulation models, including the 95% confidence intervals of the simulations results. Note that the results obtained by the hypercube model are within the corresponding intervals. In particular, the mean region wide travel time calculated by simulation is $\overline{T}$ = 4.5604 min (Conf. interv. 4.4663 – 4.6545), and the result obtained by the hypercube model is $\overline{T}$ = 4.5117 min

Table 4. Mean travel time of type 1 calls (minutes) and ambulance workloads.

Ambulance i	Model	$\overline{TU}_i^{[1]}$ (min)	ρ_i
1	hypercube simulation (Conf. interv.)	8.127 8.090 7.767 – 8.413	0.0243 0.0246 0.0221 – 0.0267
2	hypercube simulation (Conf. interv.)	3.068 3.123 3.065 - 3.171	0.0223 0.0217 0.0203 – 0.0231
3	hypercube simulation (Conf. interv.)	1.942 1.974 1.868 – 2.079	0.1024 0.1015 0.0978 – 0.1052
4	hypercube simulation (Conf. interv.)	10.380 10.357 10.022 – 10.692	0.0188 0.0215 0.0204 – 0.0226
5	hypercube simulation (Conf. interv.)	6.320 6.325 6.275 – 6.375	0.0321 0.0326 0.0311 – 0.0341

Trade-off curve: Similar to the multi-objective analysis for the first case study, Figure 9 depicts the approximate trade-off curve between $\overline{T}$ and σ_ρ obtained by the location hybrid GA/hypercube algorithm. In this case, we use different values of ε varying from 0.032 to 0.011, based on the results obtained in the previous experiments in Table 3 (using Δ = 0.01).

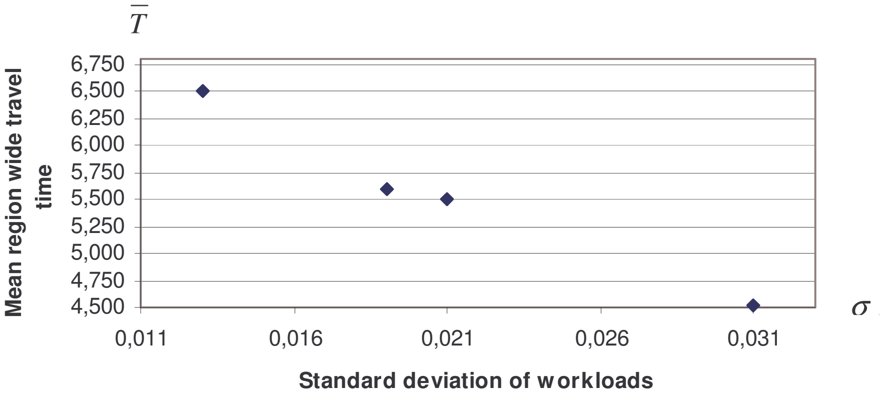

Figure 9: Approximate trade-off frontier between $\overline{T}$ and σ_ρ obtained by the location hybrid GA/hypercube algorithm.

6.2 Results of the location and districting hybrid GA/hypercube algorithm

Results of the first case study: The original configuration of the system is represented by chromosomes $y = (y_1, y_2, y_3, y_4, y_5, y_6) = (0.0, 0.2192, 0.3315, 0.4973, 0.7807, 1.0)$ for the location of ambulance bases and $x = (x_1, x_2, x_3, x_4, x_5) = (0.50, 0.50, 0.50, 0.50, 0.22)$ for the districting (as illustrated in Figure 1). As discussed previously, the following results are found in the original configuration: $\overline{T}$ = 7.912 min and σ_ρ = 0.0551. Tables 5 and 6 present the results of the first and second solutions obtained by the location and districting hybrid GA/hypercube model. We performed 10 runs, minimizing $\overline{T}$ and σ_ρ, respectively. The parameters utilized are: G_1 = 10; G_2 = 10; Pop_1 = 10; Pop_2 =10; Δ_1 = 0.01, Δ_2 = 0.01, Δy = 0.01 and $d_{\min}$ = 20km. The solution in Table 5 is represented by the chromosomes: $y = (y_1, y_2, y_3, y_4, y_5, y_6)$ = (0.07, 0.23, 0.37, 0.56, 0.74, 0.88) for the location of ambulance bases and $x = (x_1, x_2, x_3, x_4, x_5)$ = (0.45, 0.45, 0.59, 0.39, 0.50) for the districting. This solution is also better than the results obtained by applying the location and districting standard GA/hypercube algorithm (using G_1 = 1000; G_2 = 1000; Pop_1 = 200; Pop_2 =100). For example, the final solution obtained by that algorithm minimizing $\overline{T}$ is 6.1616 min.

Table 5. Results of the location and districting GA/hypercube algorithm minimizing $\overline{T}$

Solution of step 1: location	6.2311 min
Solution of step 2: districting	6.1548 min
% Improve to step 1	1.22%

Table 6. Results of the location and districting GA/hypercube algorithm minimizing σ_ρ

Solution of step 1: location	0.0218
Solution of step 2: districting	0.0152
% Improve to step 1	30.27%

An additional experiment was performed to analyze the behavior of the variation of the location and districting GA/hypercube algorithm, which takes into account the best *nc* configurations from step 1 instead of only the best one (see section 5). We arbitrarily used *nc* = 10 configurations. Table 7 presents the 10 best solutions obtained in step 1 (location GA/hypercube algorithm), and the respective solutions obtained in step 2 (districting GA/hypercube algorithm), where the objective is to minimize $\overline{T}$. Note that, the ranking of the best solutions found in step 1 does not correspond to the ranking of the best solutions of step 2. For example, the best solution of step 2 (solution 2 – 6.1375 min) is not a result of the best solution of step 1 (solution 1 – 6.2311 min), which shows that the variation of the location and districting GA/hypercube algorithm can be more effective, despite requiring higher computing time. Regarding the computing time, the hybrid GA/hypercube algorithm with *nc* = 1 took an average of 30 seconds while with *nc* = 10 took an average of 250 seconds. In comparison the standard GA/hypercube algorithm with *nc* = 1 took an average of 490 seconds

(8.17 minutes) and the algorithm with nc = 10 took an average of 2,250 seconds (37.5 minutes) to find similar solutions.

Table 7. Results of the variation of the location and districting hybrid GA/hypercube algorithm (with $nc = 10$ configurations in step 1), minimizing $\overline{T}$.

	Solution of step 1 (min)	Solution of step 2 (min)
1	6.2311	6.1548
2	6.2328	6.1375
3	6.2354	6.1492
4	6.2395	6.1634
5	6.2468	6.1802
6	6.2512	6.1706
7	6.2535	6.1813
8	6.2635	6.1898
9	6.2748	6.1580
10	6.2851	6.2071

Results of the second case study: The original configuration of the system is represented by chromosomes $y = (y_1, y_2, y_3, y_4, y_5) = (0.135, 0.622, 0.368, 0.750, 0.930)$ and $x = (x_1, x_2, x_3) = (0.5, 0.5, 0.5)$, where $n_1 = 2$ and $n_2 = 3$ (see Figure 7). Tables 8 and 9 present the results of the first and second step of the location and districting hybrid GA/hypercube algorithm. We performed 10 runs, minimizing $\overline{T}$ and σ_ρ, respectively. The parameters utilized are: $G_1 = 10$; $G_2 = 10$; $Pop_1 = 20$; $Pop_2 =20$; $\Delta_1 = 0.01$, $\Delta_2 = 0.01$, $\Delta y = 0.01$ and $d_{\min} =$ 20km. The chromosomes that represent the solution in Table 8 are: $y = (y_1, y_2, y_3, y_4, y_5) = (0.59, 0.87, 0.0, 0.55, 0.87)$ for the location of ambulance bases and $x = (x_1, x_2, x_3) = (0.51, 0.43, 0.57)$ for the atoms size between two adjacent bases.

We also conducted experiments with the variation of the location and districting GA/hypercube algorithm with nc = 10 solutions, minimizing $\overline{T}$. The results obtained in step 1 and 2 of the algorithm are presented in Table 10. Similar to the first case study, the best solution from step 2 (solution 2 – 4.4533) is not generated by the best solution from step 1 (solution 1 – 4.5117). The hybrid GA/hypercube algorithm with nc = 1 took an average of 46 seconds of computational time and the algorithm with nc = 10 took an average of 122 seconds. The standard GA/hypercube algorithm with nc = 1 took an average of 158 seconds (2.6 minutes) of computational time and the algorithm with nc = 10 took an average of 680 seconds (11.3 minutes).

Table 8. Results of the location and districting GA/hypercube algorithm minimizing $\overline{T}$.

Solution of step 1: location	4.5117 min
Solution of step 2: districting	4.4561 min
% Improve to step 1	1.23%

Table 9. Results of the location and districting GA/hypercube algorithm minimizing σ_{ρ}.

Solution of step 1: location	0.0094
Solution of step 2: districting	0.0065
% Improve to step 1	30.85%

Table 10. Results of the variation of the location and districting GA/hypercube algorithm (with *nc* = 10 configurations in step 1), minimizing $\overline{T}$.

	Solution of step 1 (min)	Solution of step 2 (min)
1	4.5117	4.4561
2	4.5131	4.4533
3	4.5196	4.4666
4	4.5198	4.4634
5	4.5209	4.4604
6	4.5210	4.4666
7	4.5228	4.4673
8	4.5231	4.4677
9	4.5245	4.4649
10	4.5252	4.4705

6.3 Results of other problem instances

Initially, we considered test problems with *N* = 3, 4, 5 and 6 ambulances (bases) and $N_A = 2N - 2$ atoms. We generated 10 random instances for each problem size, based on the data set of the first EMS case study. For example, the arrival rate λ_j of each atom *j* was randomly generated sorting a value in the interval ($\lambda_{\min}$, $\lambda_{\max}$), where $\lambda_{\min}$ = 0.00008 and $\lambda_{\max}$ = 0.00375 are the minimum and maximum arrival rates in the case study. Similarly, the service rate μ_i of each server *i* was randomly sorted in the interval ($\mu_{\min}$, $\mu_{\max}$), where $\mu_{\min}$ = 0.0101 and $\mu_{\max}$ = 0.0241 are the minimum and maximum service rates in the case study. We applied both the enumerative and the location GA/hypercube algorithms to solve these data sets, optimizing the objective (fitness) function: $\min f(x) = \overline{T}(x)$ discussed in section 4.2.

Table 11 presents the results for the mean best solutions, runtimes and optimality percentages of each problem set (using Δ = 0.01 and $d_{\min}$ = 20km). For the location hybrid GA/hypercube, we used the parameters *G* = 10, *Pop* = 10. Note all problems of sizes *N* = 3 and 4, and 9 out of 10 problems of size N = 5 ambulances are optimally solved by the GA/hypercube algorithm, indicating that it is effective for solving problems of moderate size. For problems of size N = 6 or larger, the enumerative algorithm becomes too expensive computationally (see last row of the table). Since we are not aware of tight lower bounds for the optimal solution values of these examples, we are unable to provide the gaps of optimality.

Table 11. Results for the mean best solutions, runtimes and optimality percentages of 10 random problem instances (using Δ = 0.01).

Data set	Enumerative algorithm		GA/hypercube algorithm		Optimality percentage
	Objective $\min \overline{T}(x)$	Runtime (hours)	Objective $\min \overline{T}(x)$	Runtime (hours)	
$N = 3$	4.8053	6.55×10^{-4}	4.8053	7.22×10^{-5}	10 / 10
$N = 4$	5.3891	0.015	5.3891	2.08×10^{-4}	10 / 10
$N = 5$	5.8612	0.338	5.8679	9.89×10^{-4}	9 / 10
$N = 6$	-	> 11 hours	6.2602	0.0040	-

We also examined the performance of the location and districting hybrid GA/hypercube algorithm for other larger problem instances than the first case study. We considered test problems with N = 6, 8 and 10 ambulances (bases) and $N_A = 2N - 2$ atoms, randomly generated from the first case study in the same way as discussed previously. In these experiments we also optimize the objective (fitness): $\min f(x) = \overline{T}(x)$. The parameters utilized by the hybrid GA/hypercube are: G_1 = 10; G_2 = 10; Pop_1 = 10; Pop_2 =10; Δ_1 = 0.01, Δ_2 = 0.01, $\Delta y = 0.01$ and $d_{\min}$ = 20km. We summarize our findings in Table 12.

Table 12. Results of the GA/hypercube algorithm minimizing $\overline{T}$

Problem instance	Measures	Original Config.	Step 1: location	Percent Improved	Runtime (hours)	Step 2: districting	Percent Improved to step 1	Runtime (hours)
$N = 6$	$\overline{T}(x)$	7.7549	**6.2063**	19.97%	0.0038	**6.1768**	0.47%	0.0034
	$\sigma_\rho(x)$	0.0860	0.0457	46.86%		0.0438	4.16%	
	$P_{t>10}(x)$	0.2791	0.1699	39.12%		0.1704	-0.29%	

$N = 8$	$\overline{T}(x)$	7.5307	**6.5897**	12.49%	0.0828	**6.5583**	0.48%	0.156
	$\sigma_\rho(x)$	0.0519	0.0347	33.14%		0.0397	-14.41%	
	$P_{t>10}(x)$	0.2592	0.1831	29.36%		0.1856	-1.36%	
$N = 10$	$\overline{T}(x)$	7.6331	**6.3428**	16.90%	1.518	**6.3218**	0.33%	2.120
	$\sigma_\rho(x)$	0.0751	0.0303	59.65%		0.0319	-5.28%	
	$P_{t>10}(x)$	0.2700	0.1837	31.96%		0.1863	-1.41%	

The results obtained by the hybrid GA/hypercube in step 1 and step 2 of Table 12 are better than the results obtained by applying the standard GA/hypercube in both steps (using G_1 = 1000; G_2 = 1000; Pop_1 = 100; Pop_2 =100). For example, the best results found by the standard GA/hypercube algorithm for the objective function in step 1 in the three experiments are: 6.2241, 6.6647 and 6.8499, respectively, whereas the results found by the hybrid GA/hypercube in step 1 are: 6.2063, 6.5897 and 6.3428, respectively (Table 12). Similarly, in step 2 the best solutions found by the standard GA/hypercube algorithm in step 2 (districting) in the three experiments are: 6.1882, 6.5721 and 6.8072, respectively, whereas the results found by the hybrid GA/hypercube in this step are: 6.1768, 6.5583 and 6.3218, respectively.

Nevertheless, it is important to emphasize that the standard GA/hypercube algorithm requires significantly higher computational time than the hybrid GA/hypercube algorithm. For example, a single run of the standard GA algorithm including steps 1 and 2 in the three experiments took on average: 0.154, 2.84 and 42.3 hours, respectively.

Note that, even when using an iterative method to solve the linear systems of the hypercube model, the computer storage and, perhaps more importantly, runtime requirements of the location and districting GA/hypercube algorithm increase significantly for $N \geq 10$ servers. For example, a single run of the location and districting hybrid GA algorithm for the problem instance with N = 10 servers took more than 3 hours. Nevertheless, there are few EMS on Brazilian highways with $N \geq 10$ servers. Conversely, if this approach is used to support decisions at a strategic level, it seems reasonable that the decision maker(s) will spend more time studying it. The current version of the location and districting GA/hypercube is less promising if the EMS operators wish to dynamically reconfigure the system (atom sizes) in response to significant variations in the demand patterns by day of the week or even by hour of the day.

7. Conclusions

In this study we combined extensions of the hypercube model with hybrid genetic algorithms to optimize the configuration and operation of EMS on highways. In our first approach (location GA/hypercube algorithm), we study the location of the ambulance bases along the highway, in order to minimize the mean user response time, the imbalance of the ambulances workloads and the fraction of calls not serviced within a predetermined threshold. Since these performance measures may be in conflict, we presented how to adapt the algorithm to carry out a trade-off analysis and generate an approximate Pareto efficient frontier between these measures.

This approach was also extended to arrive at decisions regarding of the location of ambulance bases and the districting of coverage areas of ambulances along the highway (location and districting GA/hypercube algorithm). Computational results were analyzed applying the methods to two cases studies: The first case study (with single dispatch and partial backup) is an EMS operating on a highway linking the cities of Sao Paulo and Rio de Janeiro, which was initially studied by Mendonça and Morabito (2001). The second case study (with multiple dispatch and partial backup) is an EMS on stretches of highway in the state of Sao Paulo, which was recently studied by Iannoni et al. (2008) and Iannoni and Morabito (2007).

Our study showed that the main performance measures (objectives), such as the mean user response time, imbalance of ambulance workloads, and the fraction of calls not serviced within a time limit could be improved by relocating the ambulance bases and simultaneously determining the district (atom) sizes of the system. The methods require reasonable computational time to solve problems of moderate size (e.g., with less than 10 ambulances). For larger problem instances, the approaches (in their current versions) would take a prohibitive amount of CPU time, since they require the solution of exact hypercube models (models 1 and 2) in order to evaluate each chromosome. Therefore, an interesting perspective for future research is the use of hypercube approximation algorithms methods (not based on systems of 2^N linear equations), such as the methods proposed in Larson (1975), Jarvis (1985), Goldberg and Szidarovszky (1991) and recently, in Atkinson et al. (2006, 2007).

Acknowledgments: The authors thank the anonymous referees for the useful comments and suggestions. This research was partially supported by CNPq and Fapesp.

References

Arroyo J.C.; Armentano V.A. (2005) Genetic local search for multi-objective flowshop scheduling. *European Journal of Operational research* 167, p.717-738.

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), p.379-391.

Atkinson J.B., Kovalenko I.N., Kuznetsov N., Mykhalevych K.V. (2007) A hypercube queueing loss model with customer-dependent service rates. *European Journal of Operational Research* doi:10.1016/j.ejor.2007.08.014.

Ball M.O., Lin F.L. (1993) The reliability model applied to emergency service vehicle location. *Operations Research* 41, p.18-36.

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

Beasley J.E. (2002) Population heuristics. In: Pardalos, P.M. Resende, M.G.C. (eds). *Handbook of Applied Optimization*. University Press. Oxford, p. 138-157.

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 Science* 22, Elsevier, p.121-153.

Brotcorne L., Laporte G., Semet F. (2003) Ambulance location and relocation models. *European Journal of Operational Research* 147, p. 451-63.

Burwell, T.H.; Jarvis, J.P.; Mcknew, M.A.(1993) Modeling co-located servers and dispatch ties in the hypercube model. *Computers & Operations Research* 20(2), p.113-119.

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

Chiyoshi F., Galvão R. D., Morabito R. (2003) A note on solutions to the maximal expected covering location problem. *Computers & Operations Research* 30(1), p. 87-96.

Cohon J. L. (1978). *Multiobjective programming and planning.* Academic Press, New York.

Galvão R.D., Chiyoshi F., Morabito R. (2005) Towards unified formulations and extensions of two classical probabilistic location models. *Computers & Operations Research* 32, p. 15-33.

Goldberg D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison Wesley.

Goldberg D. E., Szidarovszky F. (1991) Methods for solving nonlinear equations used in evaluating emergency vehicle busy probabilities. *Operations Research* 39, 6, p. 903-916.

Hertz A., Kobler D. (2000) A framework for the description of evolutionary algorithms. *European Journal of Operational Research* 126, p.1-12.

Holland J.H. (1975) *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.

Iannoni A.P., Morabito R. (2006) A discrete simulation analysis of a logistics supply system. *Transportation Research* E 42 (3), p. 191-210.

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), p. 755- 771.

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

Jaszkiewicz A. (2002) Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research* 137, p. 50-71.

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

Larson R.C. A (1974) Hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research* 1, p. 67-95.

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

Larson R.C., Odoni A.R. (1981) *Urban operations research*. Prentice Hall. New Jersey.

Larson R.C. (2004) OR models for homeland security. *OR/MS Today* 31, 22-29.

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, p. 261- 268.

Michalewicz Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer-Verlag. Berlin.

Owen, S.H., Daskin, M.S.(1998) Strategic facility location: A review. *European Journal of Operational Research* 111, p. 423 – 447.

Rajagopalan H.K., Saydam C., Xiao J. (2007) A multiperiod set covering location model for a dynamic redeployment of ambulances. *Computers & Operations Research. In press.*

Sacks S. R., Grief S. (1994) Orlando Police Department uses OR/MS methodology, new software to design patrol districts. *OR/MS Today*, Baltimore, p.30-32.

Saydam C., Aytug H. (2003) Accurate estimation of expected coverage: revisited. *Socio-Economic Planning Sciences* 37, p. 69-80.

Swersey A.J. (1994) *Handbooks in OR/MS*. Amsterdam: Elsevier Science B.V., v. 6, p. 151-200.

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