

Spatial Tessellation of Infectious Disease Spread for Epidemic Decision Support

Runsang Liu  and Hui Yang , *Senior Member, IEEE*

Abstract—Infectious diseases such as COVID-19 have severe impacts on both economy and public health in the US and the world. Due to the heterogeneity of virus spread, there are spatial variations in the demand for medical resources such as personal protective equipment (PPE), testing kits, and vaccines. The availability of such medical resources is critical to effective epidemic control. Although these resources can be readily transported to designated areas for fighting an epidemic, the demand is increasing and varying in space that places significant stress on the supply and allocation of medical resources. However, little has been done on the tessellation of infection distributions for resource management. In this letter, we develop new tessellation algorithms for decision support in epidemic resource allocation and management. The objective is to estimate resource locations and coverage based on the spatial analysis of heterogeneous infection distribution. First, spatial tessellation centroids are initialized through either greedy or cluster-centric approaches. Next, the locations of tessellation centroids are calibrated through a gradient learning algorithm. Lastly, the spread tessellation is computed to provide an estimation of resource coverages under the heterogeneous infection distribution. The proposed methodology is evaluated and validated using a COVID-19 case study of infection data in Pennsylvania. Experimental results show the proposed methodology effectively tessellates the spread of infectious diseases. The new spread tessellation algorithms are shown to have strong potentials for epidemic decision support in infection modelling and resource allocation.

Index Terms—Agent-based systems, decision making, spatial analysis, resource allocation, infectious diseases.

I. INTRODUCTION

THE outbreak of infectious diseases can cause severe impacts on both the economy and public health. Recently, Coronavirus Disease 2019 (COVID-19) has been spreading throughout multiple countries and is declared as a pandemic by the World Health Organization (WHO) in March 2020. The disease spreads quickly and heterogeneously in space, leaving little time for healthcare professionals to react. As shown in Fig. 1, there are only a few cases in the US in March, but the disease has spread to almost all major cities on the east and west coast in less than one month. The situation keeps deteriorating

in the following months and has become a pandemic, causing lockdowns and a high mortality rate for aged individuals [1].

Due to the heterogeneity of virus spread, there are spatial variations in the demand for medical resources such as personal protective equipment (PPE), testing kits, and vaccines. PPEs such as facemasks and protective coveralls protect healthcare professionals and individuals from infection. Testing kits help identify positive cases, thereby enabling tracing protocols to slow the spread of the disease. Vaccines help develop immunity and thus reduce the risk of infection. The availability of such medical resources is critical to effective epidemic control. Although these resources can be readily transported to designated areas for fighting the epidemic, the demand is increasing and varying in the space that places significant stress on the supply and allocation of medical resources [2].

The complexity of infection and demand distribution poses great challenges to infection modeling and resource management during an epidemic. As shown in Fig. 2, the spread of COVID-19 in the state of Pennsylvania has large spatial variability. The number of infections is encoded by a colormap, where red and blue color represents more and less infected cases. Such spatial variability of infected cases over the region is referred to as heterogeneous infection distribution. While the demand for medical resources is closely related to the spread of an epidemic, such complexity of the infection distribution poses great challenges to resource decision making because directly allocate resources to highly infected areas will create large disparities across the spatial region. Therefore, spatial analysis should be used to support the resource allocation process.

Traditionally, spatial analysis of resource allocations is performed using locational optimization models such as the p-median model [3] and set covering model [4]. Locational optimization models often use integer programming to find an assignment of demand locations to resource locations by minimizing the sum of weighted distances, which is computationally expensive. The infection map creates a large-scale decision space and thus discrete location models are less favored under this setting. On the other hand, although traditional clustering and Voronoi methods are computationally efficient, they cannot be directly applicable due to the lack of considerations about both demand locations and the heterogeneous infection distribution. As a result, there is an urgent need to account for both heterogeneous infection distribution and large-scale problem settings.

In this letter, we develop new tessellation algorithms for decision support in epidemic resource management. The objective is to estimate resource locations and coverage based on the spatial analysis of heterogeneous infection distribution.

Manuscript received July 14, 2021; accepted November 22, 2021. Date of publication December 1, 2021; date of current version December 9, 2021. This work was supported by NSF IIP under Grant 2026875 RAPID COVID19. This letter was recommended for publication by Associate Editor N. Kong and Editor J. Yi upon evaluation of the reviewers' comments. (*Corresponding author: Hui Yang.*)

The authors are with the Complex Systems Monitoring, Modeling and Control lab, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: rzl275@psu.edu; huy25@psu.edu).

Digital Object Identifier 10.1109/LRA.2021.3131699

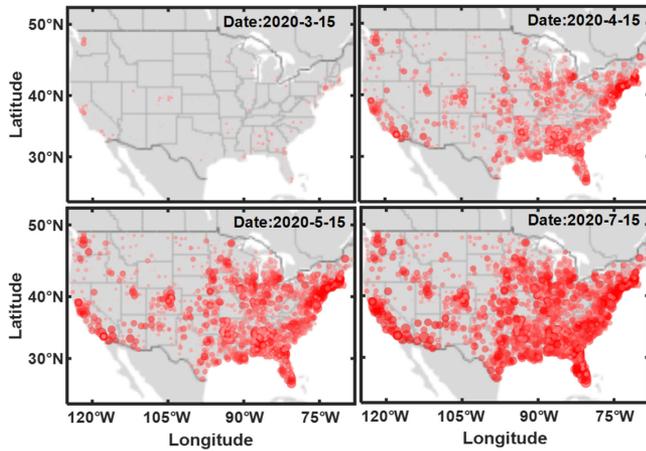


Fig. 1. COVID-19 infection map of the United States in a timeline.

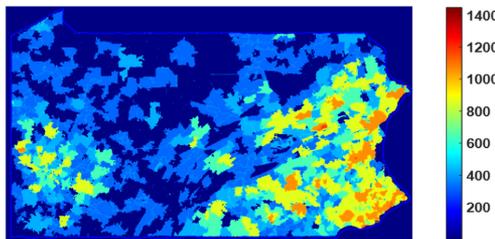


Fig. 2. COVID-19 infection map of Pennsylvania.

First, spatial tessellation centroids are initialized through either greedy or cluster-centric approaches. Next, the locations of tessellation centroids are calibrated through a gradient learning algorithm. Lastly, the spread tessellation is computed to provide an estimation of resource coverages under the heterogeneous infection distribution. The proposed methodology is evaluated and validated using a COVID-19 case study of infection data in Pennsylvania. The proposed tessellation algorithm is benchmarked with modified Voronoi tessellation and 3D K-means clustering algorithm. Experimental results show the proposed sequential optimization framework is effective to tessellate the spread during the pandemic.

Furthermore, as the number of infections is changing over time during an epidemic, spread tessellation should also be adjusted according to the new infection data. As shown in Fig. 3, the proposed tessellation algorithm can be integrated into a decision support framework for mobile resource allocation. The infection data in the region of interest are acquired to monitor and update the spatial distribution of infected cases. Such infection distributions are fed to the proposed algorithm to tessellate infection regions for the decision support of optimal intervention policies (e.g., resource allocation), which is critical for the coverage and control of the epidemic. While the infection data are updated each day, the tessellation is also re-computed to cope with the latest situation. The proposed methodology is flexible to support decision-makings during an epidemic.

The rest of the letter is organized as follows: In Section II, we discuss the research background and the concept of spatial tessellation. Section III shows the formulation of the proposed methodology. We detail the experimental design and results in

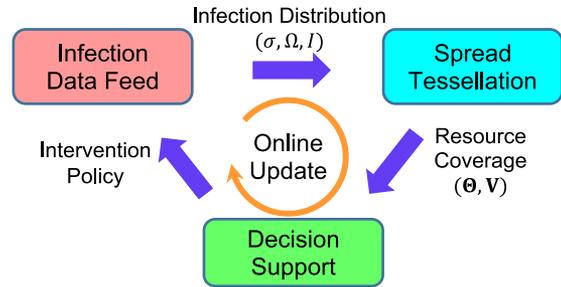


Fig. 3. Spread tessellation for decision support during an epidemic. The infection data are acquired to monitor and update the spatial distribution of infected cases. Such distributions are then processed by the spread tessellation algorithm, which derives tessellation regions for decision makers to optimize intervention policies (e.g., resource allocation) towards the coverage and control of infectious disease spread.

Section IV and Section V. The closing remarks are encompassed in Section VI.

II. RESEARCH BACKGROUND

Locational optimization models are originated where industries need to leverage the locations of production warehouses, distribution centers, and retail outlets [5]. In healthcare systems, locational optimization models can be used to find an assignment of medical centers during disasters [6], management of medical resource supply [7], localization of public health services [8], and analysis of optimal organ allocation boundaries [9].

A. Discrete and Continuous Locational Optimization

The p -median model is a discrete location model that originally aims to find the allocation of resources with the lowest cost—a total of p resource locations is determined such that the average of weighted distances between demand locations and supply locations is minimized [10]. The modified p -median model formulates the discrete location problem with integer programming and uses simulated annealing metaheuristics to find the optimal health center locations [11]. The large-scale Emergency Medical Service (LEMS) framework [3] also utilizes an integer programming framework that can be translated into classical models such as the maximum coverage model and set covering model.

However, the optimization process of integer programming models is computationally expensive. Increased dimensionality not only places computational burdens on the optimization process, but the complicated allocation policy is also hard to implement. On the other hand, there are fast algorithms based on metaheuristics (e.g., genetic algorithm [12]). Although integer programming problems may be complex, most of them are not computationally expensive due to the use of metaheuristics and matheuristics. However, discrete location models only compute locations on networked nodes and are limited in the ability to tessellate infection spread when the number of infected cases is densely distributed over the region.

The proposed work consists of a continuum approximation (CA) model in two-stage algorithms for handling the computational complexity of discrete facility location problems. CA models rely on distribution functions instead of exact locations

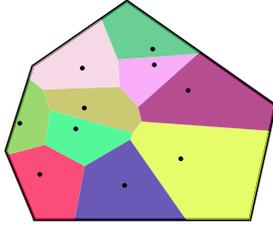


Fig. 4. 2-D Voronoi tessellation with 10 regions.

to optimize the cost objective. Therefore, they provide easier and faster implementations than exact location models in the presence of spatial densities [13]. Ouyang and Daganzo proposed a CA-based algorithm to find terminal locations and their coverage to minimize the total logistic cost by sliding influence areas within the service region [14]. Nonetheless, the spread of COVID-19 is different from slowly varying distributions of demands considered in the literature, but rather brings highly heterogeneous distribution of infected cases in a large spatial region. Therefore, new algorithms that can handle heterogeneous infection distributions are urgently needed.

B. Clustering and Voronoi Tessellation

Traditional clustering algorithms such as K-means clustering and variable clustering [15] group a set of data points into homogeneous subsets based on a predefined similarity function. Data points are “closer” to each other within the same cluster than those in other clusters. Also, traditional clustering methods cannot be directly used in spread tessellation because not only spatial locations are involved, but also the infection distribution should be considered, where the correlation between the number of cases and 2-D spatial locations cannot be readily estimated [16]. Notably, instead of using correlation-based metrics for clustering, it is not uncommon that distance measures (e.g., Euclidean distance, and Manhattan distance) can be used to compute the cluster of data points. These clustering methods focus more on the spatial relationship between two data samples, but are less concerned about the heterogeneous distribution of infected cases over spatial locations. Through our experiments, direct usage of the infection data as the third dimension may not generate desired results.

Traditional Voronoi tessellation computes partitions in a spatial domain using Lloyd’s algorithm. The Voronoi tessellation is a partition of space into several sub-regions, where each region is spawned by a Voronoi center θ . Anywhere within the region satisfy the following inequality:

$$\text{dist}(p, \theta_i) \leq \text{dist}(p, \theta_j), \forall p \in V_i, i \neq j \quad (1)$$

where p is a location inside the i^{th} Voronoi region V_i . Equation (1) indicates that anywhere within a Voronoi region has a smaller distance to its center than centers in other Voronoi regions. Common distance functions include 1-norm (Manhattan distance) and 2-norm (Euclidean distance).

As shown in Fig. 4, a convex polygon can be divided into several Voronoi regions. Note that any location from a color-coded region has the smallest distance to its corresponding Voronoi center than other centers. This indicates if we consider Voronoi centers to be the location of resources, the spatial locations can be partitioned into a number of sub-areas that satisfy this property.

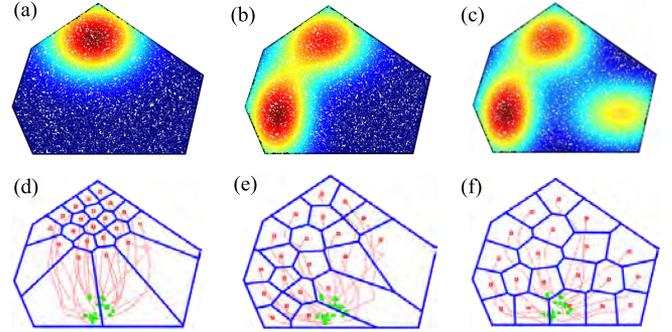


Fig. 5. Illustrations of spread tessellation with weight using different complexity levels of infection distributions, a) One Gaussian distribution; b) two Gaussian mixtures and; c) three Gaussian mixtures; d)-f) are tessellation results of traditional Voronoi tessellation with weights for Gaussian mixtures in a), b) and c), respectively.

Therefore, spatial tessellation is conducive to infection modeling and resource allocation.

However, traditional Voronoi tessellation only partitions a polygonal space into regions based on spatial locations but does not account for the number of infections. In other words, the operation can only be done in the same dimension, e.g., from $\mathbb{R}^2 \rightarrow \mathbb{R}^2$, while we want an assignment of \mathbb{R}^3 (i.e., with the added dimension of infection distribution) to \mathbb{R}^2 . Thus, the consideration of infection distribution is needed in the design of tessellation algorithms for partitioning the spread regions of an infectious disease (e.g., COVID-19).

C. Traditional Voronoi Tessellation With Weights

The infection distribution can be incorporated into the objective function of the tessellation algorithm as the weight of the distances between spatial locations and centroids. The optimization proceeds by updating the centroids of the tessellations to the center of mass of the Voronoi region [17]. Fig. 5 shows the optimization results with different levels of spatial variability (complexity) of infection distributions, i.e., single Gaussian distribution (low), two Gaussian mixtures (medium), and three Gaussian mixtures (heterogeneous). Red and blue color indicates a higher and lower number of infected cases. Green and orange squares are initial and final locations of tessellation centroids. The trajectory shows the optimization process of tessellation centroids.

As shown in Fig. 5, when the spatial variability of the infection distribution is low and medium, more regions are allocated near highly infected areas. However, when the complexity is heterogeneous, traditional Voronoi tessellation with weights tends to partition these regions uniformly. This is because the optimization is stuck at a local minimum. A high level of spatial variability of the infected cases poses significant challenges to the ability of the existing algorithms to handle heterogeneous infection distributions. It may be noted that there are existing Voronoi tessellation methods that also consider spatial heterogeneity. For example, heterogeneous Voronoi tessellation (HVT) uses dynamic programming to estimate the shortest heterogeneous distance for emergency drone delivery from fixed agencies to demand locations [18]. This present investigation focuses on spread tessellation algorithms for decision support of mobile resource allocation, where the

location of tessellation centroids can vary as the epidemic evolves.

III. RESEARCH METHODOLOGY

In this section, we propose two spread tessellation algorithms to handle heterogeneous infection distribution for resource decision support during an epidemic. Because the spatial analysis of infection distribution is conducive to infection modeling and resource allocation, we study the spread tessellation on an infection map. Given the number of partitions, two proposed algorithms both follow a sequential optimization framework—in the first stage, initial locations of tessellation centroids are roughly estimated using either greedy or cluster-centric approach. In the second stage, all tessellation centroids are being moved through a gradient learning algorithm. The partition is further being updated while the locations of tessellation centroids are optimized, and the stable tessellation is obtained when all locations of centroids are converged. The proposed methodology is evaluated and validated with both simulated data and a real-world case study with COVID-19 infection data in Pennsylvania.

A. Weighted Spatial Tessellation Formulation

Let $\sigma(s)$ be an injective mapping function: $\Omega \mapsto \mathbb{R}^+$ that maps a spatial location s inside a convex polytope space Ω to a real value (the number of infected cases at a location s) and $\Theta = \{\theta_1, \theta_2, \dots, \theta_i\}$ be the set of locations of tessellation centroids, each is allowed to move inside the convex region of Ω . Consider $dist(\theta_i, s)$ to be the distance function between a centroid location θ_i and a spatial location s . Here we define accessibility to be the sum of weighted distances between spatial locations and tessellation centroids. Therefore, the cost function is formulated as follows:

$$\operatorname{argmin}_{\theta} C(\Theta) = \sum_{i \in I} \int_{V_i} dist(\theta_i, s) \sigma(s) ds \quad (2)$$

where V_i is the i^{th} Voronoi region, θ_i is the i^{th} Voronoi center, I is the number of partitions. Here, we use the squared Euclidean distance to be our distance function, i.e., $dist(\theta_i, s) = \|\theta_i - s\|^2$. Given the properties of Voronoi tessellation, it may be noted that $\min_k \|\theta_k - s_j\|^2 \sigma(s_j) = \|\theta_i - s_j\|^2 \sigma(s_j)$ for all $s_j \in V_i$. This indicates the minimum weighted distance between a spatial location and a centroid can only be found as the Voronoi center of that region. It is worth noting that this formulation is different from traditional clustering because both distance functions and the number of infections are taken into consideration.

Because the Voronoi tessellation can be treated as a continuous function of the tessellation centroid location set $\Theta = \{\theta_1, \theta_2, \dots, \theta_i\}$, the objective function $C(\Theta)$ is continuously differentiable with respect to centroid location θ_i . Thus, we can find the partial derivatives with respect to the location of the tessellation centroid θ_i as:

$$\frac{\partial C}{\partial \theta_i} = \int_{V_i} \frac{\partial}{\partial \theta_i} \|\theta_i - s\|^2 \sigma(s) ds \quad (3)$$

Because the integral and summation are both convex preserving operations, and the squared Euclidean distance function is convex, thus the resulting objective function is also convex.

Therefore, the objective function converges to a minimum location where the partial derivative vanishes. We can further compute the partial derivatives to simplify the gradient. Recall that the center of mass of the i^{th} Voronoi region can be computed as:

$$CM_i = \frac{1}{\int_{V_i} \sigma(s) ds} \int_{V_i} s \sigma(s) ds \quad (4)$$

The moment of inertia (MOI) of the i^{th} region is defined as:

$$I_i = \int_{V_i} \|\theta_i - s\|^2 \sigma(s) ds \quad (5)$$

According to the parallel axis theorem:

$$I_{V_i} = I_{V_i, CM_i} + \int_{V_i} \sigma(s) ds \|\theta_i - CM_i\|^2 \quad (6)$$

which indicates the MOI of the i^{th} Voronoi region can be divided into two parts: the MOI associated with the center of mass I_{V_i, CM_i} , and the parallel axis passing through the center of mass. See more details about the parallel axis theorem in [19]. Using equation (6), the partial derivatives can be computed as:

$$\frac{\partial C}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left(\sum_{i \in I} I_{V_i, CM_i} + \sum_{i \in I} \int_{V_i} \sigma(s) ds \|\theta_i - CM_i\|^2 \right) \quad (7)$$

$$= 2 \int_{V_i} \sigma(s) ds (\theta_i - CM_i) \quad (8)$$

which indicates the gradient direction of Voronoi center points at the center of mass of that Voronoi region. To minimize sum of weighted distances, a gradient learning algorithm is developed to update Voronoi centers as:

$$\theta_i(t+1) = \theta_i(t) - \alpha \frac{\partial C}{\partial \theta_i} \quad (9)$$

where t is the current step, and α is the learning rate. During each step, the tessellation is also updated using new Voronoi centers. When Voronoi centers are moved towards centers of mass of their corresponding region, new Voronoi centers will have closer spatial distances to locations with more infections, and greater distances from less infected areas. Therefore, highly infected areas will be covered by more Voronoi regions, and areas with less infection will be covered by fewer regions. Through our experiments, optimizing such spatial tessellation purely relies on gradient descent tends to be limited in the ability to handle heterogeneous infection distributions. Therefore, two sequential optimization algorithms are proposed to tackle this issue in the next section.

B. Proposed Tessellation Algorithms to Address the Heterogeneous Infection Distribution

Greedy-Voronoi tessellation (GVT): The greedy-Voronoi tessellation first estimates the locations of partition centroids one at a time until the number of partitions is equal to a fixed value I . Next, all locations of the centroids are calibrated to find the optimal spread tessellation.

As shown in Table I, the greedy Voronoi tessellation is executed in two stages—sequential optimization and global calibration. The sequential optimization stage divides the original problem into sub-problems and makes the optimal decision at each sub-problem. As such, the GVT finds the best tessellation

TABLE I
PSEUDOCODE FOR GREEDY-VORONOI TESSELLATION (GVT)

Infection mapping function $\sigma(s)$, Polygon space Ω , number of partitions I	
1:	Place the first centroid θ_1 at the center of mass of the Ω with density $\sigma(s)$
2:	For $i = 2$ to I
3:	Randomly place a new centroid θ_i in the Voronoi region with the largest mass
4:	Compute Voronoi tessellation \mathbf{V} based on the location of current centroids using equation (1)
5:	Compute the cost $C(\Theta) = \sum_i \int_{V_i} dist(\theta_i, s) \sigma(s) ds$
6:	Compute the gradient $\frac{\partial C}{\partial \theta_i}$ for this newly added centroid θ_i
7:	Update θ_i according to $\theta_i = \theta_i - \alpha \frac{\partial C}{\partial \theta_i}$
8:	Repeat 5-7 until convergence
9:	Update Voronoi tessellation \mathbf{V}
10:	End For
11:	Compute the cost function $C(\Theta) = \sum_{i=1}^I \int_{V_i} dist(\theta_i, s) \sigma(s) ds$
12:	Compute the gradient $\frac{\partial C}{\partial \theta_i}$, $i = 1, 2, \dots, I$ for all centroid locations
13:	Update all θ_i 's according to $\theta_i = \theta_i - \alpha \frac{\partial C}{\partial \theta_i}, i = 1, 2, \dots, I$
14:	Update Voronoi tessellation \mathbf{V}
15:	Repeat 11-14 until convergence
16:	Return spread tessellation under tessellation centroid locations $\theta_i, i = 1, 2, \dots, I$

centroid location one at a time. After placing an initial centroid at the center of mass in the polygon space, the location of the next centroid is randomly placed inside the Voronoi region with the largest total sum of weighted distances, and a new Voronoi tessellation and cost are computed. The sequential optimization proceeds by moving a newly placed tessellation centroid towards its gradient direction until the variation of the location are converged. Note that during this process the locations of all other centroids are fixed. This placement and refinement process repeats until all I tessellation centroid locations are estimated.

After that, the global calibration stage is initiated where all centroids are free to move. The cost and gradient for all centroids are computed, and the corresponding locations are updated according to step 13. With each movement of the centroid, the tessellation is also updated. This process terminates when the locational differences of tessellation centroids are converged, and the cost function is on a plateau. Upon convergence, the algorithm returns a tessellation of the infection distribution.

Cluster-Voronoi tessellation (CVT): The cluster-Voronoi tessellation first utilizes clustering methods to estimate initial tessellation centroid locations. Next, these locations are calibrated using a gradient learning algorithm to find the optimal tessellation of the infection distribution.

As shown in Table II, the infection distribution inside polygon space Ω is first being filtered using a threshold parameter ξ , which can be determined empirically. This step is to filter out areas with the number of infections below a certain level (i.e., set the number of infections to zero) and prioritize highly infected areas. For example, using $\xi = 0.5$ will filter out regions with

TABLE II
PSEUDOCODE FOR CLUSTER-VORONOI TESSELLATION (CVT)

Infection mapping function $\sigma(s)$, Polygon space Ω , number of partitions I	
1:	Filter the infection distribution with threshold ξ
2:	For $i = 1$: t
3:	Randomly choose I locations as the initial cluster centers
4:	(Re)assign each spatial location in Ω to the cluster where the cluster center is the closest
5:	Update the location of the cluster center as the mean of the spatial location within
6:	Repeat 3-5 until convergence
7:	End for
8:	Assign the initial location of the tessellation centroids to be the mass center of the I clusters
9:	Compute Voronoi tessellation \mathbf{V} based on the location of I centroids
10:	Compute the cost function $C(\Theta) = \sum_i \int_{V_i} dist(\theta_i, s) \sigma(s) ds$
11:	Compute the gradient $\frac{\partial C}{\partial \theta_i}$, $i = 1, 2, \dots, I$ for all centroid locations
12:	Update all θ_i 's according to $\theta_i = \theta_i - \alpha \frac{\partial C}{\partial \theta_i}, i = 1, 2, \dots, I$
13:	Repeat 9-12 until convergence
14:	Return spread tessellation under tessellation centroid locations $\theta_i, i = 1, 2, \dots, I$

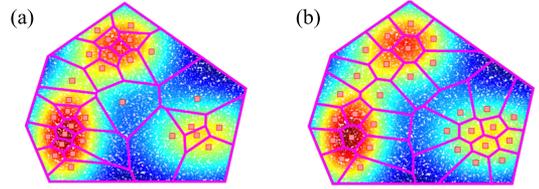


Fig. 6. Illustration of tessellating the spread into 30 regions on the simulated heterogeneous infection distribution in Fig. 5 using a) GVT and b) CVT.

the number of infected cases that are less than average. Next, a total number of I tessellation centers are initialized in the polygon space and spatial locations are assigned to its nearest cluster centroid based on their spatial closeness. After that, we iteratively move the location of cluster centers by averaging the number of infections within the cluster until the sum of distances between the spatial location and cluster center are minimized. These cluster centers are treated as initial locations of tessellation centroids. Next, the Voronoi tessellation is computed given current centroid locations, and the centroid locations are updated using the gradient descent method. Each movement of the centroid locations will result in another tessellation in the polygon space. Upon convergence, we can obtain the spread tessellation of the epidemic.

The proposed method is first examined with a simulated infection distribution. Three Gaussian mixtures are simulated as the underlying infection distribution inside the polygon space. As shown in Fig. 6, both GVT and CVT places tessellation centroids densely near the peak of infections and sparsely over regions with a lower number of infections. The proposed methodology shows promises to handle the simulated heterogeneous infection distribution. In the next section, we conducted a case study using

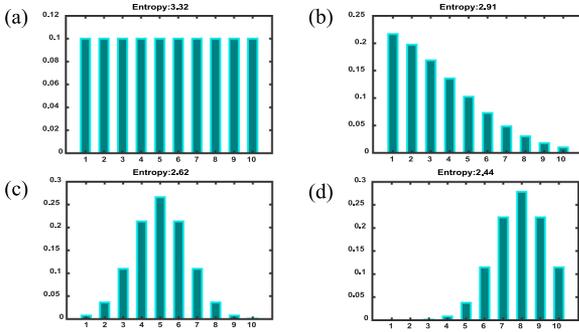


Fig. 7. Entropies of a) uniform distribution, b) normal distribution centered at 0, c) normal distribution, and d) skewed normal distribution.

COVID-19 infection data in Pennsylvania to further investigate and evaluate the performance of proposed algorithms.

IV. EXPERIMENTAL DESIGN AND RESULTS

A. Case Study on COVID-19 Infection Map

The proposed methodology is evaluated using a case study of COVID-19 infection data from the Pennsylvania Department of Health website as shown in Fig. 2. Red color indicates more cases of infections, while blue color means fewer cases. The pixel locations of the infection map are set to be infection locations and decision space for tessellation centroid locations to create a large-scale problem setting. The total number of infection locations are exceeding 860000, where algorithms with poor scalability certainly encounter issues in computational tractability.

B. Performance Measures

The performance of the proposed methodology is evaluated using the accessibility score, and equity score:

- 1) **Accessibility score:** We compute the sum of distances between spatial locations to centroid locations with the weight of the number of infected cases: $\sum_{i \in I} \int_{V_i} dist(\theta_i, s) \sigma(s) ds$. The s here is the spatial locations, with the corresponding number of infected cases $\sigma(s)$. The θ_i 's are the locations of tessellation centroids and V_i is the i^{th} Voronoi region generated using θ_i . The **Accessibility score** is the reciprocal of the sum of weighted distance. As such a higher accessibility score means the algorithm is better at tessellating the spread of an epidemic.
- 2) **Equity score:** Equity measures the fairness of infection severities among partitions within the region. To quantify this concept, we use the entropy of the covered cases within the Voronoi regions to be the measure of equity.

Entropy is a measure of information content in signals $Entropy = -\sum_i p_i \log(p_i)$, where p_i here is the probability that element i can be observed. As shown in Fig. 7, the entropy varies when the data follow different distributions. When the data follows a uniform distribution, it has the highest entropy. When the data follows normal distributions with different mean and standard deviation, we can observe that entropy is correlated

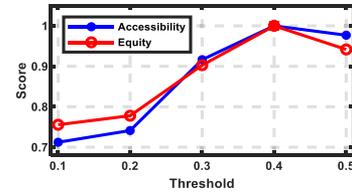


Fig. 8. Variations of accessibility and equity scores of the CVT algorithm when the threshold is increased from 0.1 to 0.5.

with the variation of the data. The larger the variation in distribution, the smaller the entropy. To evaluate the spread tessellation, p_i is computed as the normalized accessibility score in the i^{th} region.

The **Accessibility score** and **Equity score** are further normalized to be in the range between 0 and 1, for the sake of comparisons. We evaluate the performance of two proposed algorithms using both scores by varying the parameters, i.e., the threshold ξ for the CVT, and the number of partitions I . Tessellation results are compared when the threshold is varied from 0.1 to 0.5. Next, we choose the best performing threshold for CVT in the following experiments. After picking the best threshold ξ , we benchmark both GVT and CVT algorithms with traditional Voronoi tessellation with weights and 3D K-means clustering with a varying number of partitions $I = 40, 60, 80, 100$ and 120 .

C. Experimental Results

The CVT algorithm leverages the threshold to prioritize highly infected areas. Therefore, the choice of ξ will impact the performance of CVT algorithm, but not for the GVT algorithm since it does not have the threshold parameter. If the threshold is too large, the initialization of the tessellation centroids is too radical, and if the threshold is too small, the optimization will be trapped at a local minimum. Thus, tuning the threshold is the first step to achieve better performance for the CVT algorithm.

Fig. 8 shows the comparison of tessellation results when the threshold is varied among 5 levels ($\xi = 0.1, 0.2, 0.3, 0.4$, and 0.5) to identify the optimal threshold for the PA infection map with 100 tessellations. Also, the number of tessellations can be flexibly adjusted in our proposed algorithms to support the decision-making process of policy makers. Experimental results show that the CVT yields the best performance at $\xi = 0.4$ in terms of both accessibility and equity scores. Therefore, we choose the optimal threshold to be 0.4 for the CVT algorithm in the following experiments.

Fig. 9 shows the comparison among GVT, CVT, as well as Voronoi tessellation with weights and 3D K-means. The 3D K-means algorithm considers both spatial coordinates and the number of infected cases at a location s when computing the Euclidean distance as the similarity measure. The number of tessellations is set as parameter K . In the practice, parameter K is often predetermined empirically by the policy maker, e.g., the number of resource locations under the budget constraint to maximize the coverage and control of COVID-19 spread. In this investigation, the value of K is set as the number of tessellations for the consistence of performance comparison with

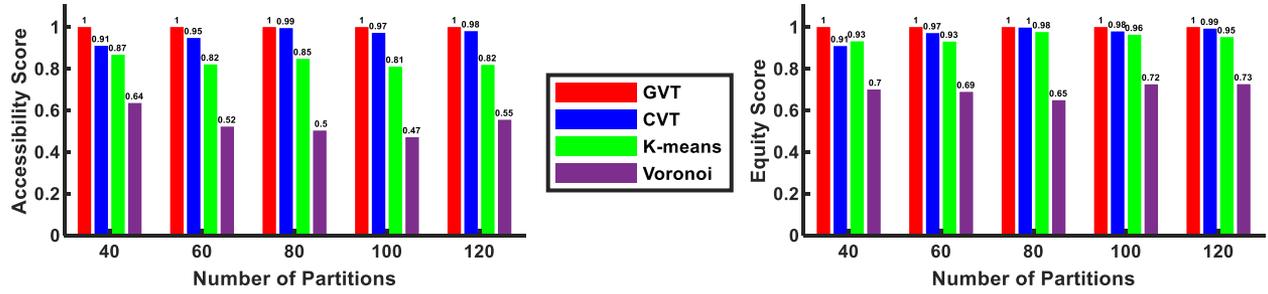


Fig. 9. Performance comparison of four methods (i.e., GVT, CVT, traditional 3-D K-means, and traditional Voronoi tessellation with weights) when the number of partitions is increased from 40 to 120.

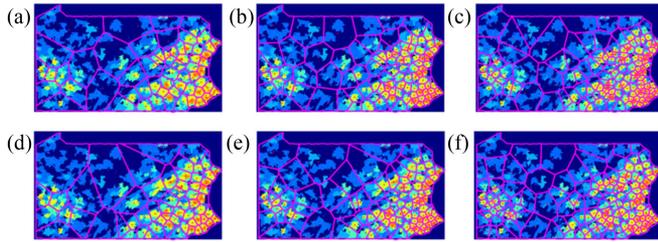


Fig. 10. Spread tessellation results with the GVT (a, b, and c) and CVT (d, e, and f) when the number of partitions is varied from (a and d) 60 to (b and e) 100 and (c and f) 140.

other algorithms. On the other hand, if the objective is to identify the best parameter K automatically, there is a need to set up a new objective function and screen the possible range of K for the optimality. Experimental results show that the proposed GVT and CVT algorithms yield significantly better results than traditional algorithms when the number of partitions is increased from 40 to 120. The traditional Voronoi tessellation yields the lowest accessibility and equity score among all four methods. Although 3D K-means has a decent equity score comparing with other methods, its accessibility scores are not satisfactory. This is because traditional clustering methods focus more on the spatial relationship between data samples, while concerning less on the distribution of infections. The GVT yields the best performance for different numbers of partitions because optimal learning is performed on every sub-problems. On the other hand, the CVT is better in terms of computational time. The runtimes for the CVT with 40, 60, 80, 100, and 120 partitions are 132.96, 150.13, 156.62, 171.55, and 179.43 seconds, while the GVT requires 654.11, 1290.56, 2117.52, 3019.04, and 3971.41 seconds, respectively. The computation time is estimated with the use of a laptop computer with Intel Core i7 2.60GHz, 16GB RAM. Because all tessellation centroids are initialized at the same time, the CVT has the advantage of computation efficiency over the GVT. Moreover, when the number of partitions is increased, the difference in the performance measures between GVT and CVT algorithms is decreasing. In our experiment, when the number of partitions is large, CVT can be used as a fast alternative to GVT.

Fig. 10 shows the comparison of spread tessellation results using the GVT and CVT algorithms when the number of partitions is varied among 60, 100, and 140. Note that more regions are obtained near highly infected areas for both approaches but

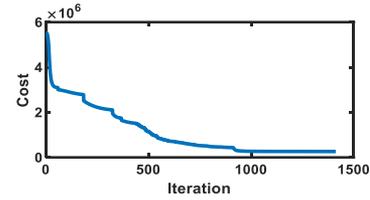


Fig. 11. Convergence curve of the cost function for 100 partitions using the greedy-Voronoi tessellation (GVT) algorithm.

are much sparser in areas with fewer infected cases. It may also be noted that tessellation results are slightly different between the two proposed algorithms, achieving different levels of performance scores. The GVT yields the accessibility of 0.292, 0.382, 0.459, and entropy of 5.72, 6.48, 7.02 when the number of partitions is varied from 60 to 100 and 140. However, the CVT yields accessibility of 0.287, 0.382, 0.433, and entropy of 5.70, 6.44, 6.76 when the number of partitions is varied from 60 to 100 and 140. Recall that the tessellation with higher accessibility and equity scores is better. Therefore, the GVT algorithm yields better performance than the CVT.

Fig. 10 also shows the trend in the shape of tessellations when the number of partitions is increased from 60 to 100 and 140. For the GVT algorithm, when the number of partitions is increased, new centroids are increasingly being placed near highly infected areas, while regions with lower infected cases also have minute shifts. This is also similar to the CVT algorithm. More partitions are obtained in highly infected areas and less in areas with lower infected cases. Such difference in the shape of Voronoi regions is beneficial for spatial analysis of the infection distribution and resource allocation decision support.

Fig. 11 shows the convergence curve for the GVT algorithm with 100 partitions that demonstrate the tractability of the sequential optimization process. The curve of cost function keeps decreasing with respect to the change of the location of centroids. When a new tessellation centroid is being placed, the cost is decreased to a larger extent when compared with the calibration of tessellation centroid locations. After 1000 iterations, the cost function is on a plateau, which indicates algorithmic convergence. The optimization process can be controlled by the step length α , the system may converge faster when the step length is increased. The optimization process will be more conservative

when the step length is smaller with the sacrifice of a higher computation time.

Experimental results show the proposed GVT and CVT algorithms are robust against heterogeneous infection distribution in a spatial region and yield better performance than traditional Voronoi tessellation with weights and 3D K-means clustering. The proposed methods also provide zoning for the irregular and complex distribution of infected cases. When the number of partitions is increased, the difference in performance measures is decreased between CVT and GVT. Therefore, a trade-off can be established between two proposed algorithms between performance and efficiency. The CVT can be first used to analyze the situation as a fast solution, and the GVT can be used to fine-tune and search optimal tessellation of the spread.

V. CONCLUSION

Infectious diseases have severe impacts on both public healthcare and the economy. For example, COVID-19 causes a global pandemic and brings economic challenges. Due to the heterogeneity of virus spread, the demand for medical resources such as PPEs, testing kits, and vaccines have large spatial variations. The availability of such resources is critical to epidemic control. To better analyze the spread and support epidemic decisions, new tessellation methods of infectious diseases are urgently needed. However, traditional discrete location models only compute the locations on networked nodes and do not provide zoning which is important to equity regulation and resource decision making. In addition, integer programming-based discrete location models are computationally expensive and are thus not suitable for large-scale problem settings, e.g., partitioning the infection distribution. On the other hand, traditional clustering and tessellation algorithms cannot be directly applicable due to the lack of considerations about both spatial distances and the heterogeneous infection distribution.

This letter presents two sequential optimization algorithms to solve the spread tessellation problem, namely greedy-Voronoi tessellation (GVT) and cluster-Voronoi tessellation (CVT). We implement and evaluate the proposed algorithms using both simulation data and a real-world case study of COVID-19 infection data in the state of Pennsylvania. The objective is to estimate resource locations that consider both accessibility and equity based on the heterogeneous infection distribution. The performance measures are defined to be accessibility and equity scores, in which accessibility leverages spatial distances and equity evaluates the fairness of coverage among Voronoi regions. Next, we benchmark the proposed algorithms with traditional Voronoi tessellation with weight and 3D K-means using different numbers of partitions. Experimental results show the GVT yields superior performance when compared with benchmark models, but the performances of two proposed

algorithms are getting closer when the number of partitions is increased. In addition, the CVT is shown to have a better computation time compared to the GVT. The proposed methodologies are shown to have strong potentials for epidemic decision support in infection modeling and resource allocation.

REFERENCES

- [1] H. Yang *et al.*, "Epidemic informatics and control: A holistic approach from system informatics to epidemic response and risk management in public health," in *Proc. AI Analytics for Public Health - Proc. INFORMS Int. Conf. Service Sci.*, 2021, pp. 1–46.
- [2] E. J. Emanuel *et al.*, "Fair allocation of scarce medical resources in the time of covid-19," *N. Engl. J. Med.*, vol. 382, no. 21, pp. 2049–2055, 2020.
- [3] H. Jia, F. Ordóñez, and M. Dessouky, "A modeling framework for facility location of medical services for large-scale emergencies," *IIE Trans.*, vol. 39, no. 1, pp. 41–55, 2007.
- [4] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Oper. Res.*, vol. 19, no. 6, pp. 1259–1551, 1971.
- [5] S. Salhi and M. Daskin, "Network and discrete location: Models, algorithms and applications.," *J. Oper. Res. Soc.*, 1997.
- [6] M. K. Oksuz and S. I. Satoglu, "A two-stage stochastic model for location planning of temporary medical centers for disaster response," *Int. J. Disaster Risk Reduct.*, vol. 44, pp. 101426, 2020.
- [7] M. Dillon, F. Oliveira, and B. Abbasi, "A two-stage stochastic programming model for inventory management in the blood supply chain," *Int. J. Prod. Econ.*, vol. 187, pp. 27–41, 2017.
- [8] S. U. Rahman and D. K. Smith, "Use of location-allocation models in health service development planning in developing nations," *Eur. J. Oper. Res.*, vol. 123, no. 3, pp. 437–452, 2000.
- [9] H. Yang and E. K. Lee, *Healthcare Analytics: From Data to Knowledge to Healthcare Improvement*. Hoboken, N.J., Wiley, 2016.
- [10] M. S. Daskin and L. K. Dean, "Location of health care facilities," in *Operations Research and Health Care*, 2006.
- [11] T. Jia, H. Tao, K. Qin, Y. Wang, C. Liu, and Q. Gao, "Selecting the optimal healthcare centers with a modified P-median model: A visual analytic perspective," *Int. J. Health Geogr.*, vol. 13, pp. 42, 2014.
- [12] E. Villeneuve *et al.*, *A Framework to Address Key Issues of Neonatal Service Configuration in England: The NeoNet multimethods Study*. Southampton, U.K.: NIHR Journals Library, 2018.
- [13] A. Langevin, P. Mbaraga, and J. F. Campbell, "Continuous approximation models in freight distribution: An overview," *Transp. Res. Part B Methodol.*, vol. 30, no. 3, pp. 163–188, 1996.
- [14] Y. Ouyang and C. F. Daganzo, "Discretization and validation of the continuum approximation scheme for terminal system design," *Transp. Sci.*, vol. 40, no. 1, pp. 89–98, 2006.
- [15] G. Liu and H. Yang, "Self-organizing network for variable clustering," *Ann. Oper. Res.*, vol. 263, no. 1, pp. 119–140, 2018.
- [16] Y. Chen and H. Yang, "A novel information-theoretic approach for variable clustering and predictive modeling using dirichlet process mixtures," *Sci. Rep.*, vol. 6, no. 1, pp. 38913, 2016.
- [17] J. Cortés, S. Martínez, T. Karataş, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.
- [18] X. Feng and A. T. Murray, "Allocation using a heterogeneous space voronoi diagram," *J. Geogr. Syst.*, vol. 20, no. 3, pp. 207–226, 2018.
- [19] A. R. Abdulghany, "Generalization of parallel axis theorem for rotational inertia," *Am. J. Phys.*, vol. 85, no. 10, pp. 791, 2017.