Optimal Spatial Partitioning for Resource Allocation

Kostas Kolomvatsos
Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens
e-mail: kostasks@di.uoa.gr

Kakia Panagidi
Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens
e-mail: mop10297@di.uoa.gr

Stathes Hadjiefthymiades
Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens
e-mail: shadj@di.uoa.gr

ABSTRACT

Spatial partitioning consists of the problem of finding the best segmentation of an area under specific conditions. The final goal is to identify parts of the area where a number of resources could be allocated. Such cases are common in disaster management scenarios. In this paper, we consider such a scenario and propose a methodology for the resource allocation for emergency response. We utilize an intelligent technique that is based on the Particle Swarm Optimization algorithm. We define the problem by giving specific formulations and describe the proposed algorithm. Moreover, we provide a method for separating the area into cells and describe a technique for calculating cell weights based on the underlying spatial data. Finally, we present a case study for allocating a number of ambulances and give numerical results concerning the run time and the total coverage of the examined area.

Keywords

Spatial Partitioning, Resource Allocation, Emergency Management, PSO, Segmentation.

INTRODUCTION

The segmentation of a geographical region into pieces is a common problem in distributed resource allocation. Such segmentation is necessary in cases where developers need to split the area into parts in order to find the optimal location for specific resources. Some typical problems on this scenario include the assignment of public health services, waste management, snow plowing operations, setting up postal boxes or bus stops, etc. Resource allocation is usually done according to the patterns of population in a community or the spatial characteristics of the examined area. Usually, a limited amount of resources are available and should be assigned to meet the demands which are spatially distributed in a bounded region, such as a town, a city or a state.

One can meet spatial partitioning for the optimal resource allocation problem in emergency management systems. Methodologies for handling such a problem are based on crucial spatial data (e.g. roads information, population, etc) as well as on resources information (e.g. maximum speed, capacity, etc). Resources could be either vehicles (e.g. ambulances, etc) or other important items or equipment for disaster response (e.g. materials, food, water, etc). The optimal distribution of such limited number of resources in a specific area is known as the *partitioning problem for optimal resource allocation*. Many times, the discussed problem is referred as the *location allocation problem*. The location allocation problem is different from the classical resource allocation problem (Moller-Jensen & Kofie, 2001). Actually, the location allocation problem is consisted by two parts (Shanbhag et al., 2005): a) where to locate the new resources, and b) which area each resource will cover. Additionally, in the discussed problem the number of the resources also plays a crucial role. The final objective is to maximize the area that the limited number of resources will cover under a number of constraints. For example, a firefighting department could want to place a number of vehicles in specific points in a city in order to have the minimum response time when an emergency situation is present.

In this paper, we provide a solution to the discussed problem based on the known *Particle Swarm Optimization* (PSO) (Kennedy & Eberhart, 1995) technique. We consider that every particle is a vector containing the resources with the corresponding locations. Thus, particles should be moved to places in order to maximize a reward function. In our case, the reward is the maximum area coverage. It should be noted that we split the area in to a number of cells and for each cell a specific weight is devoted according to the underlying spatial characteristics.

The rest of the paper is organized as follows. In Section '*Related Work*', we present important research efforts in the field while in Section '*Spatial Partitioning Problem*', we describe our approach. We formulate the problem and analyze the required data for such cases. In Section '*Proposed Approach*', we fully describe our model and give specific descriptions for each part. We present our approach for defining weights for every cell in a grid and describe our algorithm that is based on the widely known *PSO* technique. In Section '*Case Study: Optimal Allocation for Ambulances*', we describe the allocation of a number of ambulances having specific characteristics. Finally, we present some plots depicting the required time for the execution of our model while in Section '*Conclusions*', we conclude our paper.

RELATED WORK

A number of methodologies and techniques have been proposed for handling the discussed problem. In literature, one can find clustering techniques (Joshi et al., 2009; Tu et al., 2005; Liu et al., 2011), graph partitioning techniques (Tu et al., 2005; Wu & Leahy, 1993), Voronoi polygons methods (Yang & Gold, 1996; Zhao et al.,1999; Riol et al., 2011; Rezayan et al., 2008; Riviere & Schmitt, 2007; Wu et al., 2007) or techniques based on multi criteria analysis (Shanbhag et al., 2005). All of them result a number of locations in the space where the resources will be placed. A visualization part provides a more comprehensible interface as it provides a map and marked the resulted locations. Many of these techniques have been used to consist of the basis of creating hierarchical structures for the retrieval of spatial information (Frank & Timpf, 1994; van Oosterom & Schenkelaars, 1995). In general, we can identify two groups of theories used for handling the discussed problem. Based on the first group, researchers pay attention into the resources characteristics that should optimally be placed in the *Area of Interest (AoI)*. Such theories are usually applied to mobile resources like mobile sensors. Based on the second group of theories, researchers are trying to divide the AoI into sub-areas (clusters). Clustering utilizes specific rules in order to have each sub-area satisfying a minimum set of resources (e.g. k-means algorithm).

Zou and Chakrabarty (2003) presented the *Virtual Force Algorithm* (VFA) algorithm that is inspired by disk packing theory and the virtual force field concept from robotics. For a given number of resources, VFA attempts to maximize the covered field using a combination of attractive and repulsive forces. The covered field is represented by a two-dimensional grid. The coverage of the resources is modeled as a circle on the grid. The center of the circle denotes the resource location while the radius denotes the coverage range. In (Aziz et al., 2010), an evolutionary algorithm is the solution for the coverage problem. The proposed algorithm, WSNPSO_{con} uses PSO or WSNPSO (Wireless Sensor networks PSO) to find the best locations of the sensors according to a penalty. PSO optimizes the problem by having a population of candidate solutions, here particles, and moving them around in the search-space according to simple mathematical formulation over the particle's position and velocity. Each particle movement is influenced by its local best position and is also guided toward the best known position in the search-space, which are updated as better positions found by other particles. The fitness function uses a Voronoi diagram to measure the quality of the coverage.

The clustering algorithms are based on two approaches: hierarchical and partitional (Frigui & Krishnapuram, 1999; Leung et al., 2000). In hierarchical clustering, the output is a tree showing a sequence of clustering with each cluster being a partition of the data set (Leung et al., 2000). Hierarchical algorithms can be agglomerative (bottom-up) or divisive (top-down). Hierarchical algorithms have two basic advantages (Frigui & Krishnapuram, 1999; Leung et al., 2000). Firstly, the number of classes need not be specified a priori and secondly, they are independent of the initial conditions. However, the main drawback of hierarchical clustering techniques is that they are static. In addition to that, they may fail to separate overlapping clusters due to lack of information about the global shape of the clusters (Jain et al., 1999). Partitional clustering algorithms attempt to decompose the data set directly into a set of disjoint clusters. They try to optimize certain criteria. The criterion function may emphasize to the local structure of data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters. The advantages of the hierarchical algorithms are the disadvantages of the partitional algorithms and vice versa. Clustering can also be performed in two different modes: crisp and fuzzy. In case of fuzzy clustering, a pattern may belong to all the classes with a certain fuzzy membership grade (Jain et al., 1999). The most widely used iterative K-means algorithm (MacQueen, 1967) for partitional clustering aims at minimizing the ICS (Intra-Cluster Spread).

In general, the disadvantage of clustering algorithms is the complexity of the calculations required for retrieving the final result. In such approaches, the AoI is separated in a number of cells and accordingly clustering algorithms result the final groups. Therefore, a data preprocessing step is necessary in order to result the final value for each cell and, thus, to be handled by the algorithms. Additionally, in the majority of the above described approaches, the main goal is the provision of a method that minimizes the frequent repartitioning. For example, in (Shanbhag et al., 2005), the authors propose a visualization technique for depicting temporal changes in the underlying data (e.g. population density) giving the users the opportunity to choose the appropriate place for building schools or other important infrastructures. However, in our approach, we focus on a more dynamic scenario where the underlying data can be changed due to various reasons. Specific roads could not be available or temporal changes on data could affect the final result. Disaster managers simply re-run the algorithm and take the new resource locations. More importantly, we focus on the combination of resources information with the underlying spatial data. The location of each resource is affected by the characteristics of the resources (may be different according to the type – e.g. ambulances, firefighting vehicles, etc) as well as the spatial data like number of roads, population density and so on.

SPATIAL PARTITIONING MODEL

The proposed model was developed due to the need of resources optimal placement before a physical disaster happens. Combining the features of resources, such as speed, and geographical characteristics of an AoI, an optimal allocation is produced according to the user requirements. Every area is presented as a grid. An algorithm for spatial partitioning in AoI places optimal the resources with the best coverage of cells without holes or overlapping cells.

Problem Formulation

A number of N_j, j=1,2,3,…,R, (R is the number of resources), are available to be allocated in a specific area A. The discussed resources are of type T_j. We consider that the area A has an orthogonal scheme with width W_0 and height H_0. The result of the proposed model is the optimal placement for each resource in order to meet pre-defined conditions C_{jk} (k=1,2,…,K). The process is repeated for each resource type T_j. The number N_j differs for each T_j. Moreover, resources of different types could be placed in different positions in A, covering different regions. The number of conditions for each T_j differs as well. Figure 1 depicts an example allocation for 6 resources. Actually, the proposed model should split the area into N_j sub-areas (*spatial partitioning*) and each resource of the same type will be placed in each sub-area. The resource should be placed in a road or in an accessible place and not in a position not accessible (e.g. mountain, forest). We do not know in advance the size of each shape and probably do not know the exact shape of each sub-area (rectangle, circle, etc). For simplicity, we can assume that each area can be represented through a rectangle or polygon of known dimensions (grid-modelling). The dimension of each area is variable and depends on a number of parameters (see the list below). Moreover, overlaps should be eliminated (*optimal spatial partitioning*). Thus, in the ideal case, the following holds true:

$$\sum_{l=1}^{N_j} A_l = W_0 \cdot H_0 \tag{1}$$

where A_l is the area covered by every resource. The interesting is that the size of each shape should be based on parameters which describe: a) resources properties, and b) the covered area. Hence, parameters that should be taken into consideration for the optimal resource allocation are the following:

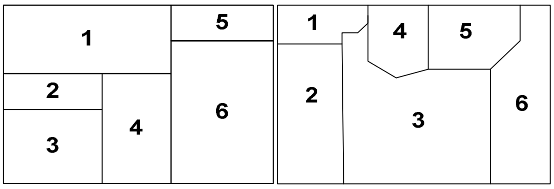
Area Related Parameters: Population attributes, density of population (percentage), type (hilly, flat, etc), traffic, roads – road segments (type: urban, highway, etc - speed limit, width, etc), places of interest (POIs) (schools, hospitals, industrial areas possibly flammable, dangerous buildings, etc).

Resource Related Parameters: Type (vehicle, vehicle type, rescue team, etc), maximum allowed speed for emergency situations, maximum travel distance, capacity (for persons, if applicable, or rescue material, etc).

It should be noted that for some of the abovementioned parameters apart from the spatial aspect of the information there is also the time aspect. For example, the density of population or information related to POIs (e.g. schools) could change over time during the day. Pupils are present at schools mainly every day morning while there is no presence in evenings.

Data Organization

The data required by the proposed model are spatial data that are stored in a spatial database. Spatial data can be found either in the Web (e.g. OpenStreetMap) or they can be defined by developers. Developers are able to define such data through GIS tools. Moreover, in some countries, public authorities have available data related to the cities or other AoIs. Spatial data are very important as they define an objective view of the area A. For example, if we want to place a number of ambulances in A then the best solution is to cover first the most dangerous areas. Such areas could be places where a lot of people gather together (e.g. schools, factories and so on) or places where the danger just after a disaster is high (e.g. fuel stations, etc).



(a) (b)

Figure 1. Examples for resources allocation (Rectangles and Polygons)

We consider that every resource has specific characteristics that affect the area it will cover. For example, the area covered by a vehicle is affected by the maximum speed and travel distance. The same ambulance will cover a wider range in rural area than in an urban area. This depends on the population density. If this *AoI* is close to national roads, where probably there is not any traffic jams, an ambulance can cover a larger area than in the city center. In our case, concerning resources, we take into consideration the following information:

- Resource Type (e.g. vehicle, building, team, etc)
- Crew or Personnel (it depends on the resource type)
- Maximum Speed
- Maximum travel distance
- Current location
- It should be noted that, especially for moving resources (i.e. vehicles and rescue teams), we consider that there is a maximum allowed response time. This means that the location of such resources should be optimal in order to respond in the minimum time. Following the same rationale, we define the information related to the examined area. We focus on critical data as the proposed system aims to have the best response in emergency situations. Thus, the information taken into consideration is the following:
- Amenity type (e.g. schools, fuel stations, fire stations, hospitals, forests, etc).
- Area type (e.g. city, village, hamlet, etc).
- Roads and road segments characteristics (e.g. length, lanes number, oneway, maximum allowed speed, etc).
- Road type (e.g. highway, primary, secondary, tertiary or track).

PROPOSED APPROACH

Our proposed model is based on an intelligent technique like the *PSO* algorithm. The aim is to find the optimal solution for the above discussed problem. The steps of our methodology are the following:

- At first, we split the area in a number of cells creating a grid.
- We define a specific weight for every cell according to the underlying spatial information.
- We apply the PSO algorithm for finding the best location of each resource in order to have the maximum coverage. This is done under specific conditions that are defined by restrictions on the resources.

Below, we describe every step and give specific formulations.

Grid Partitioning

Our area *A* is defined by the coordinates [(x_{UL}, y_{UL}), (x_{LR}, y_{LR})] where (x_{UL}, y_{UL}) is the upper left corner and (x_{LR}, y_{LR}) is the lower right corner. The area *A* is divided into N_c X N_c cells. We consider that the number N_c is defined by the developer and it is positive. In our approach, we split the axes of the 2D space in N_c parts. At the beginning, we do not care about the spatial characteristics of the area. An example of this process is depicted by Figure 2. For each cell in the grid, we store information related to the corners that define the specific cell. The size of each cell is equal to:

$$A_c = \frac{x_{LR} - x_{UL}}{N_c} \cdot \frac{y_{UL} - y_{LR}}{N_c} = \frac{(x_{LR} - x_{UL}) \cdot (y_{UL} - y_{LR})}{N_c^2} \quad (2)$$

Actually, each cell is the conceptual part of the area between the two corners. In the area covered by cells there is a number of spatial characteristics (e.g. roads, schools, hospitals, etc) playing an important role in the definition of the weight (see the upcoming section). Finally, for each cell, we store the center by using the real coordinates.

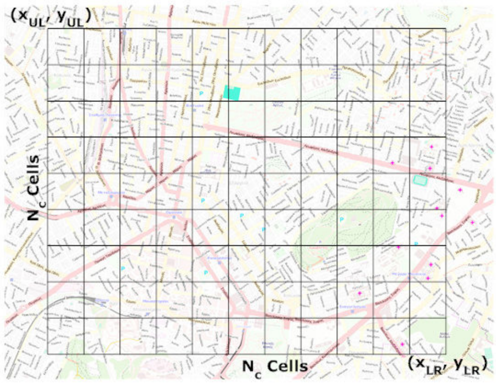


Figure 2. Grid definition example.

Weights Definition

With all these attributes, we would try to ascribe degrees of priority to *N* sub-areas. The number of sub-areas is equal to the number of resources. The higher is the priority, the smaller is the radius of resources. For example, in case of an earthquake or, in general, of a natural disaster, as many schools and hospitals are in an AoI, the greater is the need for evacuation. Our method to compute the priority of the sub-areas is based on the *Analytic Hierarchy Process (AHP)*. The AHP method has been developed by Saaty (1990) and is one of the best known and most widely used in multi-criteria approaches. It allows users to assess the relative weight of multiple criteria or multiple options against given criteria in an intuitive manner. In case quantitative ratings are not available, policy makers or assessors can still recognize whether one criterion is more important than another. The basic process to carry out the AHP consists of the following steps:

1. *Structuring a decision problem and selection of criteria:* The first step is to decompose a decision problem into its constituent parts. In its simplest form, this structure comprises a goal or focus at the topmost level, criteria (and sub-criteria) at the intermediate levels, while the lowest level contains the options.
2. *Priority setting of the criteria by pair wise comparison (weighting):* For each pair of criteria, the decision maker is required to respond to a question such as "How important is criterion A relative to criterion B?" Rating the relative "priority" of the criteria is done by assigning a weight between 1 (equal importance) and 9 (extreme importance) to the more important criterion.
3. *Pair wise comparison of options on each criterion (scoring):* For each pairing within each criterion the better option is awarded a score, again, on a scale between 1 (equally good) and 9 (absolutely better), whilst the other option in the pairing is assigned a rating equal to the reciprocal of this value.
4. *Obtaining an overall relative score for each option:* In a final step the option scores are combined with the criterion weights to produce an overall score for each option.

In our algorithm, we suppose that the developer will decide the number and the type of the resources that will be placed in the AoI. The developer is going to decide the ranking of the criteria. For example, the developer may decide that the hospitals are two times as important as fuel or fire stations and fuel stations are four times as important as banks. Using pair wise comparisons, the relative importance of one criterion over another can be expressed, as Table 1 indicates. A future work is the provision of a tool where disaster managers could define their own pair wise ratings.

	Fuel	**School**	**Hospital**
Fuel	1/1	2/5	1/2
School		1/1	3/4
Hospital			1/1

⇒

Fuel	**School**	**Hospital**
1/1	2/5	1/2
5/2	1/1	3/4
2/1	4/3	1/1

Table 1. Pair wise comparisons

Having categorized the attributes which will possibly exist in an area, the weight of each cell (WC_j) is calculated as follows:

$$WC_j = w_i \cdot \frac{A_{ij}}{\sum A_{ij}}, \begin{cases} 0 \le i \le NA \\ 0 \le j \le N_c \end{cases} \tag{3}$$

where w_i is the weight of i^{th} attribute which is derived by AHP, A_{ij} is the value of the specific attribute i in the cell j (e.g. schools, hospitals, etc) and NA is the attributes number. If no special attributes are present in the cell then the weight is defined equal to 1.

Particle Swarm Optimization

The PSO algorithm is a population-based search algorithm based on the simulation of the behavior of birds within a flock. In such systems, each individual has a very simple behavior: to follow the success of its own and its neighbors. Thus, the collective behavior that is the result of such simple behavior finally leads to discover the optimal regions of a search space. The position of each individual is adjusted according to its own experience and the experience of its neighbors. Let x_k^t denote the position of the k^{th} individual at time t. The position is changed by adding a velocity (v_k^{t+1}) as the following equation indicates:

$$x_k^{t+1} = x_k^t + v_k^{t+1} \tag{4}$$

The velocity is a very important parameter as it leads to the optimal solution as the algorithm evolves. In PSO, the velocity is calculated as follows:

$$v_{kj}^{t+1} = v_{kj}^t + c_1 \cdot r_{1j}^t \cdot (y_{kj}^t - x_{kj}^t) + c_2 \cdot r_{2j}^t \cdot (\hat{y}_{kj}^t - x_{kj}^t) \tag{5}$$

where, v_{kj}^t is the velocity of individual k in dimension j at time t, x_{kj}^t is the position of the individual k in dimension j at time t, c_1 and c_2 are positive constants used for acceleration and r_{1j}^t, r_{2j}^t are random values in the region [0, 1]. Moreover, in the above defined equation, the algorithm uses the personal and the global best positions (y_{kj}^t and $\hat{y}_{kj}^t$ respectively). The personal best position at every step t+1 is calculated as follows:

$$y_k^{t+1} = \begin{cases} y_k^t & \text{if } f(x_k^{t+1}) \ge f(y_k^t) \\ x_k^{t+1} & \text{if } f(x_k^{t+1}) < f(y_k^t) \end{cases} \tag{6}$$

where f(·) is the fitness function. Therefore, the global best position is equal to:

$$\hat{y}_k^t = y_i^t \mid i \in \{1, .., N\} \text{ and } f(\hat{y}_k^t) = \min\{f(y_1^t), f(y_2^t), ..., f(y_N^t)\} \tag{7}$$

where N is the number of individuals. PSO can provide an efficient solution to the discussed problem. Our aim is to maximize the covered area based on the spatial and resources information. Actually, we want to estimate the exact coordinates where resources will be located. Each possible solution is a vector:

$$p = [(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)] \tag{8}$$

with all potential coordinates of the resources. We can adopt PSO in order to generate *M* particles, that is, *M* vectors *p* of all the coordinates of the resources. The fitness function *F(p)* for the PSO algorithm refers to the percentage of covered areas A_i out of the entire area *A*. The PSO algorithm calculates at every iteration the current best solution *p* which maximizes *F(p)* from all *M* possible solutions. The best solution *p** which maximizes *F(p*)* refers to the optimal placement of the resources in the terrain such that the coverage of the area is maximized provided that each resource has maximized its covered sub-area A_i.

In more detail, at first, we define the number of particles *M*. In the beginning, we randomly generate the location of each resource in the particle. This location is the center of a specific cell in the grid. We consider that each resource has limited capabilities that are specifically defined. For example, an ambulance has specific maximum speed, capacity and so on. For each resource, we take the related information as well as the underlying spatial information of the cell. Accordingly, we calculate the area covered by the resource (Eq.(9)). This is done based on a simple mathematical model depended on the resource type (i.e. maximum covered distance based on the time restriction and the maximum speed). If we consider that the examined resources are vehicles then the covered area mainly depends on the available speed and the population characteristics. Such characteristics affect the cell weight. As mentioned, for each one, we define a specific weight. The weight affects the distance that the resource (in the vehicles case) can cover. For example, if a resource is in a cell and neighbors (cells) are '*important'* (there are schools, hospitals, etc located in it) then the resource will cover smaller area than in other cases. This is done even if the resources could cover a greater area (much more cells). Moreover, specific restrictions could be used in resources manipulation. For example, a user could define a condition of time in order to have resources response in a disaster situation. Probably, in ambulances or firefighting vehicles could be an upper time limit. These restrictions affect the number of cells that will be covered by resources. Thus, the distance that every resource will cover is calculated by the following equation:

$$C_j = \frac{D}{\sum_{i=1}^{NH} w_i - NH + 1} \tag{9}$$

where

$$D = \frac{T}{60} \cdot S \tag{10}$$

and *T* is the time restriction defined by the user (in minutes), *S* is the resources maximum speed, w_i is the weight of each cell in the neighborhood, *j* is the resource number, and *NH* is the number of neighbors that could be initially covered without taking into consideration the cells weights. The cells in the area C_j will be covered by the specific resource and, thus, new neighbors are calculated. Finally, the total covered area is defined as follows:

$$C = \frac{\sum_{i=1}^{N_j} |Ns_i|}{N_c^2} \tag{11}$$

where $|Ns_i|$ is the number of the new list of the resource neighbors. A snapshot of the described model is depicted by Figure 3. Pins describe resources location for a specific particle. The described process is used for every particle. Particles try to maximize the area coverage. At the end of every iteration, the local best and the global best are calculated and particle locations are updated. If a particle should change location, we update the resource locations by selecting a cell in the neighborhood as the new location.

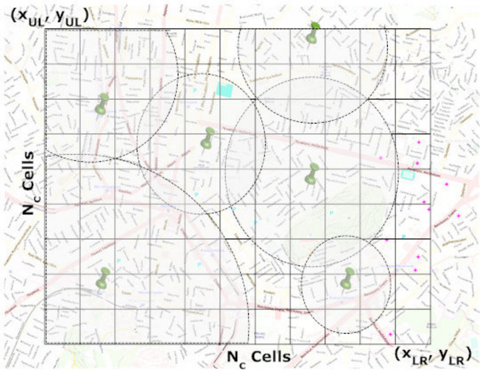


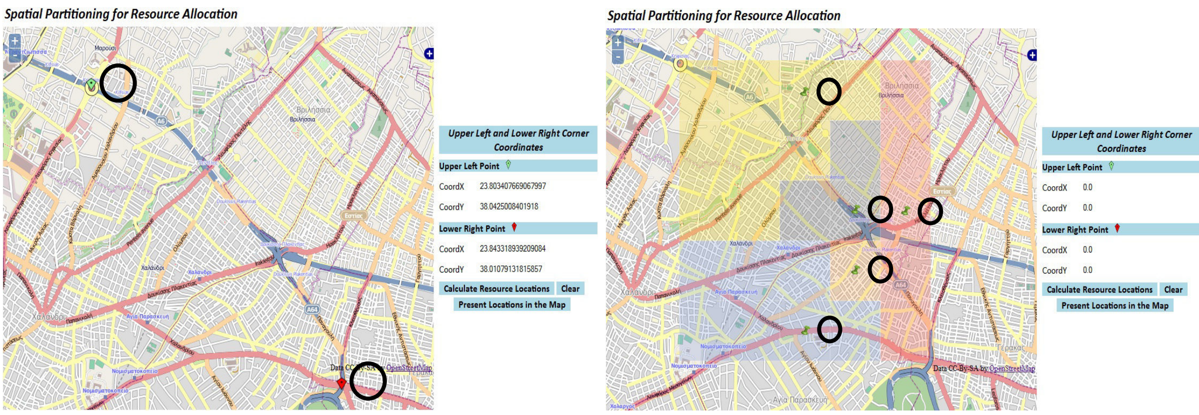
Figure 3. Resource allocation example

CASE STUDY: OPTIMAL ALLOCATION FOR AMBULANCES

In this section, we provide a case study of the proposed framework for a hypothetical scenario. We suppose that a number N_j of ambulances are available to be allocated in an area. We consider that $N_j = 5$. For these ambulances, we define their characteristics as follows:

No	Capacity	Max speed (Km/h)	Max travel distance (Km)
1	2	60	200
2	4	180	40
3	1	160	900
4	3	150	100
5	1	5	20

Moreover, we define a maximum response time T = 5 minutes. In the map, we select the desired area *A* by adding the green and the red marker (Figure 4a). The green marker defines the upper left corner of *A* and the red marker the lower right corner. For this, we provide a simple interface created by using the Openlayers API (http://openlayers.org/) based on maps provided by OpenStreetMap (http://www.openstreetmap.org/). Our functionality is provided by a Web Service uploaded in a Tomcat 7.0 server. After clicking on the appropriate button, the area is separated into five sub-areas with different colors (Figure 4b). The user has the opportunity to click on the pins representing resources and see their characteristics (including their actual locations). In Figure 4b, we can easily identify the ambulance No 5 which covers only one cell due to its poor capabilities. It should be noted that we utilize $N_c = 10$. When changes happen in the underlying data, disaster managers should re-run the algorithm in order to have up-to-date results. However, this implies that a module updating the database works on the background. Then, users simply choose again the upper left and the lower right corner of the AoI and run the algorithm. This will result the new resource locations. The same scenario stands for post-disaster cases. After a disaster happens, probably roads and sub-areas could be affected (e.g. roads could be closed or flooded, etc). In such cases, managers should rerun the algorithm after the underlying database is updated. This means that our module is closely connected with a module that is responsible to update the underlying information.

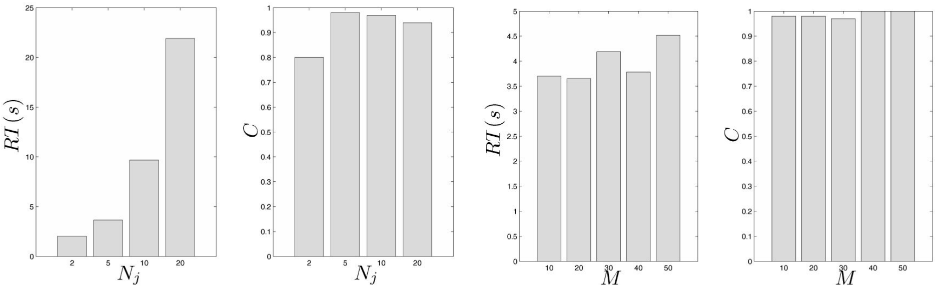


(a) (b)

Figure 4. Spatial partitioning result

Comparing our model, with other proposed frameworks, we could note that our proposed spatial partitioning module does not require any complex mathematical calculations for retrieving the final result. It is based only on a spatial database where information related to the target AoI is available. For example, there is no need for doing any image analysis and identifying specific sub-areas on the map. Moreover, it does not require any users' intervention as the final result is automatically calculated. Additionally, our approach does not need any pre-processing phase as the required calculations are made during the execution of the algorithm. In the majority of the other research efforts, data related to the required run time of the application are not available in order to make a comparison. Finally, to the best of our knowledge, our model is the first using the PSO algorithm for resulting the appropriate location for a number of resources in the pre-disaster phase.

In this point, we give some numerical results related to the required time (*RT*) for executing our algorithm as well as related to the total covered area (*C*). This way, we aim to provide to the reader with an insight of the performance of the proposed model. We run a number of experiments for different N_j (resources number) and *M* (particles number) values. In Figure 5a, we plot the *RT* and *C* for different N_j values. The greater the N_j is the grater the *RT* becomes. However, the maximum *RT* value is equal to 22 seconds approximately (when $N_j = 20$). Moreover, the N_j value affects the covered area as well. We see that for only 2 resources the area is covered by 80%. In the rest of the cases, the area is covered by 97% in average.



(a) (b)

Figure 5. Required time and coverage for different resources number

In Figure 5b, we plot our results for different *M* values. The *RT* values remain at the same levels while the area is covered by 100% approximately especially when M = 40 or 50. The reason is that due to the large particles number the algorithm has the opportunity to find the optimal solution (full area coverage). Finally in Figure 6, we see our results for different N_c values. In the first row of the plot we consider $N_j = 5$ and in the second row of the plot $N_j = 10$. We see that the *RT* is greater when N_j is large. However, in average, *C* is greater when N_j is small. In general, our model performs well achieving a large number of *C* in parallel with a small amount of time for calculating the final results.

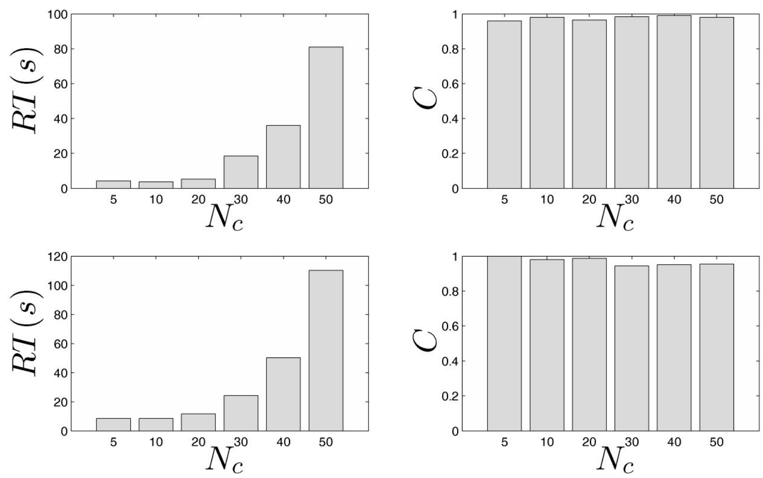


Figure 6. Required time and coverage for different grid cells number

CONCLUSIONS AND FUTURE WORK

For handling emergency situations an efficient technique is to place a number of resources in optimal locations before disaster happens. Thus, we are able to have the best response in emergencies. In this paper, we propose an algorithm for finding the optimal locations for resource placement. We present a methodology for solving the discussed problem based on the known PSO algorithm. The contribution of our work is an efficient algorithm that combines resource characteristics with the underlying spatial information for retrieving the final result. Moreover, our module does not require users' involvement in the results definition as well as it does not require any complex models and mathematical calculations. In our experiments, we provided a case study for a hypothetical scenario where five ambulances should be allocated in an area defined by the user. Numerical results show that our technique is efficient as it does not require a lot of time for the discussed calculation while the area coverage remains at high levels.

A limitation of the proposed model is that the definition of the relative importance of each criterion over another for cells weight calculation is manually done by developers. A future work is the provision of a tool where disaster managers will define the relative importance of each criterion and, thus, cell weights will be adapted to their experience. Furthermore, final users' evaluation will be very important as through this process, the proposed framework will be adapted to their needs. Incorporating a more intelligent technique for handling temporal changes in the database, like holidays where some buildings or roads are not crowded is another future extension of our work. For now, we can handle every change in the underlying information as the proposed module is combined with a module responsible to update the underlying spatial database when changes happen. For example, when some roads are closed the underlying database is updated by the responsible module and, thus, the proposed algorithm can be adapted to the new situation. The same happens in the post-disaster phase. In such situations, disaster managers can simply rerun the algorithm and take the new resource locations.

ACKNOWLEDGMENTS

This work was partially supported by the European Commission through the FP7 SEC Programme in the scope of the project IDIRA (Interoperability of data and procedures in large-scale multinational disaster response actions), contact FP7-ICT- 261726 (http://www.idira.eu).

REFERENCES

1. Aziz, N.A. A., Mohemmed, A. W. and Zhang, M. Particle Swarm Optimization for Coverage Maximization and Energy Conservation in Wireless Sensor Networks. EvoApplications 2010, Part II, pp. 51–60.
2. Frank, A. & Timpf, S. (1994). Multiple representations for cartographic objects in a multiscale tree: an intelligent graphical zoom. Computers and Graphics Special Issue: Modelling and Visualization of Spatial Data in Geographic Information Systems, 18(6), 823-829.
3. Frigui H and Krishnapuram R, (1999). A Robust Competitive Clustering Algorithm with Applications in Computer Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5), pp. 450-465.
4. Jain AK, Murty MN and Flynn PJ, (1999). Data clustering: a review. ACM Computing Surveys, vol. 31, no.3, pp. 264-323.
5. Joshi, D., Soh, L. K., and Samal, A., (2009). Redistricting Using Heuristic Based Polygonal Clustering. 9th IEEE International Conference on Data Mining.
6. Kennedy, J. & Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948.
7. Leung Y, Zhang J and Xu Z, (2000). Clustering by Space-Space Filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12), pp. 1396-1410.
8. Liu, L., Fong, S., and Ip, A., (2011). Grid Based Hierarchical Clustering for Spatial Resource Allocation. In Proc. of the IADIS International Conference on E-Society, Avila, Spain.
9. MacQueen J, (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297.

10. Moller-Jensen, L., and Kofie, R. Y., (2001). Exploiting Available Data Sources: Location/Allocation Modeling for Health Service Planning in Rural Ghana. Geografisk Tidsskrift: Danish Journal of Geography, vol. 101, pp. 145-154.
11. Rezayan, H., and Najian, A. H., (2008). Land Use Allocation Optimization Using Advanced Geographic Information Analyzes. World Applied Sciences Journal, vol. 3, 2008, pp. 136 – 142.
12. Riol, E., Puche, J. C., Delgado, F. J., Finat, J., & Martinez, R., (2011). Weighted Voronoi Diagrams for Optimal Location of Goods and Services in Planar Maps. In 8th Int. Symp. on Voronoi Diagrams in Sc. and Engineering.
13. Riviere, S., and Schmitt, D., (2007). Two Dimensional Line Space Voronoi Diagram. In Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering, pp. 168-175.
14. Saaty, T.L. (1990). How to make a decision: The Analytic Hierarchy Process. European Journal of Operational,Research, vol. 48, pp. 9−26.
15. Shanbhag, P., Rheingans, P., and desJardins, M., (2005). Temporal Visualization of Planning Polygons for Efficient Partitioning of Geo Spatial Data. In Proc. of the IEEE Symp. on Information Visualization, USA.
16. Tu, J., Chen, C., Huang, H., & Wu, X. (2005). A visual multi-scale spatial clustering method based on graph-partition. In Geoscience and Remote Sensing Symposium, vol. 2, pp. 745-748.
17. van Oosterom, P. & Schenkelaars, V. (1995). The development of an interactive multi-scale GIS. International Journal of Geographical Information Systems, 9(5), 489-507.
18. Wu, A. & Leahy, R. (1993). An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. IEEE Trans. on Pattern Analysis & Machine Intelligence, 15(11).
19. Wu, H., Pan, M., Yao, L., & Luo, B. (2007). A partition-based serial algorithm for generating viewshed on massive DEMs. International Journal of Geographical Information Science, 21(9), 955-964.
20. Yang, W. & Gold, C.M. (1996). Managing spatial objects with the VMO-Tree. In Proceedings Seventh International Symposium on Spatial Data Handling, pp. 711-726, Delft, The Netherlands.
21. Zhao, X., Chen, J., & Zhao, R. (1999). Dynamic Spatial Indexing Model Based on Voronoi. In Proceedings of the International Symposium on Digital Earth, Science Press.
22. Zou Yi and Chakrabarty Krishnendu, (2003). Sensor Deployment and Target Localization Based on Virtual Forces. INFOCOM 2003, IEEE , Volume: 2 , 30 March-3 April 2003, Pages:1293 - 1303 vol.2.