A Distributed Convergent Solution to the Ambulance Positioning Problem on a Streetmap Graph

Yuquan Wang, Michele Colledanchise, Alejandro Marzinotto, Petter Ögren

The authors are with the computer vision and active perception lab, royal institute of technology (kth), se-100 44 stockholm, sweden.
e-mail: {yuquan|miccol|almc|petter}@kth.se

Abstract: In this paper, we combine ideas from multi-agent cooperative coverage control, with problem formulations from the resource allocation field, to create a distributed convergent approach to the ambulance positioning problem. Inspired by coverage control we use the graph version of so-called Voronoi regions, making the solution distributed and reactive, thereby freeing computational resources. The solution is distributed in the sense that each vehicle only needs to know the positions of its neighbors, and the computations of each vehicle only depend on the size of its Voronoi region/set. This implies that considering a problem of twice the size, using twice the number of vehicles will leave the computational load per vehicle unchanged. The freed resources are used to capture the allocation problem in more detail: maximizing an estimate of the victim survival probability instead of more coarse measures of ambulance availability. Using real city street map data from OpenStreetMap (OSM), we provide simulation results illustrating the applicability of our approach. Finally, we prove that the proposed distributed algorithm is convergent in the sense that it finds a local optimum in finite time.

---

1. INTRODUCTION AND RELATED WORK

In the near future, the distribution of emergency response vehicles, such as ambulances, fire trucks, and police cars, could be controlled in a turn-by-turn fashion which is similar to the navigation applications available for most smart phones today. The real time positions of the vehicles, as well as the street maps, population densities, and estimated incident likelihoods are all available, together with the communication technology needed to support such a system.

The problem addressed in this paper is that of reactively distributing a set of ambulances in real time in an urban area, such as the one in Fig. 1 (All map data are taken from OSM[1]). The objective of the distribution is to maximize the survival chances of the next victim, assuming we have data on how survivability varies with time-to-service and the other city data mentioned above.

Efforts in operations research have already made significant contributions to emergency response resource allocation. When the availability is prescribed by law[2] the problem formulation is quite straightforward. Successful implementations include the one reported by Repede and Bernardo (1994), where the use of a probabilistic ambulance location model led to an increase from 84% to 95% in the number of calls covered in 10 minutes or less, and a

decrease in the total response time by 36%. We will now describe the related work in more detail.

Fig. 1. OSM map of Södermalm Stockholm. All the map figures through this paper have latitude as y axis unit and longitude as x axis unit.

As mentioned above, this work lies in the intersection of coverage control and resource allocation, and the most important related work from both fields is summarized in Table 1. Resource allocation research applied to ambulance location problems dates back to the 1970s, with important contributions by Toregas et al. (1971); Church and Velle (1974); Gendreau et al. (1997); Daskin (1983); Ball and Lin (1993). Extensive overviews are given by Brotcorne et al. (2003); Pillac et al. (2012) and more recent contributions include Erwig and Hagen (2000); Daskin (2008) and Azizan et al. (2012). Most of the early approaches are based on Mixed Integer Linear Programming (MILP) formulations. Two fundamental models used in these optimization problems are the *Location Set Covering Model* (LSCM) by Toregas et al. (1971) and *Maximal Covering*

---

[1] http://wiki.openstreetmap.org/
[2] The *United States Emergency Medical Services Act* states that 95% of requests should be serviced within 10 minutes in urban areas (Brotcorne et al., 2003), whereas in Montreal, Canada, the required time is 7 minutes and it must hold for more than 90% of the population (Gendreau et al., 2001).

Table. 1. Problems investigated in related work.

Reference	Objective	Constraints	Problem Space	Reactive	Distributed
LSCM (Toregas et al., 1971)	Min no. of ambulances	All $d < r$	Set covering	No	No
MCLP (Church and Velle, 1974)	Max coverage	All $d < r$	Set covering	No	No
DDSM^t (Gendreau et al., 2001)	Max double coverage $< r_1$	All $d < r_2$	Set covering	✓	No
MCLP for Streetmaps (Azizan et al., 2012)	Max coverage	All $d < r$	Graph (streetmap)	No	No
(Cortes et al., 2004)	Min sum squared distance	none	$\mathbb{R}^n$	✓	✓
(Breitenmoser et al., 2010)	Min sum squared distance	none	Graph (3D tessellation)	✓	✓
(Erwig and Hagen, 2000)	Find nearest hospital	none	Graph	No	No
This paper	Max survival probability	none	Graph (street map)	✓	✓

Location Problem (MCLP) by Church and Velle (1974). LSCM provides a lower bound on the number of ambulances required to ensure the full coverage and therefore makes itself suitable as a planning tool, while the objective of MCLP is to make the best possible use of the available amount of ambulances. A reactive approach building upon Tabu search was later presented by Gendreau et al. (2001), where two different coverage radii were taken into account. An outer r_2 covers the basic demand, and an inner r_1 covers the advanced demand. The allocation was then performed with the objective of maximizing the double coverage within r_1. All these approaches are rooted in the set covering problems by Toregas et al. (1971) and are inherently centralized. The idea of using detailed street maps was proposed by Azizan et al. (2012), but the problem formulation was still centralized and building upon MCLP. Probabilistic extensions to LSCM and MCLP have been proposed by Daskin (1983); Ball and Lin (1993). In the *Maximum Expected Covering Location Problem* (MEXCLP) given by Daskin (1983) the MCLP is extended by maximizing the sum of the probabilities of being served of all the vertices. In the Rel-p method by Ball and Lin (1993), LSCM is extended by incorporating a linear constraint on the number of vehicles required to achieve a given reliability level. A common drawback of all approaches described above is the lack of decentralized control, and the use of somewhat coarse notions of coverage, based on different time limits, instead of estimates from real data of how survivability depends on time-to-service.

In control theory, a vast amount of work on coverage control problems have grown from the seminal paper by Cortes et al. (2004), where the sensor coverage problem was solved using Voronoi partitions, and a distributed reactive controller was proposed, moving each agent towards the centroid of its own Voronoi region. This problem was formulated in continuous space, $\mathbb{R}^n$ and the objective minimized was the sum of squared distances from agents to measured instances. A graph version of such Voronoi partitions called Voronoi diagrams, had earlier been proposed by Erwig and Hagen (2000), where it was noted that such partitions were useful for finding the nearest hospital in case of an emergency. When addressing the coverage problem on surfaces embedded in 3D, Breitenmoser et al. (2010) built upon the results of Cortes et al. (2004) using a graph version of the Voronoi partitions where the objective function was once again the sum of squared distances. Recent efforts in multi-agent distributed optimization given by Nedic and Ozdaglar (2009) solve constrained optimization problems cooperatively by agents. However this method is not applicable to optimization problems with a large set of constraints, such as LSCM or MCLP.

The contribution of this paper is an approach that is reactive, convergent and distributed, solving a localization problem on a real street map graph, with an objective function that actually captures survivability, instead of more coarse notions of coverage. As shown in Table 1 this has not been done before.

In practice, we suggest that initial ambulance positions are first found using e.g., a MCLP method as described in the references. Then, the proposed approach is applied to refine these positions in a reactive and distributed fashion.

The outline of this paper is as follows. First, we formulate the problem and review some notation on graphs and Voronoi diagrams in Sec. 2. Then, we propose a solution in Section 3. The solution is analyzed in Section 4, and implementation details are given in Section 5. Finally, we provide a set of simulation examples in Section 6 and present the conclusions of the paper in Section 7.

2. PROBLEM FORMULATION

In this section we define the problem and review some notation on graphs and Voronoi diagrams.

2.1 Definitions

Let the road network be given by graph, i.e., an ordered pair $G = (V, E)$ comprising a set V of vertices together with a set $E \subset V \times V$ of edges, and let P be a set of ambulances. We restrict the ambulance positions to the vertices, therefore $P \subset V$. We denote the cardinality of a set by $|.|$, e.g. the number of vertices is $|V|$ and the number of ambulances is $|P|$. At each vertex $v_i \in V$ we have a population size of $M(v_i)$, and each edge $e_j \in E$ has a distance $d(e_j)$. Let this distance be measured in traversal time, i.e., $d(e_j) = l_j/v_j$ where l_j is length and v_j is average speed. A path P_{ab} of length l from vertex v_a to v_b is a sequence of vertices $\langle v_0, \ldots, v_n \rangle$ with $v_0 = v_a$, $v_n = v_b$ and $(v_{i-1}, v_i) \in E$ for $1 < i \leq n$. We denote $d(v_a, v_b)$ the shortest distance from v_a to v_b. We reserve k to denote the discrete time step. For instance, let the positions of the ambulances at time step k be given by $\boldsymbol{p}(k) = (p_1(k), \ldots, p_{|P|}(k))$, where $p_i(k) \in V$.

2.2 Demand Distribution

We assume that the demand of ambulance services at a given location is proportional to the instant population size, and therefore time varying. For example, during

weekends, the demand at a shopping mall is high and the demand at an elementary school is low. Inspired by the *degree of dynamism* described by Pillac et al. (2012), we denote the time-varying demand probability mass of each vertex v_i as

$$\rho(v_i) = \frac{M(v_i)}{\sum_{j=1}^{|V|} M(v_j)}. \tag{1}$$

Note that here, and in the rest of the paper we suppress the explicit time dependence in the notation.

2.3 Voronoi Diagram

A Voronoi diagram partitions the vertices V of a graph into a number of Voronoi sets V_i, given a set of pre-specified vertices called Voronoi nodes $\{\bar{v}_i\} = \bar{V} \subset V$. Each set V_i consists of all the vertices closer to a given Voronoi node $\bar{v}_i$ than any of the other Voronoi nodes. In this paper, we use the ambulances' positions p_i as the Voronoi nodes $\bar{v}_i = p_i$. Thus, given a connected graph $G = (V, E)$ and a set of ambulances $\boldsymbol{p}$, a Voronoi diagram is the partition

$$\boldsymbol{V} = \{V_1, \ldots, V_{|P|}\} \tag{2}$$

of the vertices of the graph where for each vertex $v \in V_i$ we have

$$d(v, p_i) \leq d(v, p_j) \tag{3}$$

for $j \neq i$, $j = 1, \ldots, |P|$ and a Voronoi set V_i associated with an ambulance p_i is given by:

$$V_i(\boldsymbol{p}) = \{v : d(v, p_i) \leq d(v, p_j), \ \forall j \neq i\}. \tag{4}$$

We call the ambulance j a neighbor to ambulance i, if some node in another ambulance's Voronoi set $v_j \in V_j$ is connected by an edge in E to some node $v_i \in V_i$. We denote the set of neighboring ambulances to ambulance i as N_i.

Remark 2.1. If $d(v, p_i) = d(v, p_j)$ in (3), we can choose either $v \in V_i$ or $v \in V_j$ as long as $V_i \cap V_j = \phi$ and $V_1 \cup \ldots \cup V_{|P|} = V$. ∎

2.4 How to Compute a Voronoi Diagram

Given (3) and (4), computing a Voronoi diagram on a graph can be done by first solving a shortest path problem for all pairs of vertices, which can be done in polynomial time using e.g. the Floyd-Warshall algorithm (Cormen et al., 2001). We store the shortest distance between all pairs of vertices in a distance matrix

$$D_G = \begin{pmatrix} 0 & d(v_1, v_2) & \cdots & d(v_1, v_{|V|}) \\ d(v_2, v_1) & 0 & \cdots & d(v_2, v_{|V|}) \\ \vdots & \vdots & \ddots & \vdots \\ d(v_{|V|}, v_1) & d(v_{|V|}, v_2) & \cdots & 0 \end{pmatrix}, \tag{5}$$

where the i-th row gives the shortest distances from v_i to all its connected vertices, and the j-th column gives the shortest distances from all the connected vertices to v_j.

Assume ambulance j knows its position p_j on the graph and the neighboring ambulances positions p_i, for $i \in N_j$. We construct a sub-distance matrix D_{V_j} which contains the shortest distances from p_j and p_i for $i \in N_j$ to their connected vertices:

$$D_{V_j} = \begin{pmatrix} d(p_j, v_1) & d(p_j, v_2) & \cdots & d(p_j, v_{|V|}) \\ \vdots & \vdots & \ddots & \vdots \\ d(p_i, v_1) & d(p_i, v_2) & \cdots & d(p_i, v_{|V|}) \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}, \tag{6}$$

where D_{V_j} contains $|N_j| + 1$ rows. In each column of D_{V_j}, if $D_{V_j}(p_j, y)$ is the minimum element, then it means the y-th vertex belongs to the Voronoi set V_j. More details are given in Algorithm 1 in Section 5.

2.5 Maximizing Survival Probability

In this section, we will describe the problem considered in detail, i.e. the problem of maximizing the expected survival probability with respect to the ambulance positions.

As noted in Sec. 1, legislation sometimes requires a certain ambulance coverage in terms of time-to-service for a given percentage of the population. However, these numbers are somewhat arbitrarily chosen, as can be seen from the fact that the numbers differ between countries, as well as between rural and urban areas. We propose to replace such times and percentages by maximizing an estimate of the average survivability. Assuming that either a medical expert, or hospital data, can be used to create an estimate on how average survivability varies with waiting time. Let $S : \mathbb{R}_+ \to [0, 1)$ be that survivability. Clearly, S is non-increasing; the sooner the ambulance arrives, the better. Furthermore, in the light of data reported in (Brotcorne et al., 2003; Gendreau et al., 2001) S is fairly large for the first 7 to 10 minutes, and smaller, but not zero, at 30 minutes. For the sake of the simulations in Sec. 6, we model the survivability with an affine function, but stress the fact that in a real operational system, $S(d)$ would be given by data and/or medical experts.

Given S above, we are now ready to formulate the objective function of the ambulance positioning problem. The average survivability for the whole population of the graph is as follows

$$s_{avg}(\boldsymbol{p}) = \sum_{v_i \in V} \rho(v_i) S(\min_j d(v_i, p_j)), \tag{7}$$

where $d(v_i, p_j)$ is the shortest distance from v_i to p_j on the graph G and $\rho(v_i)$ is the demand probability mass at v_i. Therefore our main problem is the following:

Problem 1. Centralized Ambulance Positioning

$$\max_{p_1, \ldots, p_{|P|}} s_{avg}(\boldsymbol{p}).$$

3. PROPOSED SOLUTION

Inspired by the continuous analysis of Cortes et al. (2004), we let the following definition be a core part of our proposed decentralized reactive ambulance control scheme.

Definition 1. Service center of a Voronoi set

$$V_j^0(V_j(\boldsymbol{p})) = \operatorname{argmax}_{p_j \in V_j} \sum_{v_i \in V_j} \rho(v_i) S(d(v_i, p_j)). \tag{8}$$

Thus the service center is the ambulance position that would maximize the survival probability of V_j if the Voronoi sets were fixed and did not depend on the ambulance positions.

We now propose the following solution to Problem 1.

Solution 1. Apply the following three steps iteratively.

(a) Compute the Voronoi partitions according to (4).
(b) Compute the Service center according to (8).
(c) Move to the Service center: $p_j(k+1) = V_j^0(k)$.

If a steady state is reached, we stop the iteration until either demand, or the number or positions of the ambulances change due to some outer influence. ∎

Remark 3.1. Note that the proposed solution above is indeed distributed and reactive. To compute the Voronoi set (a), an individual ambulance only needs to know the positions of its neighbors, i.e. the vehicles that are reasonable contenders to being closest to the considered vertices, and to compute the Service center (b) it only needs to know its own Voronoi set.

4. THEORETICAL ANALYSIS

In this section, we will show that the proposed solution is convergent. In fact, we will even show that it will reach a local optimum in a finite set of steps.

First, we need the following Lemma, stating that the inner minimization of the objective function in (7) can be removed by explicitly summing over the Voronoi sets.

Lemma 1. The average survivability for the whole population of the graph $s_{avg}(\boldsymbol{p})$ is equal to the sum of the average survivability of the population of each Voronoi set:

$$s_{avg}(\boldsymbol{p}) = \sum_{j \in \bar{V}} \sum_{v_i \in V_j} \rho(v_i) S(d(v_i, p_j)). \qquad (9)$$

Proof 1. Using property (4) of a Voronoi set V_j, we have $\forall v_i \in V_j, p_j = \arg\min_{p_k} d(v_i, p_k)$ for $k = 1, \ldots, |P|$. Therefore we have

$$s_{avg}(\boldsymbol{p}) = \sum_{v_i \in V} \rho(v_i) S(\min_j d(v_i, p_j))$$
$$= \sum_{j \in \bar{V}} \sum_{v_i \in V_j} \rho(v_i) S(d(v_i, p_j)),$$

using the fact that $V_i \cap V_j = \phi$ and $V_1 \cup \ldots \cup V_{|P|} = V$. ∎

Now we define the following function

$$h(q, W) = \sum_{v_i \in W} \rho(v_i) S(d(v_i, q)), \qquad (10)$$

which corresponds to the average survivability of the population in W if they were attended to by a vehicle at q. We are now ready to prove convergence of the proposed approach.

Theorem 2. The algorithm in Solution 1 will converge to a local optimum in a finite number of steps.

Proof 2. First we state the following set of inequalities.

$$s_{avg}(\boldsymbol{p}(k)) = \sum_{j \in \bar{V}} \sum_{v_i \in V_j} \rho(v_i) S(d(v_i, p_j)) \qquad (11)$$

$$= \sum_{j \in \bar{V}} h(\boldsymbol{p}(k), V_j(\boldsymbol{p}(k))) \qquad (12)$$

$$\leq \sum_{j \in \bar{V}} h(V_j^0(V_j(\boldsymbol{p}(k))), V_j(\boldsymbol{p}(k))) \qquad (13)$$

$$= \sum_{j \in \bar{V}} h(\boldsymbol{p}(k+1), V_j(\boldsymbol{p}(k))) \qquad (14)$$

$$\leq \sum_{j \in \bar{V}} h(\boldsymbol{p}(k+1), V_j(\boldsymbol{p}(k+1))) \qquad (15)$$

$$= s_{avg}(\boldsymbol{p}(k+1)). \qquad (16)$$

Above, (11) holds due to Lemma 1, (12) holds due to the definition of h, (13) holds due to the definition of the service center, (14) holds due to step (c) of the update law (15) holds due to the fact that a new Voronoi partition can only re-assign vertices to vehicles that are closer, and S is non-increasing, and finally, (16) holds due to the definition of h.

The fact that the sequence of improvements is finite can be seen from the fact that the problem is discrete, and bounded, which means that there is a smallest possible improvement, which in turn implies that there cannot be an infinite number of such improvements. Therefore, the sequence $s_{avg}(\boldsymbol{p}(k))$ will converge to a value corresponding to either a unique local maximum, or a limit cycle of local maxima positions with identical average survivability. In both cases, no improvements can be achieved by moving to new service centers. ∎

Remark 4.1. Note that we can only prove the convergence to a local optimum. A case where this local optimum is different from the global one can be found in Fig. 2. Suppose there are two clusters of highly populated vertices, and two ambulances. The optimal solution is to deploy the two ambulances (dashed red circle) in the left cluster, if the population of the vertices to the left is orders of magnitude larger than the population of the vertices to the right: $\sum_{i=1}^{5} \rho_i \gg \sum_{i=6}^{9} \rho_i$ and the distance between the two clusters, which is $d(n_1, n_8)$ in this case, is large compared to distances inside a cluster. However, if one ambulance starts in the smaller, right cluster, then the Voronoi sets will assign one cluster to each ambulance, and they will both position themselves relative to their cluster. Locally, there will be no incentive for the rightmost ambulance to move towards the larger cluster, since all of it will still be assigned to the leftmost ambulance.
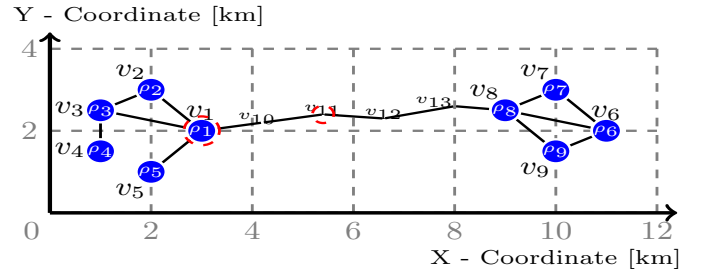

Fig. 2. An example in which the proposed solution will reach a local, not global, optimum.

5. IMPLEMENTATION

The Voronoi diagram partition is based on the distance matrix D_G which contains the shortest distances between any pair of vertices on the map. Applying the Floyd Warshall algorithm (Cormen et al., 2001, p559) we obtain D_G by solving the *all-pairs shortest-paths problem* on the Graph G in time $\mathcal{O}(|V|^3)$. This gives us the pre-computed shortest distances $d(v_i, v_j)$ and the shortest paths $p(v_i, v_j)$, for $v_i, v_j \in V$. In algorithm 1, we list the steps to obtain a Voronoi set for any ambulance according to Sec. 2.4.

Algorithm 1. Distributed Voronoi set partition and maintenance algorithm.
Goal: To partition or maintain a Voronoi set.
Requires:
(1) Pre-computed distance matrix D_G.
Input:
(1) Ambulance position $p_j(k)$.
(2) Neighboring ambulances positions $p_i(k)$, for $i \in N_j(k)$.
Output: Updated Voronoi set $V_j(k)$.
if $V_j(k)$ *and* $V_i(k)$ *for* $i = 1, \ldots, |P|$ *are all initialized.*
then
    Let $\tilde{V} = V_j \cup_{i \in N_j(k)} V_i$.
    Extract the rows in D_G correspond to $p_j(k)$ and $p_i(k)$, for $i \in N_i(k)$, to initialize $D_{V_j}(k)$.
    Keep columns of D_{V_j} correspond to every $v \in \tilde{V}$.
end
foreach *column x of D_{V_j}* **do**
    if $d(p_j, v_x) = \min D_{V_j}(:, v_x)$ **then**
       $v_x \in V_j(k)$.
    end
end

Remark 5.1. In Algorithm 1 and later in Algorithm 2, we use ':' in the way as it is defined in MATLAB script. ■

In order to look for the service center V_j^0, we define a square survivability matrix. If we place the ambulance at v_j, the survivability of v_i weighted by the population is

$$w(v_i, v_j) = \rho(v_i)S(d(v_i, v_j)). \qquad (17)$$

We define the square survivability matrix W_{V_j} for the Voronoi set V_j as

$$W_{V_j} = \begin{pmatrix} 0 & w(v_1, v_2) & \cdots & w(v_1, v_{|V_j|}) \\ w(v_2, v_1) & 0 & \cdots & w(v_2, v_{|V_j|}) \\ \vdots & \vdots & \ddots & \vdots \\ w(v_{|V_j|}, v_1) & w(v_{|V_j|}, v_2) & \cdots & 0 \end{pmatrix}. \qquad (18)$$

Due to the definition (17), the service center V_j^0 defined by (8) is equivalent to the vertex with the maximum row sum in W_{V_j}.

We can evaluate every single square matrix $W_{V_i}(k)$ in time $\mathcal{O}(|V_i|^3)$ similarly as we evaluate D_G. Nevertheless we want to avoid redundant use of the Floyd Warshall algorithm, so we evaluate the survivability matrix for the entire graph G as

$$W_G = \begin{pmatrix} 0 & w(v_1, v_2) & \cdots & w(v_1, v_{|V|}) \\ w(v_2, v_1) & 0 & \cdots & w(v_2, v_{|V|}) \\ \vdots & \vdots & \ddots & \vdots \\ w(v_{|V|}, v_1) & w(v_{|V|}, v_2) & \cdots & 0 \end{pmatrix} \qquad (19)$$

and extract the required rows and columns from W_G to construct $W_{V_i}(k)$ as in Algorithm 1. We list the detailed steps in Algorithm 2.

Algorithm 2. Voronoi set service center calculation.
Goal: Calculation of the service center $V_j^0(k)$.
Requires: The weighted distance matrix W_G.
Input: The Voronoi set $V_j(k)$.
Output: The service center $V_j^0(k)$.
if $V_j(k)$ *and* $V_i(k)$ *for* $i = 1, \ldots, |P|$ *are all initialized.*
then
    Extract the rows of W_G that correspond to $\forall v \in V_j(k)$ to initialize W_{V_j}.
    Keep the columns of W_{V_j} that correspond to $\forall v \in V_j(k)$.
    Let $sum = W_{V_j}(1, :)$ and $V_j^0(k) = 1$.
end
foreach $x = 2 : |W_{V_j}|$ **do**
    if $\sum W_{V_j}(x, :) > sum$ **then**
       Let $sum = \sum W_{V_j}(x, :), V_j^0(k) = x$.
    end
end

Given the steps for Voronoi diagram partition and service center calculation, we list the steps of the distributed reactive positioning algorithm for a particular ambulance in Algorithm 3. Notice that there is no stopping criterion, as it is designed for a reactive behavior. So if at any time any ambulance is dispatched, the other ambulances will automatically re-allocate themselves.

Remark 5.2. Since the ambulance position is restricted to be on a vertex of the graph G, we make the following compromise to achieve real time performance: if the ambulance is on the way to its next position $p(t) \neq p(k+1)$, we use its latest recorded position on the graph $p(t) = p(k)$ and update $p(t)$ only when it reaches its goal V_j^0. ■

6. EXAMPLE

In this section, we use the graph extracted from the OSM map of Södermalm island (with a population of around 100 000 people) located in central Stockholm. Our plots include the Voronoi diagram of the graph, the optimum ambulance positions given by Solution 1 and how ambulances re-allocate themselves in the case that one ambulance is dispatched.

6.1 OSM map and parsing

OSM is a widely used open-source map. The downloaded OSM model is an Extensible Markup Language (XML) file. We first need to extract only the "street" nodes based on the tags (for instance where the tag is 'highway' etc.) and then remove the unconnected nodes such that the graph is connected and the distances are finite. A further simplification is to filter out the non-intersection nodes. In Fig. 3, we overlap the extracted graph on the OSM map of Södermalm. Each node is marked as a black dot and each edge is marked with a black line. For this map we reduced the number of nodes from 38084 to 3964.

Algorithm 3. Distributed reactive ambulance allocation.

Goal: Determine the j-th ambulance's next position $p_j(k + 1)$ on the graph such that eventually it reaches its optimum location V_j^*.

Requires:
(1) Algorithm 1 and 2.
(2) Pre-computed distance matrix D_G.
(3) Pre-computed weighted distance matrix W_G.
(4) Initialized $V_j(0)$ and $V_j^0(0)$ for $j = 1, \ldots, |P|$.

Input: $p_i(k)$, for $i \in N_j(k)$.

Output: $p_j(k + 1)$.

if $p_i(k) = p_i(k - 1)$, for $i \in N_j(k)$ **then**
  if $p_j(k) = V_j^0(k)$ **then**
    | $p_j(k + 1) = p_j(k)$.
  else
    | Apply: $p_j(k + 1) = V_j^0(k)$.
  end
else
  Update $V_j(k + 1)$ using algorithm 1.
  if $V_j(k + 1) \neq V_j(k)$. **then**
    | Calculate $V_j^0(k + 1)$ using algorithm 2.
  else
    | $V_j^0(k + 1) = V_j^0(k)$.
  end
  if $p_j(k) = V_j^0(k)$ **then**
    | $p_j(k + 1) = p_j(k)$.
  else
    | Apply: $p_j(k + 1) = V_j^0(k)$.
  end
end

6.2 Validation

We choose to use 4 ambulances and select their initial positions randomly. For simplicity, we choose the allowed speed v_k to be 80km/h for all the edges and we use a affine survival probability function S to calculate W_G. Given the graph shown in Fig. 3 and its distance matrix D_G, we applied algorithm 1 and got four Voronoi sets shown in Fig. 4.
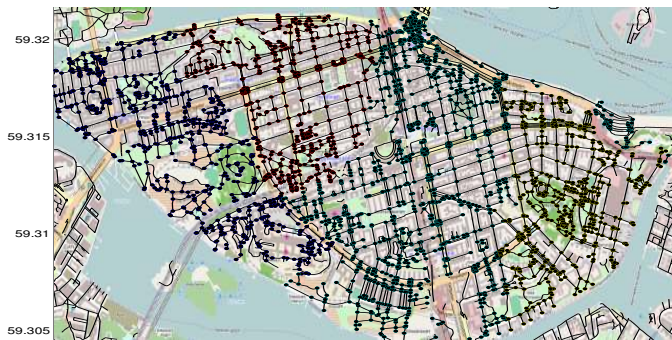

Fig. 3. OSM Map of Södermalm overlapped by graph.

Different Voronoi sets are labeled by colors and the ambulance position is annotated with a big dot inside a black ellipse. Using algorithm 3, the four ambulances reached the corresponding service centers V_j^0 as shown in Fig. 5. Note that due to the changes of the ambulances positions, the Voronoi sets are re-partitioned. For instance there are a few nodes in the upper right of Fig. 4 are initially covered
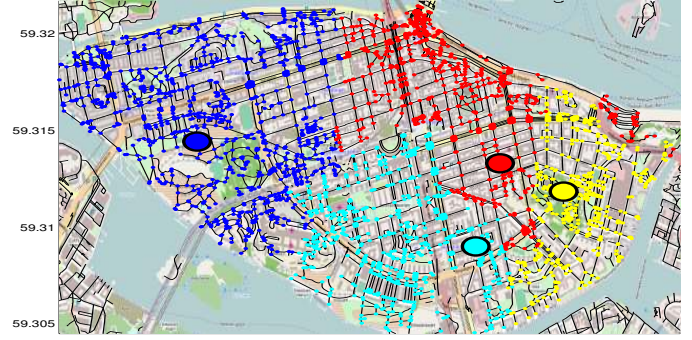

Fig. 4. Initial ambulance positions and the corresponding Voronoi diagram

by the red ambulance and later they are covered by the yellow ambulance in Fig. 5. We refer to this set of nodes as set X.
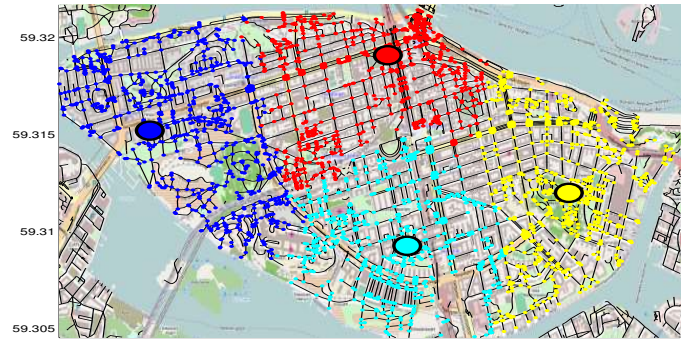

Fig. 5. Stable ambulance positions and the corresponding Voronoi sets.

Suppose we received a call of service by the red node shown in Fig. 6, the corresponding ambulance (red one) is dispatched. The remaining three ambulances then apply algorithm 3 and move to the new optimum as shown in Fig. 6. Note that now the nodes in set X are covered
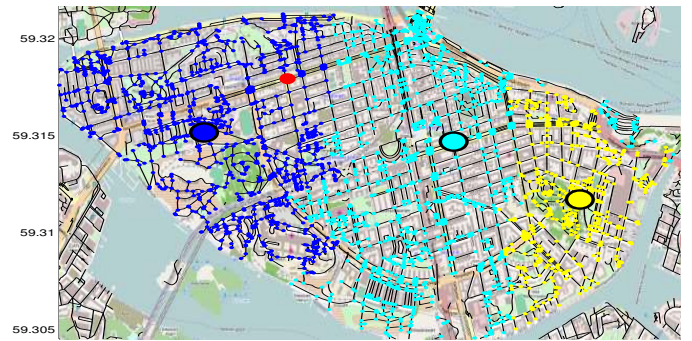

Fig. 6. Ambulances re-allocation in case that one ambulance dispatched

by the cyan ambulance in Fig. 6 rather than the yellow ambulance.

The reason that the set is assigned in a somewhat unintuitive way, is that there is a sharp height difference in this area of the island, making X accessible only by roads that follow the waterline to the west.

7. CONCLUSIONS

The proposed method provides a distributed reactive solution to the ambulance positioning problem. Given the pre-calculated shortest distance information, there is no need of solving a general optimization problem in real time for all ambulances. Instead, each ambulance is only required to keep track of its neighbors, to compute its Voronoi set, and then do an optimization based on this set, which is of average size $|V|/|P|$. Thus, the computational requirements for each ambulance are not changed as long as increases in problem size, $|V|$ are matched by more vehicles $|P|$. It was also shown that the proposed method will result in a convergence of the average survivability to a local maximum.

8. ACKNOWLEDGMENT

REFERENCES

Mohd Hafiz Azizan, Cheng Siong Lim, WA Hatta, and Lian Chai Gan. Application of openstreetmap data in ambulance location problem. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on*, pages 321–325. IEEE, 2012.

Michael O Ball and Feng L Lin. A reliability model applied to emergency service vehicle location. *Operations Research*, 41(1):18–36, 1993.

Andreas Breitenmoser, J Metzger, Roland Siegwart, and Daniela Rus. Distributed coverage control on surfaces in 3d space. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5569–5576. IEEE, 2010.

Luce Brotcorne, Gilbert Laporte, and Frederic Semet. Ambulance location and relocation models. *European journal of operational research*, 147(3):451–463, 2003.

Richard Church and Charles R Velle. The maximal covering location problem. *Papers in regional science*, 32(1):101–118, 1974.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2001.

Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, 2004.

Mark S Daskin. A maximum expected covering location model: formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, 1983.

Mark S Daskin. What you should know about location modeling. *Naval Research Logistics (NRL)*, 55(4):283–294, 2008.

Martin Erwig and Fernuniversitat Hagen. The graph voronoi diagram with applications. *Networks*, 36(3):156–163, 2000.

Michel Gendreau, Gilbert Laporte, and Frédéric Semet. Solving an ambulance location model by tabu search. *Location science*, 5(2):75–88, 1997.

Michel Gendreau, Gilbert Laporte, and Frédéric Semet. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing*, 27 (12):1641–1653, 2001.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, 2009.

Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 2012.

John F Repede and John J Bernardo. Developing and validating a decision support system for locating emergency medical vehicles in louisville, kentucky. *European Journal of Operational Research*, 75(3):567–581, 1994.

Constantine Toregas, Ralph Swain, Charles ReVelle, and Lawrence Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.