Dividing a territory among several vehicles

John Gunnar Carlsson*

October 29, 2011

Abstract

We consider an uncapacitated stochastic vehicle routing problem in which vehicle depot locations are fixed and client locations in a service region are unknown, but are assumed to be independent and identically distributed samples from a given probability density function. We present an algorithm for partitioning the service region into sub-regions so as to balance the workloads of all vehicles when the service region is simply connected and point-to-point distances follow some "natural" metric, such as any L_p norm. This algorithm can also be applied to load-balancing of other combinatorial structures, such as minimum spanning trees and minimum matchings.

1 Introduction

Optimal assignment of a workload between several agents is a common objective that is encountered in resource allocation problems. Frequently, workloads are assigned in such a way as to minimize the total amount of work done by all agents. In other situations, one may want an *equitable* assignment that balances the workload evenly across all agents. Equitable assignment policies are commonly encountered in queueing theory [3, 18, 21], vehicle routing [11, 17, 26], facility location [2, 4, 7, 15], and robotics [19, 24, 25], among others.

Our motivation for this research comes from an industrial affiliate in the form of a stochastic vehicle routing problem. Our objective is to partition a geometric region so as to assign workloads to vehicles in an equitable fashion. Partitioning and routing occupy two different strategic tiers in the optimization hierarchy; partitioning is done at a (high) tactical management level, while routing optimization is operational and made on a day-to-day basis. Hence, a natural strategy, especially in the presence of uncertainty, is to segment the service region into a collection of sub-regions and then to solve each routing sub-problem induced at the sub-regions independently of the others. This approach was used, for example, by [17], who treated the problem as a two-stage optimization problem (partitioning and routing) and implemented a tabu search and multistart heuristic to consider the problem of partitioning a planar graph optimally. This problem is also often considered in the context of facility location [2, 4, 15] and robotics [25].

In this paper, we give an algorithm that takes as input a planar, simply connected region R, together with a probability density $f(\cdot)$ defined on R. Contained in R is a collection of *n depot points* $P = \{p_1, \ldots, p_n\}$, representing the starting locations of a fleet of vehicles. We assume (purely for expositional purposes, since points p_i may be arbitrarily close to each other) that each point p_i corresponds to exactly one vehicle. The vehicles must visit clients whose exact locations are unknown, but are assumed to be independent and identically distributed (i.i.d.) samples from the density $f(\cdot)$. Our goal is to partition R into n disjoint sub-regions, with one vehicle assigned to each sub-region, so that the workloads in all sub-regions are asymptotically equal when a large number of samples is drawn. For each sub-region R_i, we will solve a travelling salesman problem, in which the point set consists of a depot point plus all points in R_i. See Figure 1.

As we will show, our problem turns out to be a special case of the *equitable partitioning problem*, in which we are given a pair of densities $\lambda(\cdot)$ and $\mu(\cdot)$ on a region R and we want to partition R into n sub-regions R_i with $\iint_{R_i} \lambda(\cdot)\, dA = \frac{1}{n} \iint_R \lambda(\cdot)\, dA$ and $\iint_{R_i} \mu(\cdot)\, dA = \frac{1}{n} \iint_R \mu(\cdot)\, dA$ for all i. In our problem, one density is an atomic

*Industrial and Systems Engineering, University of Minnesota. The author gratefully acknowledges the support of the Boeing Company. This research is supported in part by NSF Grant GOALI #0800151.

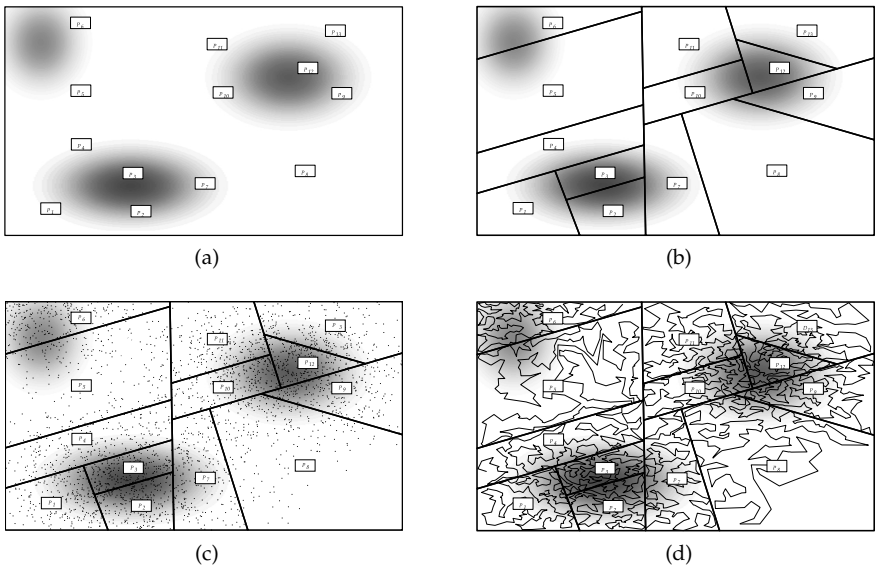


Figure 1: Inputs and outputs to our problem. We begin with a depot set and a density $f(\cdot)$ defined on a region R (1a), which we then partition (1b). This partition should be constructed so that, when points are sampled independently from $f(\cdot)$ (1c), the TSP tours of all the points in each sub-region are asymptotically equal (1d).

measure that represents the set of depots and the other represents the TSP workload over a sub-region when points are sampled from $f(\cdot)$. Theorem 12 of [8] states that an equitable partition always exists when $\lambda(\cdot)$ and $\mu(\cdot)$ are absolutely continuous, R is convex, and all R_i are required to be convex. Theorem 1 of [6] implicitly extends this result for the case that R is simply connected and all R_i must be *relatively* convex to R. The case where $\lambda(\cdot)$ and $\mu(\cdot)$ are both atomic measures consisting of gn and hn points for some positive integers g and h is a well-studied problem in computational geometry known as a *red-blue* partition [6, 8, 20], and several fast algorithms are already known for this problem.

The case where $\lambda(\cdot)$ is an atomic distribution, $\mu(\cdot)$ is a uniform distribution, and R is convex (as opposed to merely simply connected as in this paper) was solved in [10], which also suggests an extension to the non-uniform (convex) case. As was noted previously in [10], one approximate approach to our vehicle routing problem is to discretize the input data and use a red-blue partitioning algorithm. However, Section 2 of that same paper shows that this approach requires at least $\mathcal{O}\left(\epsilon^{-8/3}\right)$ time to find an ϵ-approximate solution. The algorithm presented in this paper uses the method of bisection to find partitions and its running time is shown to be $\mathcal{O}\left(nN\log\frac{N}{\epsilon}\right)$, where N is the total number of depot points and vertices of the input polygon. Furthermore, our approach does not require any discretization. When certain conditions are met (described at the end of Section 3), the running time is in fact shown to be $\mathcal{O}(nN\log N)$.

The outline of this paper is as follows: in Section 2, we describe a necessary condition for optimality of a partition of R that follows immediately from well-known results from geometric probability. In Section 3 we give an algorithm that finds an optimal partition of R when R is a simply connected polygon. In Section 4 we present some simulation results that show the solution quality of our algorithm when applied in a practical setting.

2 Summary of key facts and findings from related work

In this section we summarize the important theoretical results that form the basis of our partitioning algorithm. We consider the travelling salesman problem (TSP) in a planar region R, where the distance between two points is Euclidean, or any other "natural" metric such as the Manhattan or sup norm. The well-known *BHH Theorem* [5] says that the length of an optimal TSP tour of a set of points follows a law of large numbers:

Theorem 1. *Suppose that $\{X_i\}$ is a sequence of random points i.i.d. according to a probability density function $f(\cdot)$ defined on a compact planar region R. Then with probability one, the length* $\mathrm{TSP}(\{X_1, \ldots, X_k\})$ *of the optimal travelling salesman tour traversing points $\{X_1, \ldots, X_k\}$ satisfies*

$$\lim_{k\to\infty} \frac{\mathrm{TSP}(\{X_1, \ldots, X_k\})}{\sqrt{k}} = \beta \iint_R \sqrt{f_c(x)}\, dA \tag{1}$$

where β is a constant and $f_c(\cdot)$ represents the absolutely continuous part of $f(\cdot)$.

It is additionally known that $0.6250 \leq \beta \leq 0.9204$ [1]. This result was subsequently improved in [27], which showed that a similar law of large numbers holds for any *subadditive Euclidean functional*, such as a minimum-weight matching, minimum spanning tree, Steiner tree, or Delaunay triangulation, with different constants β. Applying a standard coupling argument to (1) gives the following result:

Theorem 2. *Let R be a compact planar region and let $f(\cdot)$ be an absolutely continuous probability density defined on R. Let $\{X_i\}$ be a collection of i.i.d samples drawn from $f(\cdot)$. Let $\{R_1, \ldots, R_n\}$ be a partition of R. If a partition of R into n disjoint pieces $R_1, \ldots, R_n$ satisfies*

$$\iint_{R_i} \sqrt{f(x)}\, dA = \frac{1}{n} \iint_R \sqrt{f(x)}\, dA \tag{2}$$

for $i \in \{1, \ldots, n\}$, then asymptotically, the lengths of the TSP tours $\mathrm{TSP}(\{X_1, \ldots, X_k\} \cap R_i)$ *will differ by a term of order $o(\sqrt{k})$, where k is the number of points sampled. Hence, the maximum tour length over any sub-region R_i differs from the optimal solution by a term of order $o(\sqrt{k})$.*

The uniform case of (2) is particularly well-known and has been used extensively to estimate TSP tour lengths and to validate heuristics for solving TSP [16, 22]:

Corollary 3. *As above, when $f(\cdot)$ is the uniform distribution on R, if a partition of R into n disjoint pieces $\{R_1, \ldots, R_n\}$ satisfies*

$$\mathrm{Area}(R_i) = \mathrm{Area}(R)/n$$

then asymptotically, the lengths of the TSP tours $\mathrm{TSP}(\{X_1, \ldots, X_k\} \cap R_i)$ *will differ by a term of order $o(\sqrt{k})$.*

Remark 4. Section 3.7 of [28] also proves that β satisfies

$$\mathbf{E}\,\mathrm{TSP}(\{X_1, \ldots, X_k\}) - \beta\sqrt{k}\,\mathrm{Area}(R) = \mathcal{O}(1)$$

for all k, when $\{X_i\}$ is a sequence of uniform independent samples on R. This says that the expected tour lengths in Corollary 3 will differ by a constant.

Remark 5. We can extend Theorem 2 slightly to more general metrics than those described. Specifically, if we are given a "roughness function" $r(\cdot)$, so that the straight-line distance from point u to point v is the line integral $\int_{\overline{uv}} r(x)\, ds$, then using a coupling argument similar to page 36 of [28], we find that the necessary optimality condition is

$$\iint_{R_i} r(x) \sqrt{f(x)}\, dA = \frac{1}{n} \iint_R r(x) \sqrt{f(x)}\, dA\,.$$

This is useful for modelling situations with variable terrain.

3 The equitable partitioning problem on a simply connected service region

In the preceding section we described a necessary optimality condition for optimally partitioning a region. Here we describe the precise formulation of our partitioning problem, including the additional constraints that are imposed by the depot locations and the shape that we require our partition sub-regions to take. We then present our fast algorithm for solving the problem.

3.1 Analysis

The optimality condition defined in Theorem 2 is easy to achieve, in the absence of other criteria; for example, a partition might consist exclusively of vertical lines, with each vertical strip cutting off $\iint_{\text{strip}} \sqrt{f(x)}\, dA = \frac{1}{n} \iint_R \sqrt{f(x)}\, dA$. For this reason, we will impose additional constraints on our algorithm that should, in principle, give a better solution. Recall that in our original problem statement, we assumed that our service region R contained a set of *depot points* $P = \{p_1, \ldots, p_n\}$. A natural constraint to impose is that each sub-region R_i should contain the depot point that we have assigned to it.

This still leaves us with considerable freedom; we have not yet imposed any constraints on the shape of the sub-regions. For example, one would expect that sub-regions thus assigned should at least be connected. A further property that might be desired is that for any two points $u, v \in R_i$, the shortest path between u and v be contained in R_i. When the input region R is convex, this constraint is equivalent to requiring that each sub-region R_i also be convex. When R is not convex, the property that we desire is called *relative convexity* [6]: each sub-region R_i should be convex "relative" to the input region R, so that the shortest path between $u, v \in R_i$ (which may not be a straight line) must itself be contained in R_i. Hence, the algorithm we desire is expressed as follows:

Input: A simply connected polygon S, a point set $P = \{p_1, \ldots, p_n\} \subset S$, and a probability density $\mu(\cdot)$ defined on S.

Output: A partition of S into n sub-regions, each satisfying the following properties:

1. Each sub-region R_i contains exactly one element p_i of P.
2. For any two points $u, v \in R_i$, the shortest path from u to v in R is contained in R_i.
3. All sub-regions satisfy $\iint_{R_i} \mu(x)\, dA = \frac{1}{n} \iint_R \mu(x)\, dA$.

If $f(\cdot)$ describes the density of the client locations, we set $\mu(\cdot) \equiv \sqrt{f(\cdot)}$ in the above, as in Theorem 2. In the following section we give an algorithm that solves our above problem where $\mu(\cdot)$ is an absolutely continuous probability density that satisfies certain natural conditions (described in Section 4).

3.2 Our algorithm

The algorithm that we describe is recursive, constructing *equitable partitions* at each of $\leq n$ iterations:

Definition 6. Let R be a compact planar region and let $\mu(\cdot)$ be an absolutely continuous probability density defined on R. Let $P = \{p_1, \cdots, p_n\} \subset R$ denote a set of n points. A partition $\{R_1, \ldots, R_k\}$ of R into k sub-regions is said to be an *equitable k-partition* if we have

$$\frac{\iint_{R_i} \mu(x)\, dA}{|P \cap R_i|} = \frac{\iint_R \mu(x)\, dA}{n}$$

for all sub-regions R_i. If each sub-region R_i is (relatively) convex, then $R_1, \ldots, R_k$ is said to be an *equitable (relatively) convex k-partition*.

To begin, we let $S \subset \mathbb{R}^2$ be a simply connected polygon with m vertices and $P \subset S$ the set of depot points, with $|P| = n$. Let $r < m$ be the number of *reflex vertices* of S, that is, the vertices that form an interior angle exceeding 180°. As mentioned earlier, each sub-region that our algorithm returns must be relatively convex.

Definition 7. An *S-geodesic* between two points u and v in a simple polygon S, written $G(u, v \,|\, S)$, is the shortest path between u and v contained in S.

Definition 8. A sub-region $\tilde{S}$ of a simple polygon S is *relatively convex* to S if, for every pair of points $u, v \in \tilde{S}$, the S-geodesic $G(u, v \,|\, S)$ lies in $\tilde{S}$.

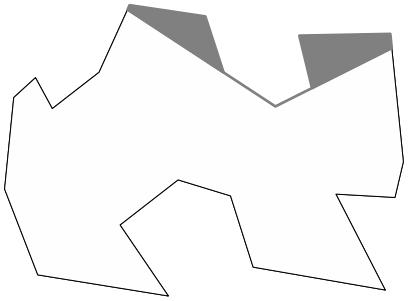


Figure 2: The shaded region is relatively convex to the polygon containing it.

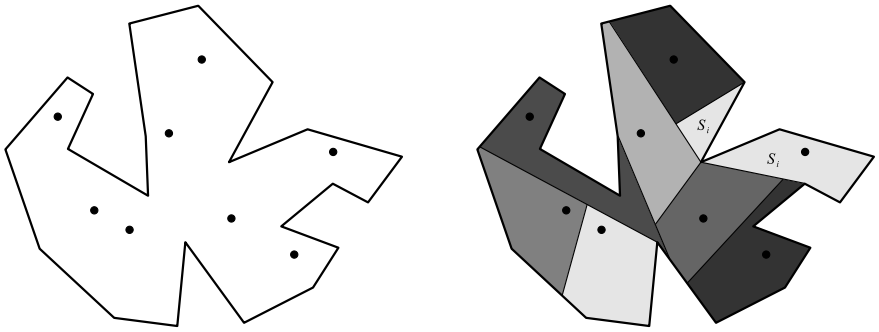


Figure 3: Inputs S and P (3a) and output (3b) to our problem, where $\mu(\cdot)$ is the uniform distribution on S. Note that the region marked S_i consists of two polygons joined at a vertex, but still satisfies our relative convexity constraint.

3.2.1 Preliminaries

Note that a relatively convex sub-region may contain degeneracies, as shown in Figure 2. In this section we will prove the following result:

Theorem 9. *Given a simple polygon S with m vertices with a probability density $\mu(\cdot)$ defined on S and a collection of points $P = \{p_1, \ldots, p_n\} \subset S$ where the vertices of S and the points in P are all in general position, there exists a partition of S into n relatively convex sub-regions $S_1, \ldots, S_n$ with disjoint interiors, where each sub-region S_i contains exactly one point from P and satisfies $\int_{S_i} \mu(x)\, dA = 1/n$. Furthermore, we can find such a partition in running time $\mathcal{O}(nN \log N)$, where $N = m + n$.*

Remark. We require the interiors be disjoint (as opposed to the regions S_i themselves) so as to allow ourselves to interpret boundaries between sub-regions liberally; a point p_i lying on the boundary of two sub-regions S_i, S_j can be assigned to either region (but only one of the two). In addition, it may be necessary for two sub-regions to share a reflex vertex or edge. In practice, the sub-region boundaries can be "perturbed" at each step of the algorithm, so as to ensure that $p_i \in \text{int}(S_i)$ provided $p_i \in \text{int}(S)$.

For the remainder of this section, we will assume for ease of exposition that $\mu(\cdot)$ is the uniform distribution, so that $\iint_{S_i} \mu(x)\, dA = \text{Area}(S_i)$. In the end of Section 4 we describe the necessary conditions on $\mu(\cdot)$ for the running time described in Theorem 9 to be achievable. We also assume without loss of generality that $\text{Area}(S) = 1$. An example of the input and output of our algorithm is shown in Figure 3. We let ∂ denote the *boundary* operator, e.g. ∂S denotes the boundary of S. The closure and interior are written $\text{cl}(\cdot)$ and $\text{int}(\cdot)$.

Definition 10. Given two points u and v on ∂S, the *left shell* $\mathcal{L}(u, v \,|S)$ consists of all elements of S lying on or to the left of $G(u, v \,|S)$. If u or v does not lie on ∂S, then we define $\mathcal{L}(u, v) = \mathcal{L}\left(u', v'\right)$, where u' and v' are obtained by extending the endpoints of $G(u, v \,|S)$ via straight lines to ∂S (see Figure 4).

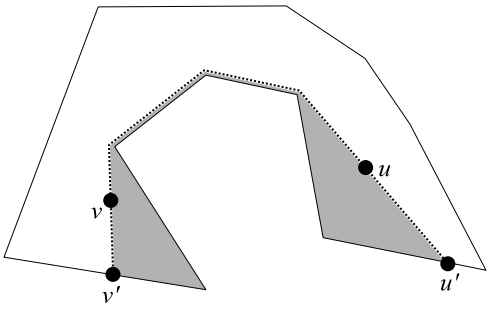


Figure 4: The geodesic $G(u, v|S)$, its extension points u' and v', and the induced left shell $\mathcal{L}(u, v|S) = \mathcal{L}\left(u', v'|S\right)$.

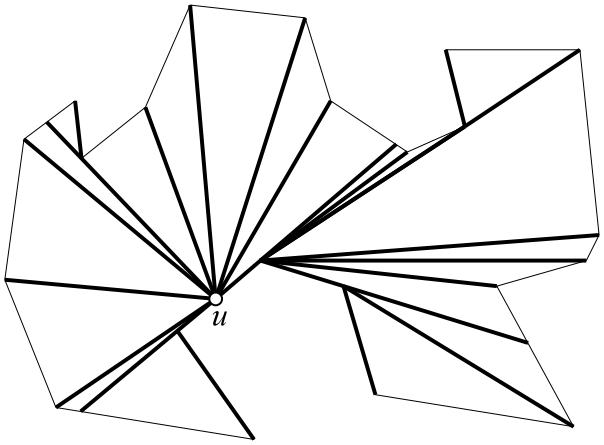


Figure 5: The geodesic shortest-path tree rooted at u.

Remark. It is clear from their definitions that $\mathcal{L}(u, v|S)$ and $\text{cl}(S \setminus \mathcal{L}(u, v|S)) = \mathcal{L}(v, u|S)$ are relatively convex (to S).

Our algorithm will proceed by successively constructing equitable relatively convex 2- and 3-partitions, dividing S into relatively convex sub-regions at each iteration. The same approach is used for the case where $\mu(\cdot)$ is an atomic measure in [6, 8]. The next two lemmas, 11 and 12, are well-known, making use of Chazelle's trapezoidal decomposition algorithm [12] and the *funnel algorithm* of [13, 23]; see [6, 9], for example.

Lemma 11. *We can pre-process S and P so that, for any pair of points u and v on ∂S, we can determine the subset of P lying in $\mathcal{L}(u, v|S)$ in $\mathcal{O}(N)$ time. This pre-processing takes $\mathcal{O}(N \log r)$ time.*

Lemma 12. *Given a point u on ∂S and a positive integer $k \leq n$, we can determine a point v so that $|\mathcal{L}(u, v|S) \cap P| = k$ in $\mathcal{O}(N \log r)$ time.*

Proofs. Omitted. □

Lemma 13. *Given a point u on ∂S and a positive integer $\alpha < 1$, we can determine a point v so that* $\text{Area}(\mathcal{L}(u, v|S)) = \alpha$ *in $\mathcal{O}(m)$ time.*

Proof. Using the funnel algorithm of [13], we can obtain an ordered geodesic shortest-path tree rooted at u to every vertex of S in $\mathcal{O}(m)$ time. Each pair of adjacent siblings in this tree defines a triangle, as shown in Figure 5. Since the vertices of S are sorted initially, we can determine the triangle containing v in $\mathcal{O}(m)$ time (by adding the areas of adjacent triangles until the running sum exceeds α), and then solving a linear equation at that triangle in constant time. □

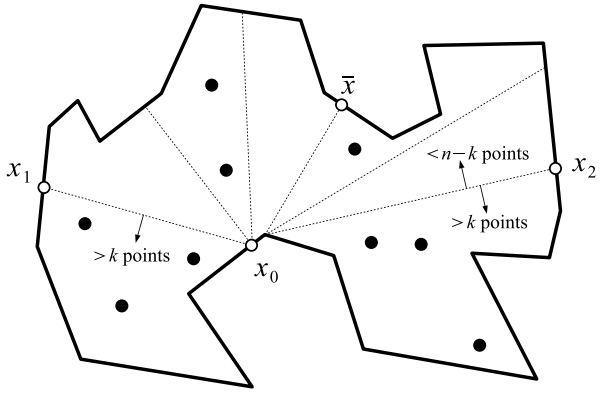


Figure 6: A family of left shells cutting off area $\frac{k}{n}, \frac{k+1}{n}, \ldots, \frac{n-k}{n}$, with $k = 2$ and $n = 9$.

Definition 14. Given a point u on ∂S and a positive integer $\alpha < 1$, define $\text{LShell}_{\alpha}(u) := v$ as in Lemma 13.

Theorem 15. *Let x_0 and x_1 be two points on ∂S. If $\text{Area}(\mathcal{L}(x_0, x_1 | S)) = k/n$ for some integer $k \leq n/2$ and $|\mathcal{L}(x_0, x_1 | S) \cap P| > k$, then we can find a relatively convex equitable 2-partition of S and P in running time $\mathcal{O}(N(\log n + \log r)) = \mathcal{O}(N \log N)$, where $N = m + n$.*

Proof. Construct another point x_2 on ∂S so that $\text{Area}(\mathcal{L}(x_2, x_0 | S)) = k/n$. Then either $|\mathcal{L}(x_2, x_0 | S) \cap P| < k$ or $|\mathcal{L}(x_2, x_0 | S) \cap P| > k$ (if we have equality then we are finished), and in either case we can derive an equitable 2-partition:

Case 1 Suppose that $|\mathcal{L}(x_2, x_0 | S) \cap P| > k$. Then $|\mathcal{L}(x_0, x_2 | S) \cap P| < n - k$ and $\text{Area}(\mathcal{L}(x_0, x_2 | S)) = (n-k)/n$. Hence, $\mathcal{L}(x_0, x_1 | S)$ contains too many points (relative to its area) and $\mathcal{L}(x_0, x_2 | S)$ contains too few points. Consider a family of left shells $\mathcal{L}(x_0, x | S)$, where x traverses ∂S clockwise from x_1 to x_2; see Figure 6. The function $\phi(x) := \text{Area}(\mathcal{L}(x_0, x | S)) - \frac{k}{n} |\mathcal{L}(x_0, x | S) \cap P|$ is piecewise continuous, increasing on each of its components, and decreasing at each discontinuity. Since $\phi(x_1) < 0$ and $\phi(x_2) > 0$, the intermediate value theorem guarantees the existence of a point $\bar{x}$ where $\phi(\bar{x}) = 0$ and our equitable 2-partition is obtained. We can find this by performing a binary search for $i \in \{k, \ldots, n-k\}$, where for each i we compute the point $\text{LShell}_{i/n}(x_0)$ and the number of points contained therein. The preceding argument guarantees that we must find an equitable 2-partition somewhere in this procedure, which can be performed in running time $\mathcal{O}(N \log r)$ (preprocessing) + $\mathcal{O}(N \log n) = \mathcal{O}(N \log N)$ running time.

Case 2 Suppose that $|\mathcal{L}(x_2, x_0 | S) \cap P| < k$. Then, as $|\mathcal{L}(x_0, x_1 | S) \cap P| > k$, we have a left shell containing too many points (relative to its area) and another left shell containing too few points. Hence, there must exist some pair of points $\bar{x}, \tilde{x}$ in ∂S such that $\bar{x} \in \partial\mathcal{L}(x_0, x_2 | S)$ and $\tilde{x} \in \partial\mathcal{L}(x_1, x_0 | S)$ (see Figure 7), where $\text{Area}(\mathcal{L}(\bar{x}, \tilde{x} | S)) = k/n$ and $|\mathcal{L}(\bar{x}, \tilde{x} | S) \cap P| = k$. This is because the function $\text{LShell}_{k/n}(x)$ is continuous in x (for $x \in \partial S$), and the assumption that our points lie in general position ensures that as x traverses ∂S from x_0 to x_2, the elements of P will enter and exit $\mathcal{L}(x, \text{LShell}_{k/n}(x))$ one by one. See the appendix for the proof of the running time of this subroutine. □

Corollary 16. *Let x_0 and x_1 be two points on ∂S. If $\text{Area}(\mathcal{L}(x_0, x_1 | S)) < k/n$ for some integer $k \leq n/2$ and $|\mathcal{L}(x_0, x_1 | S) \cap P| = k$, then we can find a relatively convex equitable 2-partition of S and P in running time $\mathcal{O}(N \log N)$.*

Proof. If $\text{Area}(\mathcal{L}(x_0, x_1 | S)) < k/n$, then $\mathcal{L}(x_0, x_1 | S) \subset \mathcal{L}(x_0, \text{LShell}_{k/n}(x_0) | S)$ so that $|\mathcal{L}(x_0, \text{LShell}_{k/n}(x_0) | S) \cap P| \geq k$ and we apply Theorem 15. □

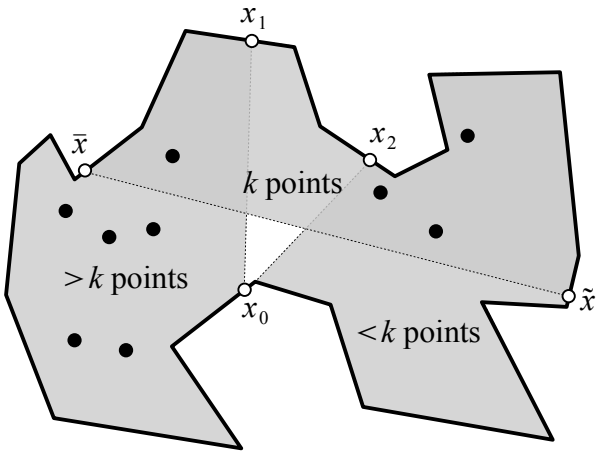


Figure 7: An equitable geodesic shell exists between $\bar{x}$ and $\tilde{x}$ with $k = 4$ and $n = 9$.

3.2.2 The algorithm

Using Theorem 15, we will describe an algorithm in this section that finds either an equitable 2- or 3-partition of a simple polygon and a point set. We can apply this algorithm in a "divide-and-conquer" fashion and thus make an equitable partition of the entire region S after repeated applications. The remainder of this section consists of the proof of the following theorem:

Theorem 17. *Given a simple polygon S with m vertices with a probability density $\mu(\cdot)$ defined on S and a collection of points $P = \{p_1, \ldots, p_n\} \subset S$ where the vertices of S and the points in P are all in general position, there exists either an equitable 2-partition or an equitable 3-partition of S and P, and we can find such a partition in running time $\mathcal{O}(N \log N)$, where $N = m + n$.*

This will suffice to prove Theorem 9, for the following reasons: first, observe that our partitioning process must never introduce any new reflex vertices (otherwise we would lose relative convexity). Hence, none of the sub-regions defined by our algorithm will ever have more than $2r$ reflex vertices (degenerate polygons may count a reflex vertex twice in traversing the boundary). Furthermore, the procedure we outline below will never introduce more than 3 new non-reflex vertices in any sub-region in any iteration (since it constructs either equitable 2- or 3-partitions); therefore, at any iteration of our algorithm we are working with a polygon having at most $m + 3n + 2r = \mathcal{O}(N)$ vertices, containing at most n points. Therefore, every iteration of our algorithm can be performed in running time $\mathcal{O}(N \log N)$, and there are at most n iterations that we must perform, giving a final running time $\mathcal{O}(nN \log N)$.

Notice that if n is even, then Theorem 15 immediately guarantees the existence of a *ham-sandwich* cut, that is, an equitable 2-partition of S and P containing $n/2$ points in either side and we are done. Therefore, we can assume without loss of generality that n is odd, and set $k = (n-1)/2$. To begin, we form the *geodesic hull* of all points P in running time $\mathcal{O}(N \log N)$ (using the optimal algorithm of [29]):

Definition 18. Given a simple polygon S and a collection of points $P \subset S$, the *geodesic convex hull* $GH(P|S)$ of P in S is the minimum perimeter circuit that lies in S and encloses P.

We construct $GH(P|S)$, obtaining a sorted list of boundary points $p_0, \ldots, p_{n'}$ in clockwise order and interior points $p_{n'+1}, \ldots, p_{n-1}$ as shown in Figure 8. We choose boundary point $p_0 \in P$ and determine a geodesic $G(x_0, x_1|S)$ through p_0 that has k points lying strictly to its right and left, using Lemma 12 . If either $\text{Area}(\mathcal{L}(x_0, x_1|S)) < k/n$ or $\text{Area}(\mathcal{L}(x_1, x_0|S)) < k/n$, then we have a certificate for the existence of an equitable 2-partition and we apply Corollary 16. Hence, we assume without loss of generality that

$$\frac{k}{n} < \text{Area}(\mathcal{L}(x_0, x_1|S)), \text{Area}(\mathcal{L}(x_1, x_0|S)) < \frac{k+1}{n} \tag{3}$$

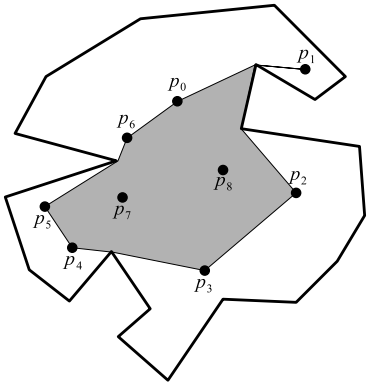


Figure 8: The geodesic convex hull of a point set in a simple polygon, with $n' = 6$.

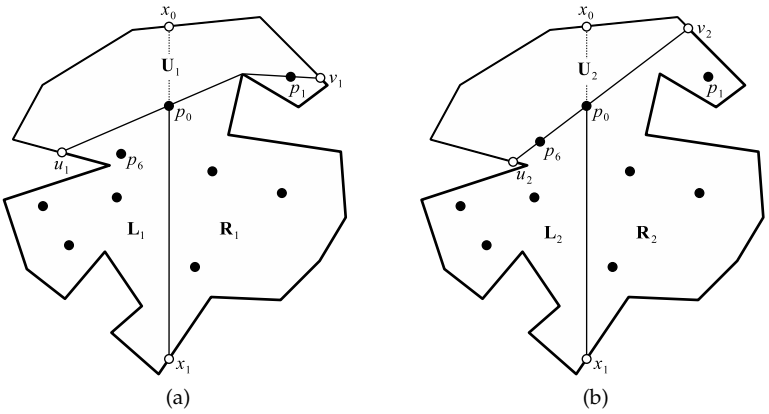


Figure 9: The partitions $\{\mathbf{U}_1, \mathbf{L}_1, \mathbf{R}_1\}$ and $\{\mathbf{U}_2, \mathbf{L}_2, \mathbf{R}_2\}$.

so that both shells are too big to contain k points, but too small to contain $k+1$ points. Since we have failed to find an equitable 2-partition, our algorithm will seek an equitable 3-partition.

Assume that x_0 is on the same "side" of $GH(P|S)$ as p_0, i.e. that $G(x_0, p_0|S) \cap GH(P|S) = p_0$. Extend the geodesic $G(p_0, p_1|S)$ until it intersects ∂S at points u_1 and v_1, where u_1 is on the same side of $GH(P|S)$ as p_0 and v_1 is on the same side of $GH(P|S)$ as p_1. We similarly extend the geodesic $G(p_0, p_{n'}|S)$ until it intersects ∂S at points u_2 and v_2. Consider the 3-partitions $\{\mathbf{U}_1, \mathbf{L}_1, \mathbf{R}_1\}$ and $\{\mathbf{U}_2, \mathbf{L}_2, \mathbf{R}_2\}$ centered at p_0, defined by

$$\begin{aligned}
\mathbf{U}_1 &:= \mathcal{L}(u_1, v_1|S) \\
\mathbf{L}_1 &:= \mathcal{L}(x_1, x_0|S) \cap \mathcal{L}(v_1, u_1|S) \\
\mathbf{R}_1 &:= \mathcal{L}(x_0, x_1|S) \cap \mathcal{L}(v_1, u_1|S) \\
\mathbf{U}_2 &:= \mathcal{L}(u_2, v_2|S) \\
\mathbf{L}_2 &:= \mathcal{L}(x_1, x_0|S) \cap \mathcal{L}(v_2, u_2|S) \\
\mathbf{R}_2 &:= \mathcal{L}(x_0, x_1|S) \cap \mathcal{L}(v_2, u_2|S),
\end{aligned}$$

as shown in Figure 9.

Since $\mathbf{U}_1$ and $\mathbf{U}_2$ can be regarded as containing only one point p_0, we can assume without loss of generality that Area$(\mathbf{U}_1)$, Area$(\mathbf{U}_2) > 1/n$, since otherwise Corollary 16 applies. Hence for $i \in \{1, 2\}$ it must be the case that either Area$(\mathbf{R}_i) < k/n$ or Area$(\mathbf{L}_i) < k/n$ (or both). There are a total of three cases we will consider separately:

1. Area$(\mathbf{L}_1) < k/n$ and Area$(\mathbf{R}_1) < k/n$ (or equivalently Area$(\mathbf{L}_2) < k/n$ and Area$(\mathbf{R}_2) < k/n$).

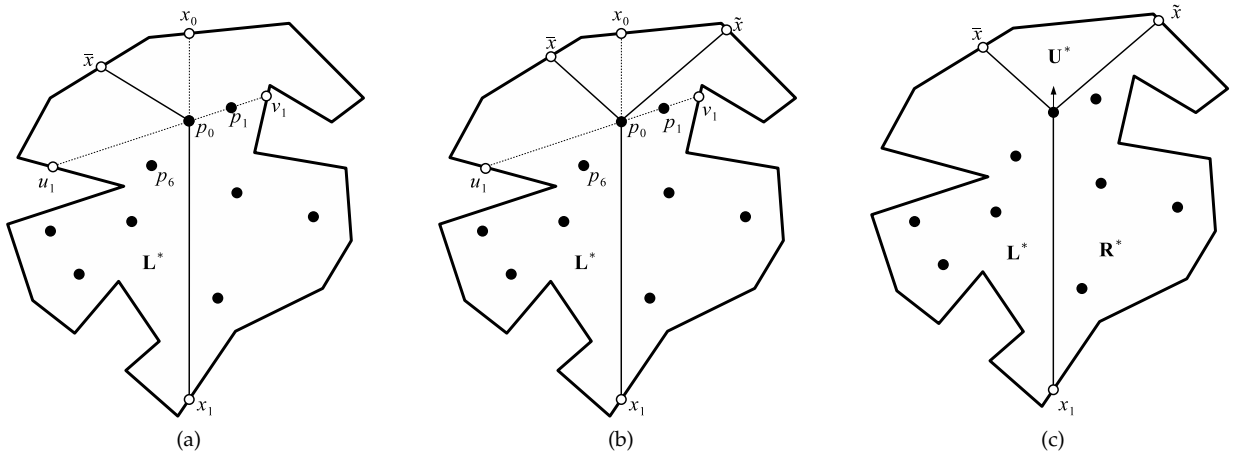


Figure 10: Making an equitable 3-partition $\{\mathbf{U}^*, \mathbf{L}^*, \mathbf{R}^*\}$. The arrow in 10c implies that we are assigning p_0 to $\mathbf{U}^*$.

2. Area $(\mathbf{L}_1) > k/n$ and Area $(\mathbf{L}_2) < k/n$ (or equivalently Area $(\mathbf{R}_1) > k/n$ and Area $(\mathbf{R}_2) < k/n$), and Case 1 does not apply.
3. Area $(\mathbf{L}_2) > k/n$ (or equivalently Area $(\mathbf{R}_1) > k/n$).

We will include a proof for Case 1 here and provide the other two cases in the appendix, since the ideas therein are essentially the same.

Case 1 Suppose without loss of generality that Area $(\mathbf{L}_1) < k/n$ and Area $(\mathbf{R}_1) < k/n$. As Area $(\mathcal{L}(x_0, x_1 | S)) > k/n$, there exists a point $\bar{x}$ on ∂S between u_1 and x_0 (traversed clockwise) such that

$$\text{Area}\left(\mathcal{L}(x_1, x_0 | S) \cap \mathcal{L}(p_0, \bar{x} | S)\right) = \frac{k}{n}$$

(see Figure 10a). Similarly, there exists a point $\tilde{x}$ between x_0 and v_1 (traversed clockwise) such that

$$\text{Area}\left(\mathcal{L}(x_0, x_1 | S) \cap \mathcal{L}(\tilde{x}, p_0 | S)\right) = \frac{k}{n}$$

(see Figure 10b). Setting

$$\begin{aligned} \mathbf{U}^* &:= \mathcal{L}(p_0, v_1 | S) \cap \mathcal{L}(u_1, p_0 | S) \\ \mathbf{L}^* &:= \mathcal{L}(x_1, x_0 | S) \cap \mathcal{L}(p_0, \bar{x} | S) \\ \mathbf{R}^* &:= \mathcal{L}(x_0, x_1 | S) \cap \mathcal{L}(\tilde{x}, p_0 | S) \end{aligned}$$

and assigning p_0 to $\mathbf{U}^*$ we obtain an equitable relatively convex 3-partition and we are finished. See Figure 10. We complete this section with the pseudo-code for our algorithm.

Input: A simply connected polygon S, a point set $P = \{p_1, \ldots, p_n\} \subset S$, and a probability density $\mu(\cdot)$ defined on S.

Output: A partition of S into n sub-regions S_i, each satisfying the following properties:

1. Each sub-region S_i contains exactly one element p_i of P.
2. For any two points $u, v \in S_i$, the shortest path from u to v in S is contained in S_i.
3. All sub-regions satisfy $\iint_{S_i} \mu(x)\, dA = \frac{1}{n} \iint_S \mu(x)\, dA$.

if $n = 1$ **then**
  return S;
else
  if *n is even* **then**
    Using Theorem 15, let S_1, S_2 be a ham-sandwich cut of S, P, and $\mu(\cdot)$;
    Let P_1 and P_2 denote the subsets of P in S_1 and S_2 and let $\mu_1(\cdot)$ and $\mu_2(\cdot)$ denote the restrictions of $\mu(\cdot)$ to S_1 and S_2, normalized so that they integrate to 1;
    return RegionPartition $(S_1, P_1, \mu_1) \cup$ RegionPartition (S_2, P_2, μ_2);
  else
    Let p_0 be a point on $GH(P|S)$ and determine a geodesic $G(x_0, x_1|S)$ through p_0 that has $k = (n-1)/2$ points lying strictly to its right and left;
    if *$GH(P|S)$ gives a certificate for an equitable 2-partition S_1, S_2* **then**
      Let P_1 and P_2 denote the subsets of P in S_1 and S_2 and let $\mu_1(\cdot)$ and $\mu_2(\cdot)$ denote the restrictions of $\mu(\cdot)$ to S_1 and S_2, normalized so that they integrate to 1;
      return RegionPartition $(S_1, P_1, \mu_1) \cup$ RegionPartition (S_2, P_2, μ_2);
    else
      Attempt to find an equitable 3-partition S_1, S_2, S_3;
      We will either find S_1, S_2, S_3, or our search will be interrupted by finding a certificate of the existence of an equitable 2-partition;
      if *we find an equitable 3-partition* **then**
        Let P_1, P_2, and P_3 denote the subsets of P in S_1, S_2, and S_3 and let $\mu_1(\cdot)$, $\mu_2(\cdot)$, and $\mu_3(\cdot)$ denote the restrictions of $\mu(\cdot)$ to S_1, S_2, and S_3, normalized so that they integrate to 1;
        return RegionPartition $(S_1, P_1, \mu_1) \cup$ RegionPartition $(S_2, P_2, \mu_2) \cup$ RegionPartition (S_3, P_3, μ_3);
      else
        Let P_1 and P_2 denote the subsets of P in S_1 and S_2 and let $\mu_1(\cdot)$ and $\mu_2(\cdot)$ denote the restrictions of $\mu(\cdot)$ to S_1 and S_2, normalized so that they integrate to 1;
        return RegionPartition $(S_1, P_1, \mu_1) \cup$ RegionPartition (S_2, P_2, μ_2);
      end
    end
  end
end

Algorithm 1: Algorithm RegionPartition (S, P, μ) equitably partitions a simply connected polygon S, a point set $P = \{p_1, \ldots, p_n\} \subset S$, and a probability density $\mu(\cdot)$ defined on S.

Computability conditions on $\mu(\cdot)$ Our algorithm can be performed with any density $\mu(\cdot)$ of a sub-region in place of its area. The two conditions that we impose on computing this density are:

1. For any simple polygon $\tilde{S} \subset S$ with $\tilde{m}$ vertices, we can compute $\iint_{\tilde{S}} \mu(\cdot)\, dA$ in running time $\mathcal{O}(\tilde{m})$.
2. For any hourglass region consisting of two triangles $\triangle apb$ and $\triangle cpd$, with $\alpha := \iint_{\triangle apb} \mu(x)\, dA < \beta := \iint_{\triangle cpd} \mu(x)\, dA$ and for any $\gamma \in [\alpha, \beta]$, we can compute a pair of points u^* on segment ab and v^*on segment cd such that segment u^*v^* contains p and $\iint_{\triangle apu^*} \mu(x)\, dA + \iint_{\triangle v^*pd} \mu(x)\, dA = \gamma$ in constant time. See

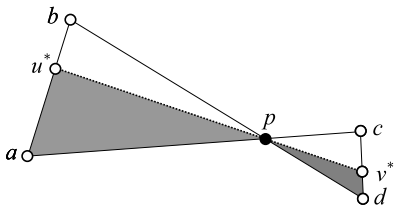


Figure 11: The region described in computability condition 2.

Figure 11.

It is straightforward to verify that these are the only two conditions that Section 3.2.2 requires. Note that in our algorithm, it may be the case that $c = d = p$, so that one of the triangles described in condition 2 may be a single point. In practice, when condition 3.2.2 is not met, the desired points u^* and v^* may be approximated using the method of bisection, in which case the running time of our algorithm becomes $\mathcal{O}\left(nN \log \frac{N}{\epsilon}\right)$. The appendix contains a formal proof that $\mu(\cdot) \equiv \text{Area}(\cdot)$ satisfies condition 2.

Other applications and extensions to arbitrary metrics Recall that our motivation for this algorithm is to distribute the workloads of a fleet of vehicles that visit a set of clients. As we have described earlier, if client locations are assumed to be i.i.d. samples from a probability density $f(\cdot)$, then by setting $\mu(\cdot) := \sqrt{f(\cdot)}$ and equitably partitioning $\mu(\cdot)$ and S, we will find an asymptotically load-balancing partition. Using the result of [27] (also mentioned in Section 2 of this paper), we see that this partitioning result also holds when the underlying combinatorial structures are minimum-weight matchings, minimum spanning trees, Steiner trees, and Delaunay triangulations, as opposed to TSP tours. To be precise, our partitioning algorithm is asymptotically optimal (within $o(\sqrt{k})$) whenever the underlying combinatorial structure being constructed for each depot is a *subadditive Euclidean functional*, as defined in [27]. In certain situations it may be advantageous to set $\mu(\cdot) = f(\cdot)$ instead; this is the case, for example, if each client requires a randomly distributed service time, and these service times dwarf the amount of time spent in transit.

One situation in which this algorithm does not generate load-balancing tours is the case where point-to-point distances are defined by an arbitrary metric $d(\cdot,\cdot)$ (representing, for example, a hilly terrain or an area with heavily congested traffic), rather than a "natural" norm such as the Euclidean distance or an L^p norm. When this is the case, the law of large numbers (1) from Section 2 no longer applies (since the shortest path between two points is not necessarily a straight line) and we cannot guarantee asymptotic optimality. However, we can still define $G(u, v \,|S)$ for any points u and v under such a metric, and hence $\mathcal{L}(u, v \,|S)$ must still be "relatively convex" to S when we consider the shortest path between any two points u, v under $d(\cdot,\cdot)$.The following theorem is proven directly from Theorem 9:

Theorem 19. *Let $S \subset \mathbb{R}^2$ be a simple polygon with m vertices, let $\mu(\cdot)$ be a probability density defined on S, let $d(\cdot,\cdot)$ be a metric defined on S, and let $P = \{p_1, \ldots, p_n\} \subset S$ be a collection of points, where the vertices of S and the points in P are all in general position. Suppose that, for any simple sub-region $\tilde{S} \subset S$, for any point $u \in \partial\tilde{S}$, and for any $\alpha \in (0,1)$, there exists a point $v \in \partial\tilde{S}$ such that $\iint_{\mathcal{L}(u,v|\tilde{S})} \mu(x)\, dA = \alpha \iint_{\tilde{S}} \mu(x)\, dA$, where the boundary component $G\left(u, v \,|\tilde{S}\right)$ of $\mathcal{L}\left(u, v \,|\tilde{S}\right)$ is taken with respect to $d(\cdot,\cdot)$. Then there exists a partition of S into n disjoint sub-regions $S_1, \ldots, S_n$ satisfying the following three properties:*

1. *Each sub-region S_i contains exactly one point from P.*
2. *For any two points $u, v \in S_i$, the shortest path $G(u, v \,|S)$ is contained in $\text{cl}(S_i)$.*
3. *For each sub-region S_i, $\iint_{S_i} \mu(x)\, dA = 1/n$.*

We will refrain from detailing the computability conditions on $d(\cdot,\cdot)$ for a running time of $\mathcal{O}(nN \log N)$ to be obtainable, as there are many of them and they are outside the scope of this paper. However, the methods in Theorem 9 can still be used in practice to obtain an equitable partition of the type described in Theorem 19.

4 Computational results

Theorem 2, our criterion for optimal partitioning, is an asymptotic result. We are guaranteed that vehicle workloads will differ by terms of order $o(\sqrt{k})$, but we have not yet established that workloads are in fact balanced when this algorithm is employed (e.g., that the convergence in k may be slow in practice). In this section we give some examples that suggest that vehicle workloads will in fact be balanced in a practical setting when point-to-point distances are Euclidean.Specifically, we present the results of a simulation in which we construct a synthetic data set with $n = 9$ depots where $f(\cdot)$ is a mixture of three Gaussian distributions, truncated to lie within a simple polygon $S \subset [0,1]^2$.

The two polygons that form the input and output to our simulation are shown in Figure 12. For each polygon, we generate 20 scenarios, with each scenario consisting of 30 samples of k points in S, for k between 50 and 1500 (and hence we performed a total of 600 simulations per polygon). TSP tours were computed using the Lin-Kernighan heuristic from Concorde [14]. Tour lengths for a particular scenario, and the average vehicle tour lengths over all scenarios, are shown in Figure 13. As the plots show, the vehicle workloads are well balanced by partitioning; these suggest that the $o(\sqrt{k})$ term of Theorem 2 may be negligible, although the variability between vehicle tours for small k is still high. This is not surprising since our partition is "asymptotically optimal" and makes no guarantees for the tour lengths when the number of points is small. One possibility for small k would be to ignore the depot set and instead require that each sub-region equitably partition both $\sqrt{f(\cdot)}$ and $f(\cdot)$, since this would ensure that on average the sub-regions will contain the same number of points. One can find an approximate solution to this problem using a discrete equitable partitioning algorithm such as [6].

A second observation is that our algorithm performs well when many scenarios are averaged, as suggested in Figures 13b and 13d. This is supported theoretically by Remark 4; our results additionally suggest that the $\mathcal{O}(1)$ constant may exist (and be small) for certain non-uniform densities, as appears to be the case here.

One other important note is that, since our algorithm merely solves a feasibility problem, the partition sub-regions can have undesirable shape properties depending on the depot placement. Figure 14 shows a partition whose sub-regions are particularly long and skinny. In practice we find it helpful to start our algorithm with various choices of starting point p_0 to generate a partition.

For a related application, Figure 15 shows the result of this algorithm applied to a map of Hennepin County, Minnesota, where $\mu(\cdot)$ is the population density and P represents the 29 largest post offices. Rather than producing equal TSP tour lengths, the algorithm partitions so that each mail carrier services the same number of houses each day.

5 Acknowledgments

The author wishes to thank two anonymous referees for their helpful suggestions.

References

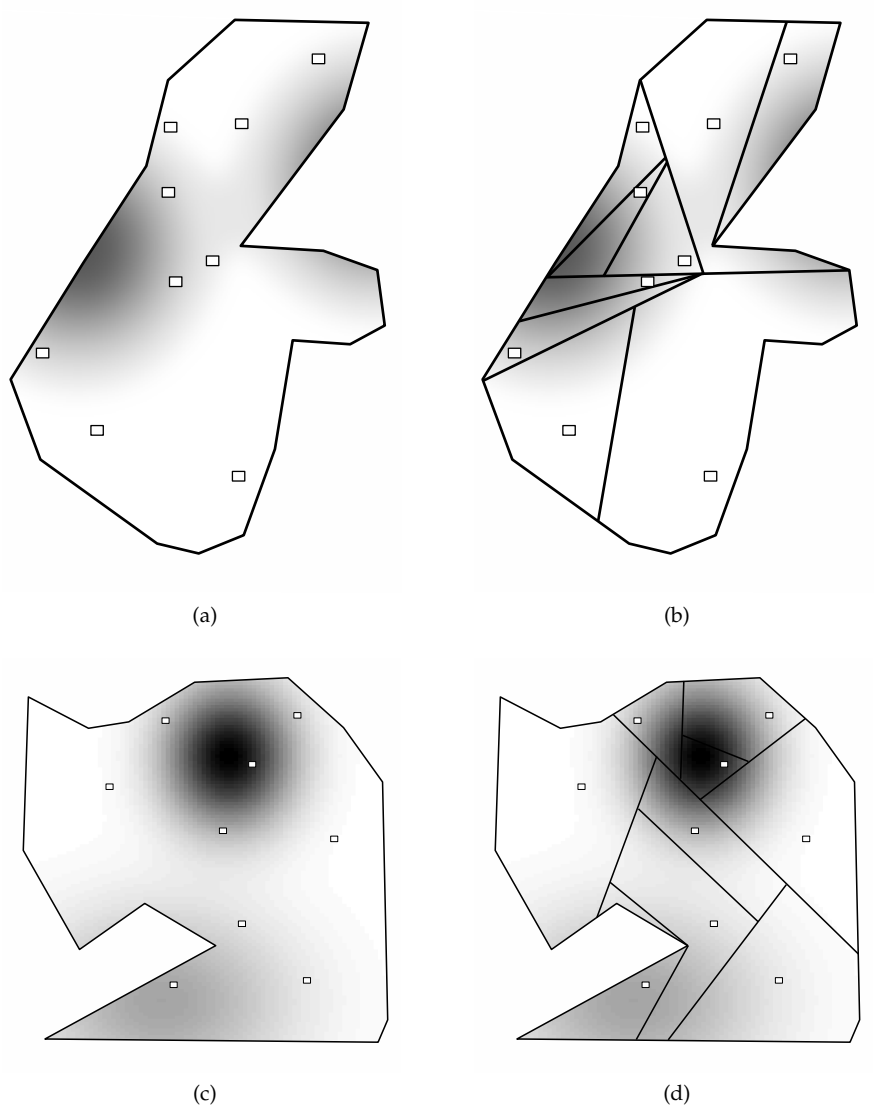
[1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA, 2006.

[2] B. Aronov, P. Carmi, and M.J. Katz. Minimum-cost load-balancing partitions. *Algorithmica*, 54(3):318–336, 2009.

[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] J. Beardwood, J.H. Halton, and J.M. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.



(a) (b)

(c) (d)

Figure 12: Inputs (12a, 12c) and outputs (12b, 12d) to our simulation.

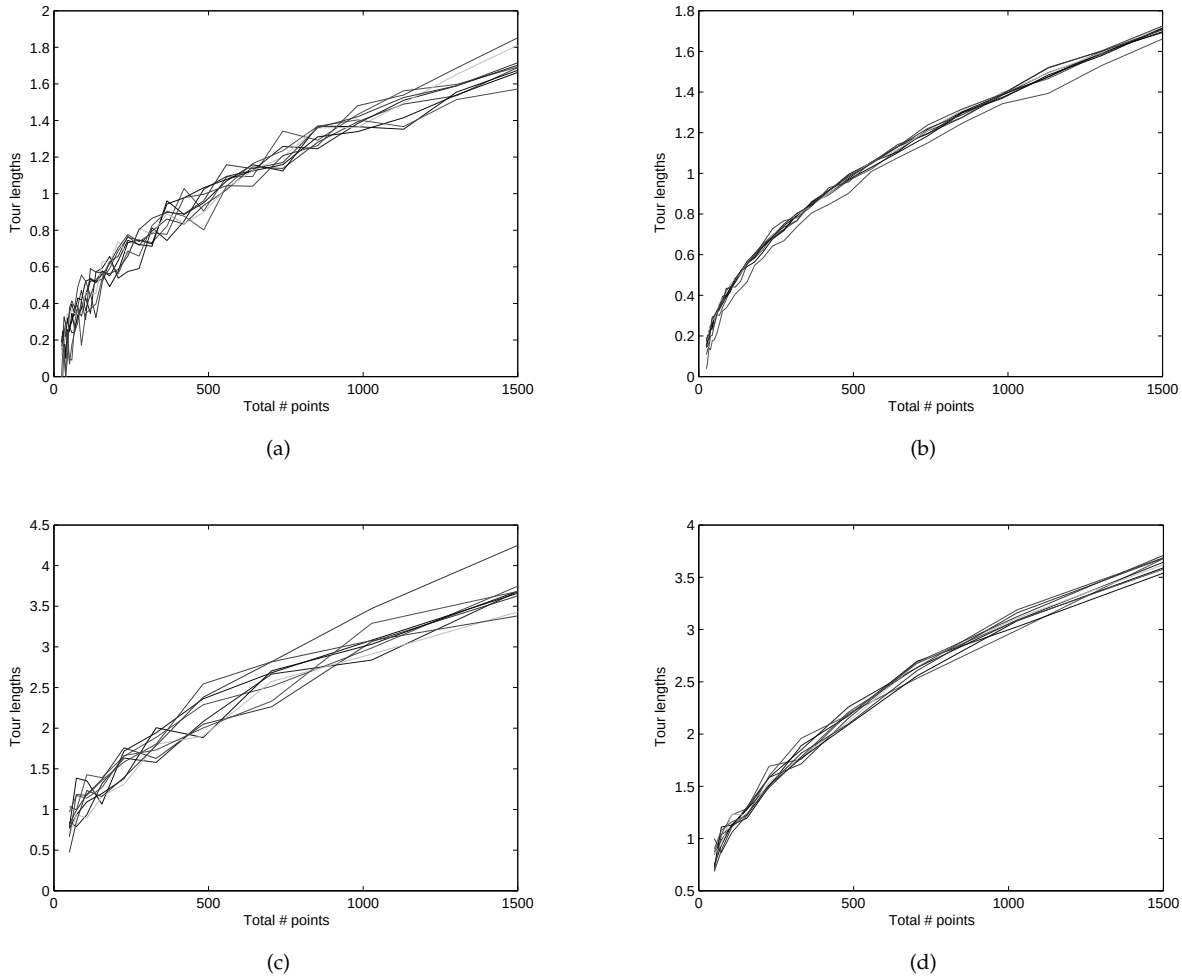
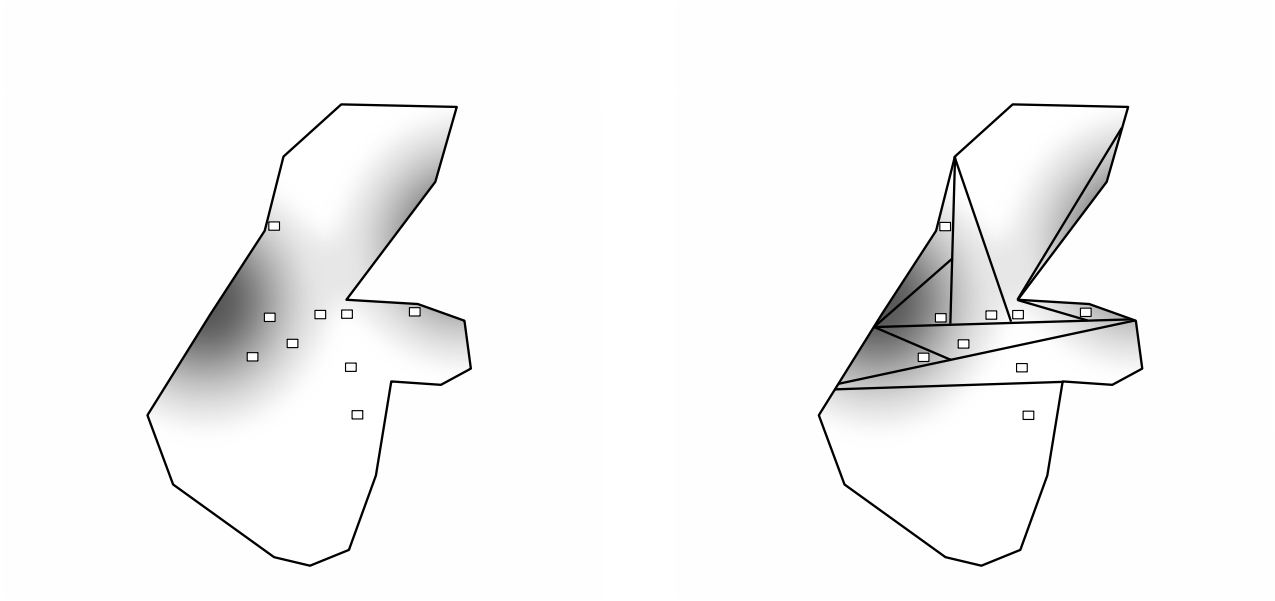


Figure 13: Tour lengths of the 9 vehicles in a particular random scenario for each of the two polygons (13a and 13c), and average tour lengths over 20 scenarios (13b and 13d).



(a) (b)

Figure 14: The clustered depot placement in the middle causes sub-regions to be particularly long and skinny.

[6] 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.

[7] O. Berman, Z. Drezner, A. Tamir, and G. O. Wesolowsky. Optimal location with equitable loads. *Annals of Operations Research*, 167(1):307–325, 2009.

[8] S. Bespamyatnikh, D. Kirkpatrick, and J. Snoeyink. Generalizing ham sandwich cuts to equitable subdivisions. *Discrete and Computational Geometry*, 24:605–622, 2000.

[9] P. Bose, E.D. Demaine, F. Hurtado, J. Iacono, S. Langerman, and P. Morin. Geodesic ham-sandwich cuts. *Discrete and Computational Geometry*, 37(3):325–339, 2007.

[10] J. G. Carlsson, B. Armbruster, and Y. Ye. Finding equitable convex partitions of points in a polygon efficiently. *ACM Transactions on Algorithms*, 6:72:1–72:19, 2010.

[11] 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.

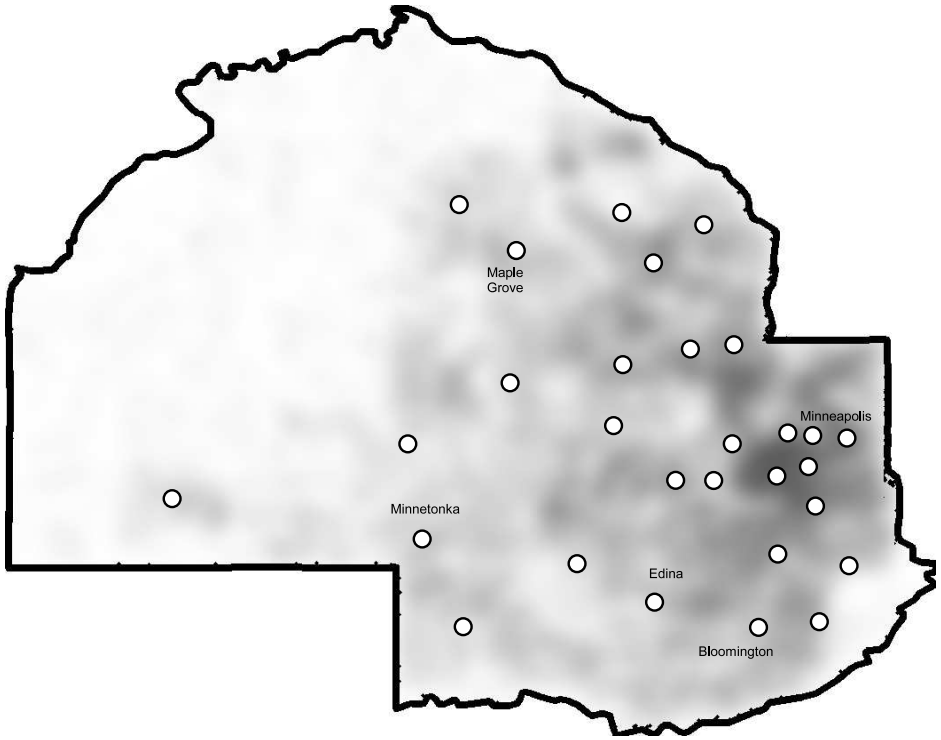
[12] B. Chazelle. A theorem on polygon cutting with applications. In *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 339–349, Washington, DC, USA, 1982. IEEE Computer Society.

[13] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6(1):485–524, 1991.

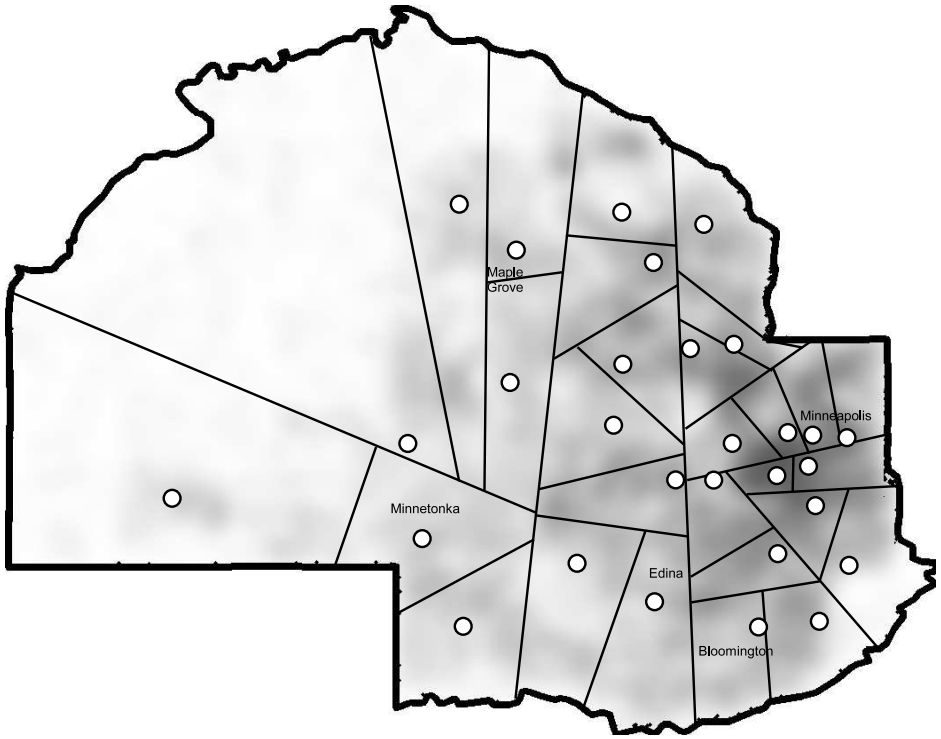
[14] W. Cook. Concorde TSP Solver. http://www.tsp.gatech.edu/concorde.html, 1997–2005.

[15] Z. Drezner and A. Suzuki. Covering continuous demand in the plane. *Journal of the Operational Research Society*, 61(5):878–881, 2010.

[16] B.L. Golden. A statistical approach to the TSP. *Networks*, 7(3):209–225, 1977.



(a)



(b)

Figure 15: An equitable partition of Hennepin County, Minnesota. All sub-regions have the same total population and each sub-region contains one post office.

[17] 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.

[18] 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.

[19] M. Jäger and B. Nebel. Dynamic decentralized area partitioning for cooperating cleaning robots. In *ICRA 2002*, pages 3577–3582, 2002.

[20] A. Kaneko and M. Kano. Discrete geometry on red and blue points in the plane - a survey. In *Discrete and Computational Geometry, The Goodman-Pollack Festschrift*, pages 551–570, Berlin, 2003. Springer.

[21] 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.

[22] O. Kwon, B. Golden, and E. Wasil. Estimating the length of the optimal TSP tour: An empirical study using regression and neural networks. *Computers & Operations Research*, 22(10):1039 – 1046, 1995.

[23] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393 – 140, 1984.

[24] M. Pavone, A. Arsie, E. Frazzoli, and F. 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.

[25] M. Pavone, A. Arsie, E. Frazzoli, and F. 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.

[26] 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.

[27] J. M. Steele. Subadditive euclidean functionals and nonlinear growth in geometric probability. *The Annals of Probability*, 9(3):365–376, 1981.

[28] J. M. Steele. *Probability Theory and Combinatorial Optimization (CBMS-NSF Regional Conference Series in Applied Mathematics)*. Society for Industrial Mathematics, 1987.

[29] G.T. Toussaint. An optimal algorithm for computing the relative convex hull of a set of points in a polygon. *Signal Processing III: Theories and Applications, Proc. EUR-ASIP-86, Part 2*, pages 853–856, 1986.

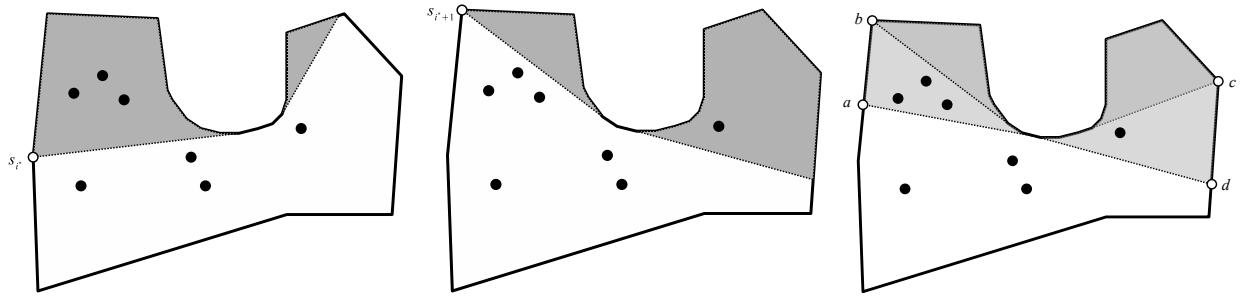


Figure 16: In 16a and 16b we isolate the desired edge $\overline{s_{i^*}s_{i^*+1}}$, which is then refined in 16c. Here we have $k = 2$ and $n = 7$.

A Additional proofs

A.1 Theorem 15, Case 2

Proof. This is essentially the same as the proof of the ham-sandwich theorem in [6, 9], with one of the two point sets replaced with the area functional. We must show that the partition $(\bar{x}, \tilde{x})$ can be computed in running time $\mathcal{O}(N(\log m + \log r)) = \mathcal{O}(N \log N)$. Suppose that we have three points x_0, x_1, and x_2 as in the hypothesis. Since the vertices of S are sorted, we let $s_1, \ldots, s_{m'}$ denote the vertices of S lying between x_0 and x_2 when S is traversed clockwise. Using Lemma 13, we perform a binary search on these vertices, computing at each query s_i the left shell through vertex s_i with area k/n (and the number of points contained therein), until we either terminate or isolate an edge $\overline{s_{i^*}s_{i^*+1}}$ that must contain $\bar{x}$. We then perform another binary search to isolate the opposite edge $\overline{s_{j^*}s_{j^*+1}}$ that must contain $\tilde{x}$. These computations can all be performed in running time $\mathcal{O}(m \log m + N \log m) = \mathcal{O}(N \log m)$.

We can further refine these two edges, $\overline{s_{i^*}s_{i^*+1}}$ and $\overline{s_{j^*}s_{j^*+1}}$, to a pair of line segments $\overline{ab}$ and $\overline{cd}$, satisfying the properties that

1. $\overline{ab} \subseteq \overline{s_{i^*}s_{i^*+1}}$ and $\overline{cd} \subseteq \overline{s_{j^*}s_{j^*+1}}$
2. $\text{Area}(\mathcal{L}(a, c\,|S)) = \text{Area}(\mathcal{L}(b, d\,|S)) = k/n$
3. $|\mathcal{L}(a, c\,|S) \cap P| > k$ and $|\mathcal{L}(b, d\,|S) \cap P| < k$

as in Figure 16. This means that our desired equitable geodesic must lie in the *funnel* $\mathcal{F} := \mathcal{L}(a, d\,|S) \cap \mathcal{L}(c, b\,|S)$. There are two cases we must consider regarding the shape of $\mathcal{F}$:

1. $\mathcal{F}$ is a "skinny funnel"; that is, $\mathcal{F}$ consists of two disjoint polygons, connected by a geodesic *reflex chain* (Figure 17).
2. $\mathcal{F}$ is a "fat funnel"; that is, $\mathcal{F}$ is a single simple polygon with no degeneracies (Figure 18).

Case 1 Let $\mathcal{F}_1$ denote the polygon of $\mathcal{F}$ containing $\overline{ab}$ and let $\mathcal{F}_2$ denote the polygon of $\mathcal{F}$ containing $\overline{cd}$. Let $h_1 \in \mathcal{F}_1$ be the reflex vertex of S in $\mathcal{F}_1$ that intersects the reflex chain, and define $h_2 \in \mathcal{F}_2$ similarly. By building the geodesic shortest-path tree between h_1 and every vertex in $\mathcal{F}_1$ in $\mathcal{O}(m)$ time, we can divide segment $\overline{ab}$ into at most $r + 1$ segments, as shown in Figure 19. We now perform a binary search on the endpoints of each such segment, determining for each query $x \in \overline{ab}$ the corresponding point $\bar{x} = \text{LShell}_{k/n}(x) \in \overline{cd}$ (using the same procedure as described in Lemma 13), and the number of points $|\mathcal{L}(x, \bar{x}\,|S) \cap P|$. Each query requires $\mathcal{O}(N)$ time and consequently this binary search requires $\mathcal{O}(N \log r)$ running time. By performing this binary search, we refine our search space $\mathcal{F}_1$ to a single triangle $\triangle a'b'r_1$, where $a', b' \in \overline{ab}$ and r_1 is a reflex vertex. We perform the same

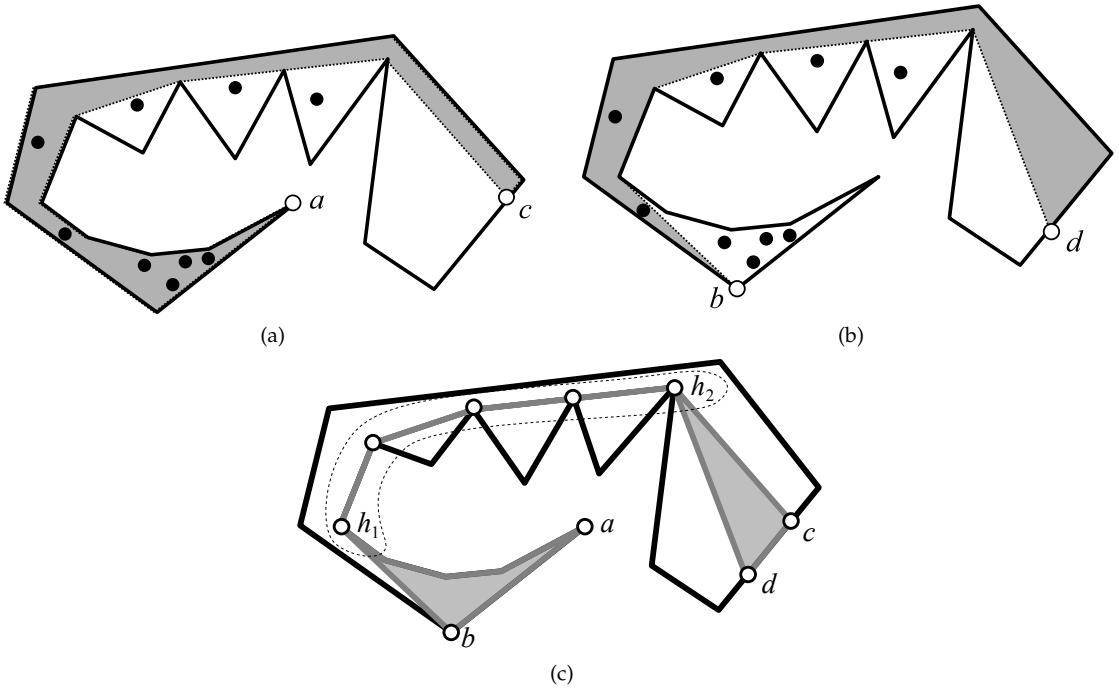


Figure 17: 17a and 17b show the two half-shells induced by $\overline{ab}$ and $\overline{cd}$; the "skinny funnel" and its reflex chain are shown in 17c.

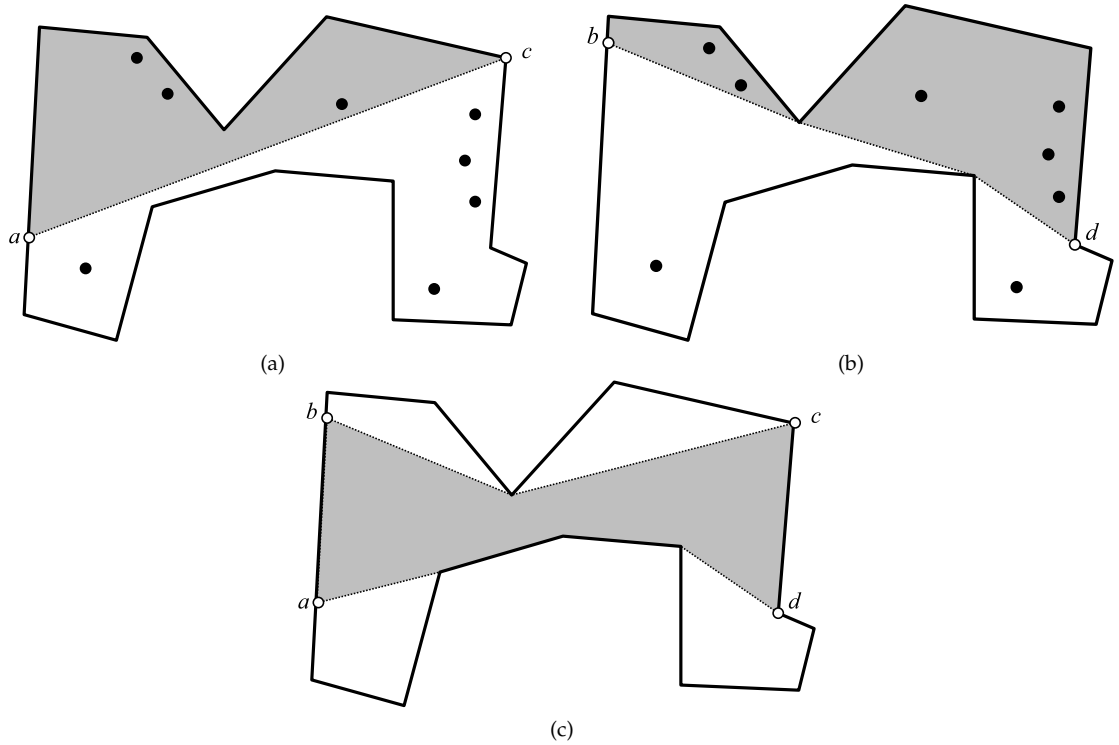


Figure 18: 18a and 18b show the two half-shells induced by $\overline{ab}$ and $\overline{cd}$; the "fat funnel"is shown in 18c.

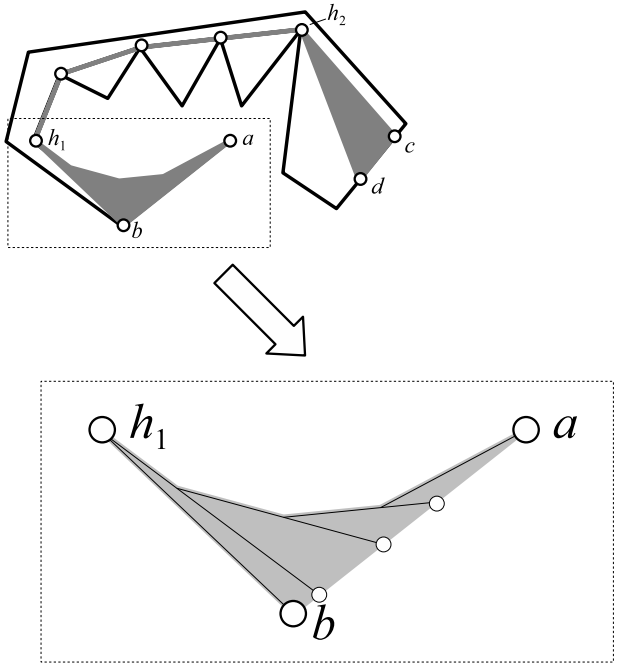


Figure 19: We divide segment $\overline{ab}$ into a collection of segments based on the reflex vertices of S.

binary search on $\mathcal{F}_2$, isolating two points c' and d' and a reflex vertex r_2, which may cause us to further refine a' and b'.

Our search space has now been refined to a pair of triangles, connected by a reflex chain; the advantage of this is that we can now make constant-time area queries. In particular, given any point $x \in \overline{a'b'}$, we can determine the corresponding $\bar{x}$ so that Area $(\mathcal{L}(x, \bar{x})) = k/n$ in constant time. Hence, for each point $p \in P \cap \left(\triangle a'b'r_1 \cup \triangle c'd'r_2\right)$, we can define a pair of points $x_p \in \overline{a'b'}$ and $\bar{x}_p \in \overline{c'd'}$ that induce a desired area, in constant time. The collection of all points x_p and $\bar{x}_p$ partitions $\overline{a'b'}$ and $\overline{c'd'}$ into at most $n+1$ segments each. Each such segment defines a family of partitions (each cutting off area k/n of S) that cut off the same subset of P, and any two adjacent segments define families of partitions that cut off a pair of subsets of P differing by only one element. We can sort the endpoints of these segments by the order in which they appear on $\overline{a'b'}$ in $\mathcal{O}(n \log n)$ time. Hence, by examining each segment sequentially (in total time $\mathcal{O}(n)$), we are guaranteed to find the segment whose induced partition is equitable;

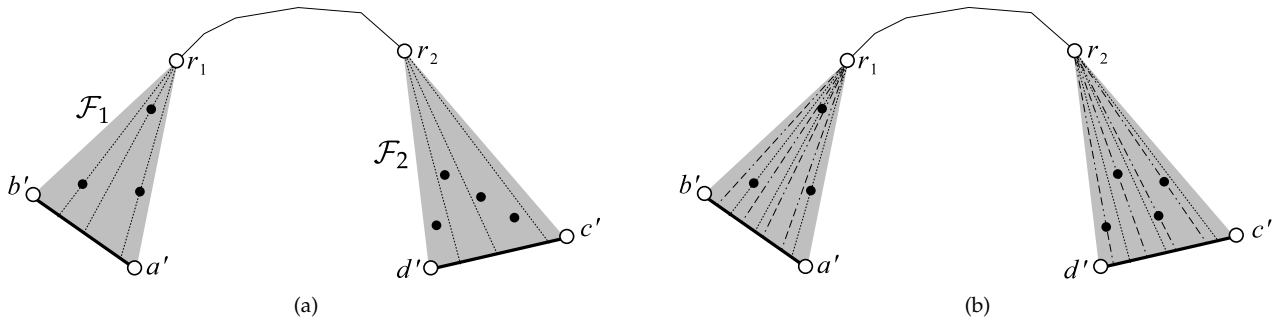


Figure 20: Dividing the two triangles $\triangle a'b'r_1$ and $\triangle d'c'r_2$ with segments that all cut off area k/n of S in constant time.

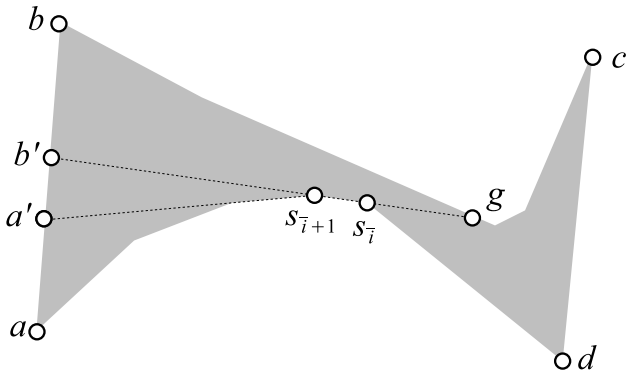


Figure 21: The diagrams above show the funnel $\mathcal{F}$ (we omit S and P for clarity). If point g does not intersect segment $\overline{cd}$, then any geodesic from $a^{'}$ or $b^{'}$ to $\overline{cd}$ must intersect segment $\overline{s_{\bar{i}}s_{\bar{i}+1}}$ by construction.

see Figure 20. This completes the proof. The total complexity for this subroutine is $\mathcal{O}(m + N \log r + n \log n) = \mathcal{O}(N \log N)$.

Case 2 We will follow a similar procedure to Case 1, although we cannot necessarily restrict our search space to a pair of triangles. Let the *lower* (resp. *upper*) *reflex chain* of $\mathcal{F}$ be the geodesic that connects vertices a and d (resp. b and c), and assume that S is oriented so that a and d lie "below" b and c. We will show that the funnel $\mathcal{F}$ can be reduced to either a skinny funnel, or a funnel with at most six vertices. In particular, we will reduce the number of reflex vertices on the upper and lower reflex chains to one. It will suffice to prove that we can reduce $\mathcal{F}$ to a funnel with only one vertex on its lower reflex chain (since we can apply the same procedure then to the upper reflex chain).

We can extend each edge in the lower chain until it touches $\overline{ab}$ (if at all); this divides $\overline{ab}$ into at most $r + 1$ segments. We also extend the *cross tangent* (the line tangent to the two chains, if it exists) to the upper and lower reflex chains until it touches $\overline{ab}$ (if at all). We perform a binary search on the endpoints of these segments, where for each query point x we compute $\text{LShell}_{k/n}(x)$, followed by the number of points cut off, $|\mathcal{L}(x, \text{LShell}_{k/n}(x) | S) \cap P|$. On completion (which takes running time $\mathcal{O}(N \log r)$) we will isolate a segment $\overline{a'b'} \subseteq \overline{ab}$ that must contain an endpoint of our desired equitable 2-partition. Note that if $b = b^{'}$, then we are finished; this is because we know that an equitable 2-partition must be contained in the funnel defined by $\overline{a'b}$ and $\overline{c, d}$; by our construction, this is either a skinny funnel, in which case we can apply the result of Case 1, or a funnel whose lower reflex chain has only one vertex.

Assuming that $b \neq b^{'}$, it must be the case that $b^{'}$ is induced by an edge $\overline{s_{\bar{i}}s_{\bar{i}+1}}$ on the lower reflex chain. We extend this edge to the right until it intersects with $\mathcal{F}$ at a point g; if g does not lie on segment $\overline{cd}$, then we are finished, since we can reduce $\mathcal{F}$ to a skinny funnel as shown in Figure 21.

If the extension of this segment lies on $\overline{cd}$ at a point g then we compute $\text{LShell}_{k/n}\left(a^{'}\right)$ and $\text{LShell}_{k/n}\left(b^{'}\right)$. There are three cases we must consider:

1. If $\text{LShell}_{k/n}\left(a^{'}\right)$ and $\text{LShell}_{k/n}\left(b^{'}\right)$ are both below g, then we have isolated our search to a skinny funnel and we apply the result of Case 1; see Figure 22.

2. If $\text{LShell}_{k/n}\left(a^{'}\right)$ and $\text{LShell}_{k/n}\left(b^{'}\right)$ both lie above g, then we can refine our search to either a skinny funnel or a funnel whose lower reflex chain is a line segment; see Figure 23.

3. If $\text{LShell}_{k/n}\left(a^{'}\right)$ lies above g and $\text{LShell}_{k/n}\left(b^{'}\right)$ lies below g, then we compute $h = \text{LShell}_{(n-k)/n}(g)$ (which lies on $\overline{a'b'}$) . We perform one more search query at this point, counting the number of points in $\mathcal{L}(h, g)$; see Figure 24. Two cases can arise:

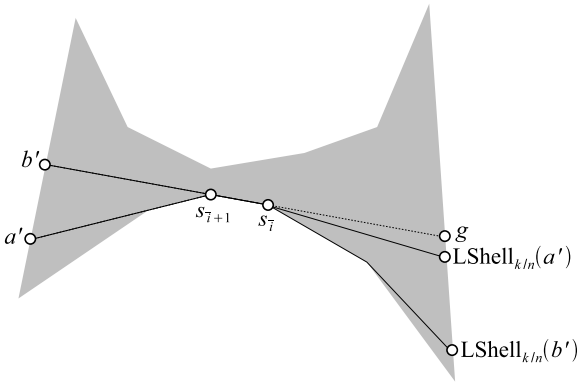


Figure 22: If $\mathrm{LShell}_{k/n}\left(a^{'}\right)$ and $\mathrm{LShell}_{k/n}\left(b^{'}\right)$ are both below g, then the equitable partition lies inside a skinny funnel, with both the lower and upper reflex chains containing $\left(s_{\bar{i}}, s_{\bar{i}+1}\right)$.

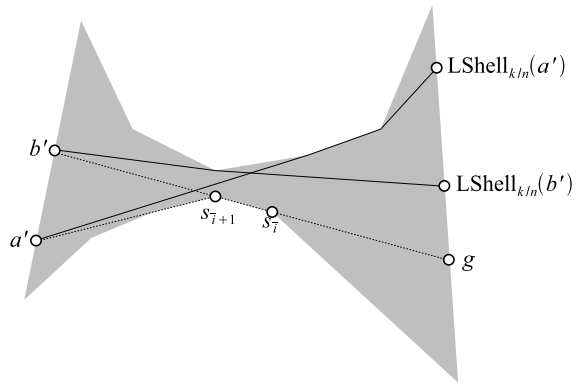


Figure 23: If $\mathrm{LShell}_{k/n}\left(a^{'}\right)$ and $\mathrm{LShell}_{k/n}\left(b^{'}\right)$ are both above g, then the geodesic $G\left(a^{'}, \mathrm{LShell}_{k/n}\left(b^{'}\right) | S\right)$ can only intersect $s_{\bar{i}+1}$ (if it intersects any of the lower reflex vertices at all). If $G\left(a^{'}, \mathrm{LShell}_{k/n}\left(b^{'}\right) | S\right)$ intersects the upper reflex chain, we have a skinny funnel.

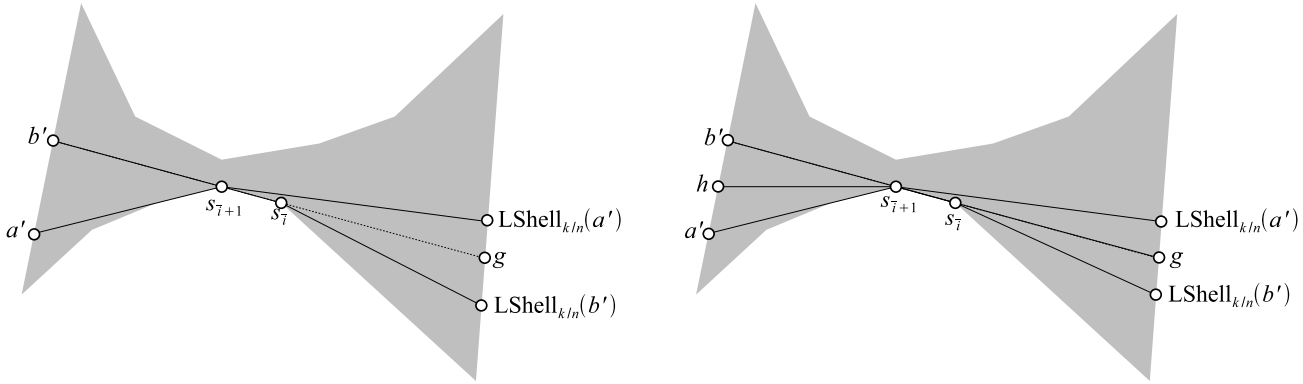


Figure 24: If $\mathrm{LShell}_{k/n}\left(a^{'}\right)$ is above g and $\mathrm{LShell}_{k/n}\left(b^{'}\right)$ is below g, then we compute $h = \mathrm{LShell}_{(n-k)/n}(g)$. Our search is then reduced to either a skinny funnel or a funnel whose bottom reflex chain is $\left(a^{'}, s_{\bar{i}+1}, s_{\bar{i}}, g\right) = \left(a^{'}, s_{\bar{i}+1}, g\right)$, which contains only one reflex vertex.

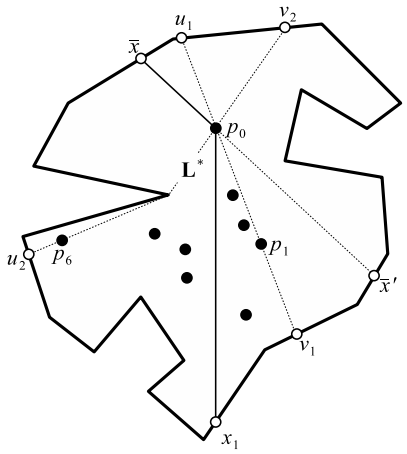


Figure 25: The points $\bar{x}$ and $\bar{x}^{'}$.

(a) If $|\mathcal{L}(h,g) \cap P| > k$, then the endpoints of the desired equitable partition must lie in the funnel joining segments $\overline{a^{'}h}$ and $\left(\text{LShell}_{k/n}\left(a^{'}\right), g\right)$. We have then isolated our search space to a funnel whose bottom reflex chain contains only one reflex vertex $s_{\bar{i}+1}$.

(b) If $|\mathcal{L}(h,g) \cap P| < k$, then the endpoints of the desired equitable partition must lie in the funnel joining segments $\overline{hb'}$ and $\overline{g\,\text{LShell}_{k/n}\left(b'\right)}$. This is a skinny funnel and we can apply Lemma 1.

To complete the proof, we require one more lemma:

Lemma 20. *Let $\mathcal{F}^{'}$ be a fat funnel with both reflex chains having only one reflex vertex (so $\mathcal{F}^{'}$ is a hexagon) as in the preceding description. We can find an equitable partition of S and P in running time $\mathcal{O}(n \log n)$.*

Proof. Let (a,b,c,d,e,f) denote the vertices of $\mathcal{F}^{'}$, with reflex vertices e and f. By our assumption, there exists an equitable partition of S whose boundary is the geodesic $G(u,v\,|S)$, where $u \in \overline{ab}$ and $v \in \overline{cd}$. For each point $p \in P$ in $\mathcal{F}^{'}$, we can compute in constant time every geodesic $G(u,v\,|S) \ni p$ cutting off a fixed area of $\mathcal{F}^{'}$. Intuitively, this is because every such geodesic is either a line segment or a path with a single reflex vertex (e or f), by our general position assumption, and we can determine the existence of all such geodesics (there can be no more than 6 of them for each p) in constant time. □

Lemma 20 completes the proof of Theorem 15 and we are finished. □

A.2 Theorem 17, Cases 2 and 3

Case 2 Suppose without loss of generality that Area $(\mathbf{L}_1) > k/n$ and Area $(\mathbf{L}_2) < k/n$. As in Case 1, there exists a point $\bar{x}$ on ∂S between u_2 and u_1 (traversed clockwise) such that

$$\text{Area}\left(\mathcal{L}(x_1, x_0\,|S) \cap \mathcal{L}(p_0, \bar{x}\,|S)\right) = k/n\,.$$

We set $\mathbf{L}^* := \mathcal{L}(x_1, x_0\,|S) \cap \mathcal{L}(p_0, \bar{x}\,|S)$. Next, let $\bar{x}^{'}$ denote the extension of $G(\bar{x}, p_0\,|S)$ to ∂S as in definition 10; see Figure 25. Note that if Area $\left(\mathcal{L}\left(p_0, \bar{x}^{'}\,|S\right)\right) \leq 1/n$ then we are guaranteed to find an equitable 2-partition by Corollary 16. Hence, we assume that Area $\left(\mathcal{L}\left(p_0, \bar{x}^{'}\,|S\right)\right) > 1/n$, in which case

$$\text{Area}\left(\mathcal{L}(x_0, x_1\,|S) \cap \mathcal{L}\left(\bar{x}^{'}, p_0\right)\right) < k/n\,.$$

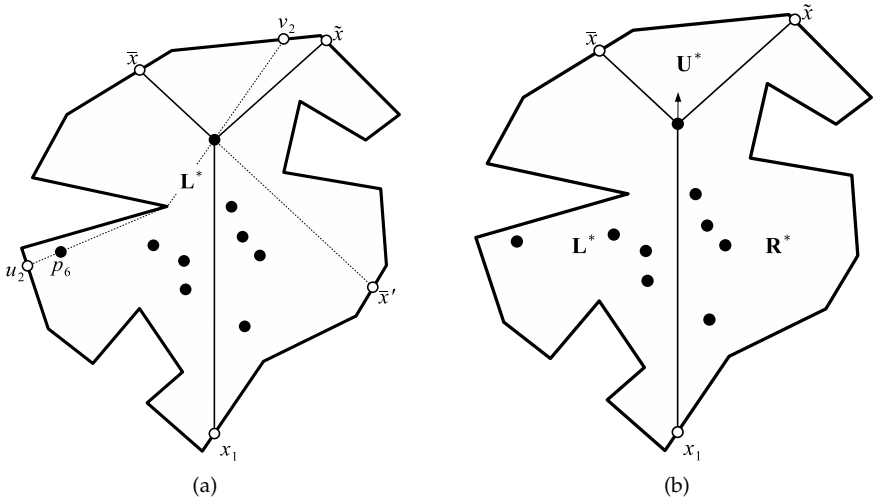


Figure 26: Finding the point $\tilde{x}$ and the partition $\{\mathbf{U}^*, \mathbf{L}^*, \mathbf{R}^*\}$.

Finally, since

$$\text{Area}\left(\mathcal{L}\left(x_0, x_1 \,|S\right)\right) > k/n$$

by our initial assumption 3, there exists a point $\tilde{x}$ on ∂S between x_0 and $\bar{x}^{'}$ (traversed clockwise) such that

$$\text{Area}\left(\mathcal{L}\left(x_0, x_1 \,|S\right) \cap \mathcal{L}\left(\tilde{x}, p_0\right)\right) = k/n\,.$$

Setting

$$\begin{aligned}
\mathbf{U}^* &:= \mathcal{L}\left(p_0, v_1 \,|S\right) \cap \mathcal{L}\left(u_1, p_0 \,|S\right) \\
\mathbf{L}^* &:= \mathcal{L}\left(x_1, x_0 \,|S\right) \cap \mathcal{L}\left(p_0, \bar{x} \,|S\right) \\
\mathbf{R}^* &:= \mathcal{L}\left(x_0, x_1 \,|S\right) \cap \mathcal{L}\left(\tilde{x}, p_0 \,|S\right)
\end{aligned}$$

and assigning p_0 to $\mathbf{U}^*$ we obtain an equitable relatively convex 3-partition and we are finished. See Figure 26.

Case 3 Suppose without loss of generality that $\text{Area}\left(\mathbf{L}_2\right) > k/n$. Let p_{i^*} with $i^* \leq n^{'}$ be the element of P such that $\mathcal{L}\left(p_{i^*}, x_0 \,|S\right)$ is a *supporting* shell to $GH\left(P \,|S\right)$, where as in definition 10 we must extend point p_{i^*} to a point x_2 on ∂S. We can find p_{i^*} in $\mathcal{O}\left(N \log r\right)$ time using Lemma 12. We assume that $\text{Area}\left(\mathcal{L}\left(x_2, x_0 \,|S\right)\right) > 1/n$, since otherwise Corollary 16 applies. Since $\text{Area}\left(\mathcal{L}\left(x_0, x_1 \,|S\right)\right) > k/n$, it must therefore be the case that

$$\text{Area}\left(\mathcal{L}\left(x_1, x_0 \,|S\right) \cap \mathcal{L}\left(x_0, x_2 \,|S\right)\right) < k/n$$

and

$$\text{Area}\left(\mathcal{L}\left(x_1, x_0 \,|S\right) \cap \mathcal{L}\left(v_2, u_2 \,|S\right)\right) > k/n\,,$$

and therefore there must exist a point $\bar{x} \in \partial S$ between x_2 and u_2 (traversed clockwise) and a point $p_{\bar{i}} \in GH\left(P \,|S\right)$ such that

$$\text{Area}\left(\mathcal{L}\left(x_1, x_0 \,|S\right) \cap \mathcal{L}\left(p_{\bar{i}}, \bar{x} \,|S\right)\right) = k/n$$

and $\mathcal{L}\left(\bar{x}, p_{\bar{i}} \,|S\right)$ is a supporting shell to $GH\left(P \,|S\right)$. See Figure 27. Next we can isolate $p_{\bar{i}}$ by performing a binary search on the ordered point set $\left\{x_2, p_{i^*}, p_{i^*+1}, \ldots, p_{n'}, p_0\right\}$. For each query, we take a pair of adjacent points $\left(p_j, p_{j+1}\right)$ (or $\left(x_2, p_{i^*}\right)$ or $\left(p_{n'}, p_0\right)$ in the boundary case) and compute $\text{Area}\left(\mathcal{L}\left(x_1, x_0 \,|S\right) \cap \mathcal{L}\left(p_{j+1}, p_j \,|S\right)\right)$. At the end of this procedure we will isolate two pairs $\left(p_{\bar{i}-1}, p_{\bar{i}}\right)$ and $\left(p_{\bar{i}}, p_{\bar{i}+1}\right)$, which gives us the point $p_{\bar{i}}$. By then performing a binary search on the vertices of S, we isolate an hourglass shape (or a skinny funnel) that contains our

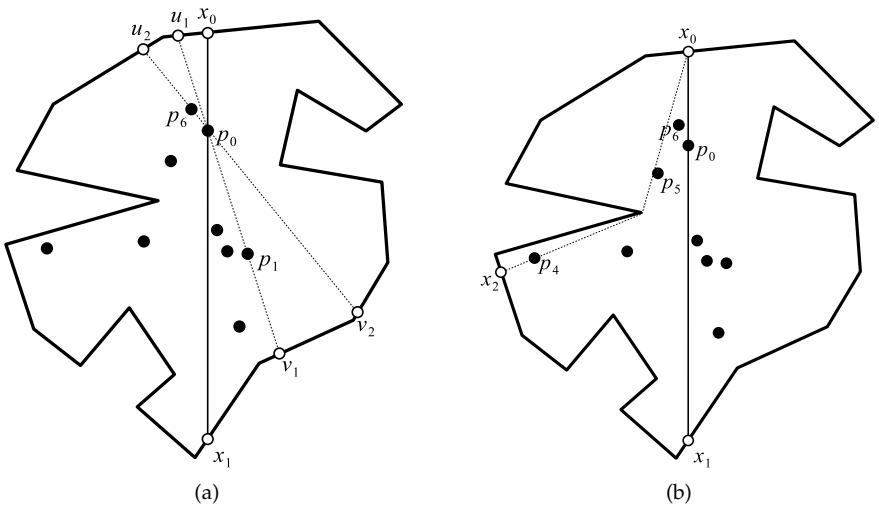


Figure 27: The inputs and solution procedure for Case 3. Beginning with Area $(\mathbf{L}_2) > k/n$ as shown in 27a, we find the point $p_4 = p_{i^*}$ in 27b.

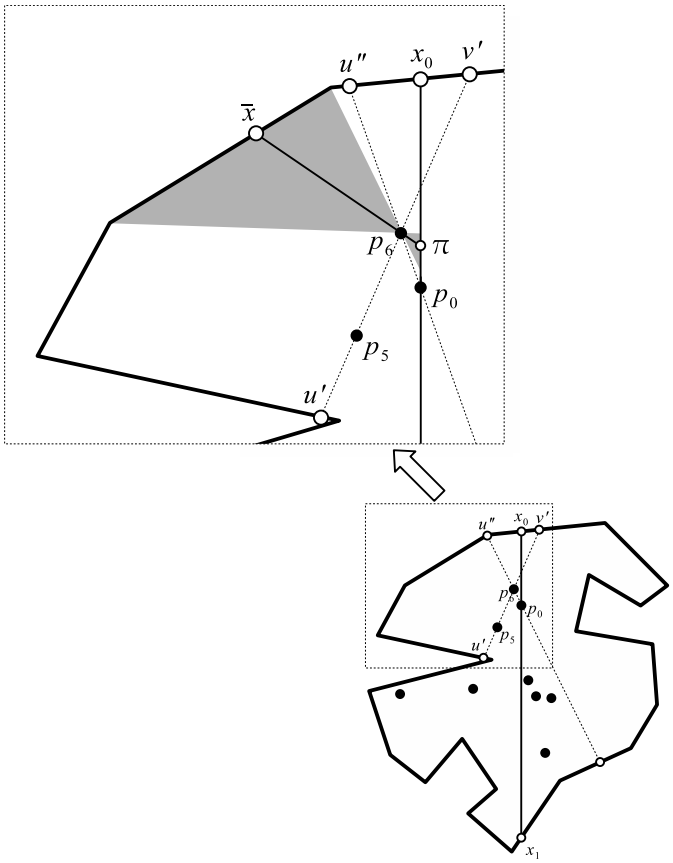


Figure 28: After isolating $p_{\bar{i}}$ in a binary search of $GH(P|S)$, we conclude that our desired point $\bar{x}$ lies between the points u' and u'' (traversed clockwise on ∂S), which are determined by intersecting $G\left(p_{\bar{i}-1}, p_{\bar{i}}|S\right)$ and $G\left(p_{\bar{i}}, p_{\bar{i}+1}|S\right)$ with ∂S. We perform a binary search on the vertices of ∂S to reduce our search space to an hourglass (or a skinny funnel in other cases), indicated by the shaded region.

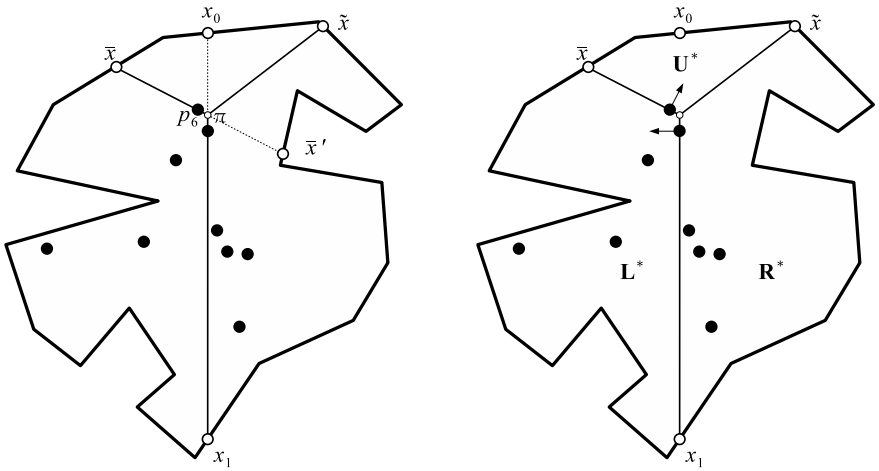


Figure 29: Finding the partition $\{\mathbf{U}^*, \mathbf{L}^*, \mathbf{R}^*\}$.

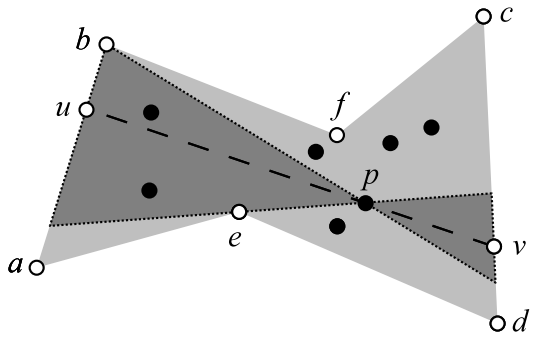


Figure A.1: The "hourglass" region corresponding to point p.

desired $\bar{x}$ as shown in Figure 28. Let π denote the intersection $G(\bar{x}, p_{i^*} | S)$ with $G(x_1, x_0 | S)$ and let $\bar{x}'$ denote the extension of $G(\bar{x}, p_{i^*} | S)$ to ∂S, also shown in Figure 28. Note that $\mathcal{L}(\bar{x}, p_{\bar{i}} | S) = \mathcal{L}(\bar{x}, \pi | S) = \mathcal{L}\left(\bar{x}', \pi | S\right)$ by definition of π.

Next, we can assume without loss of generality that Area $(\mathcal{L}(\bar{x}, \pi | S)) > 1/n$, since otherwise Corollary 16 applies. Since Area $(\mathcal{L}(x_1, x_0 | S) \cap \mathcal{L}(\pi, \bar{x} | S)) = k/n$, this implies that Area $(\mathcal{L}(x_0, x_1 | S) \cap \mathcal{L}(\pi, \bar{x} | S)) < k/n$. However, since Area $(\mathcal{L}(x_0, x_1 | S)) > k/n$ by assumption, there must exist a point $\tilde{x}$ between x_0 and $\bar{x}'$ (traversed clockwise) such that Area $(\mathcal{L}(x_1, x_0 | S) \cap \mathcal{L}(\pi, \tilde{x} | S)) = k/n$. Using Lemma 13, we can find $\tilde{x}$ with a binary search on the vertices of S, and we are finished. See Figure 29.

B Proof of Lemma 20

As in Theorem 15 and Lemma 20, we have a funnel $\mathcal{F}'$ with vertices (a, b, c, d, e, f) as shown in Figure A.1. Given any point $p \in P$ in $\mathcal{F}'$, we want to find every geodesic $G(u, v | S) \ni p$ cutting off a fixed area A of $\mathcal{F}'$, where $u \in (a, b)$ and $v \in (c, d)$. Every such geodesic is either a line segment (u, v) that contains p, or a path with a single reflex vertex (u, e, v) or (u, f, v) that contains p. We first determine all line segments that cut off a fixed area of $\mathcal{F}'$: these are all contained in an *hourglass* as shown in Figure A.1, whose bounds are determined by the placement of the funnel vertices and p (which we can determine in constant time). In order to find a segment cutting off a

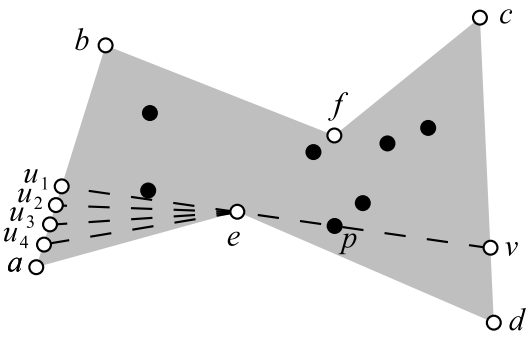


Figure A.2: A collection of geodesics (u, e, p, v), for varying u's; notice that v does not change, as it is constrained to lie on the line between e and p.

desired area, we must solve

$$F(u, v) := \frac{1}{2} \det \begin{pmatrix} 1 & 1 & 1 \\ a & p & u \end{pmatrix} + \frac{1}{2} \det \begin{pmatrix} 1 & 1 & 1 \\ p & d & v \end{pmatrix} = A$$

where a, p, u, v, and d are column vectors, subject to the constraint that p lies on the segment from (u, v), for some desired area A. It is easy to verify that the coordinates of v are a linear fractional function $G(u)$ of u (due to the constraint that $p \in (u, v)$), and therefore we are seeking a solution to

$$H(u) := \underbrace{\frac{1}{2} \det \begin{pmatrix} 1 & 1 & 1 \\ a & p & u \end{pmatrix}}_{\text{linear in } u} + \underbrace{\frac{1}{2} \det \begin{pmatrix} 1 & 1 & 1 \\ p & d & G(u) \end{pmatrix}}_{\text{linear fractional in } u} = A$$

Since $H(u)$ is the sum of a linear function of u and a linear fractional function of u, we can find a solution to $H(u) = A$ by solving a quartic equation, which has at most 4 solutions.

In addition to finding all line segments cutting off a desired area, we can also consider all paths of the form (u, e, v) or (u, f, v) that contain p; these are both straightforward. There are a total of four such paths that may exist: those of the form (u, e, p, v) and (u, p, e, v), and those of the form (u, f, p, v) and (u, p, f, v). To find a geodesic of the form (u, e, p, v) cutting off a desired area, for example, we observe that the only variable that we may change is u, since v is constrained to lie on the line determined by e and p. This is a linear equation in u; see Figure A.2.

At this point, for each point $p \in P$ in $\mathcal{F}'$, we have a collection of at most 6 points on $\partial\mathcal{F}'$ that cut off a desired area. We can sort these points. Again, the collection of all such points divides the edges (a, b) and (c, d) into $\mathcal{O}(n)$ segments. Each such segment defines a family of partitions (each cutting off area k/n of S) that cut off the same subset of P, and any two adjacent segments define families of partitions that cut off a pair of subsets of P differing by only one element. Hence, by sorting the endpoints of segments (in total time $\mathcal{O}(n \log n)$) and examining each segment sequentially (in total time $\mathcal{O}(n)$), we are guaranteed to find the segment whose induced partition is equitable. This completes the proof.