

Equitable partitioning with obstacles

John Gunnar Carlsson*, Erik Carlsson† and Raghuveer Devulapalli‡

August 4, 2012

Abstract

We consider the problem of dividing a territory into sub-regions so as to balance the workloads of a collection of facilities or vehicles over that territory. We assume that the territory is a connected polygonal region, i.e. a simply connected polygon containing a set of simply connected obstacles. The territory also contains a collection of facilities, represented by a set of n fixed points in the territory. We give a fast analytic center cutting plane method that divides the territory into n compact, connected sub-regions, each of which contains a facility, such that the workloads in each sub-region are “balanced”. Our approach can also be used to design a control policy that allows us to design the sub-regions in a decentralized fashion.

Keywords: Vector space optimization; Voronoi diagrams; geometric algorithms

1 Introduction

Efficiently dividing a workload among a collection of service facilities, vehicles, or agents is a common objective in many disciplines. The most natural objective is to distribute workloads in such a way as to minimize the total amount of work done by all agents. In practice, it is also often desirable to find a *balanced* assignment so as not to overload any particular group and to ensure uniform service levels. Such assignment policies are commonly encountered in queueing theory [3, 16, 19], vehicle routing [10, 15, 24], facility location [1, 4, 7, 11], and robotics [18, 25, 26], among others.

In this paper we consider the problem of dividing a given territory among a set of facilities in such a way as to minimize the total workload imposed on those facilities while simultaneously ensuring that their workloads are sufficiently balanced. Specifically, we are given a planar region R that contains a set of *obstacles* $\{O_1, \dots, O_m\}$, a set of n fixed landmark points (“facilities”) $P = \{p_1, \dots, p_n\}$, and a continuous demand density function $f(\cdot)$ defined on R . Our objective is to design *service districts* $\{R_1, \dots, R_n\}$, such that point p_i is assigned to provide service to R_i . The set $\{R_1, \dots, R_n\}$ should obviously be a *partition* of R , that is, we should require that $\bigcup_i R_i = R$ and that the interiors do not overlap, $\text{int}(R_i) \cap \text{int}(R_j) = \emptyset$ for all distinct pairs i, j . It is also natural that we require that $p_i \in R_i$ for all i . In addition, our partition should be *compact*, in the sense that a facility should not be assigned to service points that are far away from it, and *connected*, i.e. all districts should be contiguous. See Figure 1. We present a fast algorithm that transforms our partitioning problem into an n -dimensional convex optimization problem that determines the optimal partition boundaries using a sequence of cutting planes.

We have previously considered the problem of dividing a region into smaller pieces when the input region R is convex or simply connected. Specifically, in [9, 10], we give a fast algorithm that takes as input a convex polygon C and a point set $P = \{p_1, \dots, p_n\}$ and divides C into n sub-regions $\{C_1, \dots, C_n\}$ such that each C_i is convex, each C_i contains a point, and all sub-regions have equal area. In [8] we extend this algorithm to the case where the input region is a simply connected polygon S (i.e. a connected polygon and devoid of any obstacles) and we have an arbitrary probability density $f(\cdot)$ defined on S . Rather than producing equal-area sub-regions $\{S_1, \dots, S_n\}$, our objective is to find a partition such that $\iint_{S_i} f(x) dA$ is equal for all sub-regions (the previous paper is thus a special case of this problem in which $f(\cdot)$ is a uniform distribution). Since it is not always possible to divide S into *convex* pieces in this case, our algorithm divides S into sub-regions that are *relatively convex* to S : for any two points x, y contained in a sub-region S_i , the shortest path in S from x to y (which may not be a straight line segment) is itself contained in S_i (see Figures 2, 3, and 12 of [8] for examples). This is clearly a natural generalization of convexity, because when S happens to be convex, the shortest path from x to y in S is indeed a straight line segment, and therefore all sub-regions S_i will be convex.

*Industrial and Systems Engineering, University of Minnesota. J.G.C. gratefully acknowledges the support of the Boeing Company.

†Simons Center for Geometry and Physics, SUNY Stony Brook.

‡Industrial and Systems Engineering, University of Minnesota.

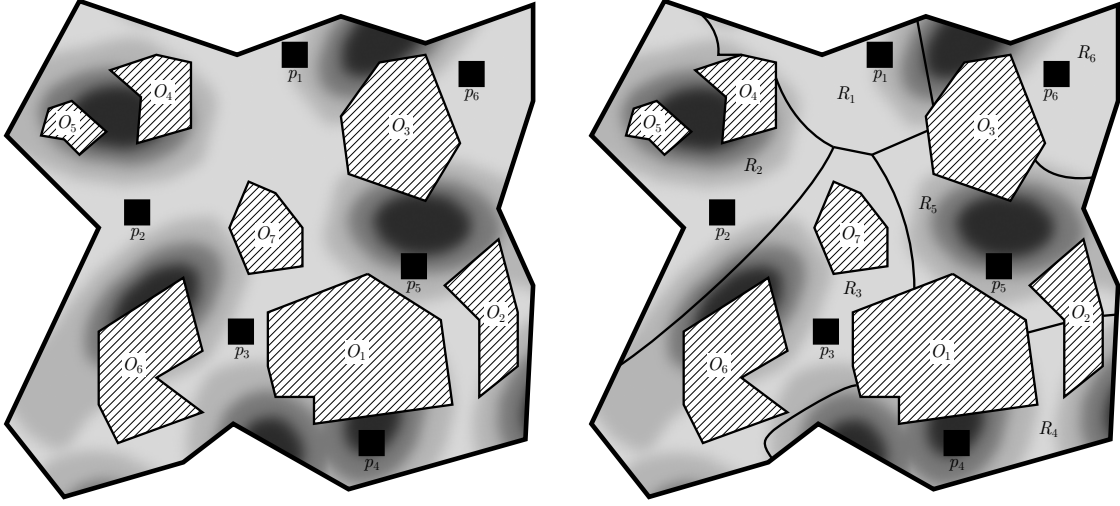


Figure 1: Inputs and outputs to our partitioning problem; the shading indicates the probability density function $f(\cdot)$. Note that we have not yet precisely defined our objective function in designing the service districts R_i .

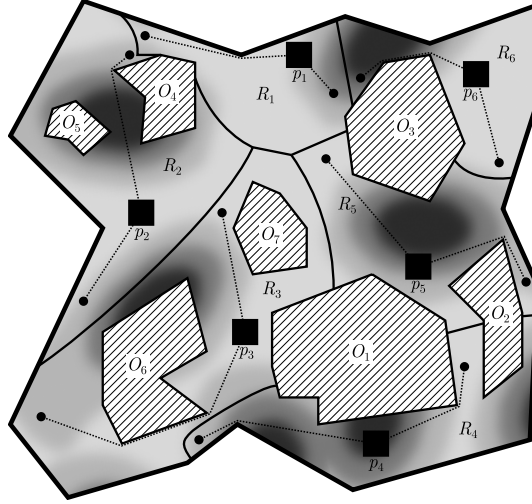


Figure 2: The partition from Figure 2 is relatively star-convex to R .

The algorithm of this paper presents two clear advantages over our previous methods: first, it is easy to show that when the input region R contains obstacles (and is thus no longer simply connected), an equitable relatively convex partition of the type just described may not exist (see Figure 1 of [5]), and thus our prior work is of little utility. Second, the algorithms of [9, 10] and [8] do not force regions to be *compact*: it is possible for the output regions to be extremely long and skinny, which is clearly undesirable from a practical standpoint (see Figure 10 of [9] and Figure 14 of [8]). Here we tackle these two problems in the following way: first, as we will show, our algorithm produces sub-regions $\{R_1, \dots, R_n\}$ that are *relatively star-convex* to R : for any point x in sub-region R_i , the shortest path in R from x to p_i is itself contained in R_i (see Figure 2). This is a weaker condition than relative convexity but still offers considerable practical utility; for example, it automatically produces connected sub-regions. Second, we are able to explicitly enforce our sub-regions to be compact by imposing a particular objective function on our partition. In fact, one interpretation of our problem is that we are trying to find the partition that is as compact as possible; we will make this explicit in Section 2. The algorithm in this paper is also novel in that our prior work generally uses well-established principles of discrete geometry such as ham sandwich cuts and binary and ternary space partitions, whereas here we use techniques from vector space optimization and linear programming complementarity.

Related work

In addition to our own previous work that we have already mentioned, the general problem of equitably partitioning a region has been studied from many perspectives. The paper [26] considers the problem of dividing a convex region into convex equal-area sub-regions using *Power diagrams*, a natural extension of *Voronoi diagrams*. The authors give a decentralized control policy that provably converges to the desired partition. More recently, [12] considers a hybrid problem in which the objective is to simultaneously place the facilities P and design coverage regions associated with the facilities using only asynchronous (and possibly noisy) pairwise communication between facilities. The authors give an algorithm that provably converges to a *centroidal Voronoi partition*, that is, a Voronoi diagram $\{V_1, \dots, V_n\}$ such that p_i is the center of mass of each sub-region V_i . Other papers making use of power diagrams as a partitioning policy include [2], [21], and [27].

One notion of “partitioning” discussed in [6] is to allow facilities to have variable “coverage radii” r_i , where the cost $\phi(r_i)$ is a monotonically increasing function; the problem is to find the optimal number, location, and coverage radii of a collection of facilities. Another paper [1] considers the problem of partitioning a convex region C so as to minimize the aggregate workload over all facilities while imposing an equal-area constraint. The authors describe a constant-factor approximation algorithm for dividing C into equal-area convex pieces to maximize the minimum “fatness” of any piece. This in turn gives an approximation algorithm for the problem of minimizing the aggregate workload over all facilities when facility placement is variable, as well as the sub-region boundaries.

Notational conventions and technical assumptions

Our notational conventions in this paper will be as follows: we let $d(x, y)$ denote the distance between points x and y in R when obstacles are taken into account. When we refer to the input region R , we implicitly take the obstacles into account (so that $R = S \setminus \bigcup_i O_i$, for some simply connected polygon S). We also assume throughout this paper that $f(x) > 0$ on R .

2 Problem formulation and applications

We begin by formally stating our optimization problem for designing the service districts R_i . Suppose that the cost of service between a demand point x and facility i is of the form $d(x, p_i) + \alpha_i$, where α_i denotes some “fixed cost” associated with facility i such as dispatching a vehicle or initiating a transaction. In the remainder of this paper, we will assume that $\alpha_i = 0$ for simplicity; our analytical results are unaffected by this assumption. If demand is distributed over R according to an absolutely continuous probability density function $f(\cdot)$, the average workload on facility i is therefore $\iint_{R_i} f(x)(d(x, p_i) + \alpha_i) dA = \iint_{R_i} f(x)d(x, p_i) dA$. It is clear that the total workload on all facilities is minimized when each point x is merely assigned to its nearest facility, i.e. when $\{R_1, \dots, R_n\}$ is a Voronoi partition of P in R . In order to balance the workloads of the facilities, we will impose the constraint that $\iint_{R_i} f(x) dA = c_i$ for each i , where $\mathbf{c} = (c_1, \dots, c_n)^T$ is a given input parameter vector such that $c_i > 0$ for all i and $\sum_i c_i = 1$; we typically expect that $c_i = 1/n$ for all i . We can thus write our optimization problem as

$$\begin{aligned} \underset{R_1, \dots, R_n}{\text{minimize}} \quad & \sum_{i=1}^n \iint_{R_i} f(x)d(x, p_i) dA \quad s.t. \\ & \iint_{R_i} f(x) dA = c_i \\ & \bigcup_{i=1}^n R_i = R. \end{aligned} \tag{1}$$

It is easy to see that the objective function of (1) forces regions to be compact because it minimizes the average distance between customers and their assigned facilities. It is also worth mentioning that (1) is a special case of the famous *Monge-Kantorovich transportation problem* [29] in the plane in which one Radon measure is continuous and the other is atomic: our objective is to “transport” the continuously distributed demand to the finite collection of facilities, while obeying capacity constraints and minimizing the aggregate transportation cost. Before describing the solution strategy for (1), we describe three practical applications of this problem.

Short trip deployment An obvious application of the above problem is the case where $f(\cdot)$ represents the locations of likely surveillance targets in R and the facilities represent launching sites of a fleet of UAVs. Suppose

that, due to short battery life, each UAV must return to its depot after visiting a target for recharging. It follows that the average workload on all facilities is precisely the objective function of problem (1).

Multi-depot vehicle routing The formulation (1) is useful even when we remove the constraint that vehicles return to their depots after visiting each destination. Suppose now that likely targets are distributed by an absolutely continuous probability density function $\tilde{f}(\cdot)$ and that, each day, a collection of N independent samples $\mathcal{X} = \{X_1, \dots, X_N\}$ from $\tilde{f}(\cdot)$ is drawn. As before, each vehicle is assigned to visit all of the samples that lie in region R_i . Thus, each vehicle's workload is a TSP tour of all of the sampled points in R_i , $\mathcal{X} \cap R_i$, plus the starting depot point p_i . As we have shown in [8], it turns out that the workload of each vehicle is approximately $\beta\sqrt{N} \iint_{R_i} \sqrt{\tilde{f}(x)} dA$, where $\beta \in (0.6250, 0.9204)$ is a constant. More precisely, with probability one as $N \rightarrow \infty$, we have

$$\text{TSP}(\mathcal{X} \cap R_i) = \beta\sqrt{N} \iint_{R_i} \sqrt{\tilde{f}(x)} dA + o(\sqrt{N}).$$

It therefore follows that, if our goal is to balance the workloads of all vehicles, we should require that $\iint_{R_i} \sqrt{\tilde{f}(x)} dA$ be equal for all vehicles (this guarantees that the TSP tours are all equal within a term of $o(\sqrt{N})$). Therefore we can formulate this problem as an instance of (1) where $f(x) = \sqrt{\tilde{f}(x)}$ (normalized so that $f(x)$ integrates to 1 on R) and $c_i = 1/n$. Note that when $\tilde{f}(x)$ is the uniform distribution, our objective is to divide the service region into equal-area sub-regions.

Carbon capture and storage The term “carbon sequestration” refers to the uptake of carbon-containing natural substances, in particular carbon dioxide, into forests, grasslands, peat swamps, and other terrestrial ecosystems. By storing this carbon in local biomass, ecosystems keep carbon dioxide out of the atmosphere. A typical Carbon Capture and Storage (CCS) scheme consists of three stages:

1. Carbon dioxide is captured at the source (power plants, refineries, or other industrial facilities) and compressed.
2. Carbon dioxide is then transported via pipelines to a storage site.
3. Finally, carbon dioxide is stored into a geologic sink or reservoir, where it is *sequestered* from the atmosphere.

The problem of optimally distributing carbon dioxide can be formulated as an instance of (1) in the following sense: suppose that P is a collection of facilities with fixed locations in R , and suppose that each facility i emits an amount of carbon c_i . Let $f(\cdot)$ denote the probability density function that represents the capacity of a portion of land to absorb carbon dioxide. Further assume that $\sum_i c_i = \iint_R f(x) dx = 1$, i.e. that the region is capable of absorbing exactly the amount of carbon being output. Our problem is to divide R into n parcels of land R_i (one per facility), so that each piece R_i can absorb an amount c_i , i.e. that $\iint_{R_i} f(x) dA = c_i$. Since pipelines for carbon transport are costly (up to \$800,000 per mile [22]), our objective is to minimize the average transport distance between a facility and the absorptive substances in R_i . Here the obstacles might represent areas in which it is prohibitively expensive or illegal to construct pipelines, such as rocky terrain or protected areas for regional flora and fauna. It is sensible that we should also require that the pipelines from different facilities should not intersect; indeed, as we will show in the following section, this is guaranteed to happen automatically by solving problem (1).

3 Optimal partitioning

We will now formulate our problem (1) as an infinite-dimensional optimization problem, which we will then transform into an n -dimensional convex optimization problem. It is clear that problem (1) admits an infinite-dimensional integer program given by

$$\begin{aligned} \underset{I_1(\cdot), \dots, I_n(\cdot)}{\text{minimize}} \quad & \iint_R \sum_{i=1}^n f(x) d(x, p_i) I_i(x) dA \quad \text{s.t.} \\ & \iint_R f(x) I_i(x) dA = c_i \quad \forall i \\ & \sum_{i=1}^n I_i(x) = 1 \quad \forall x \in R \\ & I_i(x) \in \{0, 1\} \quad \forall i, x \end{aligned} \tag{2}$$

where the $I_i(\cdot)$'s are indicator functions. The linear relaxation of this problem is obviously given by

$$\begin{aligned} \text{minimize}_{I_1(\cdot), \dots, I_n(\cdot)} \quad & \iint_R \sum_{i=1}^n f(x) d(x, p_i) I_i(x) dA \quad s.t. \\ & \iint_R f(x) I_i(x) dA = c_i \quad \forall i \\ & \sum_{i=1}^n I_i(x) = 1 \quad \forall x \in R \\ & I_i(x) \geq 0 \quad \forall i, x. \end{aligned} \quad (3)$$

It turns out that the dual of (3) can be expressed in terms of n Lagrange multipliers λ_i and that we can recover the optimal partition to the primal problem (3) via complementary slackness. Before proving our main theorem, we state an important result from Section 8.6 of [20]:

Theorem 1. (*Lagrange Duality*) Let \mathfrak{f} be a real-valued convex functional defined on a convex subset Ω of a vector space \mathfrak{X} , and let \mathfrak{G} be a convex mapping of \mathfrak{X} into a normed space \mathfrak{Z} . Suppose there exists $\mathfrak{r}_1 \in \mathfrak{X}$ such that $\mathfrak{G}(\mathfrak{r}_1) < \theta$, where θ denotes the zero element, and that $\mu_0 := \inf \{\mathfrak{f}(\mathfrak{r}) : \mathfrak{G}(\mathfrak{r}) \leq \theta, \mathfrak{r} \in \Omega\}$ is finite. Then

$$\inf_{\mathfrak{r} \in \Omega, \mathfrak{G}(\mathfrak{r}) \leq \theta} \mathfrak{f}(\mathfrak{r}) = \max_{\mathfrak{z}^* \geq \theta} \varphi(\mathfrak{z}^*)$$

where

$$\varphi(\mathfrak{z}^*) = \inf_{\mathfrak{r} \in \Omega} \mathfrak{f}(\mathfrak{r}) + \langle \mathfrak{G}(\mathfrak{r}), \mathfrak{z}^* \rangle,$$

and the maximum on the right is achieved by some $\mathfrak{z}_0^* \geq \theta$.

Our main result follows below:

Theorem 2. The dual problem to (3) is given by the n -dimensional convex optimization problem

$$\begin{aligned} \text{maximize}_{\lambda \in \mathbb{R}^n} \quad & \iint_R f(x) \min_i \{d(x, p_i) - \lambda_i\} dA \quad s.t. \\ & \mathbf{c}^T \boldsymbol{\lambda} = 0. \end{aligned} \quad (4)$$

Moreover, the optimal solution $I_1^*(\cdot), \dots, I_n^*(\cdot)$ to the primal problem (3) must satisfy

$$I_i(x) = \begin{cases} 0 & \text{if } d(x, p_i) - \lambda_i^* > d(x, p_j) - \lambda_j^* \text{ for some } j \\ 1 & \text{if } d(x, p_i) - \lambda_i^* \leq d(x, p_j) - \lambda_j^* \text{ for all } j \neq i. \end{cases}$$

If neither of the two cases above holds at a point x , then if $I_i^*(x) > 0$ at a point x , then it must be the case that $d(x, p_i) - \lambda_i^* \leq d(x, p_j) - \lambda_j^*$ for all j , i.e. the index i is among the minimal indices.

Proof. A proof sketch of this result can readily be obtained by discretizing the region R and taking the dual of the resulting linear program; the following proof is precisely the infinite-dimensional equivalent of this, which we have included for the sake of rigor (and possibly at the expense of some clarity). It is clear that an optimal solution to problem (4) exists because we must have $\lambda_i^* \leq \bar{\lambda} := \max_j \max_{x \in R} d(x, p_j)$ for all i , since otherwise we have a negative objective function value in problem (4) which is clearly sub-optimal. Thus, we can assume without loss of generality that problem (4) is defined on a compact set and hence attains its minimizer. It is also clear that problem (4) is equivalent to the infinite-dimensional linear program given by

$$\begin{aligned} \text{maximize}_{\lambda, \sigma(\cdot)} \quad & \iint_R f(x) \sigma(x) dA \quad s.t. \\ & \sigma(x) \leq d(x, p_i) - \lambda_i \quad \forall i, x \\ & \mathbf{c}^T \boldsymbol{\lambda} = 0. \end{aligned} \quad (5)$$

Since $\sum_i c_i = \iint_R f(x) = 1$, this is in turn equivalent to

$$\begin{aligned} \text{maximize}_{\lambda, \sigma(\cdot)} \quad & \mathbf{c}^T \boldsymbol{\lambda} + \iint_R f(x) \sigma(x) dA \quad s.t. \\ & \sigma(x) \leq d(x, p_i) - \lambda_i \quad \forall i, x. \end{aligned} \quad (6)$$

We now apply Theorem 1: in problem (6), the optimization variables are λ and $\sigma(\cdot)$, so we let $\mathfrak{X} = \Omega = \mathbb{R}^n \oplus L_1$, where L_1 represents all functions $h(\cdot)$ defined on R such that $|h(x)|$ is Lebesgue integrable on R . We let $\mathfrak{f}(\mathfrak{x})$ be defined by

$$\mathfrak{f} : \begin{pmatrix} \lambda \\ \sigma(\cdot) \end{pmatrix} \mapsto -\mathbf{c}^T \lambda - \iint_R f(x) \sigma(x) dA$$

and we let $\mathfrak{G} : \mathfrak{X} \rightarrow \mathfrak{Z}$ be defined by

$$\mathfrak{G} : \begin{pmatrix} \lambda \\ \sigma(\cdot) \end{pmatrix} \mapsto \begin{pmatrix} \xi_1(\cdot) + \lambda_1 \\ \vdots \\ \xi_n(\cdot) + \lambda_n \end{pmatrix}$$

where $\xi_i(x) := \sigma(x) - d(x, p_i)$, so that $\mathfrak{Z} = \underbrace{L_1 \oplus \cdots \oplus L_1}_n$. By the preceding existence argument for λ^* we can replace the infimum operator with the minimum operator. From basic functional analysis, we have $\mathfrak{Z}^* = \underbrace{L_\infty \oplus \cdots \oplus L_\infty}_n$, where L_∞ denotes all *bounded* functions on R . Let $(J_1(\cdot), \dots, J_n(\cdot))$ denote an element of \mathfrak{Z}^* . Theorem 1 says that

$$\begin{aligned} & - \min_{\mathfrak{x} \in \Omega, \mathfrak{G}(\mathfrak{x}) \leq \theta} \mathfrak{f}(\mathfrak{x}) \\ &= - \max_{\mathfrak{z}^* \geq \theta} \left\{ \inf_{\mathfrak{x} \in \Omega} \mathfrak{f}(\mathfrak{x}) + \langle \mathfrak{G}(\mathfrak{x}), \mathfrak{z}^* \rangle \right\} \\ &= - \max_{J_i(\cdot) \geq 0} \left\{ \inf_{\lambda, \sigma(\cdot)} -\mathbf{c}^T \lambda - \iint_R f(x) \sigma(x) dA + \iint_R \sum_{i=1}^n J_i(x) (\xi_i(x) + \lambda_i) dA \right\} \\ &= - \max_{J_i(\cdot) \geq 0} \left\{ \inf_{\lambda, \sigma(\cdot)} \left[\sum_{i=1}^n \left(\iint_R J_i(x) dA - c_i \right) \lambda_i \right] + \iint_R \left(\sum_{i=1}^n J_i(x) \sigma(x) - J_i(x) d(x, p_i) \right) - f(x) \sigma(x) dA \right\} \\ &= - \max_{J_i(\cdot) \geq 0} \left\{ \inf_{\lambda, \sigma(\cdot)} \left[\sum_{i=1}^n \left(\iint_R J_i(x) dA - c_i \right) \lambda_i \right] + \iint_R \sigma(x) \left(\sum_{i=1}^n J_i(x) - f(x) \right) dA - \iint_R \sum_{i=1}^n J_i(x) d(x, p_i) dA \right\}. \end{aligned}$$

Clearly we need $\iint_R J_i(x) dA = c_i$ for all i and $\sum_i J_i(x) = f(x)$ for all $x \in R$. Introducing new functions $I_i(x) := J_i(x) / f(x)$, the above is equivalent to

$$- \max_{I_i(\cdot) \geq 0} \left\{ \inf_{\lambda, \sigma(\cdot)} \left[\sum_{i=1}^n \left(\iint_R f(x) I_i(x) dA - c_i \right) \lambda_i \right] + \iint_R f(x) \sigma(x) \left(\sum_{i=1}^n I_i(x) - 1 \right) dA - \iint_R \sum_{i=1}^n f(x) I_i(x) d(x, p_i) dA \right\}$$

so that $\iint_R f(x) I_i(x) dA = c_i$ for all i and $\sum_i I_i(x) = 1$ for all $x \in R$. Thus the problem (5) and the problem

$$\text{minimize}_{I_1(\cdot), \dots, I_n(\cdot)} \iint_R \sum_{i=1}^n f(x) d(x, p_i) I_i(x) dA \quad s.t.$$

$$\iint_R f(x) I_i(x) dA = c_i \quad \forall i$$

$$\sum_{i=1}^n I_i(x) = 1 \quad \forall x \in R$$

$$I_i(x) \geq 0 \quad \forall i, x.$$

are primal-dual pairs as desired, and by Theorem 1 we know that an optimal solution $I_1^*(\cdot), \dots, I_n^*(\cdot)$ to problem (3) exists. This completes the first claim of the proof.

We now turn our attention to the second claim: consider an optimal primal-dual pair $I_1^*(\cdot), \dots, I_n^*(\cdot)$ and $\lambda^*, \sigma^*(\cdot)$. We have $\iint_R f(x) I_i^*(x) dA = c_i$, and therefore $\mathbf{c}^T \lambda^* = \sum_i \lambda_i^* \iint_R f(x) I_i^*(x) dA$, and therefore, since Theorem 1 guarantees

$$\mathbf{c}^T \lambda^* + \iint_R f(x) \min_i \{d(x, p_i) - \lambda_i^*\} dA - \iint_R f(x) \sum_{i=1}^n d(x, p_i) I_i^*(x) dA = 0,$$

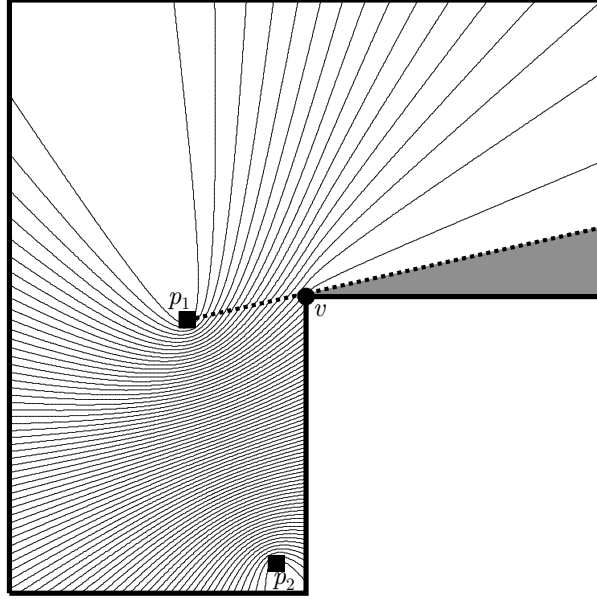


Figure 3: As λ_1 and λ_2 vary, the induced sub-regions change as indicated above. The points x in the shaded region do not have a unique minimal index i_{\min} because $\lambda_1 - \lambda_2 = d(v, p_1) - d(v, p_2)$. The point v is therefore the *reflex vertex* associated with the shaded region.

we must have

$$\iint_R f(x) \left[\sigma^*(x) - \sum_{i=1}^n (d(x, p_i) - \lambda_i^*) I_i^*(x) \right] dA = 0.$$

Recall that $\sigma^*(x) \leq d(x, p_i) - \lambda_i^*$ for all $x \in R$ and i and that $\sum_i I_i^*(x) = 1$ for all $x \in R$. For any point x , suppose that i_{\min} is the index such that $d(x, p_i) - \lambda_i^*$ is minimal (assuming such an index is unique). The integrand above is always negative unless we set $I_{i_{\min}}^*(x) = 1$ and $I_i^*(x) = 0$ otherwise. Therefore, the optimal dual vector λ^* corresponds to a solution to the primal problem (3) in which

$$I_i^*(x) = \begin{cases} 0 & \text{if } d(x, p_i) - \lambda_i^* > d(x, p_j) - \lambda_j^* \text{ for some } j \\ 1 & \text{if } d(x, p_i) - \lambda_i^* < d(x, p_j) - \lambda_j^* \text{ for all } j \neq i \end{cases}$$

as desired. It is similarly straightforward to show more generally that if $I_{\bar{i}}^*(x) > 0$ for a point x and a given index \bar{i} , then index \bar{i} must be among the minimal indices of $d(x, p_i) - \lambda_i^*$, which completes the proof. \square

Having proven Theorem 2 we now turn our attention to the problem of recovering the optimal regions R_1^*, \dots, R_n^* (equivalently $I_1^*(\cdot), \dots, I_n^*(\cdot)$) from the dual vector λ^* . Given λ^* , we define the “strict dominance regions” $R_i^+ \subseteq R_i^*$ as

$$R_i^+ = \{x \in R : d(x, p_i) - \lambda_i^* < d(x, p_j) - \lambda_j^* \text{ for all } j \neq i\}.$$

It is easy to see by construction that each R_i^+ is *relatively star-convex* to R : if $x \in R_i^+$, the shortest path (or paths) in R from x to p_i is contained in R_i^+ . From basic Euclidean geometry it is also clear that the boundary between two strict dominance regions R_i^+ and R_j^+ consists of a collection of hyperbolic arcs, since a hyperbola is the locus of points where the absolute value of the difference of the distances to two foci is constant (if we measure point-to-point distances using the ℓ_1 or ℓ_∞ norms, while still taking obstacles into account, the boundary between two strict dominance regions consists instead of a collection of line segments). Given λ^* , we can construct these arcs for all R_i^+ efficiently using the *continuous Dijkstra paradigm* [23].

The question remains of how to allocate the remaining area of R that does not lie in a strict dominance region (see Figure 3). It is not hard to show that each such region is polygonal (as opposed to being bounded by hyperbolic arcs). Since each such “ambiguous dominance region” is potentially associated with more than one facility, we associate with each such region R_k^- (not indexing by i because these regions are not yet associated with individual facilities) an index set $\mathcal{I}_k \subseteq \{1, \dots, n\}$. By construction, each R_k^- must have a *reflex vertex* r_k , that is, a point in R_k^- such that $d(x, p_i) = d(x, r_k) + d(r_k, p_i)$ for all $x \in R_k^-$ and $i \in \mathcal{I}_k$; an example of this is the point marked v in

Figure 3. Consider the functions $I_i^*(\cdot)$ (which we have not yet defined on R_k^-) for $i \in \mathcal{I}_k$. The total cost due to R_k^- is

$$\begin{aligned}
\iint_{R_k^-} \sum_{i \in \mathcal{I}_k} f(x) d(x, p_i) I_i^*(x) dA &= \iint_{R_k^-} \sum_{i \in \mathcal{I}_k} f(x) (d(x, r_k) + d(r_k, p_i)) I_i^*(x) dA \\
&= \iint_{R_k^-} f(x) d(x, r_k) \underbrace{\left(\sum_{i \in \mathcal{I}_k} I_i^*(x) \right)}_{=1} dA + \iint_{R_k^-} \sum_{i \in \mathcal{I}_k} f(x) d(r_k, p_i) I_i^*(x) dA \\
&= \underbrace{\iint_{R_k^-} f(x) d(x, r_k) dA}_{\text{constant}} + \sum_{i \in \mathcal{I}_k} d(r_k, p_i) \iint_{R_k^-} f(x) I_i^*(x) dA
\end{aligned}$$

which we observe only depends on $\iint_{R_k^-} f(x) \sum_{i \in \mathcal{I}_k} I_i^*(x) dA$, that is, the *amounts* of mass $f(x)$ of region R_k^- assigned to the facilities $i \in \mathcal{I}_k$, as opposed to the particular assignment patterns $I_i^*(x)$. Thus, in order to assign these amounts to the facilities, we can simply construct a matrix A such that

$$a_{ik} = \begin{cases} 1 & \text{if } i \in \mathcal{I}_k \\ 0 & \text{otherwise} \end{cases}$$

and then consider the problem of constructing a matrix Z that gives the amounts of mass assigned to each facility from each R_k^- , i.e. satisfying the constraints

$$\begin{aligned}
\sum_i a_{ik} z_{ik} &= \iint_{R_k^-} f(x) dA \quad \forall k \\
\sum_k a_{ik} z_{ik} &= c_i - \iint_{R_i^+} f(x) dA \quad \forall i \\
z_{ik} &\geq 0 \quad \forall i, k.
\end{aligned} \tag{7}$$

It follows from the argument above that the total cost due to these allocations is then $\sum_{i,k} b_{ik} z_{ik}$, where B is a matrix such that $b_{ik} = d(r_k, p_i)$. In fact, it is not hard to show that all feasible solutions to (7) have the same objective value $\sum_{i,k} b_{ik} z_{ik}$: this is precisely because, for any ambiguous dominance region R_k^- , we must have $d(r_k, p_i) - d(r_k, p_j) = \lambda_i - \lambda_j$ for all $i, j \in \mathcal{I}_k$. Thus, the columns of B are simply the vector λ^* , translated by a scalar, and with all indices $j \notin \mathcal{I}_k$ set to 0:

$$B = \begin{pmatrix} \left| \begin{array}{c} \lambda_{\mathcal{I}_1}^* \\ \vdots \end{array} \right| + \beta_1, & \cdots, & \left| \begin{array}{c} \lambda_{\mathcal{I}_k}^* \\ \vdots \end{array} \right| + \beta_k, & \cdots, & \left| \begin{array}{c} \lambda_{\mathcal{I}_K}^* \\ \vdots \end{array} \right| + \beta_K \end{pmatrix}$$

where K is the number of ambiguous dominance regions.

After finding a feasible (and therefore optimal) set of allocations Z^* , we merely have to find a way to assign the areas z_{ik}^* so that the final regions R_i^* are relatively star-convex. This is trivial as shown in Figure 4.

For completeness we will now verify that the partition $\{R_1^*, \dots, R_n^*\}$ that we obtain is in fact optimal, i.e. that strong duality in the sense of Theorem 1 holds. This is straightforward; since $I_i^*(x) = 1$ holds only when \bar{i} is a minimal index of $d(x, p_i) - \lambda_i^*$, we have

$$\begin{aligned}
\iint_R f(x) \min_i \{d(x, p_i) - \lambda_i^*\} dA &= \sum_{i=1}^n \iint_{R_i} f(x) (d(x, p_i) - \lambda_i^*) dA \\
&= \sum_{i=1}^n \iint_R f(x) (d(x, p_i) - \lambda_i^*) I_i^*(x) dA \\
&= \sum_{i=1}^n \iint_R f(x) d(x, p_i) I_i^*(x) dA - \underbrace{\lambda_i^* \iint_R f(x) I_i^*(x) dA}_{=c_i} \\
&= \iint_R \sum_{i=1}^n f(x) d(x, p_i) I_i^*(x) dA - \underbrace{\mathbf{c}^T \lambda^*}_{=0} = \iint_R \sum_{i=1}^n f(x) d(x, p_i) I_i^*(x) dA
\end{aligned}$$

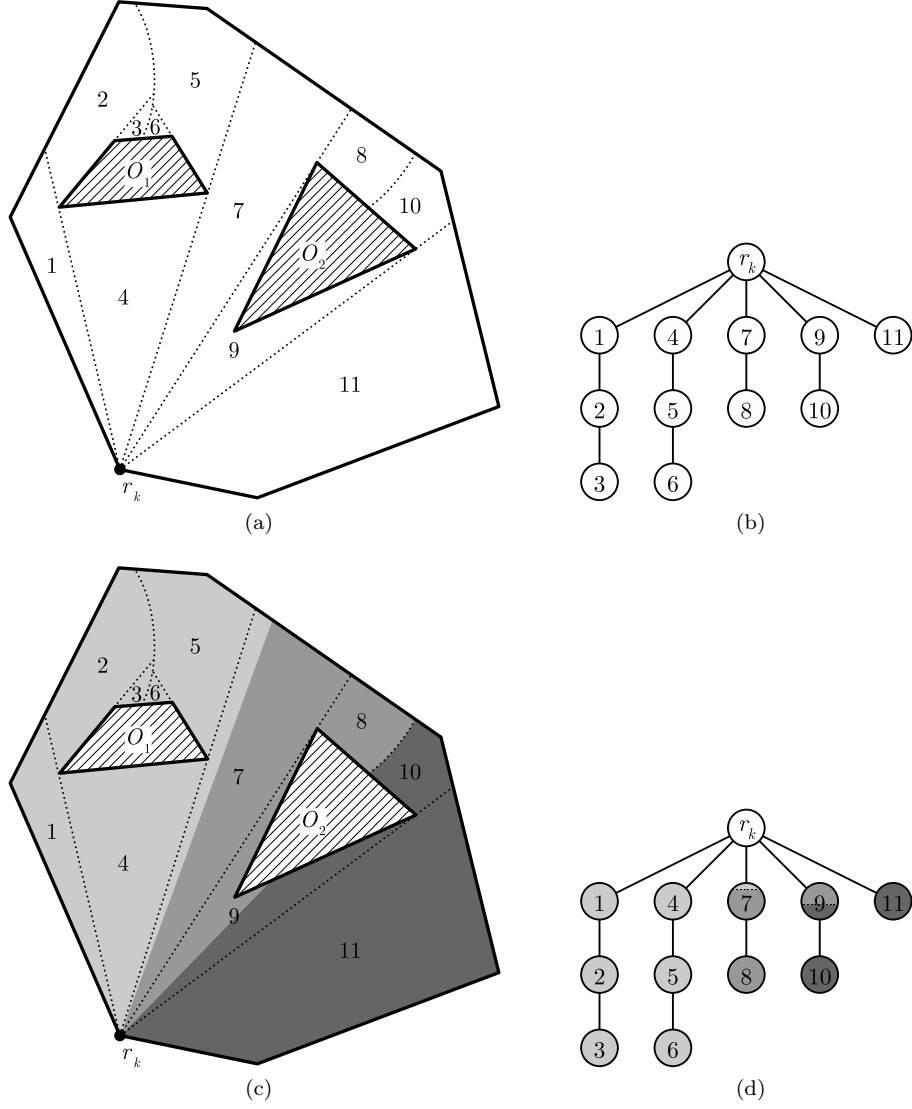


Figure 4: Given an ambiguous dominance region R_k^- (4a) and a set of allocations Z^* , it is straightforward to divide R_k^- into pieces that are relatively star-convex by constructing a shortest-path tree (4a and 4b) and then assigning regions based on (for example) a depth-first search of the tree (4c and 4d).

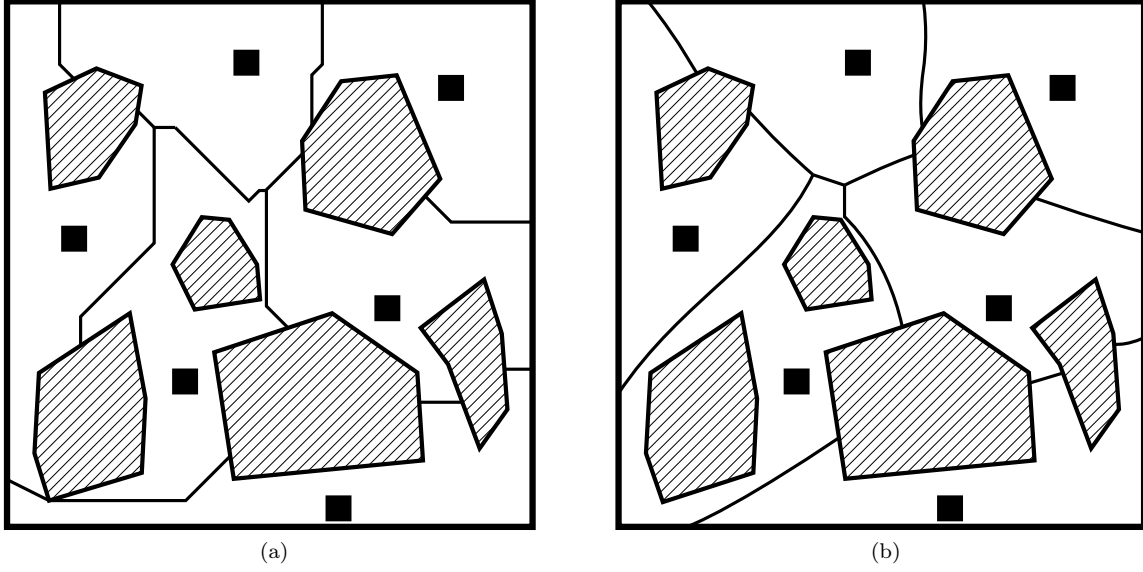


Figure 5: Equitable partitioning under the ℓ_1 norm (5a) versus the ℓ_2 norm (5b)

as desired. The following result is immediately evident:

Theorem 3. *There exists an optimal solution to the relaxation (3) in which $I_i^*(x) \in \{0, 1\}$ for all i and all $x \in R$. Thus, the integrality gap of problem (2) is unity.*

Remark 1. *A related result to Theorem 2 can be found in [2], which considers a version of the primal problem (1) without obstacles where the weights between points are the squares of the Euclidean distances. The authors show (without using complementarity) that the optimal assignment is a kind of Voronoi partition called a Power diagram.*

Remark 2. *If we desire regions with line segments as boundaries instead of hyperbolas (perhaps for computational efficiency), we may measure point-to-point distances using the ℓ_1 or ℓ_∞ norms (while still taking obstacles into account). Figure 5 shows the difference between partitions induced by using such a distance metric.*

4 Computing supergradients

In the previous section, we showed that our partitioning problem can be reduced to a low-dimensional convex optimization problem. In this section, we describe how to efficiently compute the supergradient of the dual problem (4). For the moment, for a given weight vector λ , we let R_i denote the dominance region of facility i , with ambiguous dominance regions assigned lexicographically. That is, point x is in region R_i if and only if $d(x, p_i) - \lambda_i$ is minimal and if index i is the lowest of the minimal indices (we make this assignment only so that the regions R_i can be assigned in a unique way).

Theorem 4. *The vector \mathbf{g} defined by*

$$g_i := \iint_{R_i} f(x) dA$$

is a supergradient for the concave function

$$h(\lambda) = \iint_R f(x) \min_i \{d(x, p_i) - \lambda_i\} dA.$$

Proof. Consider two vectors λ and λ' and the corresponding partitions $\{R_1, \dots, R_n\}$ and $\{R'_1, \dots, R'_n\}$. We want to show that $h(\lambda') \leq h(\lambda) + \mathbf{g}^T(\lambda' - \lambda)$, i.e. that

$$\iint_R f(x) \min_i \{d(x, p_i) - \lambda'_i\} dA \leq \iint_R f(x) \min_i \{d(x, p_i) - \lambda_i\} dA - \mathbf{g}^T(\lambda' - \lambda)$$

or equivalently that

$$\sum_{i=1}^n \iint_{R'_i} f(x)(d(x, p_i) - \lambda'_i) dA \leq \sum_{i=1}^n \iint_{R_i} f(x)(d(x, p_i) - \lambda_i) dA - g_i(\lambda'_i - \lambda_i).$$

Consider the right-hand side of the above; for each i , we have

$$\begin{aligned} \iint_{R_i} f(x)(d(x, p_i) - \lambda_i) dA - g_i(\lambda'_i - \lambda_i) &= \iint_{R_i} f(x)(d(x, p_i) - \lambda_i) dA - (\lambda'_i - \lambda_i) \iint_{R_i} f(x) dA \\ &= \iint_{R_i} f(x)(d(x, p_i) - \lambda'_i) dA \end{aligned}$$

and therefore we have

$$\sum_{i=1}^n \iint_{R_i} f(x)(d(x, p_i) - \lambda'_i) dA \geq \sum_{i=1}^n \iint_{R'_i} f(x)(d(x, p_i) - \lambda'_i) dA$$

since by construction, the sub-regions of the partition $\{R'_1, \dots, R'_n\}$ are defined by looking at the minimal value of $d(x, p_i) - \lambda'_i$ and are therefore minimal over all partitions. This completes the proof. \square

We thus have in hand a fast method for computing supergradients for problem (4). We can therefore solve this problem quickly using an *analytic-center cutting plane method* (ACCPM), as given in Algorithm 1. In the next section we will describe the computational complexity of this algorithm.

Note that each agent can compute its component of the supergradient vector using only knowledge of its own induced region, which can be determined using only local information to that agent, i.e. its neighbors in the induced partition. Therefore, an alternative to the ACCPM just described is to use a *decentralized* supergradient ascent-based control scheme; at any given instant, given a multiplier vector λ^0 , we have each agent compute its supergradient component g_i , and we then set $\lambda_i^0 \mapsto \lambda_i^0 + \epsilon(g_i - 1/n)$, where we have subtracted $1/n$ so as to retain feasibility of dual problem (4), and where ϵ is a small discretization coefficient.

5 Computational complexity

Since Algorithm 1 is simply an analytic center cutting plane method (whose complexity is well-understood [14]) for solving the convex problem (4), it will suffice to discuss the computational complexity of each iteration. For any given vector λ , we can construct the corresponding strict and ambiguous dominance regions (i.e. the weighted Voronoi diagram) using the *continuous Dijkstra paradigm* in $\mathcal{O}(N^{5/3})$ steps, where N is the total number of vertices of R and the obstacles [23]. When $f(\cdot)$ is the uniform distribution on R , we can also compute the areas of these cells in $\mathcal{O}(N^{5/3})$ steps and therefore each iteration requires only $\mathcal{O}(N^{5/3})$ computations. When $f(\cdot)$ is non-uniform, the computational complexity depends on the chosen numerical integration scheme. To describe this we use the following result (originally given in Section 7.4 of [17], but re-stated using the language of Section 9.9 of [28]):

Theorem 5. *Let $\Omega \subset \mathbb{R}^2$ be a domain of integration equipped with a triangulation \mathcal{T}_h consisting of N_T triangles, where h is the maximum edge length in \mathcal{T}_h . There exists a positive constant K_1 , independent of h , such that the error E induced using either the composite midpoint formula*

$$\iint_{\Omega} f(x) dA \approx \sum_{T \in \mathcal{T}_h} \text{Area}(T) f(\text{centroid}(T))$$

or the composite trapezoidal formula

$$\iint_{\Omega} f(x) dA \approx \frac{1}{3} \sum_{T \in \mathcal{T}_h} \text{Area}(T) \sum_{j=1}^3 f(\text{vertex}_j(T))$$

is bounded by

$$|E| \leq K_1 h^2 \text{Area}(\Omega) M_2,$$

where M_2 is the maximum value of the modules of the second derivatives of the integrand $f(\cdot)$.

Input: A polygonal region R containing a collection of obstacles O_1, \dots, O_m , a probability density $f(\cdot)$ defined on R , a collection of points $P = \{p_1, \dots, p_n\} \subset R$, a vector \mathbf{c} such that $\sum_i c_i = \iint_R f(x) dA = 1$, and a threshold ϵ .

Output: A partition of R into n compact relatively star-convex regions R_1, \dots, R_n , such that $p_i \in R_i$ and $\left| \iint_{R_i} f(x) dA - c_i \right| \in \mathcal{O}(\epsilon)$ for all i .

Note: this is simply a standard analytic center cutting plane method applied to the dual problem (4).

Define the initial polyhedron by $\Lambda = \{\lambda \in \mathbb{R}^n : \mathbf{c}^T \lambda = 0 \text{ and } |\lambda_i - \lambda_j| \leq d(p_i, p_j)\}$;

while $\text{vol}(\Lambda) > \epsilon$ **do**

 Let λ^0 be the analytic center of Λ ;

 Construct the strict dominance regions R_i^+ with multiplier vector λ^0 ;

 Allocate the remaining area in ambiguous dominance regions lexicographically; that is, assign region R_k^- to the facility i with minimal index $i \in \mathcal{I}_k$;

 /* This lexicographic allocation will not generally be feasible for the primal problem. */

for $i \in \{1, \dots, n\}$ **do**

 | Set $g_i := \iint_{R_i} f(x) dA$;

end

 Set $\Lambda := \Lambda \cap \{\lambda : \mathbf{g}^T \lambda \geq \mathbf{g}^T \lambda^0\}$;

end

Let λ^0 be the analytic center of Λ ;

Construct the strict dominance regions R_i^+ with multiplier vector λ^0 ;

Allocate the remaining area in ambiguous dominance regions so that $\iint_{R_i} f(x) dA = c_i$ for all i ;

return $\{R_1, \dots, R_n\}$;

Algorithm 1: Algorithm ObstaclePartition($R, O, P, f(\cdot), \mathbf{c}$) partitions a region with obstacles into compact relatively star-convex sub-regions.

It follows that, if our desired error in integration is ϵ in our problem, then the maximum length of any edge in the triangulation must be at most $\epsilon^{1/2}(\text{Area}(R)M_2K_1)^{-1/2}$ (since we have hyperbolic arcs separating the R_i , an exact triangulation is impossible, but the added computational complexity therein is beyond the scope of this paper). If we break R into N_T triangles, the maximum edge length will generally be $\mathcal{O}(\sqrt{\text{Area}(R)/N_T})$ and we therefore need to break R into $\mathcal{O}(\text{Area}(R)^2 M_2 / \epsilon)$ triangles. Thus, the added complexity of non-uniform $f(\cdot)$ is quadratic in $\text{Area}(R)$, linear in the maximum modules of the second derivatives of $f(\cdot)$, and inversely proportional to the desired precision ϵ . Each iteration of Algorithm 1 therefore requires $\mathcal{O}(N^{5/3} + \text{Area}(R)^2 M_2 / \epsilon)$ total steps.

6 Computational experiments

In this section, we present the results of two numerical simulations based on two natural practical applications. The first example involves designing districts for “Code Blue” emergency telephones within a university campus. Code Blue emergency telephones are located throughout many university campuses for safety enhancement. In our simulation, we assume that the phones are located in fixed positions, as obtained from a map of the St. Paul campus of the University of Minnesota; the input is shown in Figure 8a. We consider the buildings in the area as impenetrable obstacles. Assuming that the demand for these phones arises uniformly throughout the campus, our aim is therefore to divide the map into compact sub-regions so that each sub-region contains a phone and all sub-regions have equal area.

In the second simulation, we consider a military conflict zone which is located in a hilly region. Unmanned aerial vehicles (UAV’s) are used for aerial surveillance to monitor activities in such regions. A fixed number of UAV’s are available for this purpose. Our task is to strategically divide the area into sub-regions such that the total work load for all the UAV’s is minimized, i.e. such that all regions have equal area, as in the second example of Section 2. Typical low-cost UAV’s used by the military for aerial surveillance have a *service ceiling*, or maximum usable altitude, of 5000 meters. As a sample dataset we use the troubled region of Jammu & Kashmir close to the India-Pakistan border in the Himalayan mountains, whose high altitude peaks pose an obstruction for the UAV’s. Using a contour map of the Drass Sector obtained from Google Maps, we set all regions with an altitude exceeding

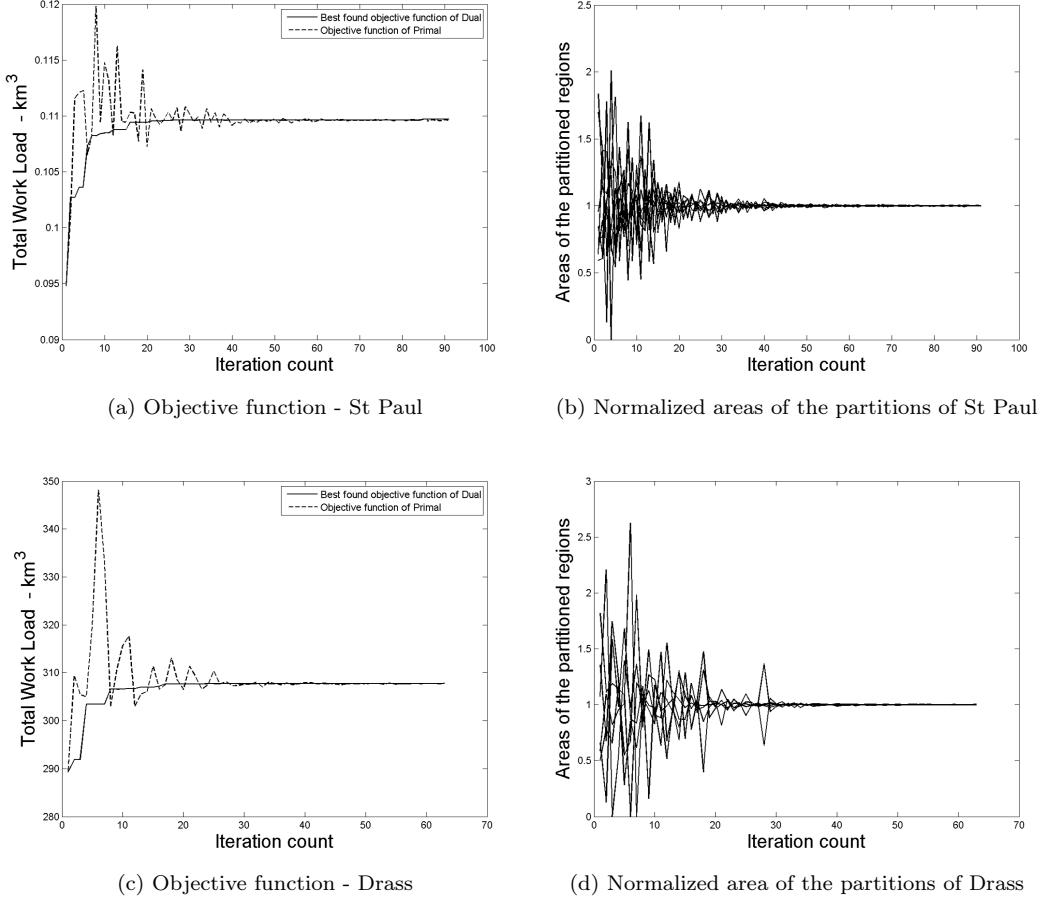


Figure 6: Convergence of the analytic center cutting plane algorithm for St Paul and Drass

5000m as “obstacles” for the UAV’s and built polygonal approximations for them. We choose six UAV launch sites randomly in the basin of the region since one generally prefers a flat terrain to launch such vehicles. The setup to this simulation is shown in Figure 9a.

6.1 Finding optimal partitions

Given a set of fixed facilities P (the emergency phones or the UAV launch sites), our objective is to find a globally optimal partition of the map C by solving the dual problem (4). We implemented Algorithm 1 in C++ on a Dell E8400 computer. We found that Algorithm 1 converged to a globally optimal partition within a tolerance of 0.01% after 89 and 48 iterations for the St. Paul and Drass cases respectively, as shown in Figure 6. The optimal partitions for these problems are shown in Figures 8b and 9b.

6.2 Variable facilities

In this section we present the results of a simulation in which facility positions are also allowed to vary, in addition to the partitions. Since determining the optimal placement of facilities is a non-convex problem, we are only able to look for locally optimal solutions. To this end, we use a natural variation of the well-known *Lloyd algorithm* for computing centroidal Voronoi partitions [30]. The variation is straightforward: for a given initial set of facilities P , we compute the optimal partition using Algorithm 1 as before. Then, we relocate each facility p_i to the geometric median (the *Fermat-Weber center*) of its associated region R_i . The new optimal partitions are then re-computed, and so on and so forth. This algorithm is guaranteed to converge to a solution because the objective function value (the total weighted distance between facilities and their assigned sub-regions) decreases at each of the two steps.

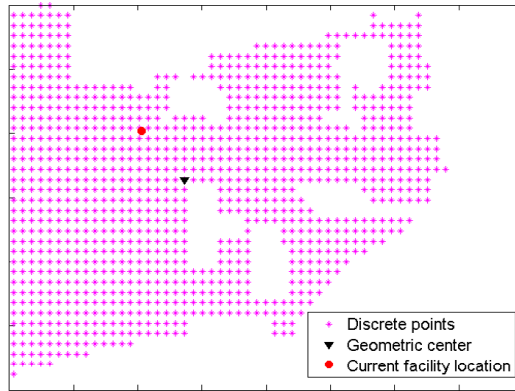


Figure 7: Moving the facility to the geometric median of the corresponding discretized partition

The problem of locating each facility p_i to the geometric median of its associated region R_i is called the *continuous Fermat-Weber problem* and was studied in [13]. For our purposes, we merely estimated the geometric median by discretizing R_i and selecting the best point in the discretization, as shown in Figure 7.

The locally optimal facility locations and their corresponding partitions for the St. Paul and Drass maps are shown in Figures 8c and 9c. The algorithm takes fewer than 15 iterations to converge to the locally optimum solution in both cases and the convergence to this local optimum is shown in Figure 10.

7 Acknowledgments

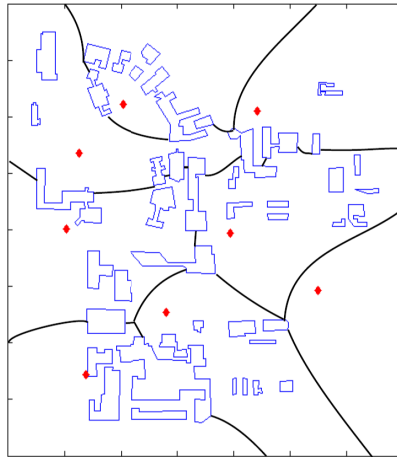
The authors thank Yinyu Ye and Shuzhong Zhang for helpful discussions.

References

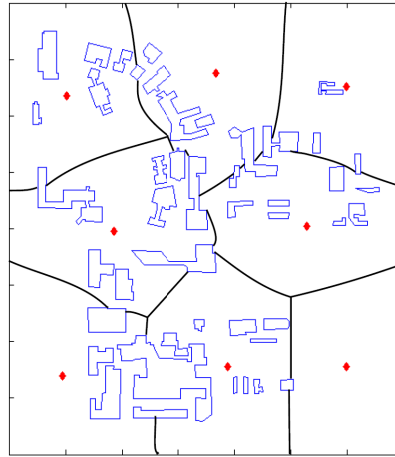
- [1] B. Aronov, P. Carmi, and M.J. Katz. Minimum-cost load-balancing partitions. *Algorithmica*, 54(3):318–336, July 2009.
- [2] F. Aurenhammer, F. Hoffmann, and B. Aronov. Minkowski-type theorems and least-squares partitioning. In *Proceedings of the eighth annual symposium on Computational geometry*, SCG ’92, pages 350–357, New York, NY, USA, 1992. ACM.
- [3] Y. Azar. On-line load balancing. In *Online Algorithms*, volume 1442 of *Lecture Notes in Computer Science*, pages 178–195. Springer Berlin / Heidelberg, 1998.
- [4] O. Baron, O. Berman, D. Krass, and Q. Wang. The equitable location problem on the plane. *European Journal of Operational Research*, 183:578–590, 2007.
- [5] S. Bereg, P. Bose, and D. Kirkpatrick. Equitable subdivisions within polygonal regions. *Computational Geometry*, 34(1):20 – 27, 2006. Special Issue on the Japan Conference on Discrete and Computational Geometry 2004.
- [6] Oded Berman, Zvi Drezner, Dmitry Krass, and George O. Wesolowsky. The variable radius covering problem. *European Journal of Operational Research*, 196(2):516 – 525, 2009.
- [7] Oded Berman, Zvi Drezner, Arie Tamir, and George O. Wesolowsky. Optimal location with equitable loads. *Annals of Operations Research*, 167(1):307–325, March 2009.
- [8] J.G. Carlsson. Dividing a territory among several vehicles. *INFORMS Journal on Computing*, To appear, 2011.
- [9] J.G. Carlsson, B. Armbruster, and Y. Ye. Finding equitable convex partitions of points in a polygon efficiently. *ACM Transactions on Algorithms*, 6, September 2010.



(a) A map of the St. Paul campus.

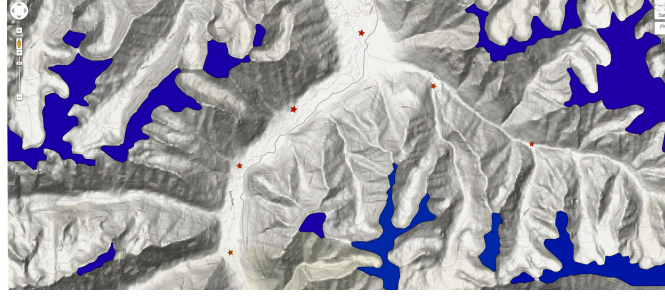


(b) Partitions based on minimizing total work load with static facilities

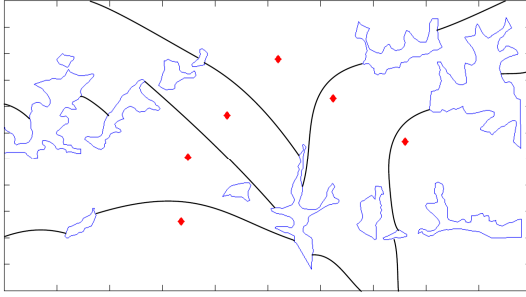


(c) Variable facilities: Lloyd's algorithm result

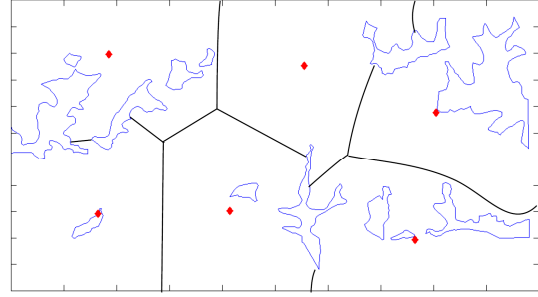
Figure 8: St Paul results



(a) A map of Drass, India, taken from Google Maps. The shaded blue regions denote the obstacles, i.e. the areas exceeding 5000m in altitude.

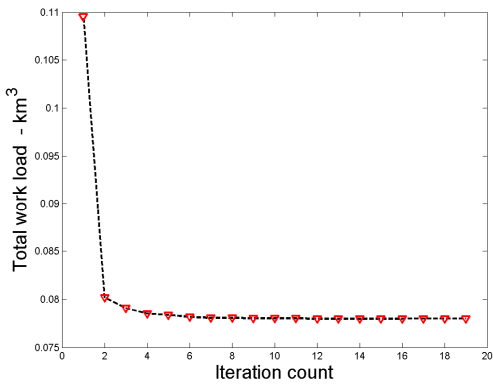


(b) Partitions based on minimizing total work load with static facilities

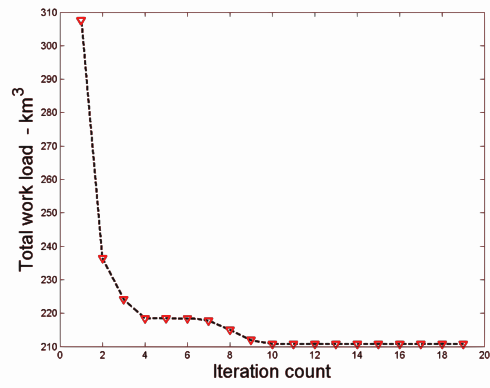


(c) Variable facilities: Lloyd's algorithm result

Figure 9: Drass results



(a) Lloyd algorithm iterations - StPaul



(b) Lloyd algorithm iterations - Drass

Figure 10: Total work load v/s iteration count

- [10] J.G. Carlsson, D. Ge, A. Subramaniam, and Y. Ye. Solving the min-max multi-depot vehicle routing problem. In *Proceedings of the FIELDS Workshop on Global Optimization*, 2007.
- [11] Z. Drezner and A. Suzuki. Covering continuous demand in the plane. *Journal of the Operational Research Society*, 61(5):878–881, 2010.
- [12] J.W. Durham, R. Carli, P. Frasca, and F. Bullo. Discrete partitioning and coverage control for gossiping robots. *IEEE Transactions on Robotics*, 99:1 – 15, 2011.
- [13] Sándor P. Fekete, Joseph S. B. Mitchell, and Karin Beurer. On the continuous Fermat-Weber problem. *Oper. Res.*, 53:61–76, January 2005.
- [14] J.-L. Goffin, Z.-Q. Luo, and Y. Ye. Complexity analysis of an interior cutting plane method for convex feasibility problems. *SIAM J. on Optimization*, 6:638–652, March 1996.
- [15] D. Haugland, S. C. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997 – 1010, 2007.
- [16] Y. He and Z. Tan. Ordinal on-line scheduling for maximizing the minimum machine completion time. *Journal of Combinatorial Optimization*, 6(2):199–206, 2002.
- [17] E. Isaacson and H.B. Keller. *Analysis of Numerical Methods*. Dover books on advanced mathematics. Dover Publications, 1994.
- [18] M. Jäger and B. Nebel. Dynamic decentralized area partitioning for cooperating cleaning robots. In *ICRA 2002*, pages 3577–3582, 2002.
- [19] H. Kellerer, V. Kotov, M. Grazia Speranza, and Z. Tuza. Semi on-line algorithms for the partition problem. *Operations Research Letters*, 21(5):235 – 242, 1997.
- [20] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., 1st edition, 1997.
- [21] Q. Mérigot. A multiscale approach to optimal transport. *Computer Graphics Forum*, 30(5):1583–1592, 2011.
- [22] R.S. Middleton and J.M. Bielicki. A scalable infrastructure model for carbon capture and storage: SimCCS. *Energy Policy*, 37(3):1052–1060, March 2009.
- [23] J.S.B. Mitchell. Shortest paths among obstacles in the plane. In *Proceedings of the ninth annual symposium on Computational geometry*, SCG ’93, pages 308–317, New York, NY, USA, 1993. ACM.
- [24] M. Pavone, N. Bisnik, E. Frazzoli, and V. Isler. Decentralized vehicle routing in a stochastic and dynamic environment with customer impatience. In *RoboComm ’07: Proceedings of the 1st international conference on Robot communication and coordination*, pages 1–8. IEEE Press, 2007.
- [25] Marco Pavone, Alessandro Arsie, Emilio Frazzoli, and Francesco Bullo. Distributed policies for equitable partitioning: theory and applications. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4191–4197, Piscataway, NJ, USA, 2008. IEEE Press.
- [26] Marco Pavone, Alessandro Arsie, Emilio Frazzoli, and Francesco Bullo. Equitable partitioning policies for robotic networks. In *ICRA’09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 3979–3984, Piscataway, NJ, USA, 2009. IEEE Press.
- [27] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira. Sensing and coverage for a network of heterogeneous robots. In *IEEE CDC’08*, pages 3947–3952, 2008.
- [28] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in applied mathematics. Springer, 2007.
- [29] C. Villani. *Topics in optimal transportation*. Graduate studies in mathematics. American Mathematical Society, 2003.
- [30] Wikipedia. Lloyd’s algorithm, 2012. [Online; accessed 05-Feb-2012].