Published in final edited form as:

IEEE Trans Syst Man Cybern A Syst Hum. 2012 September 1; 42(5): 1194–1205. doi:10.1109/TSMCA.2012.2183360.

A Novel Space Partitioning Algorithm to Improve Current Practices in Facility Placement

Tamara Jimenez,

Center of Computational Epidemiology and Response Analysis at the University of North Texas, Denton, TX

Armin R Mikler, and

Center of Computational Epidemiology and Response Analysis at the University of North Texas, Denton, TX

Chetan Tiwari

Center of Computational Epidemiology and Response Analysis at the University of North Texas, Denton, TX

Tamara Jimenez: tamara.jimenez@unt.edu; Armin R Mikler: mikler@unt.edu; Chetan Tiwari: chetan.tiwari@unt.edu

Abstract

In the presence of naturally occurring and man-made public health threats, the feasibility of regional bio-emergency contingency plans plays a crucial role in the mitigation of such emergencies. While the analysis of in-place response scenarios provides a measure of quality for a given plan, it involves human judgment to identify improvements in plans that are otherwise likely to fail. Since resource constraints and government mandates limit the availability of service provided in case of an emergency, computational techniques can determine optimal locations for providing emergency response assuming that the uniform distribution of demand across homogeneous resources will yield an optimal service outcome. This paper presents an algorithm that recursively partitions the geographic space into sub-regions while equally distributing the population across the partitions. For this method, we have proven the existence of an upper bound on the deviation from the optimal population size for sub-regions.

Index Terms

Epidemics; Pandemics and Biological Threats; Public Health Preparedness; Response Analysis; Optimization; Algorithm; Algorithmic Optimization

I. Introduction

Recent events have emphasized the necessity of emergency response plans for natural as well as man-made disasters. Naturally occurring disasters can be caused by diseases, such as H1N1 or Cholera, as well as environmental events, including earthquakes and hurricanes. Man-made emergencies include those caused by acts of terror and might involve the spread of biological agents, such as anthrax. Emergency preparedness coordinators must therefore devise contingency plans for different types of emergencies. In particular, regional public health departments must address public health threats, such as epidemics caused by seasonal influenza, or the release of anthrax. Many of the resulting emergency response scenarios involve the placement of points of dispensing (PODs) throughout the geographic region, at which the general public is provided prophylactic medications or vaccines in a timely

manner [1]. It is crucial to analyze if an underlying response plan will succeed in case of an emergency, i.e. meet mandated time-frames. We have developed the computational framework RE-PLAN (REsponse PLAN ANalyzer), which aids public health experts to determine optimal POD placement and facilitate analysis of existing response scenario. While RE-PLAN's analysis module is tailored to design and improve POD-based mitigation, related work compares different mediation strategies for specific public health events. The analysis of response plans for attacks with aerosolized smallpox is discussed by Miller et al. [2], comparing different vaccination strategies. Similarly, different emergency responses for airborne anthrax attacks are compared by Wein et al. [3]. Research presented by Kaplan et al. [4] primarily focuses on modeling general vaccination policies of large U.S. cities and estimating the number of deaths. The comparison determines mass vaccination as the most successful vaccination strategy. The success of a response scenario cannot be guaranteed if the POD operation itself is not thoroughly planned. The simulation and decision support system RealOpt focuses on POD layout design, staffing, and clinic design [5] [6]. Standards for mass antibiotic dispensing specifically for POD-based response scenarios have been recommended by RAND [7]. These standards comprise guidelines for the number and locations of PODs, internal POD operation, PODs staffing, as well as POD security. The recommendations have been implemented by various U.S. counties and represent the modus operandi. Implicitly, equal capacity PODs are assumed when generating the number of PODs. For the placement of disaster recovery centers in Florida a two-stage process has been proposed [8]. After suitable locations have been found, realistic locations nearby are identified that satisfy additional criteria. However, the underlying system limits the problem size, which restricts the number of demand points and therefore its granularity. Different aspects of disaster response planning take into account the supply chains of global humanitarian relief organizations [9]. Problems posed by such global scenarios include stocking/inventory, as well as the placement of distribution centers in the relief network.

For the design and analysis of bio-emergency contingency plans underlying geographic information, as well as resource constraints and assumptions by public health experts have to be taken into account. Geographic information is derived from census blocks and the corresponding road infrastructure. Census block data from the Census 2000 can be obtained for all U.S. counties on the U.S. Census website [10]. Detailed road information along with corresponding traffic counts and road capacities may be available from local authorities. If limited data are available, road capacities and traffic counts can be estimated as described in [11]. Constraints include available resources, such as medical staff at the POD locations, as well as Centers for Disease Control and Prevention (CDC) mandated time-frames. In the event of an anthrax attack, the population in the affected geographic region must be supplied with prophylactic medication within 48 hours [12], [13] after an initial setup phase of 12 hours [14]. This timeline is illustrated in Figure 1. Similarly, in the event of a smallpox outbreak the population has to be vaccinated within 72 hours [15], [16], [17]. Note, that for a vaccination scenario the serving time per person at a POD location is longer than for a scenario involving the distribution of medication. This parameter is part of the assumptions made by public health experts. Further, assumptions include the number of individuals traveling in a single car and the existence of base traffic.

The importance of Geographical Information Systems (GIS) for the improvement of health care has been highlighted by McLafferty et al. [18]. RE-PLAN is a GIS-based computational tool for the analysis of bio-emergency response scenarios developed by the Center for Computational Epidemiology and Response Analysis in collaboration with a local health department. It is implemented in Java and utilizes a PostGIS enabled PostgreSQL database, as well as the GIS library GeoTools [19]. RE-PLAN has been tailored to the needs of public health experts by repeatedly incorporating feedback during its development. The underlying census data, as well as road and traffic information are pre-loaded into the database. Disaster

preparedness coordinators can select POD locations via a graphical user interface on the map of their county and evaluate and compare response scenarios. The evaluation comprises bottleneck analyses of the road network and the POD locations. RE-PLAN is currently deployed at a local county's public health department and the development and addition of optimization modules is NIH funded. Optimization in this context refers to the feasible placement of PODs, such that resource constraints are met. Note that RE-PLAN's analysis module does not compute the placement of PODs, but analyzes the quality of a given placement as determined manually by emergency coordinators. A detailed description of RE-PLAN and its response plan analysis core module can be found in [20]. Catchment areas defined by RE-PLAN's analysis module resemble Voronoi tessellations. A Voronoi tessellation assigns each point in space to the closest point of a finite set of locations [21]. For a continuous space, this yields straight lines partitioning the space into k subregions, whereby k is equal to the number of locations. Figure 2 depicts an example of such a Voronoi tessellation. Note that for a continuous space, the points on each line have the same distance to at least two different locations. RE-PLAN uses a similar approach, whereby the discrete set of locations represents the POD locations and the continuous space of demand points is substituted by a discrete set of demand points. The latter corresponds to the census block of the region represented by their geographic centroids.

The analysis of an in-place response scenario may reveal that given the current resources and constraints it is likely to fail, i.e. with the given resource allocation, mandated timeframes cannot be met. Computational techniques can be used to determine a feasible placement of PODs for a given set of resources and underlying constraints. Since resource availability may be limited, public health experts try to determine a POD placement that yields a response scenario with high population coverage while approaching mandated timeframes. The algorithm presented in this paper generates a partitioning of the geographic space, such that the available resources are distributed to yield a balanced population-resource ratio for all PODs. This approach will also yield an even distribution of traffic infrastructure and traffic, since a correlation between road and population densities has been shown [22]. This task resembles problems in Continuous Location Science, whose objective is to place k PODs in a geographic area with a discrete set of demand points. In this paper, the number of PODs is dictated by the available resources, while demand points coincide with census block centroids and represent the number of individuals living in the corresponding census blocks. A solution for one facet of Continuous Location Science, the k -center problem, has been proposed in [23] by utilizing a seed point algorithm. This approach outperforms traditional iterative algorithms. Other algorithmic models to solve different facets of the continuous location allocation problem utilize genetic algorithms [24] and heuristics [25]. While the above algorithms solve the k -center problem by optimizing based on distance, the algorithm presented in this paper yields a homogeneous population partitioning.

This paper is structured as follows: In Section II a partitioning algorithm to determine a feasible resource allocation within a geographic region is presented. The algorithm takes resource availability into account and generates a geographic partitioning such that a balanced population-resource ratio for all POD locations is achieved. In Section II-A a maximum bound for the algorithm in terms of population difference is derived. Experimental results are presented in Section II-B. In Section III limitations of the algorithm are addressed and portability into practice is discussed. Section IV provides a summary of this paper.

II. Optimizing Population Distributions

Providing coverage for a geographic space necessitates the placement of sufficiently many PODs throughout the region, while not exceeding underlying resource capacities. Public health experts compile lists of potential locations that are selected based on their suitability for POD operations. From these lists a sub-set is selected for inclusion in the emergency response plan. While choosing POD locations from a list of preselected possible locations guarantees the placement of PODs strictly at feasible locations, it is essential to address the placement of PODs without constraints in terms of number of locations. This allows for the placement of any number of PODs at any point in the geographic space. The resulting assignment of PODs is an indication of the amount of resources required. Further, it determines areas that should be favored for the POD selection. Continuous models in Location Allocation Modeling do not consider a set of candidate locations, but assume demand points at discrete locations and allow for the placement of the facilities anywhere in the service area. By allowing for an unconstrained placement, limited only by the number of service facilities, a solution that can meet the service standard can be found. Nevertheless, this solution is likely to include locations that are unpractical or infeasible. Such locations may include lakes, mountains, or private and corporate properties. Further, such a solution can result in resource demand that exceeds availability.

The solution of facility placement in a specific region R requires R to be represented as a set of census blocks B . Each census block $b \in B$ is represented by its geographic centroid $centroid(b)$ and has a population size of $pop(b)$. The population of the entire geographic

region R can then be expressed as $pop(R) = \sum_{b_i \in B} pop(b_i)$. For some instance of the problem we consider k PODs (service facilities) that are to be placed within that region, such that the population is equally distributed among all facilities. Table I summarizes the definitions of the variables used in the algorithm described in the following. Algorithm 1 recursively partitions R into k sub-regions (catchment areas) CA , such that $\bigcup_{ca \in CA} = R$. First, the polygon boundary P of R is computed, which is represented as a discrete set of points. Points $p_1, p_2 \in P$ are chosen, such that $\forall p_i, p_j \in P: dist(p_1, p_2) = dist(p_i, p_j)$. For the points p_1 and p_2 corresponding lists L_1 and L_2 are created. At the beginning both lists contain identical elements, as they contain all census blocks $b \in B$ represented by their centroids. The census blocks of each list L_i are sorted in increasing distance with respect to p_i . $L_i(j)$ denotes the j -th element of list L_i . Next, R is partitioned into two regions R_1 and R_2 with $R_1 \cup R_2 = R$ and

$R_1 \cap R_2 = \emptyset$. The ratio of the population of the two regions is $c_1 : c_2$ with $c_1 = \left\lfloor \frac{k}{2} \right\rfloor$ and $c_2 = k - c_1$. This corresponds to the number of PODs each sub-region is assigned at the current recursion level of the algorithm. Each recursive step terminates after all census blocks $b \in B$ have been assigned to either R_1 or R_2 . The census blocks $b \in B$ are sequentially assigned to the region R_i for which $pop(R_i) = \min(pop(R_1), pop(R_2))$. This is achieved by assigning the first census block b_j in the sorted list L_i to its corresponding region R_i . Census block b_j is then removed from both lists L_1, L_2 . This step is repeated until both lists are exhausted. The resulting regions R_1, R_2 define new Polygons P_1 and P_2 . c_1 and c_2 PODs are then assigned to regions R_1 and R_2 , respectively. If $c_i = 1$ the algorithm is recursively executed for R_i with $i \in \{1, 2\}$.

Selecting points p_1 and p_2 on the boundary of the geographic region, such that the distance between them is maximized, avoids the generation of isolated areas during the execution of the Algorithm. If p_1 and p_2 are located close to each other, it is likely that islands are created. This is due to the fact that at each step the closest available centroid is assigned to

one of the points. The closer p_1 and p_2 , the more likely it becomes that they are competing for the same centroid at each of the steps.

The algorithm is implemented in Java and has been integrated into the RE-PLAN framework. Geographic and demographic information are loaded from the underlying database. Via the graphical user interface, preparedness coordinators can select the number of PODs that are to be placed into the underlying geographic space. RE-PLAN's optimization module then generates catchment areas and POD placement by using Algorithm 1. The results are displayed in RE-PLAN as shown in Section II-B.

Although the use of two lists may seem redundant, it should be noted that L_1 and L_2 are not necessarily ordered reversely. This is captured by Lemma 1.

Lemma 1—Lists L_1 , L_2 defined by Algorithm 1 are not always ordered in reverse order for the general case and therefore, both lists are necessary.

Proof: The validity of the Lemma can be shown by the means of the counter example shown in Figure 3. It illustrates an example, for which L_1 is not the same as L_2 read backwards. Assume p_3 and p_4 are points on a circle of radius r_1 with center p_1 . Further, p_3 is a point on a circle of radius r_2 about p_2 . p_5 has a distance $> r_1$ to p_1 and a distance $> r_2$ to p_2 . There are two permissible permutations for L_1 , as p_3 and p_4 both have the distance r_1 to p_1 :

$$L_1\{p_3, p_4, p_5\} \text{ and } L_1\{p_4, p_3, p_5\}$$

Since p_5 is very close to the circle about P_2 of radius r_2 , there is only one permissible permutation for L_2 :

$$L_2\{p_3, p_5, p_4\}$$

None of the permutations of L_1 corresponds to a reversely ordered L_2 , and therefore a single list does not suffice to keep track of the ordered distances to nodes P_1 and P_2 .

A. Maximum Error

The population of a geographic region is distributed across its census blocks B such that the sum of its population is equal to the region's population p . Further, each point in the geographic region, and therefore its population, is assigned to a unique census block:

$$\forall b_i, b_j \in B, b_i \neq b_j: b_i \cap b_j = \emptyset \text{ and } \bigcup_{b_i \in B} b_i = B$$

While a minimum population size b_{min} of 0 is permissible for census blocks, the maximum occurring value $b_{max} = \max_{b \in B}(\text{pop}(b))$ varies among different geographic regions. Let $B_{max} \subseteq B$ denote the set of all census blocks of population size b_{max} . B_{max} contains at least one element and at most all census blocks of the geographic region, i.e. $1 \leq |B_{max}| \leq |B|$.

In the following, we assume the number of PODs k to be a power of 2, i.e., $k = 2^h$ for some h . The following will then generalize to any arbitrary number of PODs. A catchment area $CA_i, i \in [1..k]$ is defined by POD_i and its population is denoted as $\text{pop}(CA_i)$. The maximum observed population difference Δ_{max} between any two catchment areas is defined as follows:

$$\Delta_{max} = \max_{i,j \in [1..k]} |pop(CA_i) - pop(CA_j)|$$

As a consequence of the algorithm, regions do not overlap and each census block is assigned to exactly one of the k PODs. Therefore, the sum of the population in each of the k catchment areas is equal to the population of the entire geographic region:

$$\sum_{i=1}^k pop(CA_i) = p$$

Lemma 2—For the placement of $k = 2$ PODs, a total population size p and largest population size of a census block b_{max} , Δ_{max} is bounded and $\Delta_{max} \in [0..b_{max}]$.

Proof: For the placement of $k = 2$ PODs the geographic region is partitioned into two sub-regions. The lower bound of 0 people difference between both catchment areas is achieved if the population is equally distributed between the PODs, which represents the best case. One instance of such a case is a region consisting of only two census blocks b_1, b_2 with equal population $pop(b_1) = pop(b_2) = \frac{p}{2}$. Algorithm 1 then assigns one census block to each of the PODs, distributing the population equally. Consequently, the resulting partitioning will result in $\Delta_{max} = 0$.

Assume that all census blocks except one census block b_{last} have been assigned to the PODs. This will result in one of the three following cases:

1. $pop(CA_1) < pop(CA_2)$: The population of CA_1 compared to the population of CA_2 is smaller and in the range of $[pop(CA_2) - b_{max}, pop(CA_2) - 1]$. The smallest current population size permissible for CA_1 is $pop(CA_2) - b_{max}$, as b_{max} is the largest possible step size and hence the largest difference that can exist between the two catchment areas at any time during the execution of the algorithm. Algorithm 1 assigns b_{last} to CA_1 , as it has been assigned a smaller population. After the assignment, the population of CA_1 is in $[pop(CA_2) - b_{max} + pop(b_{last}), pop(CA_2) - 1 + pop(b_{last})]$. As $pop(b_{last}) \in [0, b_{max}]$, the extreme cases yield $[pop(CA_2) - b_{max}, pop(CA_2) - 1]$ and $[pop(CA_2), pop(CA_2) - 1 + b_{max}]$. Hence, the maximum possible difference in population size between the catchment areas is b_{max} .
2. $pop(CA_1) > pop(CA_2)$: This case is analogous to $pop(CA_1) < pop(CA_2)$.
3. $pop(CA_1) = pop(CA_2)$: Either of the catchment areas is assigned the last census block b_{last} . Without loss of generality, assume that b_{last} is assigned to CA_1 . With $pop(b_{last}) \in [0, b_{max}]$ the population of CA_1 is then in the range $[pop(CA_2), pop(CA_2) + b_{max}]$. Hence, the maximum possible difference between the catchment areas is b_{max} .

For all of the cases, the maximum possible population difference between the two catchment areas is b_{max} , and consequently, $\Delta_{max} \in [0..b_{max}]$. Having established the maximum difference in population size between two catchment areas, we will need to bound the maximum population difference across the entire partition of the geographic space for $k > 2$ PODs.

Theorem 1—For the placement of $k = 2^h$ PODs, a total population size p and largest population size of a census block b_{max} , Δ_{max} is bounded and $\Delta_{max} \in [0..2b_{max}]$.

Proof: As stated by Lemma 2, the maximum difference of population at each recursion level is b_{max} . During each recursive step, Lemma 2 can be applied repeatedly. The resulting possible maximum and minimum populations for different recursion steps are shown in Figure 4. The left and right child of each node differ by exactly b_{max} individuals. Note that the sum of the population of all nodes at each level is p . The leftmost leaf node of the tree contains the region with the highest possible population size, whereas the rightmost leaf node of the tree contains the region with the lowest possible population size.

The maximum possible population size pop_{max} and the minimum possible population size pop_{min} for a tree of height h are calculated as follows:

$$pop_{max} = \frac{p - b_{max}}{2^h} + b_{max} = \frac{p}{2^h} + \left(1 - \frac{1}{2^h}\right) b_{max}$$

$$pop_{min} = \frac{p}{2^h} - b_{max} \sum_{i=1}^h \left(\frac{1}{2}\right)^i$$

The summation component of pop_{min} is a geometric series [26] and hence, pop_{min} can be expressed as follows:

$$pop_{min} = \frac{p}{2^h} - \left(\frac{\frac{1}{2} \left(1 - \left(\frac{1}{2}\right)^h\right)}{1 - \frac{1}{2}} \right) b_{max}$$

$$= \frac{p}{2^h} - \left(1 - \left(\frac{1}{2^h}\right)\right) b_{max}$$

The maximum difference in population sizes is calculated by subtracting the minimum population size from the maximum population size of a catchment area:

$$\Delta_{max} = pop_{max} - pop_{min}$$

$$= \frac{p}{2^h} + \left(1 - \frac{1}{2^h}\right) b_{max} - \left[\frac{p}{2^h} - \left(1 - \left(\frac{1}{2^h}\right)\right) b_{max} \right]$$

$$= \left(1 - \frac{1}{2^h}\right) b_{max} + \left(1 - \frac{1}{2^h}\right) b_{max}$$

$$= \left(2 - \frac{2}{2^h}\right) b_{max}$$

$$= \left(2 - \frac{1}{2^{h-1}}\right) b_{max}$$

For a large number of PODs k , approximating infinity, the height of the recursion tree $h = \log(k)$ also approximates infinity¹. This may be used to calculate the upper bound for the maximum difference in population sizes between any two catchment areas:

$$\lim_{h \rightarrow \infty} \left(2 - \frac{2}{2^h}\right) b_{max} = 2b_{max}$$

Hence, Δ_{max} is at most $2b_{max}$, which proves Theorem 1.

This result can be generalized to an arbitrary number of PODs, which will be shown via Lemma 3 and Theorem 2. Figure 5 illustrates such a case for $k = 5$. Figure 5(a) shows the portion of the total population of region R ideally assigned at each step to a particular sub-region. During each recursion level, Algorithm 1 assigns a proportion of a particular region to its sub-regions. This is shown in Figure 5(b). Multiplying the proportions of any path

¹All logarithms in this paper refer to logarithms with respect to base 2.

from the root node to any of the leaf nodes yields $\frac{1}{5}$. Thus, the leftmost and the rightmost paths yield $1 \times \frac{3}{5} \times \frac{2}{3} \times \frac{1}{2} = \frac{1}{5} = \frac{1}{k}$ and $1 \times \frac{2}{5} \times \frac{1}{2} = \frac{1}{5} = \frac{1}{k}$, respectively. Lemma 3 generalizes this for the general case.

Assume that at each recursion step, the last census block to be assigned by Algorithm 1 is of size b_{max} and that both sub-regions R_1 and R_2 are of equal population size: $pop(R_1) = pop(R_2)$. Further, assume that the algorithm assigns the census block to the left child. Consequently, the difference in population size between two siblings is always b_{max} . Figure 6 shows the population sizes assigned to each sub-region at a particular recursion level. The proportions by which the population is split is expressed by the coefficients α_i with $i \in [1..h_1]$ and β_j with $j \in [1..h_2]$.

Lemma 3—For coefficients α_i in any path from the root node to any leaf of a tree of height h , the product of the coefficients yields the reciprocal of the number of PODs:

$$\prod_{i=1}^h \alpha_i = \alpha_1 \times \alpha_2 \times \alpha_3 \times \dots \times \alpha_n = \frac{1}{k}$$

Proof: Let $path_i$ denote a particular path from the root node to a specific leaf and let x_1 denote the number of PODs that is assigned to the sub-region that lies on $path_i$ during the first recursion step. Then, ideally $\frac{x_1}{k}$ of the population is assigned to that sub-region. For the next partitioning step, x_1 PODs have to be assigned. Let x_2 out of the x_1 PODs be assigned to the next sub-region on $path_i$. Consequently, $\frac{x_2}{x_1}$ of the PODs at this level are assigned to it. During the last recursion step, $\frac{x_h}{x_{h-1}}$ PODs are assigned. As a leaf node has been reached, only one POD is assigned, and hence $x_h = 1$. Consequently,

$$\frac{x_1}{k} \times \frac{x_2}{x_1} \times \frac{x_3}{x_2} \times \dots \times \frac{x_h}{x_{h-1}} = \frac{1}{k}$$

Theorem 2—For the placement of k PODs, a total population size p and largest population size of a census block b_{max} , Δ_{max} is bounded and $\Delta_{max} \in [0, 2b_{max}]$.

Proof: In the general case, any number k of PODs is permissible. For each split, a coefficient denotes the proportion of the population ideally assigned to a particular sub-region. The goal of the first recursion step in Figure 6 is to assign $\alpha_1 p$ to the left sub-region and $\beta_1 p$ to the right sub-region. Assume that at each recursion level the population differs by b_{max} between the two sub-regions as shown by Lemma 2. The leftmost leaf node in Figure 6 shows the highest possible population size for a catchment area, whereas the rightmost leaf shows the smallest possible population size. The population sizes of the leftmost sub-region is calculated as follows:

$$pop_{max} = (p - b_{max}) \prod_{i=1}^{h_1} \alpha_i + b_{max}$$

Applying Lemma 3 yields

$$pop_{max} = \frac{1}{k}(p - b_{max}) + b_{max}$$

The population sizes of the rightmost sub-region is calculated as follows:

$$pop_{min} = p \prod_{i=1}^{h_2} \beta_i - b_{max} \sum_{i=1}^{h_2} \left(\prod_{j=i}^{h_2} \beta_j \right)$$

This can be simplified by applying Lemma 3:

$$pop_{min} = \frac{1}{k}p - b_{max} \sum_{i=1}^{h_2} \left(\prod_{j=i}^{h_2} \beta_j \right)$$

The right child of a particular node is at most assigned $\frac{1}{2}$ of the population, as the algorithm assigns the larger proportion to the left child. Hence $\beta_j \leq \frac{1}{2}$ with $j \in [1..h_2]$. Therefore follows

$$pop_{max} \geq \frac{1}{k}p - b_{max} \sum_{i=1}^{h_2} \left(\frac{1}{2} \right)^i$$

By applying the geometric series this can be transformed into the following inequality:

$$pop_{min} \geq \frac{1}{k}p - b_{max} \frac{\frac{1}{2} \left(1 - \left(\frac{1}{2} \right)^{h_2} \right)}{1 - \frac{1}{2}} = \frac{1}{k}p - b_{max} \left(1 - \left(\frac{1}{2} \right)^{h_2} \right)$$

The maximum difference in populations sizes is calculated by subtracting the minimum population size from the maximum population size of a catchment area:

$$\begin{aligned} \Delta_{max} &\leq \frac{1}{k}(p - b_{max}) + b_{max} - \left(\frac{1}{k}p - b_{max} \left(1 - \left(\frac{1}{2} \right)^{h_2} \right) \right) \\ &= -\frac{1}{k}b_{max} + b_{max} + b_{max} \left(1 - \left(\frac{1}{2} \right)^{h_2} \right) \end{aligned}$$

Let h_1 denote the height of the left side of the tree, i.e. the length of the path from the root node to the leftmost leaf. Then $h_1 = \lceil \log(k) \rceil$, as Algorithm 1 fills the last tree level, i.e. level h_1 , from left to right. Since the recursion tree is not necessarily complete, the height of the right side of the tree can either be $h_2 = h_1 = \lceil \log(k) \rceil$ or $h_2 = \lfloor \log(k) \rfloor$. If k is a power of 2, it follows that $h_1 = h_2 = \lfloor \log(k) \rfloor = \lceil \log(k) \rceil = \log(k)$. If k is not a power of 2, then let k' denote the largest number smaller than k that is a power of 2. This yields a height $h_2 = \log(k')$. Note that if k approaches infinity, k' also approaches infinity.

Case 1: $h_2 = h_1$ (complete tree)

$$\begin{aligned} h_2 = h_1 = \log(k) &\Rightarrow k = 2^{h_2} = 2^{h_1} \\ \Delta_{max} &\leq -\frac{1}{k}b_{max} + b_{max} + b_{max} \left(1 - \left(\frac{1}{2}\right)^{h_2}\right) \\ &= -\frac{1}{k}b_{max} + b_{max} + b_{max} \left(1 - \left(\frac{1}{2}\right)^{\log(k)}\right) \end{aligned}$$

Case 2: $h_2 < h_1$ (tree not complete)

$$\begin{aligned} h_2 = h_1 - 1 = \log(k') = \lfloor \log(k) \rfloor &\Rightarrow k' = 2^{h_2} = 2^{h_1-1} \\ k' \leq k &\iff \log(k') \leq \log(k) \\ \left(\frac{1}{2}\right)^{\log(k')} &\geq \left(\frac{1}{2}\right)^{\log(k)} \\ 1 - \left(\frac{1}{2}\right)^{\log(k')} &\leq 1 - \left(\frac{1}{2}\right)^{\log(k)} \\ \Delta_{max} &\leq -\frac{1}{k}b_{max} + b_{max} + b_{max} \left(1 - \left(\frac{1}{2}\right)^{h_2}\right) \\ &= -\frac{1}{k}b_{max} + b_{max} + b_{max} \left(1 - \left(\frac{1}{2}\right)^{\log(k')}\right) \\ &\leq -\frac{1}{k}b_{max} + b_{max} + b_{max} \left(1 - \left(\frac{1}{2}\right)^{\log(k)}\right) \end{aligned}$$

For both cases Δ_{max} is bounded by the same expression. If the number of facilities approaches infinity, the following bounds can be determined:

$$\begin{aligned} &\lim_{k \rightarrow \infty} \Delta_{max} \\ &= -\underbrace{\frac{1}{k}b_{max}}_{\rightarrow 0} + b_{max} + b_{max} \underbrace{\left(1 - \left(\frac{1}{2}\right)^{\log(k)}\right)}_{\rightarrow 1} = 2b_{max} \end{aligned}$$

For k approaching infinity, the difference between the upper and the lower bound can be calculated. Hence, the maximum difference in population size between any two catchment areas is $2b_{max}$.

Theorem 3—The deviation of the population p of any catchment area CA_i from the optimal population size p_{opt} is b_{max} .

Proof: The optimal population distribution for k PODs would result in $\frac{p}{k}$ people per catchment area.

Case 1: A catchment area has the largest possible population size. Then the deviation from the optimal size is calculated as

$$\begin{aligned} &p_{max} - p_{opt} \\ &= \frac{1}{k}(p - b_{max}) + b_{max} - \frac{p}{k} \\ &= b_{max} - \frac{b_{max}}{k} \\ &\lim_{k \rightarrow \infty} b_{max} - \frac{b_{max}}{k} = b_{max} \end{aligned}$$

Case 2: A catchment area has the smallest possible population size. Then the deviation from the optimal size is calculated as

$$\begin{aligned} &\leq \frac{p}{k} - \left[\frac{1}{k} p - b_{max} \left(1 - \left(\frac{1}{2} \right)^{h_2} \right) \right] \\ &= b_{max} \left(1 - \left(\frac{1}{2} \right)^{h_2} \right) \end{aligned}$$

a) $h_2 = \log(k)$

$$\lim_{k \rightarrow \infty} \left[b_{max} \left(1 - \left(\frac{1}{2} \right)^{\log(k)} \right) \right] = b_{max}$$

b) $h_2 = \lfloor \log(k) \rfloor$

$$\lim_{k \rightarrow \infty} \left[b_{max} \left(1 - \left(\frac{1}{2} \right)^{\lfloor \log(k) \rfloor} \right) \right] = b_{max}$$

The largest and smallest possible catchment areas have a maximum deviation of b_{max} from the optimal size and therefore, it is true for all other catchment area sizes.

B. Experimental Results

The partitioning algorithm has been applied to Denton County, Texas, sub-dividing the geographic space into three and fifteen regions, respectively. Denton County has a total population of 432,976 people. Figure 7(a) displays the census blocks of Denton County partitioned by the algorithm into three catchment areas. The geographic centroid of each catchment area is assigned a service facility. In Figure 7(b) RE-PLAN has been applied to these particular POD locations yielding Voronoi-based catchment areas, whereas Figure 7(c) shows the results of the response analysis applied to the catchment areas determined by the algorithm. RE-PLAN draws rings around each of the PODs and estimates traffic along ring borders. Intersections between ring borders and the road network are then visualized by color-coded circles. These circles correspond to different levels of emerging traffic normalized taking into account factors, such as estimated number of cars, road capacity and speed limits. The estimation of traffic is beyond the scope of this paper and has been addressed in previous publications [11], [20]. Although Figures 7(b) and 7(c) show strong resemblance, they are not identical. In Figure 7(c) the catchment area around POD 2 is smaller than in Figure 7(b), which indicates that in the lower right corner of the county a high population density prevails.

The population distribution across the catchment areas is shown in Figures 7(d) and Figures 7(e), which confirms the observation that the catchment area sizes of POD 3 differ due to nonuniform population distributions. Since the partitioning algorithm optimizes with respect to the population distribution, Figure 7(e) shows population sizes that are more equally distributed than those obtained by the Voronoi-based plan analysis (Figure 7(d)). The actual population sizes of the catchment areas are compared in Table II. The maximum size of a census block in Denton County is 3931, and therefore Theorem 2 bounds the maximum difference between any two catchment area sizes by 7862 people. The maximum difference

observed for the algorithm with $k = 3$ is 558, which only constitutes 7% of that bound. If for the same set of PODs, however, Voronoi-based catchment areas are calculated, the maximum difference between any two catchment areas results in 197271 people, which is about 25 times the difference by the theoretical bound for the algorithm.

Setting $k = 15$, Denton County is partitioned into fifteen catchment areas (Figure 8(a)). The results of the response analyses on the two different catchment area types are shown in Figures 8(b) and 8(c). The corresponding population distributions are compared in Figures 8(d) and 8(e). Similarly, to the results for $k = 3$, the algorithm yields a more uniform population distribution. The population sizes on a catchment area bases are summarized in Table III.

With a theoretical bound of 7862 and a maximum observed difference in catchment area size of 861 for the algorithm, only about 11% of the bound is reached. This stands in contrast to the Voronoi-based approach with a maximum observed difference of 50528, which is over six times the theoretical bound of the partitioning algorithm. The results indicate that catchment area types that are not Voronoi-based may be an alternative that allows for balancing the number of people across all catchment areas.

III. DISCUSSION

Comparing novel approaches in resource allocation for POD placement to current best practices can serve as an indicator whether the new approach may yield improvements in service outcome. Collaboration with local public health departments has shown that current best practices rely on the selection of POD locations within the geographic space primarily based on the infrastructures of the service locations themselves. Although this approach results in adequately equipped POD locations, it does guarantee that the underlying road infrastructure will suffice the emerging traffic. In particular, PODs with high capacities are prone to producing bottlenecks at their parking lots, entrances, and exits, as well as the areas surrounding them. Further, public health experts have devised plans that underly the assumption that individuals can be assigned to specific PODs. Such an assignment could be done based on zip codes or census blocks, similar to the assignment of the population to voting districts. Further, the POD locations are dynamic and not known by the general public ahead of time. The locations will be announced only if an emergency occurs. Counties may offer resources, such as websites that offer POD lookups based on the street address of an individual. If individuals are not assigned a specific POD, Voronoi-based scenarios would be obtained with unbalanced population distributions.

Applying Algorithm 1 to equally-capacitated PODs yields uniformly distributed demand and hence, automatically distributes the emerging traffic across the different catchment areas. Although the road network of the different catchment areas may vary, our algorithm improves current practices by spatially distributing the population versus only taking into account capacities of the actual service locations. The definition of *optimality* in the context of placing service facilities for disaster mitigation may vary for different types of incidents. For instance, the release of the biological agent anthrax, requires the entire population of the affected region to be provided with prophylactic medication within $t_{max} = 48\text{hours}$. Thus, an *optimal service outcome* is achieved by any response scenario that indicates that this criterion is met: The entire population of the region must be served in $t \leq t_{max}$. In the presence of resource constraints, the window for achieving *optimality* is further narrowed; it might not even be feasible. If increasing resources is not possible, then the *optimality criterion* may need to be relaxed, by minimizing the amount of time Δt , by which t_{max} is exceeded.

Although Algorithm 1 balances the workload uniformly across the PODs, it is likely that portions of the population may need to travel further to reach the POD they are assigned to as opposed to traveling to the closest POD. However, for facility placement in the case of disaster mitigation optimality is not expressed in terms of travel distance per individual, but in overall service time. The latter is dictated by the POD or subset of PODs that take longest to serve their demand. Therefore, in balanced catchment areas some individuals may travel further than their closest POD location, but the entire region will encounter less traffic congestion and balanced service times.

Note that the algorithm yields optimized POD locations by placing the PODs into the geographic centroids of the catchment areas that were generated. There is no guarantee that any of these locations is adequate for POD operation. Based on these locations public health experts may select feasible locations within each of the catchment areas close to the algorithm-suggested locations. In order to include PODs with higher capacities, the algorithm could be modified, such that each POD obtains an Integer-weight describing its capacity. Then coefficients c_1 and c_2 in Algorithm 1 have to be adjusted to reflect the resource distribution of each split. Alternatively, neighboring catchment areas can be merged by removing both PODs and placing a POD in the centroid of the combined catchment area. Nevertheless, changing the assumption that all PODs have equal capacities, can have adverse consequences on the service outcome in terms of traffic volume and bottlenecks around specific pod locations, thereby increasing the serving time t .

IV. SUMMARY

The analysis of an existing response plan may suggest that it will not meet government mandated time-frames, hence, posing the question whether there exists a feasible solution with the given resources. In this paper, an algorithm to determine a POD placement based on available resources is presented. The feasibility of the generated response scenarios is examined based on their population distribution across the different catchment areas.

The continuous partitioning method can be based in the area of continuous location science, since the POD locations can be placed anywhere in the geographic space. The algorithm creates sub-regions within the geographic region for each of the PODs, while maintaining a uniform population distribution. The maximum deviation from the arithmetic average of the population is bounded by the maximum census block size.

The algorithm has been demonstrated authentic census data from Denton County. Results show that the partitioning method creates catchment areas with uniform population distributions. The POD locations are determined by calculating the geographic centroids of the catchment areas. A comparison to catchment areas calculated via the Voronoi-approach on the same set of PODs show that the catchment areas determined by the algorithm provide better population distributions than the Voronoi regions. These results suggest that public health experts may need to consider partitions of the geographic space into catchment areas, that do not resemble Voronoi tessellations. Such an alternate assignment of the population to the service facilities can provide a balanced distribution of resources throughout the target region.

Acknowledgments

This research has been funded by NIH grant NIH 1R15LM010804-01.

References

1. Centers for Disease Control and Prevention. Mass antibiotic dispensing: Streamlining pod design and operations. [Online]. 2005 Apr. Available: <http://www2a.cdc.gov/phtn/antibiotic3/>
2. Miller G, Randolph S, Patterson JE. Responding to bioterrorist smallpox in san antonio. *Interfaces*. Nov; 2006 36(6):580–590.
3. Wein LM, Craft DL, Kaplan EH. Emergency response to an anthrax attack. *Proc Natl Acad Sci USA*. Apr; 2003 100(7):4346–51. [PubMed: 12651951]
4. Kaplan EH, Craft DL, Wein LM. Emergency response to a smallpox attack: the case for mass vaccination. *Proc Natl Acad Sci USA*. Aug; 2002 99(16):10 935–40. [PubMed: 11782544]
5. Lee EK, Chen C-H, Pietz F, Benecke B. Modeling and optimizing the public-health infrastructure for emergency response. *Interfaces*. Sep; 2009 39(5):476–490.
6. Lee EK, Maheshwary S, Mason J, Glisson W. Large-scale dispensing for emergency response to bioterrorism and infectious-disease outbreak. *Interfaces*. Nov; 2006 36(6):591–607.
7. Nelson, C.; Chan, E.; Chandra, A.; Sorensen, P.; Willis, HH.; Comanor, K.; Park, KARH.; Caldarone, L.; Shea, M.; Zambrano, J.; Hansell, L. Recommended infrastructure standards for mass antibiotic dispensing. Santa Monica, CA: RAND Corporation; 2008.
8. Dekle J, Lavieri MS, Martin E, Emir-Frainas H, Francis R. A florida county locates disaster recovery centers. *Interfaces*. 2005; 25(2):133–139.
9. Balci B, Beamon B. Facility location in humanitarian relief. *International Journal of Logistics Research and Applications*. 2008; 11(2):101–121.
10. Census Bureau. Fact Finder. Nov. 2008 [online] <http://www.census.gov/>. [Online]. Available: <http://www.census.gov/>
11. Schneider, T.; Mikler, AR.; O'Neill, M. Computational Tools for Evaluating Bioemergency Contingency Plans. WIT Press; 2009. Disaster Management and Human Health Risk: Reducing Risk, Improving Outcomes; p. 33-44.
12. Centers for Disease Control and Prevention. Key facts about the cities readiness initiative (CRI). Apr. 2008 [Online]. Available: <http://www.bt.cdc.gov/cri/facts.asp>
13. Orange County Health Department. Floridapamphlet. Aug. 2009 p. 1-2.
14. Young, D. CDC rolls out nerve-agent antidote program. Sep. 2004
15. Centers for Disease Control and Prevention. Smallpox response plan and guidelines (version 3.0). [Online]. 2002 Nov. Available: <http://www.bt.cdc.gov/cri/facts.asp>
16. Young, D. Transcript of "smallpox vaccine logistics: Distribution, storage, and security. Aug. 2002 p. 1-5.
17. Young, D. Smallpox response plan: Executive summary. Aug. 2003
18. McLafferty SL. Gis and health care. *Annu Rev Public Health*. Jan; 2003 24(1):25–42. [PubMed: 12668754]
19. Turton, I. Open Source Approaches in Spatial Data Handling. Springer; 2008. p. 153-169.ch. Geo Tools
20. Schneider T, Mikler AR. Re-plan: a computational framework for response plan analysis. *International Journal of Functional Informatics and Personalised Medicine (IJFIPM)*. 2010; 3(2): 103–121.
21. Okabe, A.; Boots, B.; Sugihara, K. Spatial tessellations: concepts and applications of Voronoi diagrams. New York, NY, USA: John Wiley & Sons, Inc; 1992.
22. Tang, J. Evaluating the relationship between urban road pattern and population using fractal geometry. *ucgis.org*. Jan. 2003 [Online]. Available: <http://www.ucgis.org/summer03/studentpapers/junmeitang.pdf>
23. Pelegrin B, Canovas L. A new assignment rule to improve seed points algorithms for the continuous k-center problem. *European Journal of Operational Research*. 1998; 104(2):366–374.
24. Salhi S, Gamal M. A genetic algorithm based approach for the uncapacitated continuous location-.... *Annals of Operations Research*. Jan.2003
25. Wei H, Murray A, Xiao N. Solving the continuous space p-centre problem: planning application issues. *IMA Journal of Management Mathematics*. Jan.2006

26. Cormen, T.; Leiserson, C.; Rivest, R. Introduction to Algorithms. Cambridge, MA: The MIT Press; 1990.

Biographies

Tamara Jimenez received the Diplom in Informatics from the University of Passau, Germany in 2007 and the Ph.D. in Computer Science from the University of North Texas, United States in 2010. After working as a post-doctoral research associate with the Center of Computational Epidemiology and Response Analysis (CeCERA) at the University of North Texas, Dr. Jimenez is currently employed as a Lecturer in Theoretical Computer Science at the University of North Texas. Her research includes the design of algorithms for the analysis and optimization of bio-emergency response plans, as well as the development of methodology for generation of synthetic geographic regions.

Armin R. Mikler received his Diploma in Informatics from Fachhochschule Darmstadt, Germany in 1988. He received a Fulbright Scholarship in 1986 and completed his MS and PhD in Computer Science at Iowa State University in 1990 and 1995 respectively. Since 1997, Dr. Mikler is a faculty member in Computer Science at the University of North Texas (UNT) with joint appointment in the Department of Biological Sciences. Dr. Mikler has established and is the director of the Center for Computational Epidemiology and Response Analysis (CeCERA) at UNT and leads the Computational Epidemiology Research Laboratory (CERL).

Chetan Tiwari received his Ph.D. in Geography from The University of Iowa, USA in 2008. Dr. Tiwari is a faculty member in the Department of Geography at the University of North Texas. He is also affiliated with the Center for Computational Epidemiology and Response Analysis (CeCERA) at UNT. His research interests are in the applications of Geographic Information Science and spatial analysis methods for problems in Public Health.

Fig. 1.
Timeline for the event of a bio-emergency involving anthrax

Fig. 2.
Example of Vornoi tessellation

Fig. 3.
Proof by counter-example

Fig. 4.
Maximum and minimum population sizes at each recursion level

(a) Portion of population

(b) Proportions assigned at each level

Fig. 5.
Example of $k = 5$ PODs

Fig. 6. General case: k PODs while k is not required to be a power of 2

(a) Catchment areas defined by Algorithm 1

(b) Voronoi-based response analysis on PODs determined by Algorithm 1

(c) Response analysis on catchment areas defined by Algorithm 1

(d) Population distribution of Voronoi-based catchment areas

(e) Population distribution of catchment areas determined by Algorithm 1

Fig. 7.
Results of partitioning algorithm with $k = 3$ PODs

(a) Catchment areas defined by Algorithm 1

(b) Voronoi-based response analysis on PODs determined by Algorithm 1

(c) Response analysis on catchment areas defined by Algorithm 1

(d) Population distribution of Voronoi-based catchment areas

(e) Population distribution of catchment areas determined by Algorithm 1

Fig. 8.
Results of partitioning algorithm with $k = 15$ PODs

TABLE I

Variables used in Algorithm 1

B set of census blocks
$b \in B$ a single census block in B
b_{max} population of census block with highest population count
B_i subset of census blocks with $B_i \in B$
R geographic region
P polygon boundary
$p_i \in P$ vertex on polyhon boundary P
P_i polygon boundary of a subregion with census blocks B_i
k number of PODs
L_i list containing cnesus block centroids sorted by increasing distance with respect to a point p_i
$c_1:c_2$ ratio in which the population is to be split
$size(L_i)$ number of elements in list i

Table IIPopulation of catchment areas for $k = 3$ PODS

(a) Voronoi-based		(b) Algorithm 1	
pod	population	pod	population
1	71580	1	144689
2	92545	2	144130
3	268851	3	144157

Table IIIPopulation of catchment areas for $k = 15$ pods

(a) Voronoi-based		(b) Algorithm 1	
pod	population	pod	population
1	24771	1	28803
2	51456	2	29014
3	34658	3	28952
4	9164	4	29175
5	6373	5	28478
6	19559	6	28808
7	55711	7	28802
8	50763	8	28991
9	26096	9	28630
10	29217	10	28836
11	8875	11	28778
12	5183	12	29339
13	51747	13	28928
14	34493	14	28697
15	24910	15	28745

Algorithm 1

Continuous Partitioning Algorithm

Input: A set of census blocks B forming a continuous geographic region R with polygon boundary P and with population $pop(b)$ and centroid $centroid(b)$ for each $b \in B$; number of PODs k

Output: Partition of the region R into k sub-regions such that the population sizes between any two sub-regions differ by at most $2b_{max}$

Select $p_1, p_2 \in P, \forall p_i, p_j \in P: dist(p_1, p_2) \leq dist(p_i, p_j)$

Create list $L_1: dist(p_1, b_{1,1}) \leq dist(p_1, b_{1,2}) \leq \dots \leq dist(p_1, b_{1,|B|}), b_{1,i} \in B, i \in [1..|B|]$

Create list $L_2: dist(p_2, b_{2,1}) \leq dist(p_2, b_{2,2}) \leq \dots \leq dist(p_2, b_{2,|B|}), b_{2,i} \in B, i \in [1..|B|]$

Initialize $R_1 = R_2 = \emptyset$ and $pop(R_1) = pop(R_2) = 0$

Create coefficients $c_1 = \lfloor \frac{k}{2} \rfloor, c_2 = k - c_1$

while $size(L_1) > 0$ **do**

if $c_2 \times pop(R_1) < c_1 \times pop(R_2)$ **then**

$b = L_1(0)$

$R_1.add(b)$

$L_1.remove(b)$

$L_2.remove(b)$

else

$b = L_2(0)$

$R_2.add(b)$

$L_1.remove(b)$

$L_2.remove(b)$

end if

end while

Create Polygons P_1, P_2 containing census blocks in R_1, R_2

if $c_1 > 1$ **then**

 Recursively repeat using $P = P_1, k = c_1$, and $B = R_1$

end if

if $c_2 > 1$ **then**

 Recursively repeat using $P = P_2, k = c_2$, and $B = R_2$

end if