TheoretiCS

Minimum Star Partitions of Simple Polygons in Polynomial Time

Received Mar 6, 2025
Revised Oct 6, 2025
Accepted Nov 3, 2025
Published Jan 13, 2026

Key words and phrases
Computational geometry, star partition, dynamic programming

Mikkel Abrahamsen [a]
Joakim Blikstad [b]
André Nusser [c]
Hanwen Zhang [a]

a University of Copenhagen, Denmark

b Centrum Wiskunde & Informatica, Amsterdam, Netherlands

c Université Côte d'Azur, CNRS, Inria, Sophia Antipolis, France

ABSTRACT. We devise a polynomial-time algorithm for partitioning a simple polygon P into a minimum number of star-shaped polygons. The question of whether such an algorithm exists has been open for more than four decades [Avis and Toussaint, Pattern Recognit., 1981] and it has been repeated frequently, for example in O'Rourke's famous book [*Art Gallery Theorems and Algorithms*, 1987]. In addition to its strong theoretical motivation, the problem is also motivated by practical domains such as CNC pocket milling, motion planning, and shape parameterization.

The only previously known algorithm for a non-trivial special case is for P being both monotone and rectilinear [Liu and Ntafos, Algorithmica, 1991]. For general polygons, an algorithm was only known for the restricted version in which Steiner points are disallowed [Keil, SIAM J. Comput., 1985], meaning that each corner of a piece in the partition must also be a corner of P. Interestingly, the solution size for the restricted version may be linear for instances where the unrestricted solution has constant size. The covering variant in which the pieces are star-shaped but allowed to overlap—known as the *Art Gallery Problem*—was recently shown to be $\exists\mathbb{R}$-complete and is thus likely not in NP [Abrahamsen, Adamaszek and Miltzow, STOC 2018 & J. ACM 2022]; this is in stark contrast to our result. Arguably the most related work to ours is the polynomial-time algorithm to partition a simple polygon into a minimum number of *convex* pieces by Chazelle and Dobkin [STOC, 1979 & Comp. Geom., 1985].

A preliminary version of this article appeared at STOC 2024 [3].
Mikkel Abrahamsen and Hanwen Zhang are supported by Independent Research Fund Denmark, grant 1054-00032B, and by the Carlsberg Foundation, grant CF24-1929.

Cite as Mikkel Abrahamsen, Joakim Blikstad, André Nusser, Hanwen Zhang. Minimum Star Partitions of Simple Polygons in Polynomial Time. TheoretiCS, Volume 5 (2026), Article 2, 1-68.

https://theoretics.episciences.org
DOI 10.46298/theoretics.26.2

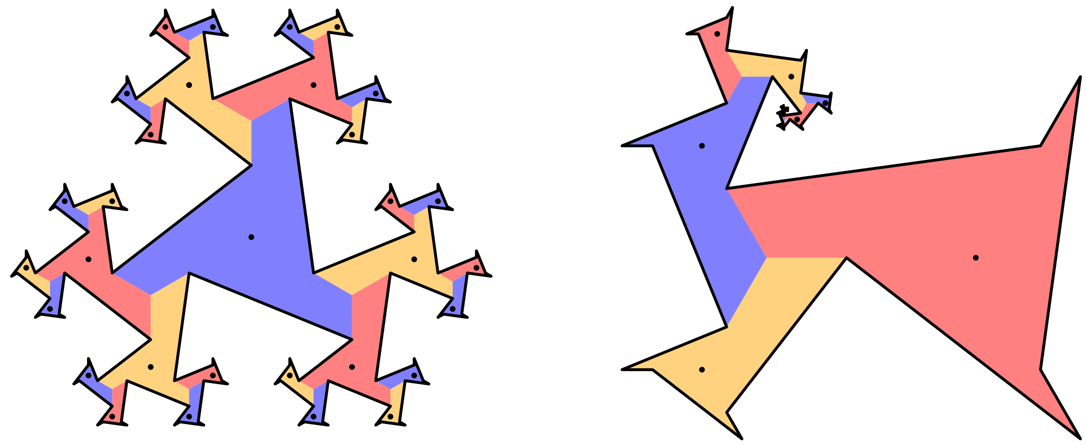

Figure 1. Repeating the patterns, we obtain polygons where star centers and corners of pieces of arbitrarily high degree are required.

1. Introduction

A simple polygon Q is called *star-shaped* if there is a point A in Q such that for all points B in Q, the line segment AB is contained in Q. Such a point A is called a *star center* of Q. A *star partition* of a polygon P is a set of pairwise non-overlapping star-shaped simple polygons whose union equals P; see Figure 1. The polygons constituting the star partition are called the *pieces* of the partition.

Avis and Toussaint [8] described in 1981, an algorithm running in $O(n \log n)$ time to partition a simple polygon (i.e., a polygon without holes) into at most $\lfloor n/3 \rfloor$ star-shaped pieces—where n denotes the number of corners of the polygon—based on Fisk's constructive proof [30] of Chvátal's Art Gallery Theorem [22]. Avis and Toussaint [8] wrote: "*An interesting open problem would be to try to find the decomposition into the minimum number of star-shaped polygons.*" This question has been repeated in several other papers [50, 44, 33, 49] and also in O'Rourke's well-known book [43]: "*Can a variant of Keil's dynamic programming approach [33] be used to find star partitions permitting Steiner points[1]? Chazelle was able to achieve $O(n^3)$ for minimum convex partition with Steiner points via a very complex dynamic programming algorithm [16], but star partitions seem even more complicated.*" Before our work, the problem was not known to be in NP and not even an exponential-time algorithm was known. In this paper, we resolve the open problem by providing a polynomial-time algorithm, thereby closing a research question that has been open for more than four decades.

THEOREM 1.1. *There is an algorithm performing $O(n^{105})$ arithmetic operations that partitions a simple polygon with n corners into a minimum number of star-shaped pieces. The number of bits used to represent each Steiner point in the constructed solution is $O(K)$ where K is the total number of bits used to represent the corners of P.*

1 A *Steiner point* is a corner of a piece in the partition which is *not* a corner of the input polygon. We discuss the challenges and importance of allowing Steiner points later in this section.

Related work. The minimum star partition problem belongs to the class of *decomposition problems*, which forms an old and large sub-field in computational geometry. In all of these problems, we want to *decompose* a polygon P into polygonal *pieces* which are in some sense simpler than the original polygon P. Here, the union of the pieces should be P, and we usually seek a decomposition into as few pieces as possible. A decomposition where the pieces may overlap is called a *cover*, and a decomposition where the pieces are pairwise interior-disjoint is called a *partition*. This leads to a wealth of interesting problems, depending on the assumptions about the input polygon P and the requirements on the pieces. There is a vast literature about such decomposition problems, as documented in several highly-cited books and survey papers that give an overview of the state-of-the-art at the time of publication [35, 15, 43, 49, 21, 34, 45]. Some of the most common variations are

- whether the input polygon P is simple or may have holes,
- whether we seek a cover or a partition,
- whether we allow Steiner points[1] or not,
- what shape of pieces we allow; let us mention that for *partitioning* a simple polygon, variants have been studied with polygonal pieces that are convex [29, 47, 19, 33, 14, 32, 36], star-shaped [29, 33, 40], monotone [31, 39], spiral-shaped [33], "fat" [51, 23, 12], "small" [5, 24], "circular" [25], triangles [7, 18], quadrilaterals [41, 42] and trapezoids [6].

Closely related to our problem is that of *covering* a polygon with a minimum number of star-shaped pieces. This is usually known as the *Art Gallery Problem* and described equivalently as the task of placing guards (star centers) so that each point in the polygon can be seen by at least one guard. Interestingly, the Art Gallery Problem has been shown to be $\exists\mathbb{R}$-complete [2] and it is thus not likely to be in NP. This is in stark contrast to our main result, which shows that the corresponding *partitioning* problem is in P. Covering a polygon with a minimum number of convex pieces is likewise $\exists\mathbb{R}$-complete [1].

If the polygon P can have holes, the minimum star partition problem is known to be NP-hard, whether or not Steiner points are allowed [43]; again in contrast to our result.

Keil [33] gave polynomial-time algorithms for partitioning simple polygons into various types of pieces where Steiner points are not allowed. Among these algorithms is an $O(n^7 \log n)$ time algorithm for finding a minimum star partition of a simple polygon without Steiner points, but the unrestricted version of the problem (with Steiner points allowed) remained open. Let us mention that there are polygons where $\Theta(n)$ pieces are needed when Steiner points are not allowed, whereas 2 pieces are sufficient when they are allowed; see Figure 2 (left). Therefore, our algorithm in general constructs partitions that are significantly smaller. This highlights an interesting difference between minimum star partitions and convex partitions: A minimum convex partition without Steiner points has at most 4 times as many pieces as when Steiner points are allowed [32]. Another difference is that an arbitrarily small perturbation of a single corner can change the size of the minimum star partition between 1 and $\Theta(n)$, whereas the

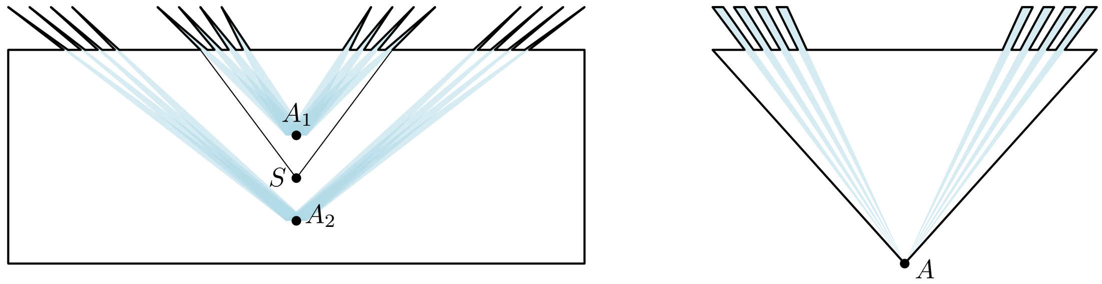

Figure 2. *Left:* A polygon that is partitioned into two star-shaped pieces using the Steiner point S. A star center that can see the two middle groups of spikes must be placed at or close to A_1, while a star center that sees the outer groups must be at or close to A_2. Without Steiner points we need $\Theta(n)$ pieces to partition at least one of the groups of spikes. *Right:* Moving the bottom corner a bit up changes the size of a minimum star partition from 1 to $\Theta(n)$.

change in size of the minimum convex partition is at most 1; see Figure 2 (right). In that sense, minimum star partitions are much more sensitive to the input.

Unrestricted partitioning problems (that is, allowing Steiner points), are seemingly much more challenging to design algorithms for. Chazelle and Dobkin [16, 19] proved already in 1979 that a simple polygon can be partitioned into a minimum number of *convex* pieces in $O(n^3)$ time, by designing a rather complicated dynamic program. Asano, Asano and Imai [6] gave an $O(n^2)$-time algorithm for partitioning a simple polygon into a minimum number of trapezoids, each with a pair of horizontal edges. However, the minimum partitioning problem has remained open for most other shapes of pieces (e.g. triangles, spiral-shaped, and—until now—star-shaped).

Liu and Ntafos [40] also studied the minimum star partition problem, but with restrictions on the input polygon. They describe an algorithm for partitioning simple monotone and rectilinear[2] polygons into a minimum number of star-shaped polygons in $O(n)$ time, and a 6-approximation algorithm for simple rectilinear polygons that are not necessarily monotone.

Challenges. As argued above, star partitions are very sensitive to the input polygon, and allowing Steiner points is in general necessary to obtain a partition with few pieces (Figure 2). In order to demonstrate the complicated nature of optimal star partitions, let us also consider Figure 1, which shows (representatives of) two families of polygons with arbitrarily many corners and unique optimal star partitions. In both examples, some star centers and Steiner points depend on as many as $\Theta(n)$ corners of P. The example to the right shows that star centers and Steiner points of *degree* $\Theta(n)$ are also needed, where points V_i of degree i are defined as follows. The points V_0 are the corners of P; and V_{i+1} are the intersection points between two non-parallel lines, each through a pair of points in V_i. The size of V_i grows as $\Theta(n^{4^i})$, so we

2 A polygon P is *monotone* if there is a line ℓ such that the intersection of P with any line orthogonal to ℓ is connected, and P is *rectilinear* if all sides are either vertical or horizontal.

cannot iterate through the possible star centers and Steiner points. This is in contrast to the problem without Steiner points studied by Keil [33]. Here, by definition, the corners of the pieces are in V_0 and it is not hard to see that the star centers can be chosen from V_1, of which there are "only" $O(n^4)$.

Since we cannot iterate through all possible star centers and Steiner points, we devise a two-phase algorithm, as follows. In the first phase, we find polynomially many relevant points, so that we are sure that an optimal solution can be constructed using a subset of those points as star centers and Steiner points. In the second phase, we use dynamic programming to find optimal solutions to larger and larger subpolygons, using only the constructed points from the first phase. We note however that the phases are intertwined as the algorithm for the first phase calls the complete partitioning algorithm recursively on subpolygons. The argument that the set of points constructed in the first phase is sufficient relies on several structural results about optimal star partitions which we believe are interesting in their own right.

1.1 Practical Motivation

Besides being interesting from a theoretical angle, star partitions are useful in various practical domains; below we mention a few examples. Many of the papers mentioned below describe algorithms for computing star partitions with no guarantee of finding an optimal one.

CNC pocket milling. Our first motivation comes from the generation of toolpaths for milling machines. CNC milling is the computer-aided process of cutting some specified shape into a piece of material—such as steel, wood, ceramics, or plastic—using a milling machine. When milling a pocket, spirals are a popular choice of toolpath, since the entire pocket can be machined without retracting the tool and sharp corners on the path can be largely avoided. Some of the proposed methods to generate spirals require the shape of the pocket to be star-shaped, for instance because they rely on radial interpolation between curves that morph a single point (a star center) to the boundary of the pocket [11, 46, 10]. When milling a non-star-shaped pocket, we therefore seek to first partition the pocket into star-shaped regions, each of which can then be milled by their own spiral. We want a star partition rather than a star cover, since it is a waste of time to cover the same area more than once. In order to minimize the number of retractions (lifting and moving the tool from one spiral to the next), we want a partition into a minimum number of star-shaped regions.

Motion planning. Star partitions are also useful in the domain of motion planning. Varadhan and Manocha [52] describe such an approach. They first partition the free space into star-shaped regions to subsequently construct a route for an agent from one point to another in the free space using the stars. In each star, we route from the point of entrance to the star center

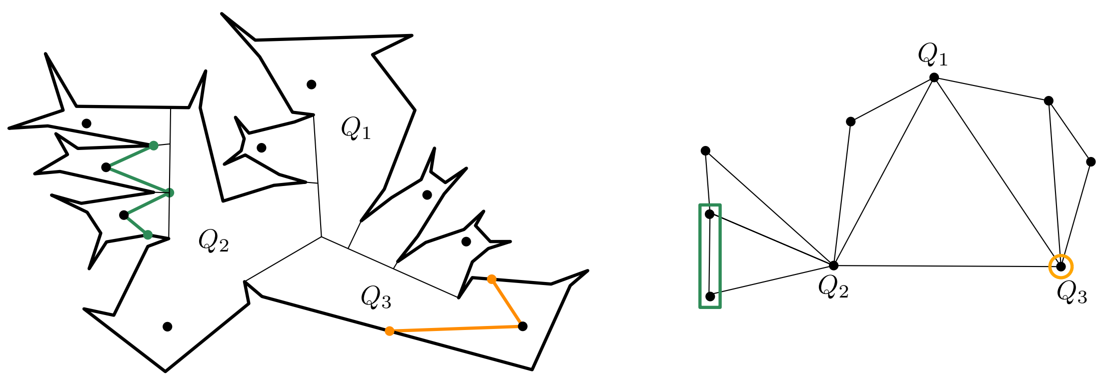

Figure 3. *Left:* A polygon with a star partition and an example of a short (orange) and a long (green) separator. *Right:* The dual graph of the partition. The short and long separators of the partition correspond to vertices, respectively edges, in the graph.

and from there to a common boundary point with the next star. Similar applications of star partitions are described in [26, 37, 38, 55].

Capturing the shape of a polygon. Star partitions can be used to blend/morph one polygon into another [48, 27], for shape matching and retrieval [54], and they are also used in shape parameterization [53].

1.2 Technical Overview

To enable our algorithm, we had to identify a multitude of interesting structural properties of optimal star partitions, which are interesting in their own right. In this section, we outline the most important of these properties and explain informally how they are used to derive a polynomial-time algorithm. Naturally, we sometimes stay vague or glance over complicated details in order to hide complexity to make the technical overview easily accessible.

Separators. Similar to the algorithms for related partitioning problems [19, 33], we use dynamic programming: We compute optimal star partitions of larger and larger subpolygons P' contained in the input polygon P. For dynamic programming to work, we need an appropriate type of separator which separates the subpolygon P' from the rest of P. To this end, a useful (and non-trivial) property is that there exists an optimal partition in which each piece shares boundary with P; as we will see in Section 3 (Corollary 3.11). This suggests that we use separators consisting of two or four segments of the following forms:

- *Short separator:* B_1-A_1-B_2. A piece with star center A_1 that shares boundary points B_1 and B_2 with P.
- *Long separator:* B_1-A_1-S-A_2-B_2. Each A_i is the star center of a piece that shares the boundary point B_i with P and the point S is a common point of the boundaries of the two pieces.

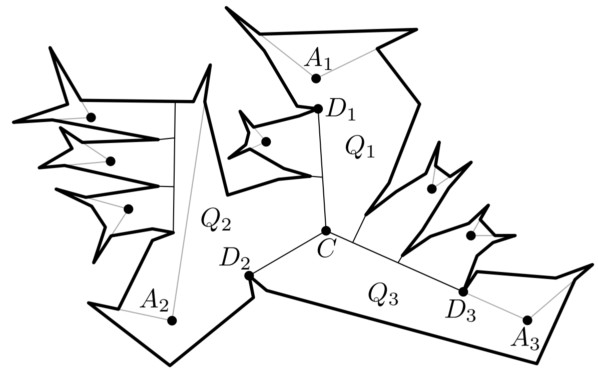

Figure 4. The same polygon and partition as in Figure 3, where the pieces Q_1, Q_2, Q_3 form a tripod with supports D_1, D_2, D_3 and tripod point C. The star centers are coordinate maximum and the gray segments show how they are constructed. The tripod is used to construct A_3.

A state of our dynamic program consists of a separator and is used to calculate how many pieces we need to partition the associated subpolygon, which is the part of P on one side of the separator. We start with trivial short separators of two types: (i) degenerate ones of the form B-A-B for a star center A that can see a boundary point B, and (ii) B_1-A-B_2 where B_1 and B_2 are points on the same edge of P so that the separator encloses a triangle. We describe a few elementary operations to create partitions of larger subpolygons from smaller ones by merging two compatible separators into one that covers the union of the two subpolygons.

The main difficulty lies in choosing polynomially many candidates for the star centers A_i, the boundary points B_i and the common points S, so that we can be sure that our algorithm eventually constructs an optimal partition. As already mentioned, our algorithm has two phases, and in the first phase we compute a set of $O(n^6)$ points that are guaranteed to contain the star centers of an optimal partition. In Section 4, we show that we can use these potential star centers to also specify polynomially many candidates for the points B_i and S. In a second phase, the algorithm uses the constructed points to iterate through all relevant separators.

Tripods. A structure that plays a crucial role in our characterization of the star centers is that of a *tripod*; see Figure 4 for an example of a partition with one tripod. Three pieces Q_1, Q_2, Q_3 with star centers A_1, A_2, A_3 form a *tripod* with *tripod point* C if the following two properties hold.

- There are concave corners D_1, D_2, D_3 of P such that $D_i \in A_iC$ for each $i \in \{1, 2, 3\}$. These corners are called the *supports* of the tripod.
- The union $Q_1 \cup Q_2 \cup Q_3$ contains a (sufficiently small) disk centered at C.

Note that it follows that the segment D_iC is on the boundary of the piece Q_i. Such a segment D_iC is called a *leg* of the tripod. Furthermore, the edges of P incident to the supports D_i are either all to the left or all to the right of the legs (when each leg D_iC is oriented from D_i to C); otherwise a disk around C could not be seen by the star centers A_i.

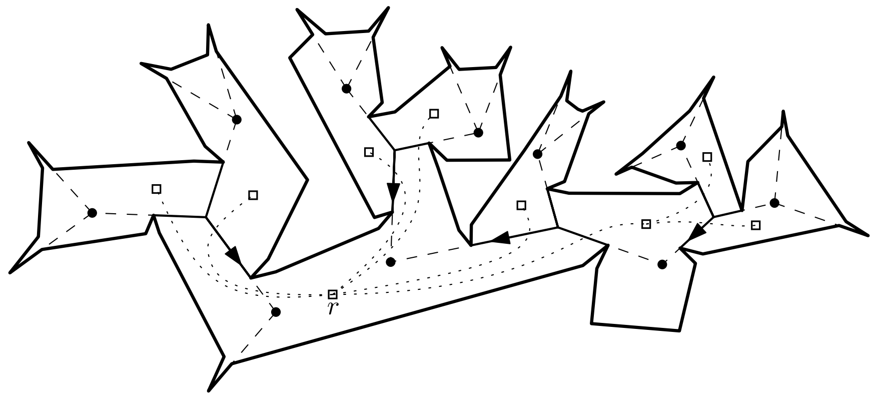

Figure 5. A polygon P with a star partition using ten pieces and four tripods. The legs of the tripods partition P into nine faces $\mathcal{F}$. The disks are star centers and the squares denote the vertices of the dual graph G of the faces $\mathcal{F}$. The dashed segments indicate how the star centers are defined or used to define other centers by tripods. The tripods have consistent orientation towards the root r and the edges of the tree $\mathcal{T}$ are shown as dotted curves.

Constructing star centers. We can define a set of points containing the star centers as follows. Let V_0 be the corners of P and define recursively V_{i+1} as the intersection points between any two non-parallel lines each containing two points from V_i. It follows that $V_i \subset V_{i+1}$. Tripods cause star centers to depend on each other in complex ways: If two of the participating star centers A_1 and A_2 are in $V_i \setminus V_{i-1}$, then the tripod point C is in general in $V_{i+1} \setminus V_i$ and the third star center A_3 will be in V_{i+2}. See for instance Figure 1 for two examples; both with unique optimal star partitions. Here, all neighbouring pieces form tripods, and in the right figure only V_i with $i = \Omega(n)$ contains all the star centers of the optimal partition.

We obtain powerful insights about the solution structure by considering a so-called *coordinate maximum* optimal partition. We can write the star centers $A_1, \ldots, A_k$ of an optimal partition in increasing lexicographic order (that is, sorted with respect to x-coordinates and using the y-coordinates to break ties). We can then consider the vector of star centers $\langle A_1, \ldots, A_k \rangle$ which is maximum in lexicographic order among all sets of star centers of optimal partitions. We show that there exists a partition realizing the maximum, which is our *coordinate maximum* partition (Lemma 2.3). The star centers of the partition in Figure 4 have been maximized in this sense. By analyzing a coordinate maximum partition, we conclude in Section 3 (Lemma 3.2) that there are essentially only two ways in which a star center A can be restricted. In both cases, A is forced to be contained in a specific half-plane H bounded by a line ℓ, and ℓ is of one of the following types: (i) ℓ contains two corners of P, (ii) ℓ contains a tripod point C and one of the associated supporting concave corners D_i. The star center A can then be chosen as the intersection point between two lines, each of type (i) or (ii). Note that in each tripod, the legs $D_1C \cup D_2C \cup D_3C$ partition P into three parts; since P is a simple polygon, it is thus impossible that the star centers depend on each other in a cyclic way. It follows that the star centers can be chosen from V_i for a sufficiently high value of i.

Orientation of tripods. Each tripod is defined from two of the participating star centers, say A_1 and A_2, and takes part in defining the third star center, A_3. Hence, we can consider the tripod to have an orientation: it is directed from (A_1, A_2) towards A_3. The legs of all tripods partition P into a set of faces $\mathcal{F}$; see Figure 5. One face can contain several pieces, since the tripod legs are in general only a subset of the piece boundaries. We will denote one of the faces as the root r. The faces $\mathcal{F}$ induce a dual graph G, in which each tripod corresponds to a triangle in G. Traversing $\mathcal{F}$ in breadth-first search order from the root r defines a rooted tree T, which is a subgraph of G. Each node u in T has an even number of children—two for each tripod for which u is the face closest to r among the three faces containing the pieces of the tripod. In order to successfully apply dynamic programming, we need the tripods to have a *consistent* orientation in the following sense: If the face u is a parent of v, then the corresponding tripod should be directed towards the star center in u. As we will see in Section 3 (part of Theorem 3.5), there exists an optimal partition where the tripods have a consistent orientation. This requires a modification to the coordinate maximum partition: Whenever a tripod violates the desired orientation, we choose a subset of the star centers and move them in a specific direction as much as possible to eliminate the illegal tripod. We describe such a process that must terminate, and then we are left with tripods of consistent orientation.

With consistent orientation, the star centers in the leaves of T belong to the set V_1 (which is constructed from lines through the corners of P), and in general, the star centers in a face u can be constructed by tripods involving centers in the children of u as well as lines through the corners of P bounding the face u. The star centers in the root face r are constructed at last, potentially depending on all previously constructed star centers.

Greedy choice. Consider three concave corners D_1, D_2, D_3 of P which are the supports of a tripod in an optimal partition Q. Let the associated star centers be A_1, A_2, A_3 and suppose that the tripod is directed towards A_3. The three shortest paths in P between these supports enclose a region which we call the *pseudo-triangle* Δ of the tripod; see Figure 6 (left). As we will see in Section 3 (part of Lemma 3.2), there is an optimal star partition where no star centers are in the pseudo-triangle of any tripod, and this is a property we maintain throughout our modifications. Consider a connected component P' of $P^- := P \setminus \Delta$. Note that P' is separated from the rest of P by a single diagonal of P which is part of the boundary of Δ. Since no star centers are in Δ, the restriction of Q to P' is a star partition of P'. Furthermore, in the optimal star partition we are working with, we can assume that this restriction is a *minimum* partition of P', since otherwise we could replace the partition in P' by one with less pieces and use an extra piece to cover Δ, thereby obtaining an equally good partition of P without this tripod.

Let P_i be the connected component of P^- containing A_i, so that the tripod is used to define the star center A_3 in P_3 using star centers A_1 and A_2 in P_1 and P_2, respectively; see Figure 6 (right). In all connected components except P_1, P_2, P_3, we can choose an arbitrary optimal

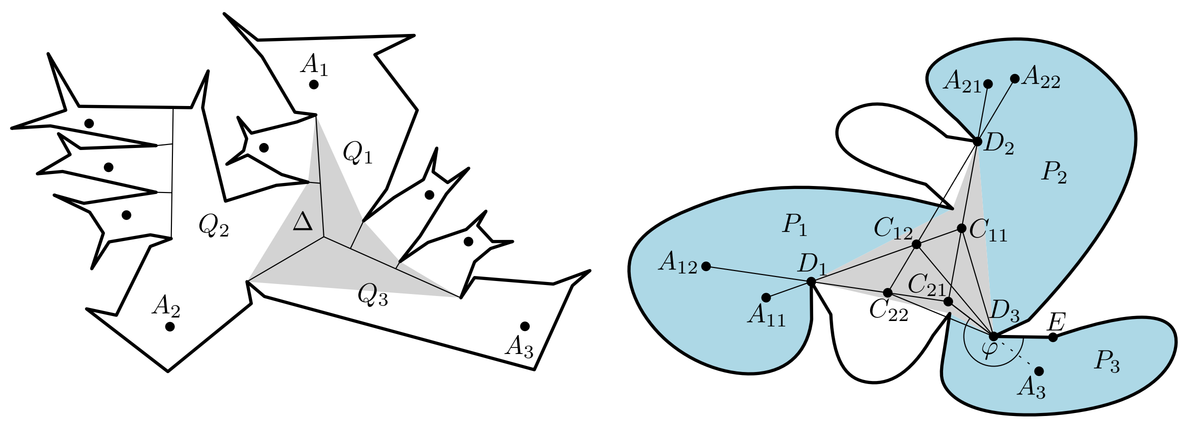

Figure 6. *Left:* A partition with a tripod and the pseudo-triangle shown in gray. *Right:* For a tripod with supports D_1, D_2, D_3 directed towards P_3, we find the optimal partitions in the subpolygons P_1 and P_2. There are two choices for the star centers that see D_1 and D_2, leading to four possible sets of legs of the tripod. We want to choose the combination that minimizes the angle φ from the edge D_3E clockwise to the leg to D_3. The angle is minimized when choosing A_{12} and A_{22}, but this choice is invalid since the legs D_2C_{22} and D_3C_{22} would intersect the boundary of P. We will therefore choose A_{12} and A_{21}, which give the second-best option with the tripod point C_{21}. Curved parts indicate that the details have not been shown.

partition. There may be several optimal partitions of P_1 and P_2, and any combination of two partitions may lead to different legs of the tripod (since A_1 and A_2 may be placed differently) and thus to different restrictions on the center A_3 in P_3. In fact, there can be an exponential number of possible restrictions on A_3. However, as shown in Section 5.2.1, we can apply a *greedy choice*: We can use the combination of partitions of P_1 and P_2 that leads to the mildest restriction on A_3, in the sense that we want to minimize the angle φ inside P between the leg D_3C and the edge of P incident to D_3 which is also an edge of P_3. Hence, we can use the greedy choice to restrict our attention to a single pair of optimal partitions of the subpolygons P_1 and P_2.

Bounding star centers and Steiner points. There are $O(n^3)$ possible triples of supports of tripods and using the greedy choice, we can restrict our attention to a specific pair of star centers that define the third star center for each tripod. Since a star center may be defined from two tripods, we get a bound of $O(n^6)$ on the number of star centers that we need to consider.

We also need polynomial bounds on the other points defining the separators, namely the boundary points B_i that the pieces share with P and the points S that neighbouring pieces share with each other. Some of these points may be corners of P, but the rest will be Steiner points, i.e., not corners of P. Suppose that we know the star centers $\mathcal{A} = \{A_1, \ldots, A_k\}$ of the pieces in an optimal partition. We can then consider a partition $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ where A_i is a star center of Q_i and we have maximized the vector of areas $\langle a(Q_1), \ldots, a(Q_k)\rangle$ in lexicographic order. As we will see (Lemma 2.5), such a partition $\mathcal{Q}$ exists (for any fixed set of star centers $\mathcal{A}$), and in Section 4 we show that $\mathcal{Q}$ has the property that each Steiner point in the interior of P is defined by at most five star centers and two corners of P. Hence, there are at most $O(n^{6\cdot 5} \cdot n^2) = O(n^{32})$ relevant Steiner points to try out. A Steiner point on the boundary ∂P will be defined by an edge of P and, in the worst case, a line through two star centers, which gives $O(n^{13})$ possibilities. Hence, we can bound the number of possible long separators by $O(n^{13} \cdot n^6 \cdot n^{32} \cdot n^6 \cdot n^{13}) = O(n^{70})$.

In Appendix B, we give an elementary proof of a structural result that reduces the number of Steiner points needed on the boundary of P to only $O(n)$. It might be possible to use this result to design a faster algorithm than the one presented here, but the proof relies on many modifications to the partition, so it is not clear if our algorithm can be modified to find the resulting partition.

Algorithm. Our algorithm now works as follows. In the first phase, we consider each diagonal of P, and we recursively find all relevant optimal partitions of the subpolygon on one side of the diagonal. Once this has been done for all diagonals, we consider each possible triple of concave corners of P supporting a tripod, and we use the greedy choice to select the pair of star centers that can be used to define the third star center of the tripod. We then construct $O(n^6)$ possible star centers by considering all pairs of (i) tripods, (ii) lines through two corners of P, and (iii) one tripod and one line through two corners of P. The set of potential star centers leads to polynomially many Steiner points and separators as described above. In the second phase, we use dynamic programming to find out how many pieces we need in the subpolygon defined by each separator. The total running time turns out to be $O(n^{107})$ or within $O(n^{105})$ arithmetic operations.

1.3 Open Problems & Discussion

Although polynomial, our algorithm is too slow to be of much practical use. Our main result is showing that the problem is polynomial-time solvable, so in order to facilitate understanding and verification of our work, we decided to give a description of the algorithm that is as simple as possible, and consequently we did not further optimize the running time. Although we believe that it is possible to optimize the algorithm significantly (for instance using structural insights from Appendix B), it seems that our approach will remain impractical. Hence, it is interesting whether a practical constant-factor approximation algorithm exists. For the minimum convex partition problem, the following wonderfully simple algorithm produces a partition with at most twice as many pieces as the minimum [15]: For each concave corner C of the input polygon P, cut P along an extension of an edge incident to C until we reach the boundary of P or a previously constructed cut. It would be valuable to find a practical and simple algorithm for star partitions with similar approximation guarantees.

Higher-dimensional versions of the minimum star partition problem are also of great interest and we are not aware of any work on such problems from a theoretical point of view. The high-dimensional problems are similarly well-motivated from a practical angle, since in motion-planning the configuration space is in general high-dimensional and a star partition of the free space can then be used to find a path from one configuration to another, as described in Section 1.1 (in fact, all the cited papers related to motion planning [52, 26, 37, 38, 55] also describe a high-dimensional setting). We note that the three-dimensional version of the minimum convex partition problem already received some attention, e.g. [20, 9, 17].

Many interesting partitioning problems of simple polygons with Steiner points are still open. Surprisingly, one problem that remains open is arguably the most basic of all problems of this type, namely, that of partitioning a simple polygon P into a minimum number of *triangles*. If P has n corners, a maximal set of pairwise interior-disjoint diagonals always partitions P into $n-2$ triangles and finding such a triangulation is a well-understood problem with a long history, culminating in Chazelle's famous linear-time algorithm [18]. In general, however, there exist partitions into fewer than $n-2$ triangles and it is an open problem whether an optimal partition can be found in polynomial time. Asano, Asano, and Pinter [7] showed that a minimum triangulation without Steiner points can be found in polynomial time. When Steiner points are allowed, they gave examples of polygons in which points from the set V_1 are needed, and they conjecture that there are instances in which points from the set V_i for arbitrarily large values of i are needed (i.e. points which have arbitrarily large degrees). Another classical open problem is to partition a simple polygon into a minimum number of spirals with Steiner points allowed. A *spiral* is a polygon where all concave corners appear in succession. The problem of partitioning into spirals was originally motivated by feature generation for syntactic pattern recognition [29] and a polynomial-time algorithm finding the optimal solution to the problem without Steiner points is known [33]. However, no algorithm is known for the unrestricted problem.

We hope that our techniques may be useful when designing algorithms to solve the above-mentioned problems. In particular, considering extreme partitions can lead to natural piece boundaries which in turn can be exploited using a dynamic programming approach. Computing such partitions in two phases, first computing potential locations of Steiner points that are subsequently used in guessing separators of pieces in an optimal solution, presents itself as a general paradigm to attack problems of this type.

1.4 Organization

The remainder of this work is organized as follows. In Section 2, we define various types of polygons, partitions, and other central concepts. We also give lemmas ensuring the existence of partitions that are extreme in terms of the coordinates of the star centers or the areas of the pieces. In Section 3, we study coordinate maximum partitions and the structures arising from tripods. These structural properties help us find a set of polynomially many potential points to use as star centers. In Section 4, we study area maximum partitions. The insights gained on their structure help us characterize all other Steiner points to use as corners in our partition, given a set of potential star centers (coming from the previous section). Finally in Section 5, we show how to use our structural results to design our two-phase dynamic programming algorithm.

2. Preliminaries

In this section we first cover some basic definitions to then turn towards partitions that are maximum with respect to either the area of the pieces or the coordinates of the star centers.

2.1 Definitions

We say that a pair of segments *cross* if their interiors intersect.

Polygons. A *simple polygon* is a compact region in the plane whose boundary is a simple, closed curve consisting of finitely many line segments. For technical reasons, we allow the pieces of a partition to be weakly simple polygons. A *weakly simple polygon* Q is a simply-connected and compact region in the plane whose boundary is a union of finitely many line segments. In particular, a simple polygon is also a weakly simple polygon, but the opposite is not true in general. For instance, a weakly simple polygon Q may have a disconnected or even empty interior. However, just as for a simple polygon, a weakly simple polygon Q can be defined by its edges in counterclockwise order around the boundary. These edges form a closed boundary curve γ of Q. Since Q is weakly simple, some corners may coincide, and edges may overlap. A perturbation of γ that is arbitrarily small with respect to the Fréchet distance can turn Q into a simple polygon [4, 13]. This perturbation may involve the introduction of more corners. For instance, if Q is just a line segment, then Q has only two corners, and one more is needed to obtain a simple polygon. We denote the boundary of a (weakly) simple polygon Q as ∂Q.

We sometimes consider points that lie on so-called extensions. Given a polygon P and a segment $CD \subset P$, the *extension* of CD is the maximal segment $C'D'$ such that $CD \subset C'D' \subset P$.

Star-Shaped Polygons. A (weakly) simple polygon Q is called *star-shaped* if there is a point A in Q such that for all points B in Q, the line segment AB is contained in Q. Such a point A is called a *star center* of Q. We denote by $\ker(Q)$ the set of all star centers of Q, and it is well-known that $\ker(Q)$ is a convex polygonal region in Q. Throughout the paper, we use the symbol "Q" to denote a star-shaped polygon and "A" to denote a fixed star center of such a polygon. When proving our structural results, we repeatedly use the following lemma to trim some of the pieces of a star partition.

LEMMA 2.1. *Let Q be a star-shaped polygon with star center A, and let H be an open half-plane bounded by a line h. Let C be a connected component of the intersection $Q \cap H$ and suppose that $A \notin C$. Then $A \notin H$ and $Q' := Q \setminus C$ is also a star-shaped polygon with star center A.*

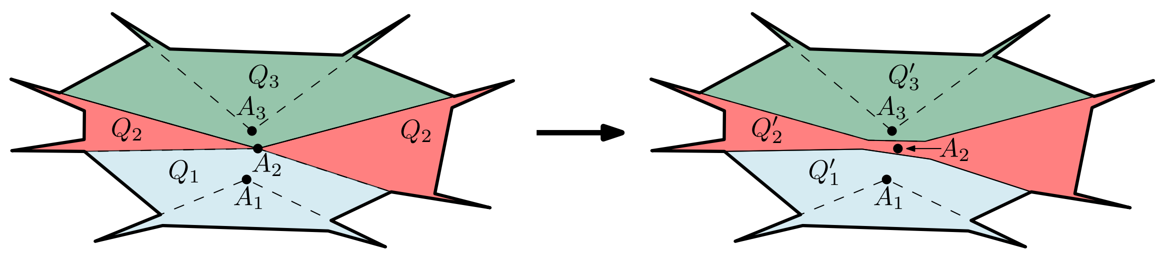

Figure 7. *Left:* A partition that our algorithm may produce, involving the piece Q_2 which is only weakly simple. *Right:* We can assign a bit of the neighbouring pieces to Q_2 and obtain a partition into simple polygons.

PROOF. First, if $A \in H$, then $Q \cap H$ consists of a single connected component as Q is star-shaped. However, this implies $A \in C$, which we assumed not to be the case. Thus, $A \notin H$.

Now consider a point $B \in Q'$. If $B \notin H$, then $AB \cap H = \emptyset$. Hence we also have $AB \subset Q'$, since $AB \subset Q$. Otherwise (if $B \in H$), then B is in a connected component D of $Q \cap H$ with D different from C. Let X be the intersection point of h and AB. Since $AB \subset Q$, we must have $XB \subset D$. As $D \subset Q'$, we then have $AB \subset Q'$. We therefore conclude that Q' is star-shaped. ■

Partitions. We will eventually consider star partitions of a modification $\tilde{P}$ of the input polygon P obtained by making incisions into the interior of P from corners. Thus, $\tilde{P}$ is a weakly simple polygon covering the same region as P, but $\tilde{P}$ has some extra edges on top of each other that stick into the interior of P. To accommodate this, we define star partitions in a way that allows both the input polygon P and the pieces to be weakly simple polygons. We define a *star partition* of a weakly simple polygon P to be a set of weakly simple star-shaped polygons $Q_1, \dots, Q_k$ such that after an arbitrarily small perturbation of P and $Q_1, \dots, Q_k$, we obtain *simple* polygons P' and $Q'_1, \dots, Q'_k$ with the following properties:

1. The polygons $Q'_1, \dots, Q'_k$ are pairwise interior-disjoint.
2. $\bigcup_{i=1}^{k} Q'_i = P'$.

Note that this implies that the weakly simple polygons $Q_1, \dots, Q_k$ must also have properties 1 and 2 (with P' replaced by P and Q'_i replaced by Q_i for all $i \in \{1, \dots, k\}$), since otherwise a large perturbation would be needed for them to be transformed into simple polygons with the required properties. However, it would not be sufficient to define a partition as a set of weakly simple polygons with properties 1 and 2 alone. This would, for instance, allow two pieces with empty interiors (such as two segments) to properly intersect each other, which is not intended. Our algorithm may produce weakly simple pieces which are not simple, since the boundary can meet itself at the star center; see Figure 7. As demonstrated in the figure, by applying Lemma 2.1, such a piece Q can "steal" a bit from the neighbouring pieces, which turns Q into a simple polygon Q', resulting in a partition consisting of simple polygons.

(Important) Sight Lines. Given a star-shaped polygon Q and a star center $A \in \ker(Q)$, each segment that connects a corner of Q with the center A is called a *sight line* of Q. A sight line ℓ is called an *important sight line* if it contains a corner D of P in its interior. We call D the *support* of ℓ. If there are multiple candidates, we define the corner farthest from the star center as the support.

Tripods. In a star partition, three pieces Q_1, Q_2, Q_3 with star centers A_1, A_2, A_3 form a *tripod* with *tripod point* C if the following properties hold.

- A_iC is an important sight line of Q_i with support D_i, for each $i \in \{1, 2, 3\}$. These concave corners D_1, D_2, D_3 of P are called the *supports* of the tripod.
- The union $Q_1 \cup Q_2 \cup Q_3$ contains a (sufficiently small) disk centered at C.
- The three pieces Q_1, Q_2, Q_3 have *strictly* convex corners at C.

Tripods can be necessary in optimal solutions, see Figure 1 for such an example.

The three segments D_1C, D_2C, D_3C are called the *legs* of the tripod. The polygon bounded by the three shortest paths in P between pairs of the supports D_1, D_2, D_3 is called the *pseudo-triangle* of the tripod, and these shortest paths are called *pseudo-diagonals*.

LEMMA 2.2. *Let $\mathcal{T}_1, \mathcal{T}_2$ be two distinct tripods in a star partition $\mathcal{Q}$. The interiors of the pseudo-triangles of $\mathcal{T}_1$ and $\mathcal{T}_2$ are disjoint.*

PROOF. The legs of $\mathcal{T}_1$ partition P into three regions R_0, R_1, R_2. Since tripod legs are boundary segment of pieces, they cannot cross each other. Hence, all the tripod legs of $\mathcal{T}_2$ must lie in one of R_0, R_1, R_2; without loss of generality, assume they lie in R_0. Then the pseudo-triangle of $\mathcal{T}_2$ is a subpolygon of R_0. Towards a contradiction, assume that the interiors of the two pseudo-triangles are not disjoint. It is impossible that $\mathcal{T}_2$ is contained in $\mathcal{T}_1$, since it would mean that the corners of $\mathcal{T}_2$ are a subset of the corners of one pseudo-diagonal of $\mathcal{T}_1$, and a pseudo-triangle cannot be made from corners on a concave chain. Hence, if $\mathcal{T}_1$ and $\mathcal{T}_2$ are not interior-disjoint, there is an edge e of the pseudo-triangle of $\mathcal{T}_2$ that crosses the boundary of the pseudo-triangle of $\mathcal{T}_1$; see Figure 8. As the pseudo-triangle of $\mathcal{T}_1$ in R_0 is bounded by the two tripod legs (which e does not cross) and a concave chain, the segment e must have an endpoint inside the pseudo-triangle. As both endpoints of e are supported by a concave corner of P, we obtain a contradiction with the fact that no vertex of P is contained in the interior of the pseudo-triangle. ■

2.2 Coordinate and Area Maximum Partitions

Coordinate Maximum Partition. We define the lexicographic order $\preceq$ of vectors $v_1, v_2 \in \mathbb{R}^d$ so that $v_1 \preceq v_2$ iff $v_1 = v_2$ or i is the first dimension that v_1 and v_2 differ in and $v_1[i] < v_2[i]$. Note that this definition carries over to star centers in a straightforward manner. For a star-shaped polygon Q, the *maximum* star center of Q is the star center (i.e. point in $\ker(Q)$) with

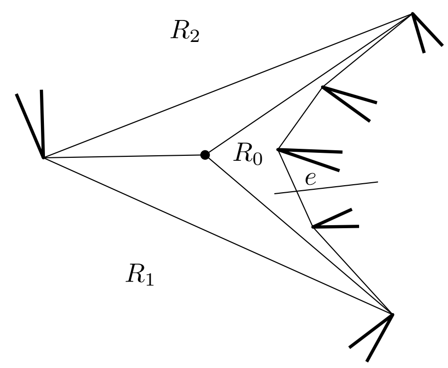

Figure 8. If the interiors of the pseudo-triangles of two tripods intersect, then either a vertex of a pseudo-triangle is in the other pseudo-triangle, or a pseudo-triangle crosses a tripod leg of another tripod.

the lexicographically largest value. For a polygon P, consider an optimal star partition Q with maximum star centers $A_1, \dots, A_k$ sorted in lexicographic order, and define $c(Q) = \langle A_1, \dots, A_k \rangle$ to be the combined coordinate vector. If $c(Q)$ is maximum in lexicographic order among all optimal star partitions of P, we say that Q is a *coordinate maximum* optimal partition. In other cases, it is useful to consider a partition with given star centers where the vector of *areas* of the pieces has been maximized. In this section, we provide lemmas that ensure the existence of such partitions. The proofs are deferred to Appendix A.

LEMMA 2.3. *For any simple polygon P, there exists a coordinate maximum optimal star partition.*

Restricted Coordinate Maximum Partitions. It will sometimes be necessary to change the direction in which we maximize a specific subset of star centers, while keeping the remaining ones fixed. Furthermore, we often have to restrict the star centers that we are optimizing to a subpolygon $F \subseteq P$. For this, we use the following generalization of Lemma 2.3, the proof of which is analogous. Given a vector $d \in \mathbb{R}^2$, we define $d^\perp \in \mathbb{R}^2$ to be the vector orthogonal to d obtained by rotating d counterclockwise by $\pi/2$.

LEMMA 2.4 (Restricted coordinate maximization). *Consider a simple polygon P and an optimal star partition with star centers $A_1, \dots, A_k$. Let $i \leq k$ and suppose that $A_i, \dots, A_k \in F$ for a polygon $F \subseteq P$. Let $d \in \mathbb{R}^2 \setminus \{(0,0)\}$ be a vector. There exists a star partition of P with star centers $A_1, A_2, \dots, A_{i-1}, A_i^*, A_{i+1}^*, \dots, A_k^*$ where $A_i^*, A_{i+1}^*, \dots, A_k^* \in F$ and $\langle A_i^* \cdot d, A_i^* \cdot d^\perp, A_{i+1}^* \cdot d, A_{i+1}^* \cdot d^\perp, \dots, A_k^* \cdot d, A_k^* \cdot d^\perp \rangle$ is maximum in lexicographic order among all star partitions with fixed star centers $A_1, \dots, A_{i-1}$ and for which the remaining star centers are restricted to F.*

The partition described in Lemma 2.4 is called the *restricted coordinate maximum* optimal star partition along d, within F and with fixed star centers $A_1, \dots, A_{i-1}$. Note that a coordinate maximum optimal partition is a restricted coordinate maximum one along $d = (1, 0)$, within P and with no fixed star center.

Area Maximum Partition. Consider a polygon P and a star partition $Q = \{Q_1, \ldots, Q_k\}$ of P with corresponding star centers $\mathcal{A} = \{A_1, \ldots, A_k\}$. We say that Q is *area maximum* with respect to $\mathcal{A}$ if the vector of areas $a(Q) = \langle a(Q_1), \ldots, a(Q_k)\rangle$ is maximum in lexicographic order among all partitions of P with star centers $\mathcal{A}$.

LEMMA 2.5. *Let P be a polygon and suppose that there exists a star partition of P with star centers $\mathcal{A} = \{A_1, \ldots, A_k\}$. Then there exists a partition which is area maximum with respect to $\mathcal{A}$.*

3. Structural Results on Tripods and Star Centers

In this section, we will present a construction process which can construct all the star centers and tripods in some optimal solution within linearly many steps. To achieve this goal, we first need to pick an optimal solution with good properties. We do this by considering *restricted coordinate maximum partitions* (see Lemma 2.4).

LEMMA 3.1. *Consider a simple polygon P and an optimal star partition $Q = \{Q_1, \ldots, Q_k\}$ with corresponding star centers $A_1, \ldots, A_k$. There exists an optimal star partition consisting of simple polygons with the same star centers, such that no four pieces meet in the same point and no star center lies in the interior of a sight line.*

PROOF. We first turn the weakly simple star partition into a star partition with simple polygons; see Section 2. We then modify the partition, not moving the star centers, so that no four pieces contain the same point. Assume that there exists a point C such that $C \in Q_1 \cap \cdots \cap Q_m$ for some $m \geq 4$. Without loss of generality, assume $Q_1, \ldots, Q_m$ appear in clockwise order around C. Let α_i be the angle of Q_i at C. Since $\sum_{i=1}^{m} \alpha_i \leq 2\pi$, we have either $\alpha_1 + \alpha_2 \leq \pi$ or $\alpha_3 + \alpha_4 \leq \pi$. Without loss of generality, assume $\alpha_1 + \alpha_2 \leq \pi$. We now decrease the number of pieces containing C, while not creating an intersection point of four or more pieces. Recall that A_i is the star center of Q_i. We consider two cases; see Figure 9:

- **A_1 is not on an extension of the shared boundary with A_2 or vice versa.** If A_1 is not on an extension of the shared boundary with A_2, then Q_1 can take a small enough triangle around C from Q_2, while the two new Steiner points in the partition are contained in at most three pieces, namely Q_1, Q_2, Q_3. A similar modification is possible if A_2 is not on an extension of the shared boundary with A_1.
- **Both A_1 and A_2 are on an extension of their shared boundary.** Without loss of generality, assume A_1 is closer to C than A_2. Then Q_1 can take a sufficiently small triangle around the segment A_1C, while not creating any new point where four pieces meet.

Hence, eventually we obtain a star partition that has no four pieces meeting in the same point.

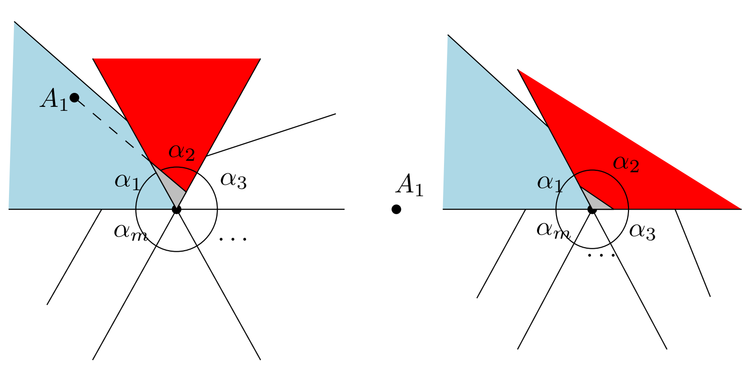

Figure 9. Reduce the number of pieces containing the same point. The two figures on the left show the modification we perform if none of the star centers are on an extension of the shared boundary with the other star center. The right figure shows the case when both star centers lie on an extension of the shared segment. Blue region marks the piece Q_1, red regions is the piece Q_2, and the gray region is going to be transferred from Q_2 to Q_1.

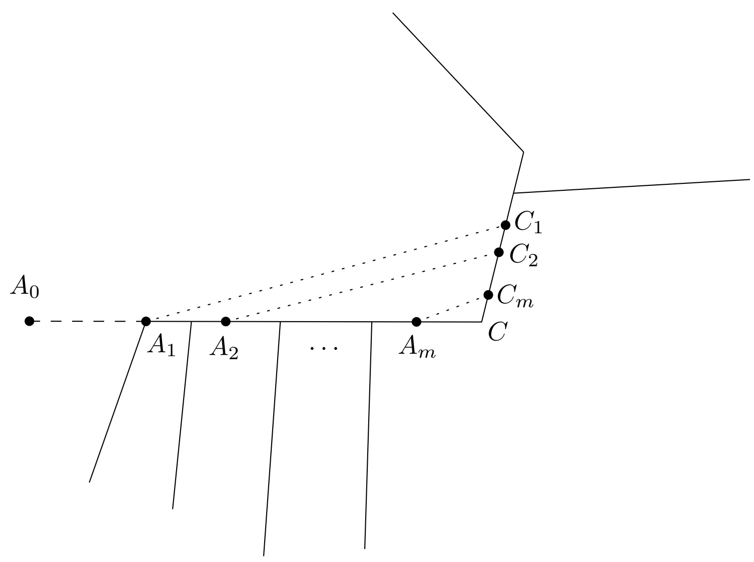

Figure 10. Redistributing pieces to remove a sight line that contains star centers in its interior.

We now modify the partition to remove all sight lines that contain some star center in their interior. Let $\ell = A_0C$ be a sight line that contains $A_1, A_2, \dots, A_m$ in its interior. We first choose a sequence of points $C_1, C_2, \dots, C_m$ along the next edge of Q_0; see Figure 10. We then give the quadrilateral $A_iA_{i+1}C_{i+1}C_i$ to piece Q_i for all $i \in \{1, 2, \dots, m-1\}$. Finally we give A_mCC_m to Q_m. This modification removes all star centers from the interior of one sight line while no newly created sight line contains a star center in its interior. It is easy to check that this modification of the partition does not make four pieces meet. ∎

The main tool in this section is *restricted coordinate maximum partitions* defined in Section 2. The following lemma captures one of our key combinatorial results on a star center in a restricted coordinate maximum partition from Lemma 2.4. Intuitively, if one moves a star center A_k of an optimal star partition in the direction d as far as possible without moving other star centers (but possibly changing what region of P each piece contains), then there are only a few reasons to get stuck.

LEMMA 3.2. *Consider the restricted coordinate maximum optimal star partition consisting of simple polygons $\mathcal{Q} = \{Q_1, \dots, Q_k\}$ along d and with fixed star centers $A_1, \dots, A_{k-1}$ (so that only the coordinates of the last center A_k have been maximized). Assume A_k is restricted within a polygon $F \subseteq P$. Suppose that no four pieces meet in the same point and no star center is in the interior of any sight line. Then A_k lies on the intersection of two non-parallel segments of the following types:*

- *an edge of F,*
- *an important sight line of Q_k, not containing any other star center, which is*
  - *an extension of an edge of P, or*
  - *on the extension of a diagonal of P that connects two concave corners, or*
  - *an extension of a tripod leg (see Section 2), and no star center is in the interior of the pseudo-triangle of this tripod.*

PROOF. We can choose A_k freely inside $\ker(Q_k) \cap F$ while all the pieces remain the same. By coordinate maximization, A_k must be a corner of $\ker(Q_k) \cap F$. Let $\mathcal{S}$ denote the set of edges e of Q_k such that A_k is on the extension of e. Let $\mathcal{S}'$ denote the set of edges of F that A_k lies on. Since all edges of $\ker(Q_k)$ come from extensions of edges of Q_k, there must be two non-parallel segments in $\mathcal{S} \cup \mathcal{S}'$.

We call a segment in $\mathcal{S}$ *good* if it is collinear to a segment that is of the described types in the lemma statement; otherwise, we call it *bad*. In the remainder of the proof, we modify the partition $\mathcal{Q}$ while *not moving any star centers* and *never creating any new important sight lines of Q_k*, which means that the two good segments we find at last are also good segment of the initial star partition $\mathcal{Q}$. At the same time, we decrease the number of bad segments in $\mathcal{S}$ until all segments in $\mathcal{S}$ are good. In the end, either A_k satisfies the lemma, or else we cannot find two non-parallel segments in $\mathcal{S} \cup \mathcal{S}'$, which would mean that A_k is not at a corner of $\ker(Q_k) \cap F$ therefore contradicting that A_k is optimal with respect to coordinate maximization.

In the remainder of the proof, all star centers and corners of P on ∂Q_k are considered as corners of Q_k, so there can be several collinear consecutive segments of Q_k. First, we consider the case that A_k is the *endpoint* of a bad segment in $\mathcal{S}$.

1. **A_k is a corner of P.** Then A_k must be a corner of F as $F \subset P$. Hence A_k is at two edges of P and the lemma holds.
2. **A_k is in the interior of an edge of P.** If A_k is a corner of F, then the lemma holds. Otherwise, A_k is in the interior of an edge of F that is collinear to the boundary of P. We can assign an arbitrarily small region around A_k to Q_k such that A_k is not an endpoint of a bad segment anymore; see Figure 11. Since we do not create any new important sight line in Q_k, we do not introduce any new good segments in $\mathcal{S}$. The only new bad segment we introduce

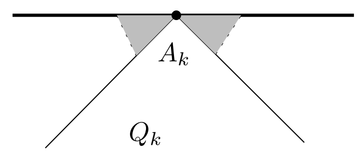

Figure 11. Dealing with the case that $A_k \in \partial P$. The gray region marks the region that we give to Q_k.

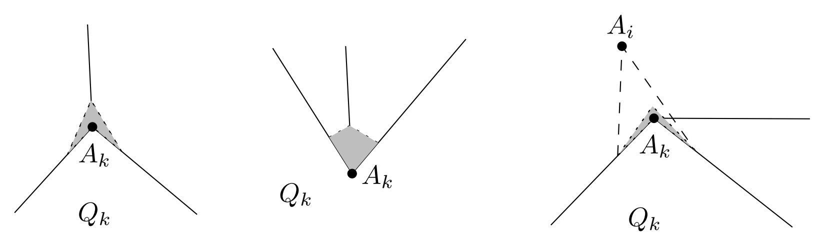

Figure 12. Dealing with the case that $A_k \in \partial Q_k$. The gray region marks the region that we give to Q_k. The first figure shows the case that A_k is a convex corner for all pieces; the second shows the case that A_k is a concave corner of Q_k; the third shows the case that A_k is a concave corner of other pieces.

is parallel to the boundary edge of P that A_k lies on, which can also be removed from S without breaking the assumption that $S \cup S'$ contains two non-parallel segments— since there must be a parallel edge of F.

3. **A_k is in the interior of P.** This can happen when A_k is either a convex corner of all pieces touching it, a concave corner of the piece Q_k, or a concave corner of some other piece. In all cases, we transfer a sufficiently small area around A_k to Q_k and thereby make A_k not be an endpoint of a bad segment; see Figure 12.

In the following A_k is not an *endpoint* of a bad segment in $\mathcal{S}$. Let $e = C_1C_2$ be a segment in $\mathcal{S}$ and let C_2 be the farther end of e from A_k. Let C_0 be the nearest vertex along A_kC_1 to A_k. Note that $C_0C_1 \subset \partial Q_k$. Note that this implies $C_0 \neq A_k$, as A_k is not the endpoint of a bad segment in $\mathcal{S}$. Furthermore, let C_3 be the farthest point from A_k in the direction of C_2 on the extension of A_kC_2 such that $C_2C_3 \subset \partial Q_k$ (it might be the case that $C_3 = C_2$). Let C_4 be the next corner of C_3 on ∂Q_k, and let C_{-1} be the previous corner of C_0 on ∂Q_k. We again consider multiple cases:

1. **Another star center A_i is on A_kC_3.** According to our assumptions in the lemma statement, no star center is in the interior of a sight line, thus we have $A_i = C_3$. We can then transfer the triangle $C_0C_3C_4$ from Q_k to Q_i and the number of bad segments in $\mathcal{S}$ is thereby reduced; see Figure 13.
2. **No corner of P is in the interior of the sight line A_kC_2.** Or equivalently, A_kC_2 is not an important sight line. In this case we give a sufficiently small triangle $C'C_0C_2$ to Q_k, where C' is sufficiently close to C_0 on the segment $C_{-1}C_0$; see Figure 14. According to Lemma 2.1, all pieces that are cut by the segment $C'C_2$ are still star shaped. This way we reduce the size of $\mathcal{S}$ by removing C_1C_2.

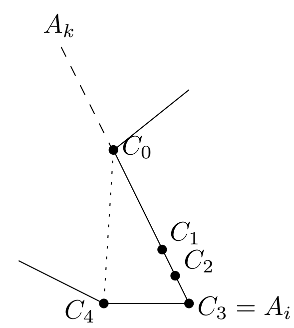

Figure 13. If the sight line ends at another star center A_i, we can give the triangle $C_0C_3C_4$ to Q_i and reduce $|\mathcal{S}|$.

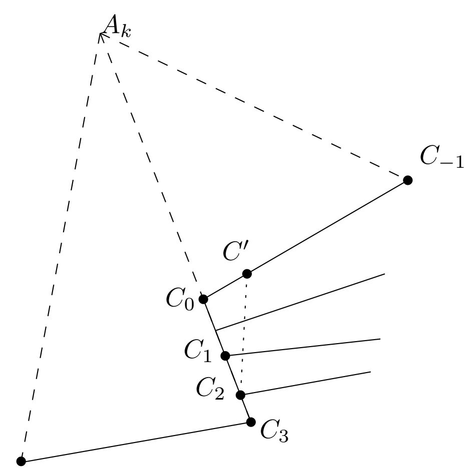

Figure 14. No corners of P on A_kC_2.

In the remainder we can assume that the sight line A_kC_2 is supported by a corner of P, i.e., it is an important sight line. Since A_kC_2 is covered by A_kC_3, A_kC_3 is also an important sight line. Let D be the support (Section 2) of A_kC_3. In the remainder we try to remove DC_3 from $\mathcal{S}$.

3. ***C_3* is a convex corner of *P* and not adjacent to *D*.** Note that if C_3 was a convex corner of P adjacent to D, then A_kC_3 would be an extension of an edge DC_3 of P, which is a good segment in $\mathcal{S}$. We can remove a sufficiently small triangle DC_3C' from Q_k for C' close enough to C_3 on segment C_3C_4, and distribute the triangle to the neighboring pieces by extending the edges that end at DC_3; see Figure 15. Since we do not create concave corners in any pieces they remain star-shaped. The same argument is also applicable if DC_3 ends in the interior of an edge of P as we can consider the intersection point as a degenerate convex corner of P.
   In the remainder C_3 is in the interior of P.
4. ***C_3* is a concave corner of some piece *Q_i*.** Then C_3 is a convex corner of $P \setminus Q_i$ and we can use a similar modification to remove DC_3 from $\mathcal{S}$ as in the previous case; see Figure 16.
   In the remainder, C_3 a convex corner of all its adjacent pieces. According to the assumption of the lemma that no four pieces meet at the same point, C_3 is contained in at most three

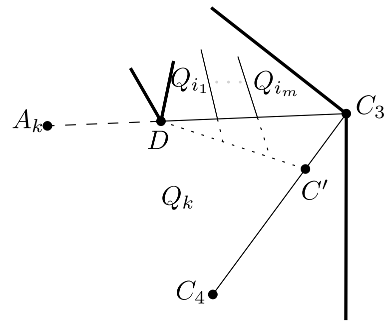

Figure 15. The case when the boundary of Q_k ends at a convex corner of P.

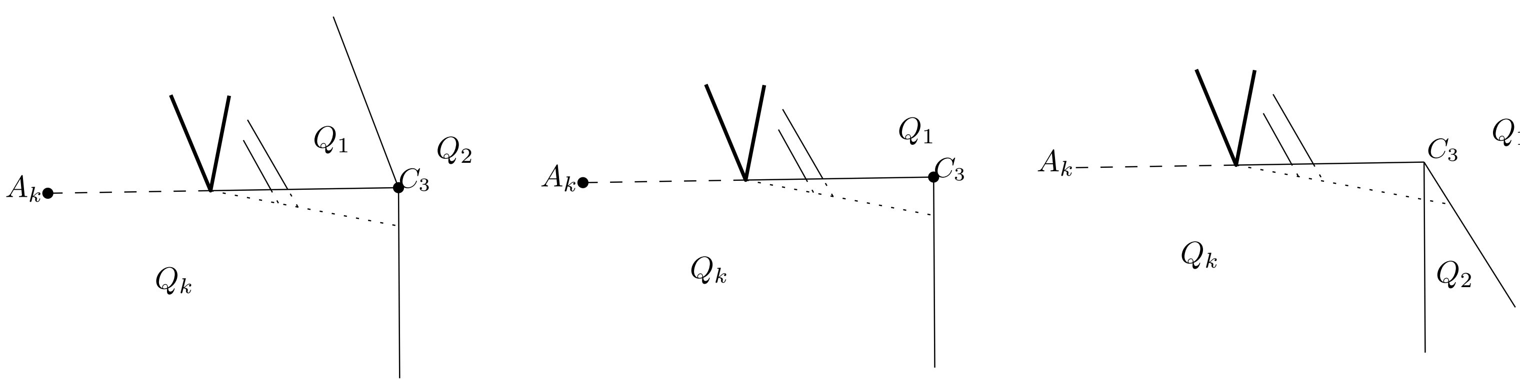

Figure 16. C_3 is a concave corner of some piece.

pieces. Since the angle of Q_k at C_3 is strictly less than π, there actually must be exactly three pieces containing C_3. With slight abuse of notation, let $Q_0 = Q_k, Q_1, Q_2$ be these three pieces in clockwise order; let α_i be the angle of Q_i at C_3; and let A_i be the star center of Q_i.

5. C_3 **is not a tripod point.** If an edge at C_3 is not covered by an important sight line, we can modify the partition and remove DC_3 from $\mathcal{S}$; see Figure 17. Otherwise, all edges at C_3 are covered by important sight lines. As C_3 is not a tripod point, we have that $\alpha_1 = \pi$ or $\alpha_2 = \pi$. If $\alpha_1 = \pi$, let D' be the support of A_1C_3. Now A_kC_3 is on the extension of the diagonal DD' that connects two concave corners of P. If $\alpha_2 = \pi$, then C_3 is a convex corner of $P \setminus Q_2$ and we can modify the partition similar to case 3.

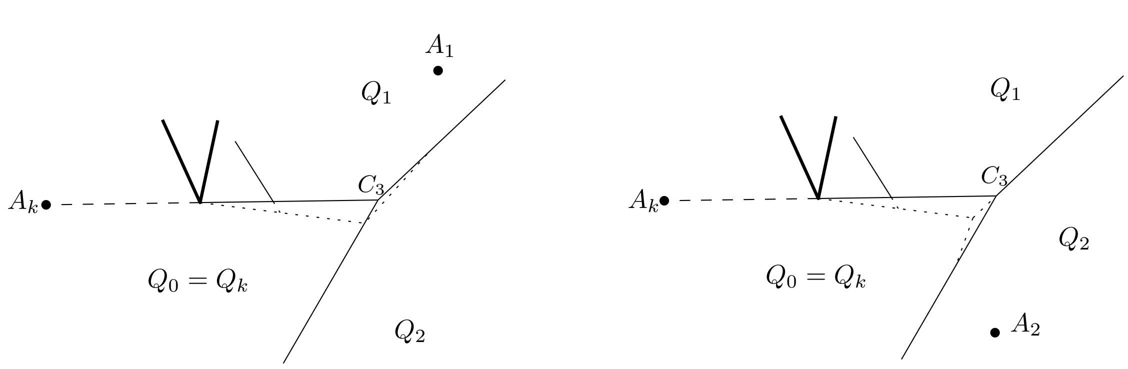

Figure 17. The case when an edge at C_3 is not covered by important sight line.

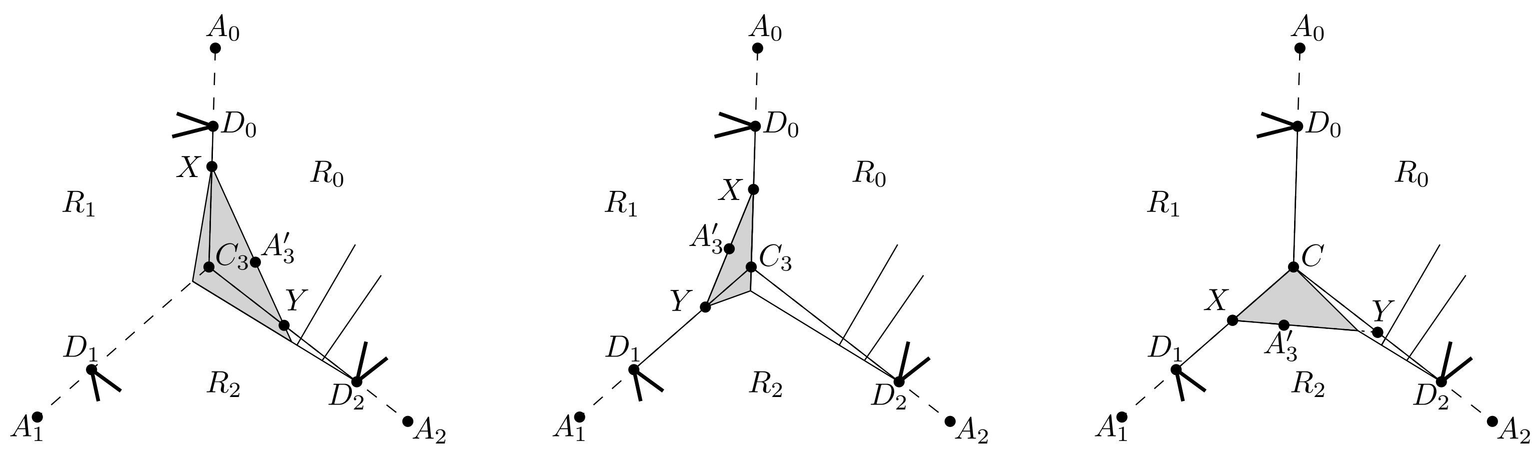

Figure 18. Three cases of where the star center A'_3 is and how we define the gray triangle to give to A'_3.

In the remainder, C_3 is a tripod point. Note that if the tripod associated with C_3 contains no star center in its pseudo-triangle, then the edge C_1C_2 is a good edge. Hence, the only remaining case is the following.

6. **C_3 is a tripod point and a star center is in the interior of the pseudo-triangle of this tripod.** Let A_3 be any star center inside the pseudo-triangle. The tripod partitions P into three regions R_0, R_1, R_2, where R_i is the region containing A_i. Let R'_i be the intersection of R_i and the pseudo-triangle. First we prove that there exists a segment XY from one leg of the tripod to another leg that contains a star center A'_3 and no star center is in the interior of triangle C_3XY. Let R'_j be the region that contains A_3. Consider the convex hull C of all corners of this pseudo-triangle and all star centers in R'_j. There exists a corner of C that is a star center A'_3, as A_3 lies in the interior of R'_j. Let ℓ be an arbitrary tangent of C at A'_3, and let XY be the subsegment of ℓ that is contained in the pseudo-triangle. Whichever region A_3 lies in, we can modify the partition so that the tripod is broken and $|\mathcal{S}|$ is not increased; see Figure 18. Since the number of possible tripods is finite, we can apply the argument a finite number of times and then either the size of $\mathcal{S}$ is decreased, or A_kC_3 becomes a good segment. ■

3.1 Tripod Trees

We now define what we call the *tripod tree*—a description of the structure of tripods in an optimal solution (see also Figure 20). Given a star partition, consider the partition that is induced by the tripod legs. Note that this partition is simply the star partition but with some pieces having been merged. We construct a bipartite graph $G = (X \cup Y, E)$ as follows:

- We add a vertex to X for each face of the partition induced by the legs.
- We add a vertex to Y for each tripod.
- We add an edge $\{x, y\}$ to E if and only if a tripod leg of y forms part of the boundary of x.

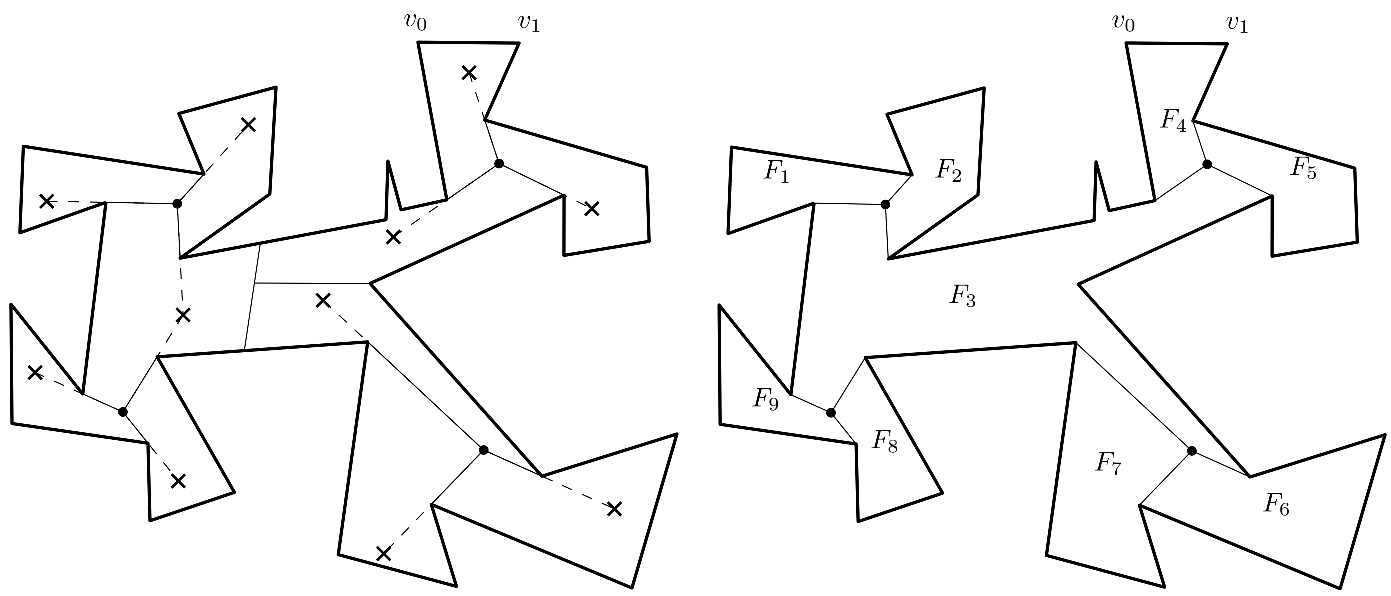

Figure 19. Illustration for a tripod tree. A cross represents a star center, a solid circle represents a tripod point. The left figure shows star partition and the right figure shows the faces splits by all the tripod legs.

OBSERVATION 3.3. *Given a star partition, the tripod tree graph G is indeed a tree.*

PROOF. If the tripods are considered degenerate pieces of the partition, then G corresponds to the dual graph of the partition induced by the legs. Thus, G is connected. Furthermore, note that every tripod cuts the polygon P into three disconnected pieces, so the corresponding vertex in Y is a cut vertex, which implies that G is a tree. ■

We choose the root of the tripod tree to be the face that contains the first edge of P, merely for consistency. For every tripod $\mathcal{T}$ formed by pieces Q_i, Q_j, Q_k where Q_i is contained in the parent face of $\mathcal{T}$, we call the star center A_i of Q_i the *parent star center* of $\mathcal{T}$ and the star centers A_j, A_k of Q_j, Q_k are both called *child star centers* of $\mathcal{T}$. Note that we can directly identify the parent star center of a tripod without the full tripod tree.

Fake tripod. In a star partition $\mathcal{Q}$, a *fake tripod* $\mathcal{T}'$ with tripod point C is defined by *two* star centers A_1, A_2 of pieces Q_1, Q_2 and *three* concave corners D_1, D_2, D_3 of P if the following properties hold.

- A_iC is an important sight line of Q_i with support D_i, for each $i \in \{1, 2\}$.
- $A_iC \subset \bigcup_{Q \in \mathcal{Q}} \partial Q$ for each $i \in \{1, 2\}$.
- The three angles $D_1CD_2, D_2CD_3, D_3CD_1$ are strictly convex.
- Let F_1, F_2, F_3 be the three connected components of P cut by $D_1C \cup D_2C \cup D_3C$, where $A_1 \in F_1, A_2 \in F_2, F_3$ contains the first edge of P. D_3 is a concave corner in F_3. The union $Q_1 \cup Q_2 \cup F_3$ contains a (sufficiently small) disk at C.

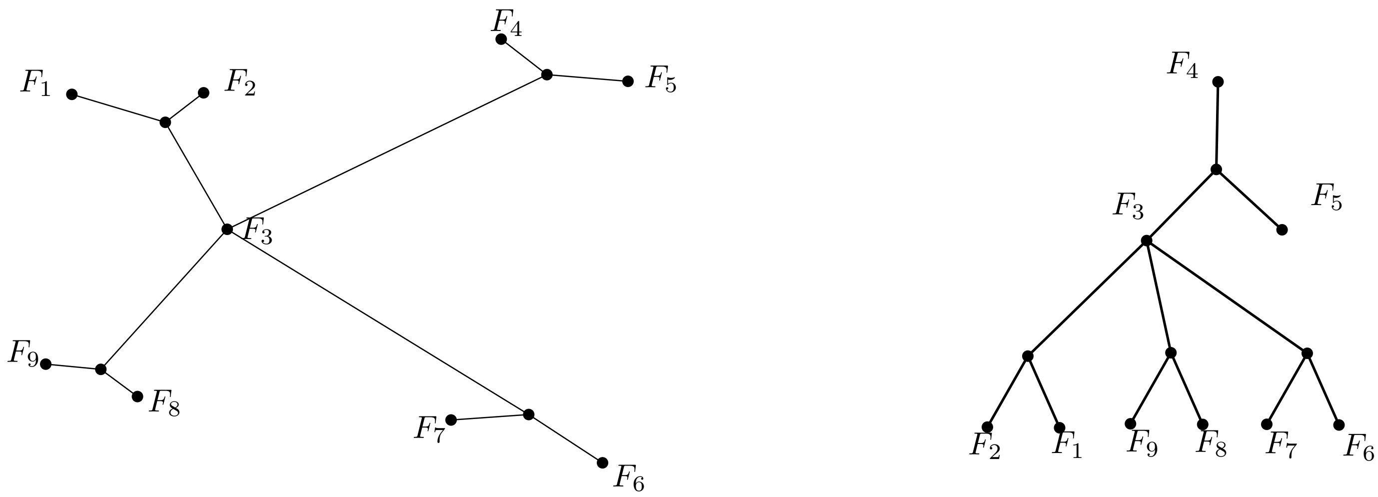

Figure 20. The combinatorial structure of tripod tree. We make it rooted by selecting a face to be the root, for example the one (F_4) containing edge v_0v_1. Note that we can easily distinguish parent and children without the full partition.

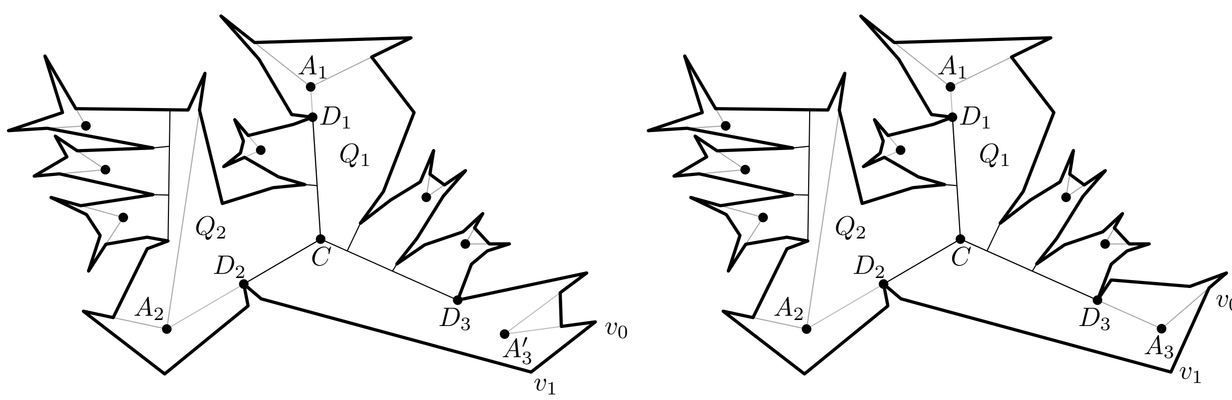

Figure 21. *Left:* a fake tripod defined by star-centers A_1, A_2 and supports D_1, D_2, D_3. Note that A_3' is not on the extension of CD_3. *Right:* a tripod and its associated fake tripod, since there exists the third star-center A_3 on the extension of CD_3. The fake tripod is the same as the one in the left figure.

Similarly as for tripods, the three segments D_1C, D_2C, D_3C are called the *legs* of $\mathcal{T}'$, A_1, A_2 are called *child star centers* of $\mathcal{T}'$, and the polygon bounded by the shortest paths between pairs of supports D_1, D_2, D_3 is called the *pseudo-triangle* of $\mathcal{T}'$.

Note that for every tripod $\mathcal{T}$, there is exactly one fake tripod $\mathcal{T}'$ with the same tripod point and legs, which is defined by its supports and the two child star centers of $\mathcal{T}$. We say $\mathcal{T}'$ is the *associated fake tripod* of $\mathcal{T}$.

REMARK 3.4. We introduce the concept of *fake* tripods (see also Figure 21) to facilitate our algorithm. The final algorithm will simulate the construction process (described in the following Section 3.2) and get a full partition in the end. We can not easily decide whether a (real) tripod exists unless we find both its two child star centers *and* its parent star center. Algorithmically, it is much easier to construct the *fake* tripods in a bottom-up fashion just using the two child star centers, without knowledge of where (or if) a potential third parent star-center might exist.

3.2 Construction Process

We now describe an iterative construction process of star centers and fake tripods. We will show that there exists an optimal star partition for which all star centers can be constructed using this process in linear steps. The construction process is with respect to a star partition $\mathcal{Q}$ and is a process to "mark" star centers and fake tripods of $\mathcal{Q}$ as "constructable". Formally, we call a star center or a fake tripod *constructable* (with respect to $\mathcal{Q}$) if it can be *marked* by the following process. At each step in the process, we can do one of the following operations:

- Mark a star center A_k at the intersection of two non-parallel segments of the following types:
  1. An edge of P;
  2. An edge of the pseudo-triangle of a marked fake tripod;
  3. An important sight line of a piece Q_k, which is on the extension of
     - an edge of P;
     - a diagonal of P that connects two concave corners of P;
     - a tripod leg of a tripod $\mathcal{T}$ whose extension contains the parent star center of $\mathcal{T}$, while the corresponding fake tripod $\mathcal{T}'$ of $\mathcal{T}$ is marked.
- Mark a fake tripod $\mathcal{T}'$ defined by two marked star centers A_i, A_j and three concave corners D_i, D_j, D_k of P. Additionally, there must be no star center (marked or unmarked) in the interior of the pseudo-triangle of $\mathcal{T}'$.

An optimal star partition $\mathcal{Q}$ is called *constructable* if all the star centers in $\mathcal{Q}$ is constructable with respect to $\mathcal{Q}$.

Now comes the major structural result in this section, which gives us a combinatorial way to describe some optimal star partition.

THEOREM 3.5 (Construction of optimal star partition). *There exists a constructable optimal star partition $\mathcal{Q}$.*

REMARK 3.6. We can also define a similar construction process if we can only mark tripod. In fact, the two definitions agree on whether a partition is constructable or not. When a fake tripod is used to mark a star center, it must be a tripod; otherwise, there is no need to mark that fake tripod.

This theorem also implies that the bit complexity of each star center is $O(n)$.

COROLLARY 3.7. *Each star center in a constructable optimal star partition can be encoded by a sequence of $O(n)$ corners of P, which specifies the process to mark it.*

REMARK 3.8. Using the same proof strategy, we can also prove $O(K)$ bits are enough to encode each star center in a constructable optimal star partition, where K is the total number of bits to encode the input polygon P.

PROOF. We will prove that we can encode each star center A_i by $4s(A_i)$ corners, where $s(A_i)$ is the size of the subtree in the fake tripod tree rooted at the face containing A_i, and encode each fake tripod point C by $4s(C)$ corners, where $s(C)$ is the size of the subtree in the fake tripod tree rooted at C.

The proof is by induction on the fake tripod tree from leaf node to root node. For each leaf node of the fake tripod tree, every star center in the corresponding face can only be marked by two lines that each of them are defined by two corners of P, since the face does not have any fake tripod point as its child. Now consider the internal nodes of the fake tripod tree. If it corresponds to a fake tripod $\mathcal{T}$, the tripod point C can be encoded by its two child star centers A_i, A_j together with two concave corners of P, so we need $s(A_i) + s(A_j) + 2 = 4s(C) - 2 \leq 4s(C)$ corners to encode C. If it corresponds to a face F, then for any star center A_k in F, it can be encoded by two lines, each of them is either defined by two corners of P or defined by a child fake tripod $\mathcal{T}$ of F. In all cases, the star center can be encoded by $4s(A_k)$ corners of P.

It remains to bound the size of the fake tripod tree. According to Chvátal's art gallery theorem, we can partition any polygon into at most $\lfloor n/3 \rfloor$ star-shape pieces. Any leaf face in the fake tripod tree contains at least one star center, so the fake tripod tree contains at most $\lfloor n/3 \rfloor$ leaves, therefore it has at most $2n/3$ nodes. ■

Instead of proving Theorem 3.5 directly, we prove the following stronger lemma, which allows us to extend a "partially constructable" optimal solution into a constructable one. This lemma also helps us to prove the correctness of the greedy choice (see Section 5.2.1) used when choosing tripods, which is the main technique to improve the running time of our dynamic programming algorithm into polynomial time in Section 5.

LEMMA 3.9. *Let Q be an optimal star partition of P such that some star centers $A_1, A_2, \ldots, A_k$ and some fake tripods $\mathcal{T}_1', \mathcal{T}_2', \ldots, \mathcal{T}_l'$ are constructable with respect to Q. Suppose there exists a star center in Q which is not constructable with respect to Q, then there exists an optimal solution Q' containing $A_1, A_2, \ldots, A_k, A_{k+1}$ as star centers and $\mathcal{T}_1', \mathcal{T}_2', \ldots, \mathcal{T}_l'$ as fake tripods, such that $A_1, A_2, \ldots, A_k, A_{k+1}$ and $\mathcal{T}_1', \mathcal{T}_2', \ldots, \mathcal{T}_l'$ are constructable.*

PROOF. A star center is called missing if it's not constructable with respect to Q. Consider the construction process of all the constructable star centers and fake tripods. Throughout the construction process we maintain what we call the *feasible region*, which initially consists of the whole polygon P. When a fake tripod with center C is marked by two marked star centers A_1, A_2 and three concave corners D_1, D_2, D_3, we partition the polygon into three parts (according to the legs of the fake tripod) and remove the pseudo-triangle of this fake tripod from the feasible region. Moreover, we add two segments A_1D_1, A_2D_2 as an incision into the boundary of both the polygon[3] and the feasible region. See Figure 22 for illustration. This way, we fix the two

3 Now the polygon becomes only weakly simple, even if it was a initially a simple polygon (see Section 2).

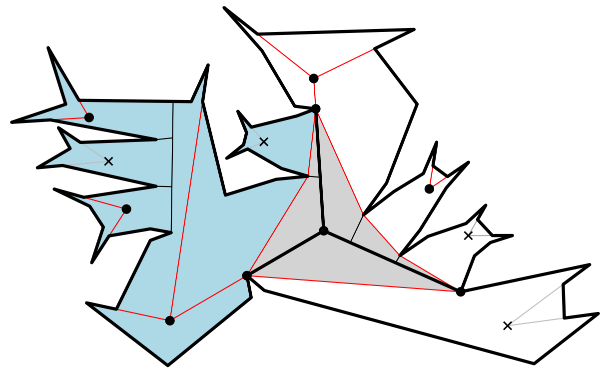

Figure 22. The figure shows the feasible region and the partition of the polygon where some star centers and fake tripods are marked. Thick black lines represent how we partition the polygon into three parts. Red segments mark the boundary of feasible region, and some of them are merely incisions. Blue region is a weakly connected components of the feasible region in the left part, with some red segments as incisions. The gray region marks the pseudo-triangle of the marked fake tripod, which is not a part of the feasible region, but the entire feasible region covers everything else. Each black circle represents a marked star center or a marked fake tripod, and each cross represents an unmarked star center.

important sight lines A_1C and A_2C, and make sure that no star centers are in the interior of the pseudo-triangle of this fake tripod in the following steps. When a star center A_i is marked by an important sight line ℓ of Q_i, we also add ℓ as an incision into the boundary of both the polygon and the feasible region.

Since the constructable fake tripods partition the polygon into disconnected pieces, we can consider the construction process within each connected part independently. Let P' be such a connected part for which at least one star center is not constructable. If there are multiple choices, we will choose one later. Let F be the feasible region inside P'. We now restrict our polygon to be P'.

We perform a restricted coordinate maximization along an arbitrary direction d as described in Lemma 2.4 within F and with all the constructable star centers fixed. By optimality, for each missing star center, the partition is also restricted coordinate-maximal along d, within F and with all the other star centers fixed. Applying Lemma 3.2, we know that all the missing star centers lie on the intersection of two non-parallel segments of certain types. We call these segments the *crucial segments*. The goal of the following discussion is to mark a new star center using crucial segments. Therefore, we enumerate the types of crucial segments and check whether they are allowed in the construction process.

Case 1: A crucial segment is an edge of F. Every edge of F is an edge of P, or a segment on the boundary of a pseudo-triangle, or an important sight line that was used to construct a constructable star center. Note that all segments on the boundary of a pseudo-triangle must be diagonals that connect two concave corners of P. Hence, all the edges of F can be used to mark new star centers.

Case 2: A crucial segment is an important sight line in P'. Note that all corners of P' are either corners of P, or constructable star centers, or tripod points of a constructable fake tripod. Each tripod point is a convex corner in any connected part separated by the tripod legs, hence they can only induce convex corners of P'. Consequently, a concave corner of P' is either a concave corner of P or a constructable star center. According to Lemma 3.2, no other star center lies on a crucial segment, so the crucial segment that the star center lies on must be supported by a concave corner of P, which implies that the crucial segment is also an important sight line when we consider the full polygon P. If the crucial segment ends at a concave corner of P', it must end at a concave corner of P, as crucial segments are not allowed to contain another star center. Hence, it is contained in an extension of a diagonal of P that connects two concave corners of P.

Thus, a crucial segment cannot be used to mark star centers only if it is an extension of a tripod leg, and either the corresponding fake tripod is not constructable, or the star center is not in the parent face of this tripod.

We now consider the tripod tree of Q. We call a tripod *illegal* if there exists a missing star center in its child face. Otherwise, we call a tripod *legal*. Then, a star center is missing only if one of its crucial segments end at the tripod point of an illegal tripod. As there exists a missing star center, there also exists an illegal tripod.

Let $\mathcal{T}$ be an illegal tripod such that all tripods contained in the subtree rooted at $\mathcal{T}$ are legal. Note that there exists a missing star center in at least one of its child faces. Hence, all star centers in the two child faces of $\mathcal{T}$, except for the child star centers of $\mathcal{T}$, are constructable. Let A be a missing child star center of $\mathcal{T}$. We choose P' to be the connected component containing A.

We perform a restricted coordinate maximization on A, where the polygon we are going to partition is the part of P' cut by $\mathcal{T}$ that A lies in, and the feasible region is F excluding the pseudo-triangle of $\mathcal{T}$ restricted to the current polygon. We choose d to be the direction perpendicular to the important sight line that A ends at the tripod point of $\mathcal{T}$ and points to the unsupported side. The choice of direction d makes it impossible that the new star center A' lies on the same important sight lines from $\mathcal{T}$, unless it is also on other two non-parallel crucial segments.

By a similar analysis of the types of crucial segments that the new star center A' lies on, A' cannot be marked in the next step only if it is a child star center of an illegal tripod contained in P', and we resolve this case recursively; see Figure 23. Since the subpolygon cut by the new illegal tripod has strictly fewer corners, this recursion will finish in finite steps. Eventually, we can find an optimal partition in which one more star center A' is constructable by the intersection of two non-parallel crucial segments. ■

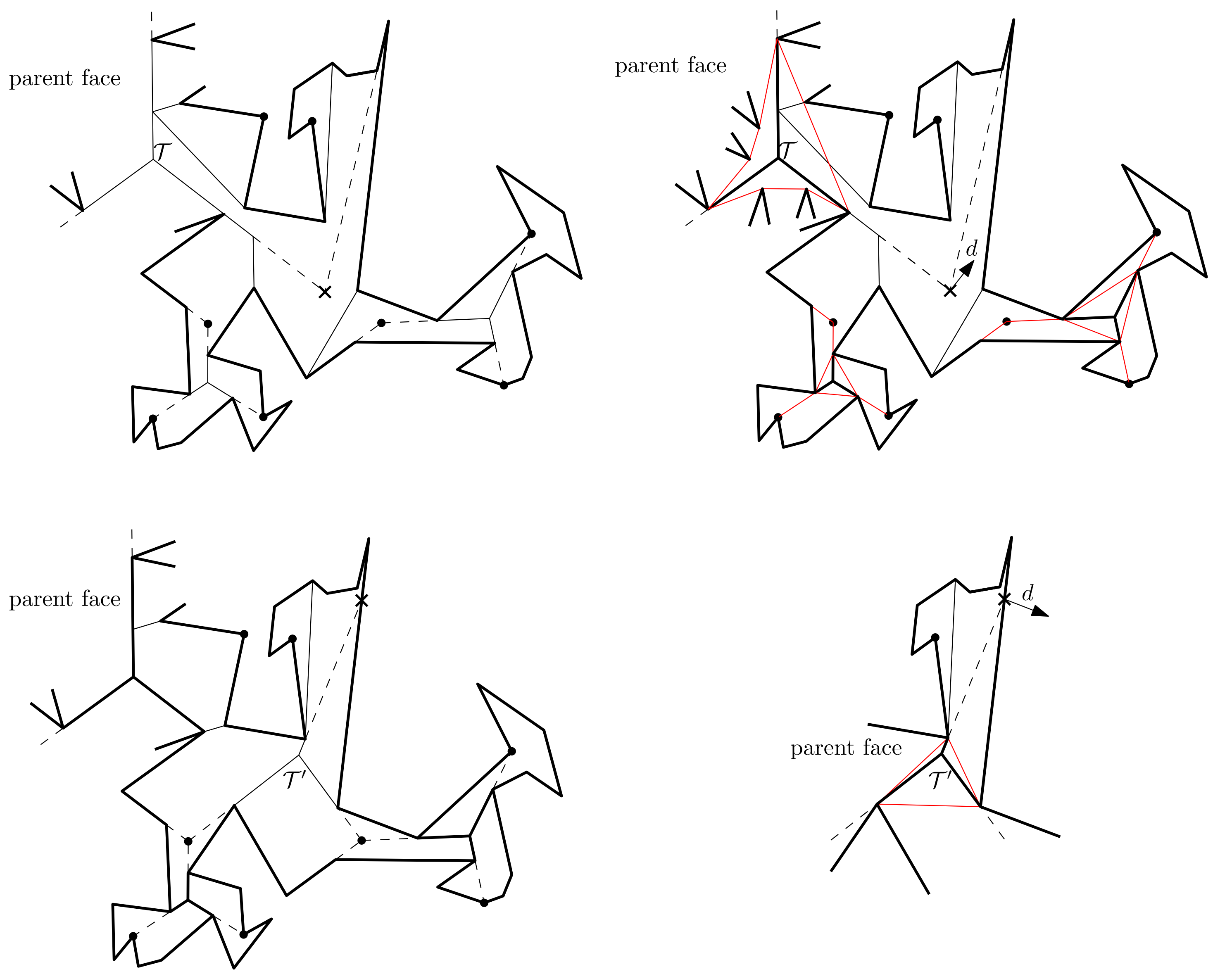

Figure 23. Illustration for the proof of Theorem 3.5. A constructable star center is represented by a solid circle, and the missing star center is represented by a cross. Red segments mark the boundary of the feasible region, and thick black segments mark how we partition the polygon into parts. The top-left figure shows the partition $\mathcal{Q}$, and the top-right figure shows the restricted coordinate maximization problem used in the proof. The bottom-left figure shows the restricted coordinate maximization partition of this problem. The old illegal tripod $\mathcal{T}$ is no longer a tripod, but we get a new illegal tripod $\mathcal{T}'$, which will be resolved recursively. The bottom-right figure shows this new subproblem that we get. Since the number of vertices on the boundary are strictly fewer, the recursion will eventually terminate.

The following lemma gives us another useful property of constructable optimal star partition, which will be used to design our dynamic programming algorithm in Section 5.

LEMMA 3.10. *Let $\mathcal{Q}$ be a star partition of P. If a star center A_k is constructable with respect to $\mathcal{Q}$, then a corner of P appears on the boundary of the piece Q_k.*

PROOF. We consider the different cases in the construction process which can lead to the star center A_k being marked. If A_k is marked by an important sight line ℓ of Q_k, then the support D' of ℓ is a corner of P in Q_k, and the lemma holds. If A_k is at the intersection of two edges of P, A_k must be a corner of P. If A_k is at the intersection of an edge of P and an edge of a pseudo-triangle,

A_k must be a corner of P, as the intersection of the pseudo-triangle and the boundary of P is just a set of concave corners of P. If A_k is at the intersection of two non-parallel edges from the same pseudo-triangle, A_k must be a corner of this pseudo-triangle, and it therefore is also a concave corner of P. If A_k is at the intersection of two non-parallel edges from different pseudo-triangles, according to Lemma 2.2, A_k must be a corner of one pseudo-triangle, therefore A_k is also a corner of P. ∎

From this lemma, we directly have the following corollary, which will be used to prove some combinatorial properties of an optimal partition in Section 5.

COROLLARY 3.11. *For any constructable optimal star partition $\mathcal{Q} = Q_1, \dots, Q_k$, we have $Q_i \cap \partial P \neq \emptyset$, that is, every star-shaped piece touches the boundary of P.*

4. Properties of Area Maximum Partitions

The objective of this section is to compute a set of polynomially many points that contains all the Steiner points for some optimal solutions, given all the star centers of an arbitrary optimal solution. We will work on a constructable optimal star partition, with all the construction lines fixed as incisions, and analyze the position of the Steiner points in each connected components independently, as we did in the proof of Theorem 3.5. Recall Lemma 3.10, all the star-shaped pieces still touch the outer boundary of the input polygon P.

Consider a weakly simple polygon P' with potentially some incisions, a sequence of points $\mathcal{A} = (A_1, \dots, A_k)$, and a star partition $\mathcal{Q} = (Q_1, \dots, Q_k)$ of the interior of P' where A_j is a star center of Q_j. Recall that $\mathcal{Q}$ is called *area maximum* with respect to $\mathcal{A}$ if the vector of areas $a(\mathcal{Q}) = \langle a(Q_1), \dots, a(Q_k) \rangle$ is maximum in lexicographic order among all partitions of P with star centers $\mathcal{A}$. In Appendix A we argue that this notion of area maximum partition is well-defined. Note that in the definition of area maximum partitions, the positions of the star centers are fixed. In this section, we prove some properties of area maximum partitions, in particular that all corners of pieces (i.e. Steiner points) are in “nice” spots.

Recall that a *sight line* of a piece Q_i is a segment of the form $r = A_i C$, where A_i is the star center of Q_i and C is a corner of Q_i. Here, the point C is called the *end* of r.

LEMMA 4.1. *Consider an optimal area maximum partition $\mathcal{Q}$ with a given set of star centers $\mathcal{A}$. For any two pieces $Q_1, Q_2 \in \mathcal{Q}$, let $\gamma := \partial Q_1 \cap \partial Q_2$ be their shared boundary. Then γ is*

- *empty, or*
- *a single point, or*
- *a single line segment contained in a sight line of Q_1 or Q_2, or*
- *two adjacent line segments, each contained in a sight line of Q_1 or Q_2.*

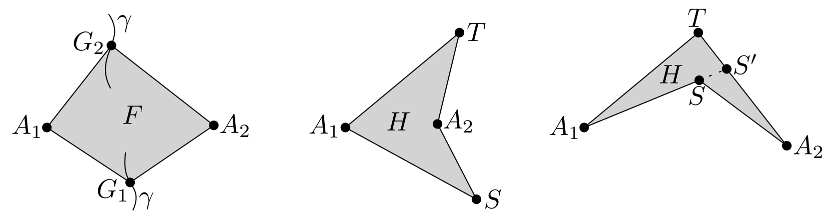

Figure 24. Situations in the proof of Lemma 4.1. *Left:* If γ is not connected, F must contain a third piece which can be subsumed by Q_1 and Q_2, which is a contradiction. *Middle:* When S and T are convex, we have $\gamma = SA_2 \cup A_2T$. *Right:* When S is concave, we have $\gamma = SS' \cup S'T$.

PROOF. We first prove that γ is connected. Otherwise, there exist points $G_1, G_2 \in \gamma$ which are not connected by γ; see Figure 24 (left). Let $A_i \in \mathcal{A}$ be the star center of Q_i. Since G_1 and G_2 are not connected by γ, the quadrilateral $F = A_1G_1A_2G_2$ is not contained in $Q_1 \cup Q_2$, but the boundary ∂F is contained. Hence, F contains a third piece from $\mathcal{Q}$. However, the quadrilateral F can be completely assigned to Q_1 and Q_2 (possibly after being split in two), so the partition $\mathcal{Q}$ was not optimal, which is a contradiction. Hence, γ is connected.

Let S and T be the endpoints of γ. If $S = T$, the shared boundary is a single point which must be a corner of at least one of the two pieces Q_1, Q_2, since otherwise their shared boundary would be longer.

Otherwise, let us traverse γ from S to T; see Figure 24 (middle and right). Since Q_1 and Q_2 are star-shaped, we move around A_1 in a monotone way, either clockwise or counterclockwise, and we move in the opposite direction around A_2. Hence, γ is contained in the quadrilateral $H = A_1SA_2T$. Assume without loss of generality that A_1 had higher priority than A_2 when we maximized the areas of the pieces in $\mathcal{Q}$.

If S and T are both convex corners of H, then all of H can be seen from A_1, so we can assign the quadrilateral H to Q_1. Then γ is a continuous part of $SA_2 \cup A_2T$ with $A_2 \in \gamma$, so γ is contained in two sight lines of Q_2.

Otherwise, assume without loss of generality that S is concave and T is convex. Let S' be the intersection between the line containing the segment A_1S and the segment A_2T. Then we maximize the area of Q_1 by assigning the triangle $A_1S'T$ to Q_1 and the rest of H (which is the triangle A_2SS') to Q_2. Hence, we have that γ is a continuous part of $SS' \cup S'T$ with $S' \in \gamma$, so γ is contained in sight lines of Q_1 and Q_2, respectively. Note that it may happen that $S' = T$, so that one of these segments is degenerate. ■

We say that a point B is *supporting* a sight line $r = AC$ from a star center A if $B \in r \setminus \{A\}$. The following lemma characterizes all edges of pieces of an area maximum partition, using Lemma 4.1.

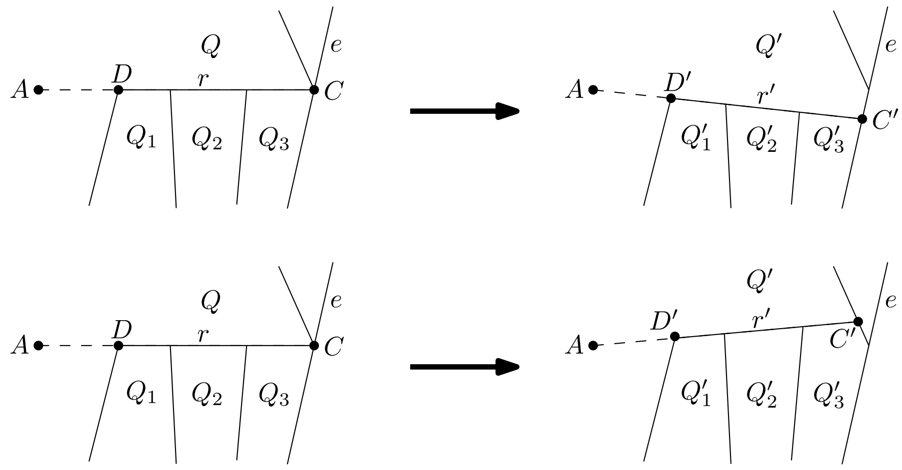

Figure 25. Situations in the proof of Lemma 4.2. In Case 1.1 or Case 1.2, we can increase the area of Q or Q_1, Q_2, Q_3 by rotating r clockwise or counterclockwise, respectively.

LEMMA 4.2. *Consider a piece Q with star center A of an optimal area maximum partition. Let $r = AC$ be a sight line of Q which contains an edge on the boundary of Q. Then r is of one of the following types:*

(i) r is supported by a corner of P', which is not a star center, or

(ii) r is supported by a star center of another piece, or

(iii) C is the end of two non-parallel sight lines of type (i) in other pieces.

PROOF. Suppose that a sight line $r = AC$ is supported by no corner of P' and no star center of another piece. We will prove that the end C must be the end of two sight lines of type (i) of other pieces. We consider multiple cases.

Case 0: C is a corner of P'. In this case, r is supported by C and of type (i) or (ii).

Case 1: C is an interior point of another sight line or an edge of P'. Let us denote this sight line or edge by e; see Figure 25. Let $f = DC$ be the edge of Q contained in r. Assume without loss of generality that r is horizontal with the end C to the right and that the interior of Q is above f. Then some other pieces $Q_1, \ldots, Q_i$ are below f. Recall that in an area maximum partition, we maximize the vector of areas in a specific lexicographic order; in other words, each piece has a distinct priority when maximizing the areas. In both of the following cases we obtain a contradiction.

Case 1.1: Q has higher priority than all of $Q_1, \ldots, Q_i$. In this case, we can expand Q a bit by rotating r a bit clockwise around A, thus "stealing" some area from the pieces $Q_1, \ldots, Q_i$ and increasing the area vector with respect to the lexicographic order.

Case 1.2: One of the pieces $Q_1, \ldots, Q_i$ has higher priority than Q. In this case, we can rotate r a bit counterclockwise, thus expanding the pieces $Q_1, \ldots, Q_i$ and increasing the area vector. We conclude that C is not an interior point of another sight line or an edge of P.

Note that if r does not fall into case 0 or case 1, C must be in the interior of P'.

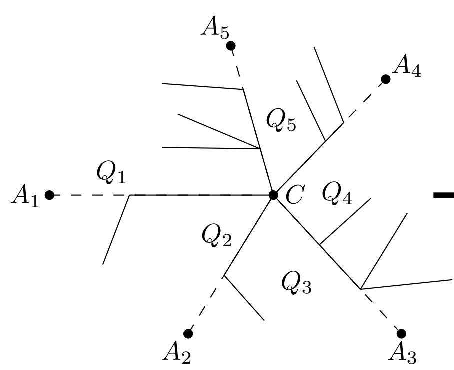

Figure 26. Case 2.1: Illustration of moving point C when none of r_i is supported by a corner or a star center. While moving C, we fix the lines containing the boundary segments that touch $r_1, \ldots, r_j$, and slide the intersection points accordingly. If more than one boundary touches r_i at the same point, we might only extend the one closest to C. Since all the new corners we create are convex, all pieces remain star shaped. We only need to move C infinitesimally to obtain a contradiction, so no new crossings will be formed during this process.

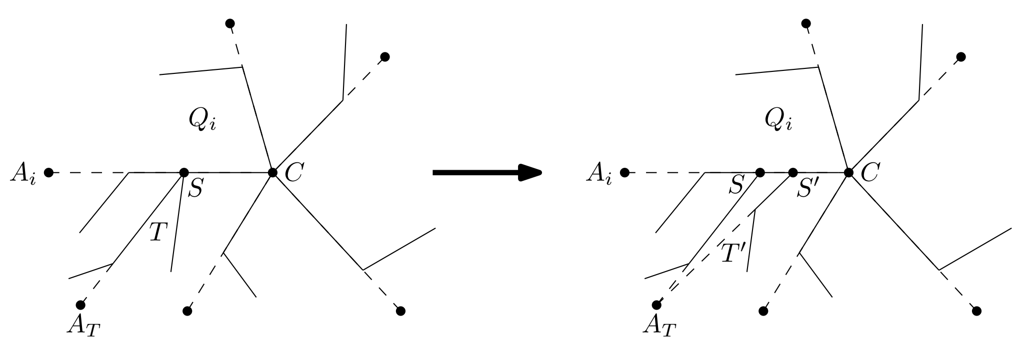

Figure 27. Case 2.1: If T is the highest priority piece and $T \cup r_i = S$ is a single point, then one side of visibility from its star center A_T to S is not blocked, therefore, T take a sufficiently small triangle near S.

Case 2: C is the end of one or more sight lines of other pieces. First, note that according to Lemma 4.1, if C is on the boundary of a piece, then C is also contained in a sight line of that piece. Let the sight lines that share the end C be $r_1, \ldots, r_j$ in counterclockwise order (one of these is r), and let the associated set of pieces and star centers be $\mathcal{Q} = \{Q_1, \ldots, Q_j\}$ and $A_1, \ldots, A_j$, respectively. We first observe that we must have $j \geq 3$: Clearly $j \geq 2$, so consider the case $j = 2$. If the two sight lines r_1 and r_2 are not parallel, then C is a concave corner of one of them that causes the piece to not be star-shaped. If the two sight lines are parallel, then C is not a corner of the pieces, so r_1 and r_2 are not (complete) sight lines of the pieces. Hence, $j \geq 3$.

Assume without loss of generality that the interior of each Q_i is to the left of r_i. This has the consequence that if r_i is supported by a corner of P, then the two incident edges are to the right of r_i and likewise, if r_i is supported by a star center, then the interior of the associated piece is also to the right. Suppose towards a contradiction that at most one of the sight lines $r_1, \ldots, r_j$ is supported by a corner of P that is not also a star center—note that otherwise r is a sight line of type (iii). Let $\mathcal{R}$ be the set of pieces R for which $R \notin \mathcal{Q}$ but $\partial R \cap r_i \neq \emptyset$ for some

$i \in \{1, \ldots, j\}$. The goal is to improve the priority of the area vector by exchanging area between pieces in $\mathcal{Q} \cup \mathcal{R}$, which leads to a contradiction.

Case 2.1: None of $r_1, \ldots, r_j$ is supported by a corner or a star center. We will show that it is always possible to move C anywhere within a sufficiently small disk; see Figure 26. We attach all sight lines $r_i = A_iC$ to the flexible point C. For each boundary segment s touching one of r_i, we fix it on the same straight line, extending or contracting with respect to the movement of C, so the intersection point of s and r_i is flexible. If there are multiple segments touching r_i at the same point, we will extend only the one closest to C if necessary. Since C is a convex corner in all of $Q_1, \ldots, Q_j$, and all new corners are formed by the intersection of some ray and some straight line, all pieces remain in star shape with respect to their initial star centers.

Now we will show that we can always improve the priority of the area vector.

- If Q_i has the highest priority among $\mathcal{Q} \cup \mathcal{R}$, we can slide C along the ray A_iC and expand Q_i.
- Otherwise, a piece $T \in \mathcal{R}$ has the highest priority among $\mathcal{Q} \cup \mathcal{R}$. Without loss of generality, assume T is touching and to the right of r_i. If T only touches r_i at a single point S, then we can expand T around S. See Figure 27. If T has a boundary segment along r_i, we can move C to the left of the original ray A_iC and expand T.

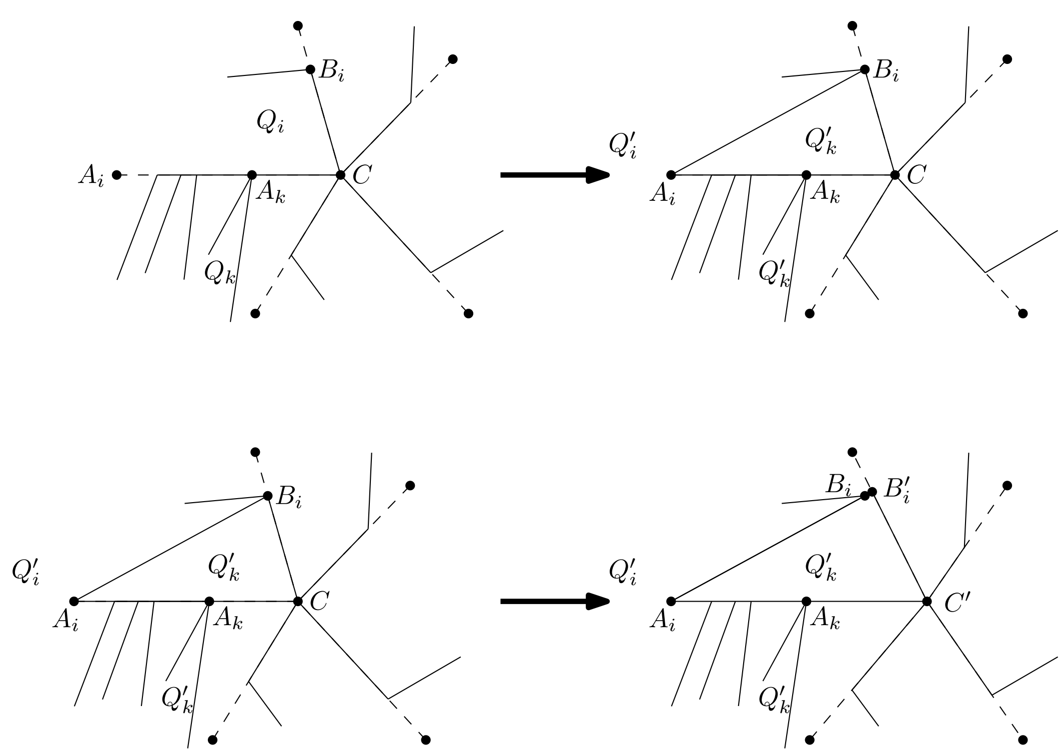

Figure 28. The top two figures illustrate the reduction from case 2.2 to case 2.1. By redistributing a triangle between pieces, the role of $r_i = A_iC$ is replaced by $r'_i = A_kC$, which is no longer a sight line of type (ii). The bottom two figures show how to expand Q_i if Q_i has the highest priority. We can slide C along the ray A_iC to C' and give the triangle $A_iC'B'_i$ back to Q_i afterward.

Case 2.2: None of $r_1, \dots, r_j$ is of type (i), but can be of type (ii). Compared with case 2.1, each sight line r_i might be supported by star centers. We will show that it's always possible to move C infinitesimally so as to increase the area vector. See Figure 28 top. We will make some local modifications along each sight line of type (ii) and reduce to the previous case. Consider any sight line r_i of type (ii). Let A_k be the farthest star center from A_i that supports r_i, B_iC be an edge of Q_i along r_{i-1}. We will give the triangle A_iCB_i from Q_i to Q_k. After this exchange, the role of r_i is replaced by a new sight line $r_i' = A_kC$, which is not supported by any star centers. After applying this modification along all the sight lines of type (ii), we reduce to case 2.1, therefore, we can move C anywhere within a sufficiently small disk.

Now we will show that we can always improve the priority of the area vector in the end. Without loss of generality, assume r_i is the initial sight line that touches the highest priority piece, and this piece is either Q_i or a piece $T \in \mathcal{R}$ to the right of r_i. If r_i is not supported by any star centers, we can make the same modification as in case 2.1. Therefore, we will only consider the case that r_i is of type (ii). Similarly, let A_k be the farthest star center from A_i that supports r_i.

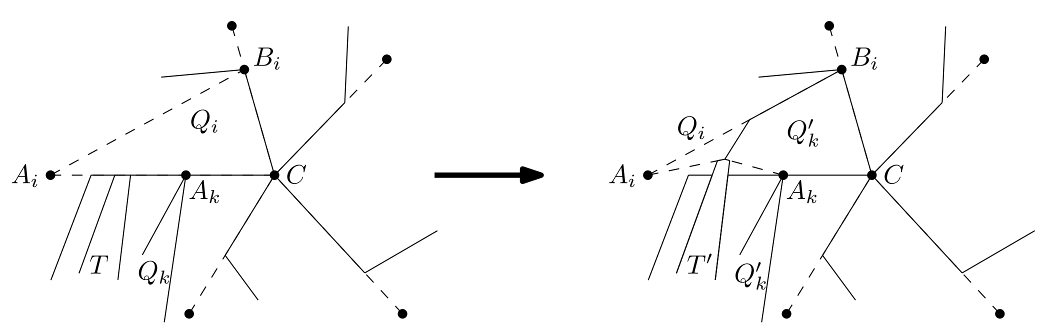

Figure 29. Case 2.2: When T has the highest priority, and T touches A_iA_k, we can give an sufficiently small pentagon to T.

— If Q_i has the highest priority, we can slide C along the ray A_iC. Let B_iC be the edge of Q_i along r_{i+1}. After sliding C along the ray A_iC to C', B_i slides to B_i', we can give the triangle $A_iC'B_i'$ back to Q_i and improve the priority of the area vector. See Figure 28 bottom.

— If some piece $T \in \mathcal{R}$ touching r_i has the highest priority. If it touches A_kC, then after transferring the triangle A_iCB_i from Q_i to Q_k, it reduces to case 2.1. Therefore, we only need to consider the case when T touches A_iA_k. If $\partial T \cap A_iA_k$ is a single point, we can apply the same modification as case 2.1. See Figure 27. If $\partial T \cap A_iA_k$ is a segment, we can extend the two boundary segments end in A_iA_k into the triangle A_iCB_i, give a sufficiently small pentagon T, and repartition the triangle A_iCB_i accordingly. See Figure 29.

Case 2.3: Some r_l is of type (i). Let D be the corner of P that supports r_l, which is not a star center. Without loss of generality, assume r_l is horizontal with C to the right. In this case, we might not be able to move C to the right of A_lD, as D blocks the visibility from A_l to C. But

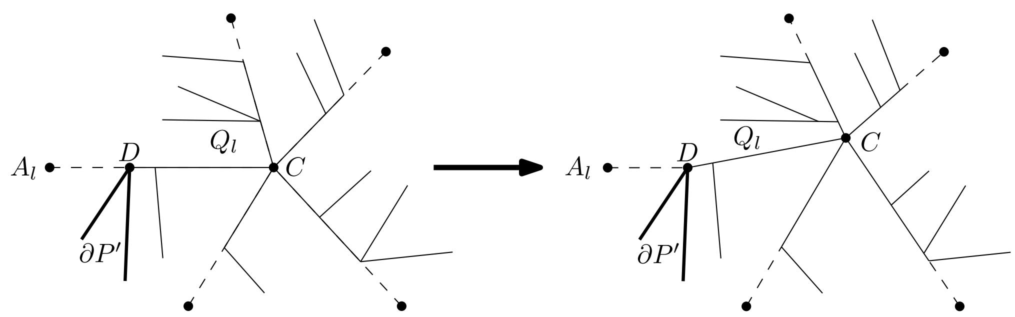

Figure 30. Case 2.3: When $r_l = A_lC$ is supported by a non-center-corner D, we treat the segment DC as r_l, and make the all the intersection points with DC flexible, as in case 2.1. If there are star centers along DC or any other r_i, we treat it in the same way as in case 2.2. This modification keeps all the pieces star shaped as long as C is on or to the left of ray A_lD and within a sufficiently small disk.

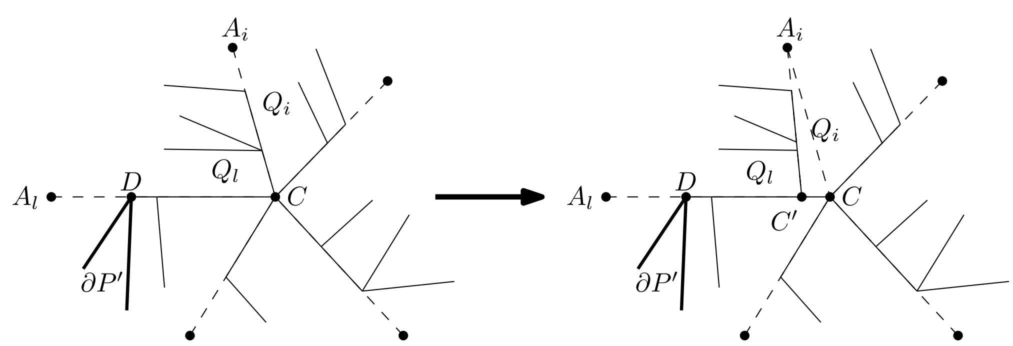

Figure 31. Case 2.3: If there is no star centers along $r_i = A_iC$, we can give the triangle A_iCC' to Q_i.

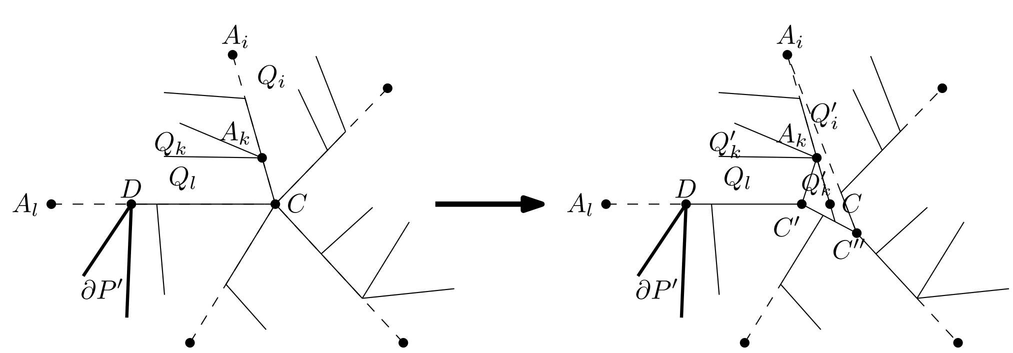

Figure 32. Case 2.3: If there are some star centers along $r_i = A_iC$, let A_k be the farthest from A_i. Since C is a convex corner at all pieces, there must be a r_m to the left of r_i. Then we can take a sufficiently close point C'' along r_m to C, and redistribute the quadrilateral $A_iA_kC'C''$ to Q_i and Q_k.

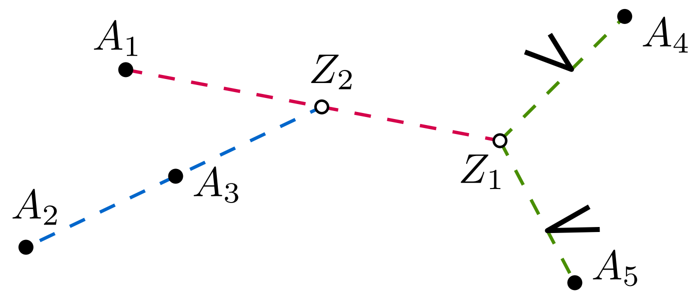

Figure 33. Examples of the different sight line types of Lemma 4.2 and different Steiner points these can give rise to, as characterized by Corollary 4.3. The sight lines A_4Z_1 and A_5Z_1 are of type (i) (supported by a corner), meeting in point Z_1. The sight line A_2Z_2 is of type (ii) (supported by star center A_3). The sight line A_1Z_1 is of type (iii) (ending in the common endpoint of two sight lines of type (i)). We note that Z_2 here is a Steiner point arising from a sight line of type (ii) ending on a sight line of type (iii). We will later on show the same diagram for the entire partition of the input polygon. There are five star centers and two corners involved in the definition of Steiner point Z_2, which turns out to be the worst case. In our algorithm, we will have $O(n^6)$ potential star centers, making for a total of $O(n^{32})$ potential such Steiner points.

we can still move C to anywhere within a sufficiently small disk while keeping on or to the left of A_lD. See Figure 30.

In this case, we will fix all the pieces in $\mathcal{R}$ touching A_lD, and extend the highest priority among the others. Let $\mathcal{R}'$ be the set of pieces touching $\{r_1, \ldots, r_j, CD\} \setminus \{r_l\}$ that is not in Q. As before, assume r_i is the sight line touching the highest priority piece.

— If there is a horizontal $r_k = A_kC$ with A_k to the right of C, then we can reduce to case 1 by taking $r_k \cup r_l$ as e and consider any other sight line ends at C;
— If $T \in \mathcal{R}'$ has the highest priority, we can apply the modification in case 2.2 and expand T.
— Otherwise, some Q_i has the highest priority. If A_i is on or below the straight line A_lD, we can slide C along the ray A_iC and expand Q_i. Let us assume A_i is above the straight line A_lD. If $r_i = A_iC$ is not supported by any star centers, then we can take a sufficiently close point C' to C on CD, and give the triangle A_iCC' to Q_i. See Figure 31. If r_i is supported by a star center, let A_k be the farthest star center that supports r_i from A_i, then Q_i and Q_k can collectively take some region around A_kC. See Figure 32. ■

See Figure 33 for an example of the different types of sight lines and Steiner points (i.e. corners of the star pieces) needed in the partition. The following corollary, which characterizes all required Steiner points, is immediate from Lemmas 4.1 and 4.2; indeed, each Steiner point must be on the beginning or end of a sight line. Together with Corollary 3.7, we can bound the complexity to encode each Steiner point.

COROLLARY 4.3. *In an area maximum partition of P', each Steiner point is of one of the following types:*

1. *The end of a sight line of type (i) or (ii) on an edge of P',*
2. *The end of a sight line of type (i) or (ii) on a sight line of type (i)–(iii),*
3. *The common end of two sight lines of type (i),*
4. *A star center.*

And each Steiner point can be encoded by a sequence of $O(n)$ corners of P'.

Combined with the result of Section 3, we can categorize all sight lines and Steiner points in some optimal constructable star partition.

LEMMA 4.4. *There exists an optimal constructable partition of any simple polygon P, such that each sight line $r = AC$ containing a segment on the shared boundary is of one of the following types:*

(a) r is supported by a corner of P;
(b) r is supported by another star center;
(c) C is the end of two non-parallel lines of type (a).

And each Steiner point is of one of the following types:

(a) The end of a sight line of type (a) or (b) on an edge of P,
(b) The end of a sight line of type (a) or (b) on a sight line of type (a), (b) or (c),
(c) The common end of two sight lines of type (a).
(d) A star center.

And each Steiner point can be encoded by a sequence of $O(n)$ corners of P.

PROOF. Consider each connected components by adding all the construction lines as incisions. Within each component, we take the area maximum partition.

We will first map the corners and edges in each connected component back to objects defined by P and the star centers. Note that in each connected component, all corners are either star centers, or tripod points, or a corner of the initial polygon P. Each edge in the connected components, is either an edge of P, or a part of a construction segment. Note that each construction segment is a sight line supported by a concave corner of P, which is of type (a).

Next we will map all the sight lines in each connected component back to objects defined by P and the star centers. Here we will map them back type by type according to Lemma 4.2.

Type (i) sight lines. All non-star-center corners of each connected component P', must be either corners of P or tripod points. Since tripod points are convex corners in any connected components, it can not support any sight line from the interior. Therefore, any sight line of type (i) must be supported by a corner of P, which is of type (a).

Type (ii) sight lines. They are still supported by star centers, so all type (ii) sight lines inside a connected component P' are of type (b).

Type (iii) sight lines. Since all type (i) sight lines are of type (a), C must be the end of two sight lines of type (a), so all type (iii) sight lines inside a connected component are of type (c).

Next we will map all the Steiner points in each connected components to objects defined by P and star centers. Here we will map back class by class according to Corollary 4.3.

Corners of a connected component. They could be Steiner points in the partition of P as well. Note that in each connected component, all corners are either star centers (class (d)), or tripod points (class (c)), or a corner of the initial polygon P (not a Steiner point), so they all fit into one of the classes.

Steiner points in class 1. Since all type (i) sight lines in each connected component are of type (a), all type (ii) sight lines in each connected component are of type (b), every Steiner point C in class 1 is the end of a sight line of type (a) or (b). Each edge in each connected component is either an edge of P, or a part of a sight line of type (a), it falls into class (a) or (b).

Steiner points in class 2. Since the types of sight lines match, all Steiner points in class 2 are in class (b).

Steiner points in class 3. Since the types of sight lines match, all Steiner points in class 3 are in class (c).

Steiner points in class 4. They are in class (d). ■

5. Algorithm

In this section we present our polynomial time algorithm to find a minimum star partition of a polygon. We restate our main Theorem 1.1 below, that we prove in this section.

THEOREM 1.1. (Restated) *There is an algorithm performing $O(n^{105})$ arithmetic operations that partitions a simple polygon with n corners into a minimum number of star-shaped pieces. The number of bits used to represent each Steiner point in the constructed solution is $O(K)$ where K is the total number of bits used to represent the corners of P.*

REMARK 5.1. Although it is polynomial time, it is not exactly efficient. Since our main result is that the problem is in P (while previously it was not clear whether the problem was even in NP), we have not tried optimizing the running time. We believe that it should not be particularly difficult to significantly improve the exponent something like ≈ 50 by a more refined analysis. For instance, using a smaller set of potential Steiner points would lead to a smaller state-space of the dynamic program (see Appendix B). Our aim here is to give the simplest possible description of an algorithm with polynomial running time. Our techniques alone might not be sufficient to bring down the exponent to, say, a single digit. We leave it as an open question to optimize the running time as far as possible, or conversely provide fine-grained lower bounds.

Overview. We begin with a brief overview of our algorithm (see also the technical overview in Section 1.2). There are two main challenges to overcome when designing a minimum star partition algorithm:

- First, even if we are given a set of potential Steiner points, it is not clear how to construct an optimal star-partition.
- Second, we need a way to find these potential Steiner points.

For the first challenge, we devise a dynamic programming algorithm. For the second, we rely heavily on our structural results in Section 3 together with a "greedy choice" lemma. In fact, in order to find the potential Steiner-points, we need to invoke the dynamic programming algorithm (which assumes that we know all the potential Steiner points already) on many smaller instances in a recursive fashion.

Dynamic program. We begin by assuming that we know a set $S^{(\text{CENTERS})}$ of potential star centers. In Section 5.1 we show a dynamic programming algorithm to find a partition of the polygon into a minimum number of star-shaped pieces *such that the star center of each piece is in* $S^{(\text{CENTERS})}$. The algorithm runs in $O(\text{poly}(n, |S^{(\text{CENTERS})}|))$ time. There are a few key properties that we show that allow us to define this dynamic programming algorithm (details can be found in Section 5.1):

- We show that using $S^{(\text{CENTERS})}$ and the corners of P, we can find a set of all potential Steiner points (e.g. internal corners of the star pieces). We do this by invoking our structural lemmas about *area maximum partitions* from Section 4. There will only be $O(\text{poly}(n, |S^{(\text{CENTERS})}|))$ many of these potential Steiner points to consider.
- We argue that each star piece touches the boundary in some optimal partition (Corollary 3.11).
- The above observation allows us to define a set of natural separators (see also Figure 34) involving at most two star pieces. For points B_1, B_2 on the boundary of P, star centers $A_1, A_2 \in S^{(\text{CENTERS})}$ and a potential internal corner Z on the shared boundary of the two star pieces, we can define a ("long") separator B_1-A_1-Z-A_2-B_2. We also consider ("short") separators of the form B_1-A_1-B_2. These separators allow us to define a sub-region P' of P on one side of the separator, that we can recursively solve using a dynamic programming approach.

Finding potential star centers. Given the above mentioned dynamic programming algorithm, the ultimate challenge is finding some relatively small (i.e. polynomial-sized) set of points $S^{(\text{CENTERS})}$ such that some optimal solution only uses star centers from $S^{(\text{CENTERS})}$. However, this turns out to be quite challenging and we present how we overcome this, together with the full algorithm, in Section 5.2.

A first attempt might be to consider $S^{(\text{CENTERS})}$ to be all the $O(n^4)$ points on the intersections of pairs of diagonals of the polygon. This turns out to not be sufficient, as can be seen in Figure 1.

Indeed, the same figure shows that the star center points can have *degree* as high as $\Omega(n)$ (in particular, the position of some star centers depend on up to $\Omega(n)$ corners of the input polygon).

Instead, here we use our crucial structural properties of optimal star partitions proven in Section 3. Essentially, we show there that the only non-trivial structure in some extreme optimal partitions must be *tripods*(see Section 2), e.g., like those in Figures 1, 4 and 6. The tripods must be supported by three corners of the polygon, so there are only $O(n^3)$ such choices where a tripod can appear. However, the location of the tripod point might depend on other tripods (again, see Figure 1 for a recursive construction capturing this). To overcome this, we need a *greedy choice* property that allows us to argue that, for each potential tripod, there is only a single arrangement of this tripod we need to care about: the one that is least restrictive for one of the involved star centers.

To find this greedy arrangement of the tripod, we need to solve the minimum star partition problem on a subregion of the polygon. For this we can recursively call our algorithm to construct potential star centers for this smaller instance, and then use the dynamic programming algorithm to find the optimal star-partition.

In Section 3, we argue that the tripods of some optimal solution are all oriented in a consistent way. Indeed, recall that each tripod is constructed by its two child star centers and used to construct its parent star center, so only one of the three subpolygons fenced off by the tripod depends on the other two subpolygons. This consistent orientation means that all the tripods can be oriented towards some arbitrary root face (see Figure 5). This is crucial for our algorithm since this allows us to bound the number of subproblems to $O(n^2)$ (each diagonal of P will define a subproblem on the side not containing this root face, that can be solved first and must not depend on the other side).

5.1 Dynamic Program

In this section we prove the following theorem.

THEOREM 5.2. *Suppose we are given a polygon P with n corners. Suppose also that we know some set $S^{(\text{CENTERS})}$ of potential star centers, such that we are guaranteed that there exists an optimal partition of P into the minimum number of weakly simple star-shaped pieces where: (i) each star piece's center is in $S^{(\text{CENTERS})}$, and (ii) each star piece contains a corner of P. Then we can find such an optimal solution in $O(\text{poly}(n, |S^{(\text{CENTERS})}|))$ arithmetic operations*[4].

REMARK 5.3. For our purposes, we will have $|S^{(\text{CENTERS})}| = O(n^6)$, and the total running time of the dynamic programming algorithm in Theorem 5.2 will be $O(n^{105})$ under the RAM model.

4 Instead of measuring running time here, we count the number of arithmetic operations. This is since points in $S^{(\text{centers})}$ might be complicated to represent exactly. In fact, we will invoke the dynamic programming algorithm with points in $S^{(\text{centers})}$ of *degree* (and hence bit-complexity) $\Omega(n)$, so we cannot assume that we can perform computation on these points in $O(1)$ time.

5.1.1 Defining Other Steiner Points

Suppose we are given a polygon P and a set of potential star centers $S^{\text{(CENTERS)}}$, as in the statement of Theorem 5.2. Using these, we will be able to identify all potential Steiner points needed for our dynamic program. The main idea is to consider an optimal partition that is *area-maximum*, and use our structural results from Section 4 (in particular Corollary 4.3, that characterizes all potential Steiner points). We will define the set $S^{\text{(INTERNAL)}}$ of potential Steiner points to be used as corners of the star-shaped pieces. Moreover, we define a smaller set $S^{\text{(BORDER)}} \subseteq (S^{\text{(INTERNAL)}} \cap \partial P)$ of potential corners of the star pieces that are also on the boundary of P.

LEMMA 5.4. *Let P be a polygon and $S^{\text{(CENTERS)}}$ a set of points satisfying the premise of Theorem 5.2. Then we can find sets $S^{\text{(BORDER)}}$ and $S^{\text{(INTERNAL)}}$ of size* $\text{poly}(n, |S^{\text{(CENTERS)}}|)$ *such that some weakly simple minimum star partition $(Q_1, Q_2, \dots Q_k)$ of P with corresponding star centers $(A_1, A_2, \dots, A_k)$ satisfies the following properties:*

1. *Each piece Q_i touches the polygon boundary ∂P.*
2. *All star centers A_i are contained in $S^{\text{(CENTERS)}}$.*
3. *All corners of Q_i are contained in $S^{\text{(INTERNAL)}}$.*
4. *All corners of Q_i that are also on the boundary of P are contained in $S^{\text{(BORDER)}}$.*

Construction of Steiner points. We use the characterization of area maximum partitions from Section 4 in order to define the sets $S^{\text{(BORDER)}}$ and $S^{\text{(INTERNAL)}}$. Figure 33 shows the “worst case” example where some Steiner point depends on five star centers and two corners of P. We begin by constructing the sight line types as in Lemma 4.2.

— Let $L^{(i)}$ be the set of lines passing through a potential star center in $S^{\text{(CENTERS)}}$ and a distinct corner of P. Note that $|L^{(i)}| = O(n|S^{\text{(CENTERS)}}|)$, and they correspond to sight lines of type (i).
— Similarly, let $L^{(ii)}$ be the set of lines passing through a pair of distinct potential star center in $S^{\text{(CENTERS)}}$. Note that $|L^{(i)}| = O(|S^{\text{(CENTERS)}}|^2)$, and they correspond to sight lines of type (ii).
— To define $L^{(iii)}$, we first define $S^{(iii)}$ to be the set of points on the intersection of two non-parallel lines in $L^{(i)}$. Then we define $L^{(iii)}$ to be the lines through a potential star center in $S^{\text{(CENTERS)}}$ and a distinct point in $S^{(iii)}$. Note that $|S^{(iii)}| = O(n^2|S^{\text{(CENTERS)}}|^2)$, so $|L^{(iii)}| = O(n^2|S^{\text{(CENTERS)}}|^3)$, and that these correspond to sight lines of type (iii).

Now we are ready to use these sight lines to construct all necessary Steiner points, specifically, the different types specified in Corollary 4.3.

— Let $S^{(1)}$ be the intersections of a segment of P and a (non-parallel) line in $(L^{(i)} \cup L^{(ii)})$. Note that $|S^{(1)}| = O(n|S^{\text{(CENTERS)}}|^2)$
— Let $S^{(2)}$ be the intersections of a line in $(L^{(i)} \cup L^{(ii)})$ and a (non-parallel) line in $(L^{(i)} \cup L^{(ii)} \cup L^{(iii)})$. Note that $|S^{(2)}| = O(n^2|S^{\text{(CENTERS)}}|^5)$

— Let $S^{(3)} := S^{(iii)}$ be the intersections of two (non-parallel) lines in $L^{(i)}$. Note that $|S^{(3)}| = O(n^2|S^{(\text{CENTERS})}|^2)$.

Finally, we can, by Corollary 4.3, define our "small" sets of potential Steiner points to consider:

— $S^{(\text{INTERNAL})} = \text{corners}(P) \cup S^{(1)} \cup S^{(2)} \cup S^{(3)}$ for internal corners of star pieces, with $|S^{(\text{INTERNAL})}| = O(n^2|S^{(\text{CENTERS})}|^5)$.
— $S^{(\text{BORDER})} = \text{corners}(P) \cup S^{(1)}$ for corners of star pieces also on the boundary ∂P, with $|S^{(\text{BORDER})}| = O(n|S^{(\text{CENTERS})}|^2)$.

We additionally note that $S^{(\text{CENTERS})} \subseteq S^{(2)} \subseteq S^{(\text{INTERNAL})}$ (since a point $A \in S^{(\text{CENTERS})}$ will lie on at least two lines in $L^{(i)}$ as P has at least three non-collinear corners). Similarly $(S^{(\text{CENTERS})} \cap \partial P) \subseteq S^{(1)} \subseteq S^{(\text{BORDER})}$.

PROOF OF LEMMA 5.4. Consider any minimum star partition $Q = (Q_1, Q_2, \dots Q_k)$—with star centers $(A_1, A_2, \dots A_k)$—of P that satisfies the premise of Theorem 5.2: that is each star center is in $S^{(\text{CENTERS})}$ and each piece touch the boundary of P at some corner.

We now consider an area maximum partition $Q' = (Q'_1, Q'_2, \dots Q'_k)$ with the same star centers $(A_1, A_2, \dots A_k)$. By Lemmas 4.2 and 2.5 and Corollary 4.3, this partition must satisfy that each corner of Q_i is in $S^{(\text{INTERNAL})}$ and if this corner is also on the boundary ∂P it must be in $S^{(\text{BORDER})}$. Indeed $L^{(i)}, L^{(ii)}, L^{(iii)}$ must contain all possible sight line types of Lemma 4.2, and so $S^{(1)}, S^{(2)}, S^{(3)}$ must contain all potential Steiner points as specified in Corollary 4.3.

What remains is to argue that each star piece touches the boundary of P. This is non-trivial, and unfortunately does not seem to follow directly from area-maximality. Instead we use the fact that the star pieces in the original partition Q touched the boundary at some corner. For each piece Q_i we can choose an arbitrary sight line $r_i = A_iB_i$ to a corner B_i of P. Intuitively, we then "fix" this sight line before doing area-maximality. That is, we instead consider Q' to be an area maximum partition where the star centers A_i are fixed, *and the chosen sight lines r_i must be contained in piece Q_i*.

Formally, we can do this by changing the input polygon P into a weakly simple polygon P' defined as follows. For each chosen sight line r_i we add it as an "incision" to P' (so now P' is a weakly simple polygon). Note that these incisions cannot intersect except at their endpoints. The partition Q is also a minimum star-partition of P', but now each star center is at a corner of P'. If Q' is chosen to be area maximum in this new polygon P', we can then look at Q' as a partition of P where we assign the "incision" r_i to piece Q'_i. This means that each piece must touch the boundary (perhaps only because of a degenerate ray from the star center to some corner at P, but this is acceptable since we allow the pieces to be weakly simple polygons).

What remains is to argue that $S^{(\text{INTERNAL})}$ and $S^{(\text{BORDER})}$ are still sufficient, i.e. that we did not introduce any new Steiner points. All new corners of P' were star centers, so we did not introduce any additional sight lines for Lemma 4.2. The additional edges of P' are the "incisions"

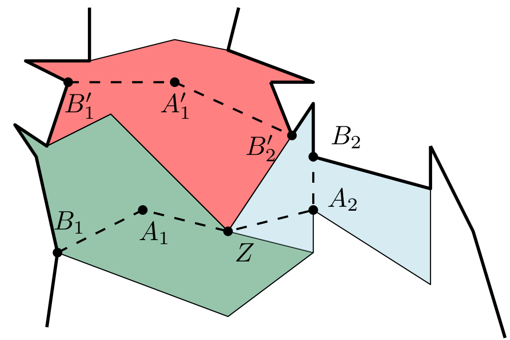

Figure 34. A short separator B_1'-A_1'-B_2' and a long separator B_1-A_1-Z-A_2-B_2 as part of a star partition.

r_i, but these will all be in $L^{(i)}$ (lines from star centers to corners of P), so they are considered as potential endpoints of sight lines in item 2 (instead of item 1) in Corollary 4.3. ∎

5.1.2 Dynamic Programming Algorithm

We now provide our dynamic programming algorithm (Algorithm 1) that will consider each possible star-partition satisfying the properties of Lemma 5.4, and thus will find an optimal partition given the set $S^{(\text{CENTERS})}$. Let B_1, B_2 be two points on ∂P, $P[B_1 : B_2] \subset \partial P$ be the chain from B_1 to B_2 along ∂P in counterclockwise order. We consider the separators (see also Figure 34):

- Short separator of the form B_1-A_1-B_2 for $B_1, B_2 \in S^{(\text{BORDER})}$, and $A_1 \in S^{(\text{CENTERS})}$.
- Long separator of the form B_1-A_1-Z-A_2-B_2 for $B_1, B_2 \in S^{(\text{BORDER})}$, $A_1, A_2 \in S^{(\text{CENTERS})}$, and $Z \in S^{(\text{INTERNAL})}$.

In the dynamic program, we will, for a given separator, calculate an optimal way to partition the subpolygon P' enclosed by $P[B_1 : B_2]$ and the separator, given that there are star centers already placed at A_1 (and A_2 in case of a long separator) on the separator. Since each piece touches the boundary, we will see that it is sufficient to consider separators passing through at most two star pieces. We describe a few elementary ways to build separators for larger and larger subpolygons P' by e.g. merging two separators or moving the common corner point Z. In figure Figure 35 and the pseudo-code Algorithm 1 we can see the different cases we consider for transitions. We also explain the cases here:

Case 0: (Base Case) In the base case we consider trivial short separators, B_1-A_1-B_2 where either $B_1 = B_2$, or B_2 is next to B_1 in counterclockwise order. Here $B_1A_1B_2$ forms a possibly degenerate triangle with one side on the boundary ∂P, that can be assigned to the star piece with center A_1

Case 1: (Merge short + short) A short separator B_1-A_1-B_2 can be seen as the "merge" of two other short separators B_1-A_1-B' and B'-A_1-B_2 for some $B' \in S^{(\text{BORDER})} \cap P[B_1 : B_2]$.

Case 2: (New star center) When a short separator B_1-A_1-B_2 is neither trivial (Case 0) or the merge of two short separators (Case 1), some other star center A' must be able to see the

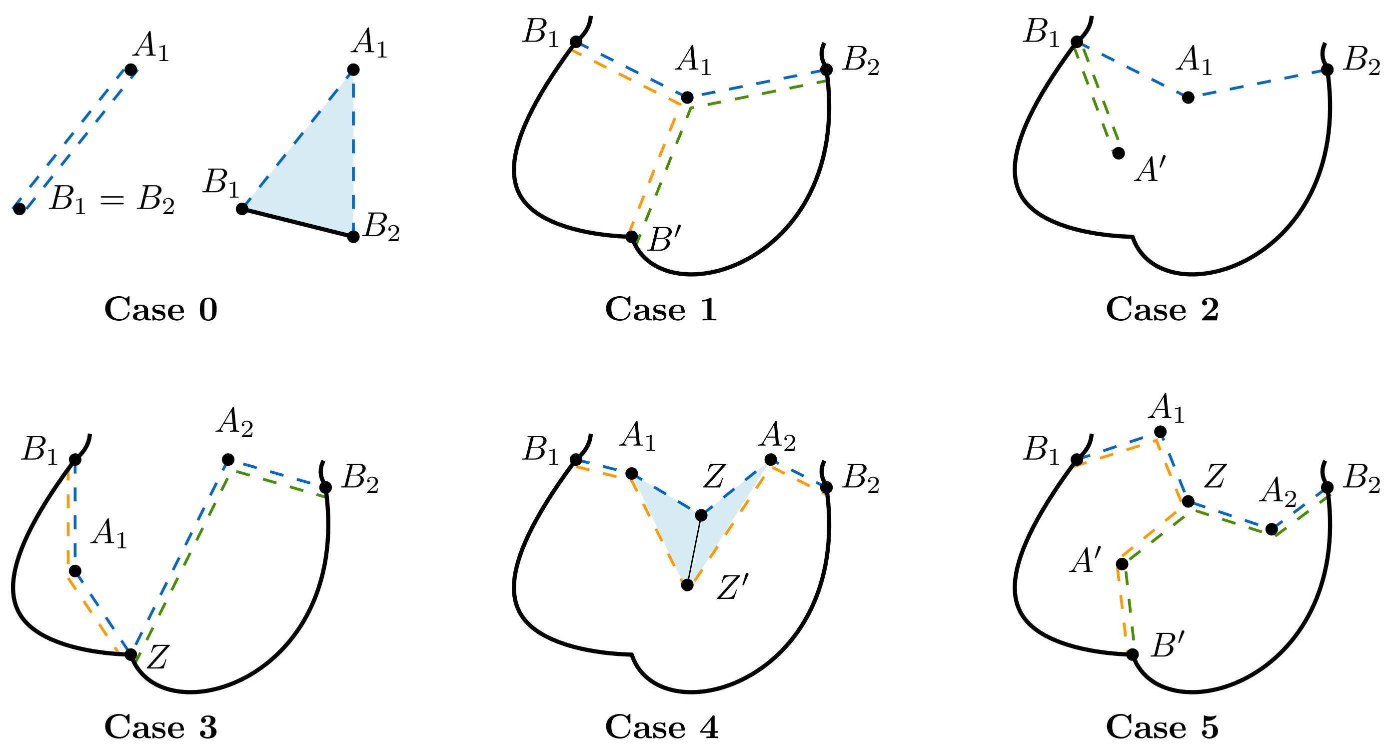

Figure 35. The different transitions we need to consider for the dynamic program algorithm. Cases 0-2 concerns short separators B_1-A_1-B_2, and Cases 3-5 concerns long separators B_1-A_1-Z-A_2-B_2, and we want to solve the subpolygon "below" these separators. Curve parts indicate that the details have now been shown.

boundary point B_1 too. This becomes a long separator B_1-A'-B_1-A_1-B_2, where the segment B_1A' is a "spike" that occurs twice.

Case 3: (Combine short + short) A long separator B_1-A_1-Z-A_1-B_2 where Z is on the boundary somewhere between B_1 and B_2 can be decomposed into two short separators B_1-A_1-Z and Z-A_2-B_2.

Case 4: (Move common corner) A long separator B_1-A_1-Z-A_1-B_2 can also arise by moving the common corner Z from some other point Z', where B_1-A_1-Z'-A_1-B_2 is also a long separator. Here the triangles $A_1Z'Z$ and A_2ZZ' can be assigned to the star piece with centers A_1 and A_2 respectively.

Case 5: (Merge long + long) For a long separator B_1-A_1-Z-A_1-B_2, if neither the common corner Z can be moved (Case 4), nor it is on the boundary (Case 3), there must exists some other star center A' that can see Z. The star piece with center A' must also touch the boundary at some point, say at B'. Then our separator is a "merge" of two long ones: B_1-A_1-Z-A'-B' and B'-A'-Z-A_2-B_2.

Note that it is only in Case 0 and Case 4 where we actually assign some positive area of P to some star piece. Whenever we say "X **can see** Y" in Algorithm 1, we mean that the segment XY is contained within the subpolygon of P restricted by the separator.

To use the dynamic programming algorithm to find an optimal star partition, we arbitrarily pick consecutive points $B_1, B_2 \in S^{(\text{BORDER})}$ on the boundary of P, where B_2 is next to B_1 in clockwise order. There must be some star piece seeing this segment, so we can simply try each possibility of star centers $A \in S^{(\text{CENTERS})}$ and call `SolveSeparator`(B_1, A, B_2) to find the optimal solution given that A sees the segment B_1B_2 (and then just return the best solution we found). Note that we do not consider B_1 and B_2 to be "adjacent" here for Case 0, as the region enclosed by $P[B_1 : B_2]$ and B_1-A-B_2 is $P \setminus B_1AB_2$.

OBSERVATION 5.5. *Note that the above actually gives us all possible positions, in optimal solutions, for star centers $A \in S^{(\text{CENTERS})}$ that see the segment B_1B_2. This will be useful later in the full algorithm.*

Correctness. We now argue that Algorithm 1 is correct, that is that the optimal solution can be constructed using the transitions (cases) in Figure 35. Suppose we have some optimal partition satisfying the properties of Lemma 5.4. We will show that the dynamic program will consider this optimal solution.

Let us first consider the case that we are looking at a short separator B_1-A_1-B_2 in this optimal partition. If either $B_1 = B_2$ or B_1 and B_2 are consecutive points on the border of P in $S^{(\text{BORDER})}$, we are in Case 0. Otherwise, in the optimal partition, either the star piece with center A_1 will also touch the boundary somewhere in between B_1 and B_2, or not. In case it does, it must touch in a point $B' \in S^{(\text{BORDER})}$, where we naturally have two short separators of sub-regions B_1-A_1-B' and B'-A_1-B_2, which is handled by Case 1. In case it does not, there must be some other star piece (say with center A') that touches B_1, and then we are in Case 2 with long separator B_1-A'-B_1-A_1-B_2.

Now suppose instead that we are looking at a long separator B_1-A_1-Z-A_2-B_2 that is part of the optimal partition. This means that the two pieces with centers A_1 and A_2 touch. Note that they will touch in a single contiguous internal boundary (Lemma 4.1 give a complete characterization of how this boundary can look; it is either a single point or up to two line segments). Note that Z must be a point on this contiguous internal boundary. If Z is not the last corner on this boundary, we can move it to the next corner Z', as in Case 4. If Z instead was the last corner on this boundary between pieces with centers A_1 and A_2, we have two sub-cases: (i) either Z is on the boundary of the polygon P, or (ii) else there must be some other star piece touching Z. In sub-case (i) it must be the case that $Z \in S^{(\text{BORDER})}$, and we have two natural short separators for sub-regions: B_1-A_1-Z and Z-A_2-B_2, as handled by Case 3. In sub-case (ii), let $A' \in S^{(\text{CENTERS})}$ be the star center of the additional piece touching Z in the optimal partition. Note that A' must also touch the boundary of P somewhere (Lemma 5.4), say in point $B' \in S^{(\text{BORDER})}$. Again, we have two natural (long) separators for sub-regions: B_1-A_1-Z-A'-B' and B'-A'-Z-A_2-B_2, which is handled by Case 5.

function SolveSeparator(B_1, A_1, B_2):
 ▷ returns the minimum number of additional (not counting A_1) star pieces needed to cover the enclosed region of $P[B_1 : B_2]$ and B_1-A_1-B_2.
  Let P' be the region enclosed by $P[B_1 : B_2]$ and B_1-A_1-Z-A_2-B_2, compute visibility of $S^{(\text{centers})} \cup S^{(\text{border})} \cup S^{(\text{internal})}$ within P'
  $\text{opt} \leftarrow n$
  if $B_1 = B_2$ or B_2 is next to B_1 in counterclockwise order **then** ▷ Case 0: base case
    $\text{opt} \leftarrow 0$
  for $B' \in (S^{(\text{border})} \setminus \{B_1, B_2\}) \cap P[B_1 : B_2]$ **do** ▷ Case 1: merge short + short
    if A_1 can see B' **then**
      $\text{opt} \leftarrow \min(\text{opt}, \texttt{SolveSeparator}(B_1, A_1, B') + \texttt{SolveSeparator}(B', A_1, B_2))$
  for $A' \in (S^{(\text{centers})} \setminus \{A_1\}) \cap P'$ **do** ▷ Case 2: new star center
    if A' can see B_1 **then**
      $\text{opt} \leftarrow \min(\text{opt}, 1 + \texttt{SolveSeparator}(B_1, A', B_1, A_1, B_2))$
  return opt

function SolveSeparator(B_1, A_1, Z, A_2, B_2):
 ▷ returns the minimum number of additional (not counting A_1 or A_2) star pieces needed to cover the enclosed region of $P[B_1 : B_2]$ and B_1-A_1-Z-A_2-B_2
  Let P' be the region enclosed by $P[B_1 : B_2]$ and B_1-A_1-Z-A_2-B_2, compute visibility of $S^{(\text{centers})} \cup S^{(\text{border})} \cup S^{(\text{internal})}$ within P'
  $\text{opt} \leftarrow n$
  if $Z \in S^{(\text{border})} \cap P[B_1 : B_2]$ **then** ▷ Case 3: combine short + short
    $\text{opt} \leftarrow \min(\text{opt}, \texttt{SolveSeparator}(B_1, A_1, Z) + \texttt{SolveSeparator}(Z, A_2, B_2))$
  for $Z' \in (S^{(\text{internal})} \setminus \{Z\}) \cap P'$ **do** ▷ Case 4: move common corner
    if all three of A_1, A_2 and Z can see Z' **then**
      $\text{opt} \leftarrow \min(\text{opt}, \texttt{SolveSeparator}(B_1, A_1, Z', A_2, B_2))$
  for $A' \in (S^{(\text{centers})} \setminus \{A_1, A_2\}) \cap P'$ **do** ▷ Case 5: merge long + long
    for $B' \in S^{(\text{border})} \cap P[B_1 : B_2]$ **do**
      if A' can see B' and Z **then**
        $\text{opt} \leftarrow \min(\text{opt}, 1 + \texttt{SolveSeparator}(B_1, A_1, Z, A', B') + \texttt{SolveSeparator}(B', A', Z, A_2, B_2))$
  return opt

Algorithm 1. Dynamic programming algorithm

To conclude, any optimal partition satisfying the properties of Lemma 5.4 must be constructable by Cases 0-5. Since Algorithm 1 considers all these cases as transitions, it will find an optimal partition of P into star-shaped pieces.

Running time. To make Algorithm 1 run in polynomial time we assume standard memoization, i.e. that if `SolveSeparator` is called several times with the same arguments it only needs to be solved once. Since $S^{(\text{BORDER})}$ and $S^{(\text{INTERNAL})}$ are both of size $\text{poly}(n, |S^{(\text{CENTERS})}|)$, it is clear that we have a polynomial many separators and polynomially many transitions, and therefore Algorithm 1 runs in $O(\text{poly}(n, |S^{(\text{CENTERS})}|))$ time (proving Theorem 5.2). Below we analyze the complexity in more detail.

Note that in the 3-parameter function `SolveSeparator`(B_1, A_1, B_2) (short separators), we have $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|)$ states, as $B_1, B_2 \in S^{(\text{BORDER})}$ and $A_1 \in S^{(\text{CENTERS})}$. Similarly, in the 5-parameter function `SolveSeparator`(B_1, A_1, Z, A_2, B_2) (long separators), we have $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|^2 \cdot S^{(\text{INTERNAL})})$ states, as $B_1, B_2 \in S^{(\text{BORDER})}$, $A_1, A_2 \in S^{(\text{CENTERS})}$, and $Z \in S^{(\text{INTERNAL})}$. We count the number of transitions in the algorithm for each "Case":

Case 0: $O(1)$ transitions for $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|)$ many separators.
Case 1: $O(|S^{(\text{BORDER})}|)$ transitions for $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|)$ many separators.
Case 2: $O(|S^{(\text{CENTERS})}|)$ transitions for $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|)$ many separators.
Case 3: $O(1)$ transitions for $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|^2 \cdot |S^{(\text{INTERNAL})}|)$ many separators.
Case 4: $O(|S^{(\text{INTERNAL})}|)$ transitions for $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|^2 \cdot |S^{(\text{INTERNAL})}|)$ many separators.
Case 5: $O(|S^{(\text{CENTERS})}| \cdot |S^{(\text{BORDER})}|)$ transitions for $O(|S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|^2 \cdot |S^{(\text{INTERNAL})}|)$ many separators.

We see that Case 4 and Case 5 dominate all other cases, for a total of

$$O(|S^{(\text{BORDER})}|^3 \cdot |S^{(\text{CENTERS})}|^3 \cdot |S^{(\text{INTERNAL})}| + |S^{(\text{BORDER})}|^2 \cdot |S^{(\text{CENTERS})}|^2 \cdot |S^{(\text{INTERNAL})}|^2)$$

many transitions. For each of these transitions, we might need to go through all $O(n)$ segments of the polygon to verify the "X can see Y" statements, adding another factor of $O(n)$ to the running time. Since $|S^{(\text{BORDER})}|, |S^{(\text{INTERNAL})}| = \text{poly}(n, |S^{(\text{CENTERS})}|)$, we have proved Theorem 5.2.

REMARK 5.6. Plugging in $|S^{(\text{BORDER})}| = O(n|S^{(\text{CENTERS})}|^2)$, $|S^{(\text{INTERNAL})}| = O(n^2|S^{(\text{CENTERS})}|^5)$, we see that the total number of transitions we have is $O(n^6|S^{(\text{CENTERS})}|^{16})$, assuming $|S^{(\text{CENTERS})}| \geq n$. In the final algorithm we will have $|S^{(\text{CENTERS})}| = O(n^6)$, making for a total of $O(n^{102})$ transitions! This means there are a total of $O(n^{103})$ arithmetic operations to run the dynamic programming algorithm (to check the "X can see Y" statements). According to Corollary 3.7 and Corollary 4.3, we will consider only points with degree $O(n)$ (i.e. all points can be described by $O(n)$ arithmetic operations from the input points), so we can perform arithmetic operations in $O(n^2)$ time naively. Therefore the total running time of Algorithm 1, for our purposes, is $O(n^{105})$, as stated in Remark 5.3.

5.2 Finding Star Centers & Full Algorithm

Now we turn to show our full polynomial time algorithm, thus proving Theorem 1.1. We note that if we can find a relatively small set of potential star centers, we can simply use our dynamic programming algorithm (Algorithm 1 and Theorem 5.2). However, we will see that in order to find such a sufficient set of potential star centers, we will need to solve smaller instances of the same problem (where we need to invoke the algorithm recursively).

Constructable partitions. Throughout this section, we will let the *root edge r* be an arbitrary edge of P. We will focus our attention to optimal partitions that can be constructed using the process defined in Section 3—specifically by Theorem 3.5—and call such a star-partition *constructable*. In particular, we recall that a *constructable* partition satisfies the following properties:

(i) It is optimal, that is it uses a minimum number of star pieces.
(ii) Each star piece touches the boundary of P at some corner.
(iii) All tripods in the partition are oriented towards the root face (the face with the root edge r).
(iv) All star centers are at the intersection of two lines, each of these lines are either an edge of P, a diagonal between two concave corners of P, or a line through a tripod point and one of its supporting corners.

Indeed, by Theorem 3.5 (and Lemma 3.10 for item (ii)) there must exist some constructable partition. However, restricting ourselves to constructable partitions is not enough to get an efficient algorithm: in general there are a double-exponential number of points that appear as star centers in some constructable partition. Therefore we seek to restrict our class of optimal partitions further, and here the *greedy choice* comes into play (defined below in Section 5.2.1).

Before presenting the *greedy choice*, we prove a simple lemma saying that the partition inside the pseudo-triangle of a tripod is not particularly important, and that any partition outside this pseudo-triangle can always be extended to cover the pseudo-triangle too. This will be useful for our algorithm, since we can then focus on solving sub-problems defined by diagonals of P (and not defined by the unknown tripod).

LEMMA 5.7. *(See Figure 36). Suppose T is a tripod supported by corners (D_1, D_2, D_3) in some constructable partition, with tripod point C. Let P' be one of the three sub-polygons T splits P into (say between corners D_1 and D_3), and say $A_1 \in P'$ is the star centers participating in T through corner D_1. Let Δ be the pseudo-triangle of the points (D_1, D_2, D_3). Note that $P' \setminus \Delta$ consists of several (at least one) sub-polygons, call them $P_1, P_2, \dots, P_k$, where $A_1 \in P_1$.*

Then any *partition of $P_1, P_2, \dots, P_k$, where A_1 is a star center in P_1 seeing corner D_1, can be extended to a partition of P' (without moving star centers) that A_1 sees C.*

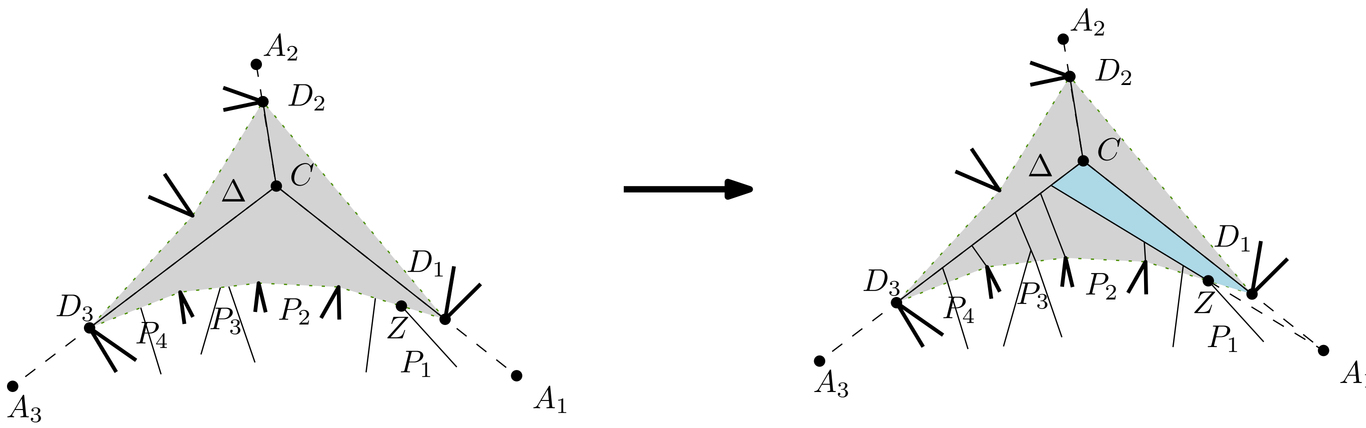

Figure 36. Modification as in Lemma 5.7. The pseudo-triangle Δ splits the polygon up into multiple sub-polygons, where we let $P_1, P_2, \ldots$ be those on the same "side" as the star-center A_1. Any partitions of $P_1, P_2, P_3, \ldots$ can be extended to the polygonal line D_3-C-D_1 in such a way that A sees the tripod point C. We first give the blue region to Q_1, then extend all the segments touching the pseudo-diagonal D_1D_3 one by one, from right to left, until it meets any existing segment.

PROOF. See Figure 36. Let $s_1, \ldots, s_m$ be the set of segments in the partition of $P_1, \ldots, P_k$ that touches $\partial\Delta$, ordered by the touching point from D_1 to D_3. We simply extend $s_1, \ldots, s_m$ one by one, until they meet the separator D_3-C-D_1 or previous extended boundaries.

We will handle the piece Q_1 with star center A_1 in a special manner, since we want it to see the tripod point C. So let Z be the point on $\partial\Delta$ so that Q_1 contains the segment D_1Z on this boundary. Then we extend the line A_1Z until it meets the line D_3C first, and assign the blue region to Q_1. This clearly leads to a partition of P' while keeping the assignment on $P_1 \cup \ldots \cup P_k$. Since C is a convex corner of P', and all new angles are intersection of a ray and a straight line (which must be convex), all pieces must remain in star shape.

What remains is to argue that everything inside Δ is covered by our extended partition. This is true as long as the pseudo-diagonal between D_1 and D_3 contains no edge of P (i.e. $\partial\Delta \cap \partial P$ just contains a finite amount of points, and no line segments). To argue this we use that T was a tripod of some constructable solution. In particular this means that there exists some partition where T is a tripod and no star center lies within the pseudo-triangle Δ. Hence, if there was some edge e of P contained in $\partial\Delta$, then the interior of this edge could not be seen by any star center in such a constructable partition, which is our desired contradiction. ∎

5.2.1 Greedy Choice

Consider three concave corners (D_1, D_2, D_3) of P, that might support a tripod in some *constructable* partition. We now argue that if there are many possibilities for how the legs of a tripod supported by (D_1, D_2, D_3) look like, then it suffices to consider a single one of these possibilities! We will call this arrangement the *greedy choice* of this tripod. Recall that the tripod point is constructed by two of the sub-polygons P_1, P_2, and used to define the third sub-polygon

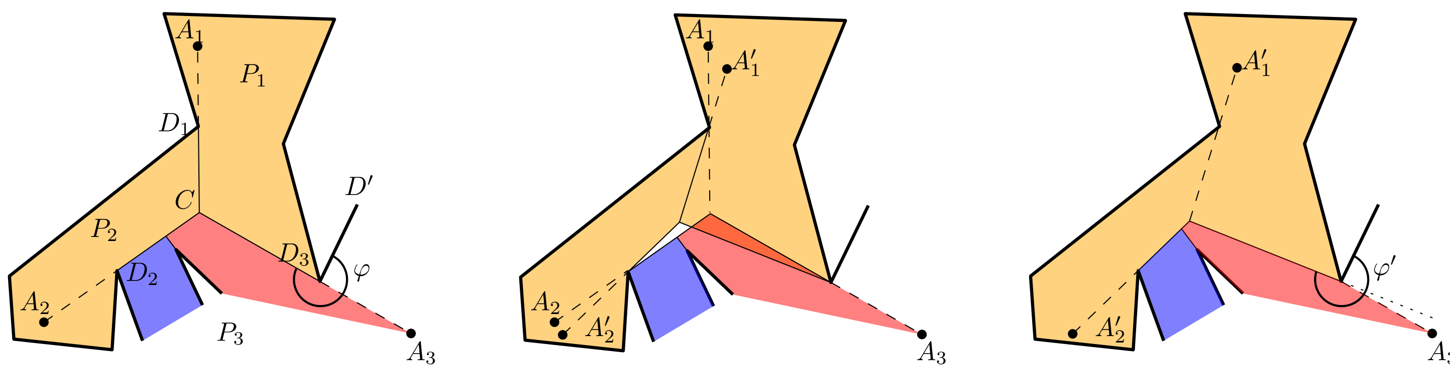

Figure 37. *Left:* Two child star-centers A_1 and A_2 define a (fake) tripod with tripod point C, splitting the polygon into three subpolygons: "childs" P_1, P_2, and "parent" P_3. The angle $\varphi = \angle CD_3D'$ of this tripod is a measure on how "restricted" a potential star-center A_3 defined by this tripod is. *Middle:* Two other star centers A'_1 and A'_2 define another fake tripod on the same support, which is less restrictive (i.e., with an angle $\varphi' < \varphi$). Inside P_3 (the parent-side of the tripod), the same partition is shown (in red and blue) as on the left. *Right:* The star partition of subpolygon P_3 can be adjusted (without moving any star centers) to also work with the less restrictive tripod.

P_3 (see Figures 6 and 37). In particular, P_1 and P_2 are the children in the *tripod tree* (see Figure 20), and P_3 the parent. We argue that the greedy choice will be the combination of optimal solutions in the two subpolygons P_1 and P_2 that give rise to the *least restrictive* tripod-center when constructing the optimal solution for P_3. With *least restrictive* we mean the one that minimizes the angle $\varphi = \angle CD_3D'$ as in Figure 37, as such a tripod will only impose the mildest restrictions on where the star center $A_3 \in P_3$ participating in the tripod lies.

We begin by proving that it never hurts replacing a tripod with a less restrictive one, see also Figure 37.

LEMMA 5.8. *Suppose that there is a tripod $\mathcal{T}$ with tripod point C supported by three corners (D_1, D_2, D_3), part of a constructable partition $\mathcal{Q}$. Let Δ be the pseudo-triangle of the tripod, and consider the three sub-polygons P_1, P_2, P_3 in $P \setminus \Delta$ participating in the tripod, such that the parent star center of $\mathcal{T}$ is contained in P_3, as in Figures 6 and 37.*

Suppose now that there is some other constructable sub-partitions of P_1 and P_2 (using the same number of star pieces as in the original one) giving rise to another fake tripod $\mathcal{T}'$ (supported on the same three concave corners D_1, D_2, D_3) with tripod point C', that is less restrictive (the angle φ in Figure 37 is smaller) for P_3. Then there also exists a constructable optimal star partition of P that contains these new sub-partitions of P_1 and P_2, and the fake tripod $\mathcal{T}'$.

PROOF. By Lemma 5.7, there exists an optimal partition $\mathcal{Q}'$ containing the same sub-partition in P_1 and P_2 that gives $\mathcal{T}'$. Since no star center is in the pseudo-triangle Δ of $\mathcal{T}$, the pseudo-triangle $\mathcal{T}'$ is also constructable with respect to $\mathcal{Q}'$. Our lemma then follows directly from Lemma 3.9. ∎

Greedy-constructable partitions. By Lemma 5.8, it never hurts to replace a tripod with a less restrictive one. The next step is to argue that we can assume that *all* tripods in our constructable partition can follow such a greedy choice—which is a very useful property when designing an algorithm. This is a bit subtle, since such an algorithm might not actually find the least restrictive version of a tripod $\mathcal{T}$, but only the least restrictive version *given that all children tripods also follow the greedy choice*. We formalize this by the notion of *greedy-constructable* partitions. A *greedy-constructable* partition is *constructable* and also satisfies the following extra property, adding to properties (i)–(iv) of a constructable partition.

(v) Any tripod $\mathcal{T}$ in the partition is *greedy* (see Definition 5.9 below).

DEFINITION 5.9 (Greedy Choice & Less Restrictive Tripods). First, we define the *angle* of a fake tripod $\mathcal{T}$ as in Figure 37, i.e., the angle $\varphi = \angle CD_3D'$, where D' is the next corner of P after D_3, in the parent-subpolygon P_3. We say that a fake tripod $\mathcal{T}$ is *less restrictive* than another fake tripod $\mathcal{T}'$ (supported by the same three corners) if the angle is smaller for $\mathcal{T}$ than for $\mathcal{T}'$, i.e. if $\mathcal{T}$ imposes a weaker restriction on the potential star center in the parent subpolygon. We break ties in an arbitrary but consistent manner.

Consider a tripod $\mathcal{T}$ supported by corners (D_1, D_2, D_3) in some constructable partition. Suppose this tripod splits the polygon into the three sub-polygons P_1, P_2, P_3 and is oriented towards P_3. Consider all the pairs of *greedy-constructable*[5] sub-partitions of P_1 and P_2, giving rise to some (fake) tripods $\mathcal{T}'$ supported by the same corners (D_1, D_2, D_3). Then the tripod T is the *greedy choice* for (D_1, D_2, D_3), or we simply say that $\mathcal{T}$ is *greedy*, if it is the *least restrictive* for P_3 among all such (fake) tripods $\mathcal{T}'$.

LEMMA 5.10. *There exists a greedy-constructable partition.*

PROOF. The general idea is to start with a constructable partition (which exists by Theorem 3.5), and replace tripods that are not greedy by their greedy version instead, using Lemma 5.8. At first it might not be apparent that this will work, since Lemma 5.8 might introduce new tripods that are not greedy. We overcome this by carefully eliminating bad tripods in a bottom-up fashion, and continuing in a recursive manner, similar to our proof that there exists *constructable* partitions (proof of Theorem 3.5).

Formally, let us consider some polygon P together with a *constructable* partition Q of P. Throughout this proof, whenever we say *tripod*, we also consider *fake tripods* (see discussion in Section 3.1 and Remark 3.4)—i.e. those that can be constructed by two children star centers but might not be used to construct a parent star center. Consider the *rooted tripod tree* (see Observation 3.3 and Figures 5 and 20) of this partition. In this tree we mark all tripods that

5 We note that the definition of *greedy choice* tripods and *greedy-constructable* partitions are mutually dependent on each other. With greedy-constructable for a sub-polygon, we mean that all the tripods used in the partition of the sub-polygon abides the greedy choice. Since the tripods form a rooted tree (see Observation 3.3), this recursive definition is well-defined, as tripods only depend on other tripods deeper down in the tree.

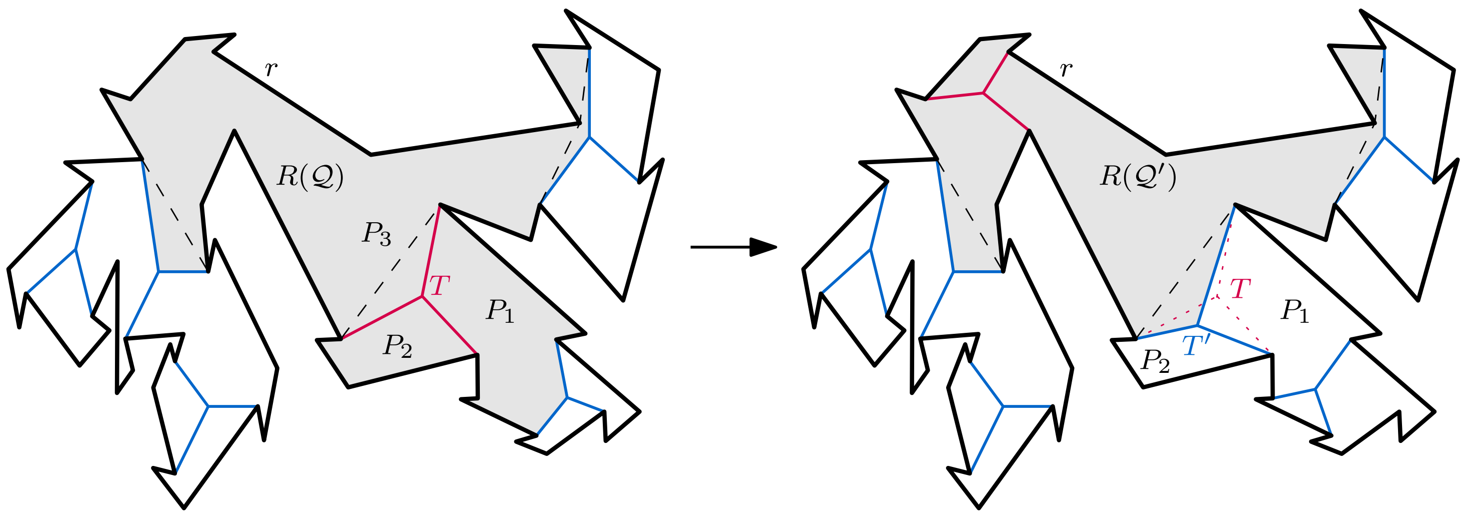

Figure 38. The process of replacing a tripod T with the *greedy* one T', as described in the proof of Lemma 5.10. Bad tripods $\mathcal{T}_{bad}$ are marked in red, good $\mathcal{T}_{good}$ in blue. The dashed lines indicates "fences" where star centers are not allowed to pass through. The gray area $R(\cdot)$ goes down, since the bad tripod T was replaced with a good one T'. The partitions (and tripods) inside P_1, P_2 and $P_3' = R(Q) \cap P_3$ can change: for example note that the tripods inside P_1 changed, and that a new bad tripod inside $R(Q')$ appeared.

are *not* greedy as bad, along with all ancestor tripods. Call this set of bad tripods $\mathcal{T}_{bad}(Q)$, and let $\mathcal{T}_{good}(Q)$ be all other tripods. That is $\mathcal{T}_{good}(Q)$ is the set of tripods T in Q for which all tripods in the subtree rooted at T (including T itself) are greedy. If $\mathcal{T}_{bad}(Q)$ is empty, Q is *greedy-constructable*, so we are done; so suppose that this is not the case. Let $R(Q)$ be the region in P reachable from the root edge r without passing through any tripod from $\mathcal{T}_{good}(Q)$ (see also Figure 38). Note that all tripods in $\mathcal{T}_{bad}(Q)$ must by definition be in $R(Q)$. We will argue that we can find some other (constructable) partition Q' such that $R(Q')$ is smaller. This is enough, since there are only finitely many possible positions for tripods in constructable solutions, so $R(\cdot)$ can only decrease a finite number of times.

Consider Figure 38. Let $T \in \mathcal{T}_{bad}(Q)$ be some non-greedy tripod. Say it is supported by corners (D_1, D_2, D_3) and splits the polygon into sub-polygons P_1, P_2, P_3 where it is oriented towards P_3. Without loss of generality, we may assume that, in our partition, the sub-partitions of P_1 and P_2 are *greedy-constructable*: that is all tripods in P_1 and P_2 abide the greedy choice (else we can instead choose T to be one of these non-greedy tripods deeper down in the tripod tree). Since T is not greedy, there must exist other sub-partitions of P_1 and P_2 that are also *greedy-constructable* and give rise to a better (less restrictive) *greedy* tripod T'. First note that these new sub-partitions of P_1 and P_2 must use the same number of pieces as the original partition used for these sub-polygons (otherwise it was not optimal; note that it never makes sense to use an extra piece in a sub-partition to get a less restrictive tripod as we might as well place this extra piece at the tripod point which would make the whole tripod redundant).

Now we want to use Lemma 5.8 to replace the tripod T with T'. However, we cannot directly apply this lemma as it might destroy some greedy tripods and introduce new non-greedy tripods inside the region P_3 in an unpredictable manner. Let us define $P'_3 = P_3 \cap R(Q)$ and $P' = P_1 \cup P_2 \cup P'_3$. We use Lemma 5.8 on the polygon P' instead of P, which makes sure that we do not destroy any greedy tripods inside the region P_3. We must be slightly careful also to not move any star centers into the pseudo-triangles of tripods in P, whose tripod points might now be corners of P'. For this, we note that it is easy to extend Lemma 5.8 to take into account these pseudo-triangles:

In the first part of the proof of Lemma 5.8 we argued that there is an optimal (not necessarily constructable) solution without moving any star centers in P'_3, and in the second part we used Theorem 3.5 (or rather Lemma 3.9) to argue that then there must also exist a constructable one. When we invoke Lemma 3.9 we can do so on the full polygon P, but on our partitions where everything outside P'_3 is already constructable (so these star pieces and tripods will not change, and no star center will be moved into the pseudo-triangles of the boundary tripods of P'_3). Then Lemma 3.9 would give us an optimal partition where tripods and star centers inside P'_3 are also constructable.

To recap, we now have a constructable partition of P' where the greedy tripod T' is used instead of T. We extend this to a partition Q' of P by using all pieces from the original partition Q which where in $P \setminus P' = P_3 \setminus R(Q)$. By design we see that Q' is *constructable*. Moreover, $R(Q') = P'_3 = R(Q) \cap P_3$, since now T' is a greedy tripod (and still all tripods in the subregions P_1 and P_2 are greedy, although the new partition of these parts can be quite different from the original one). To conclude, $R(\cdot)$ must have gone down, as it is now also restricted by the tripod T' (and the original tripod T was completely contained in $R(Q)$). Hence, by induction, there must exist some greedy-constructable partition. ■

5.2.2 Minimum Star Partition Algorithm

We are now ready to present the full algorithm, and thus proving Theorem 1.1. To optimally partition a polygon P into a minimum number of star pieces we employ the following strategy. We begin by enumerating all possible positions for tripods (that is triples of concave corners D_1, D_2, D_3 of P that might support a tripod). Now, for each of these, we can employ the *greedy choice* (see Definition 5.9) to only have a single tripod-center we need to consider. By Lemma 5.10, there must exist some optimal solution in which all tripods follow this greedy choice. For now, assume we can actually compute these greedy tripod points (we will get back to this later). That is we have $O(n^3)$ potential tripod-centers in total. By Theorem 3.5 we can now construct a set of potential star centers $S^{(\text{CENTERS})}$ by considering all intersections of pairs of lines, where each line is either (i) an extension of a diagonal of P, or (ii) a line from a (greedy) tripod-center through the corresponding concave corner in this tripod. Note that there are only $O(n^2+n^3)$ such lines, so we can bound $|S^{(\text{CENTERS})}|$ by $O(n^6)$. Additionally, we note that in (greedy-)constructable partitions, each piece touches the boundary at some corner. Given $S^{(\text{CENTERS})}$, we can hence employ the dynamic programming algorithm (Algorithm 1, Theorem 5.2) to find a minumum star partition of the polygon P.

Resolving the greedy choice. Now, let us return to the issue of actually determining the position of the greedy choice tripod point, of some tripod supported on concave corners D_1, D_2, D_3 of P. Let P_1, P_2, P_3 be the sub-regions (like in Figure 6), such that the tripod is oriented towards P_3. Note that P_1 and P_2 are defined by a diagonal of P and not by the (so far unknown) tripod. Now, we can find greedy-constructable optimal solutions for P_1 and P_2 separately, by invoking our algorithm recursively (see also Figure 39): in the subpolygons we again enumerate all potential tripods, solve using the greedy choice, and invoke the dynamic program to obtain an optimal solution. Note that there are only $O(n^2)$ subproblems, since each subproblem is defined by some diagonal between two concave corners of P (here we use the fact that the tripods in a greedy-constructable partition form a *rooted* tree). Moreover, our dynamic program allows us to find all possible positions of the star center A_1 in P_1 used to define the tripod. Indeed A_1 is the star center that sees a prefix of the pseudo-diagonal from D_1 to D_2, so we can find all possible positions of it by Observation 5.5. Similarly for the star center A_2 in P_2. Therefore we may simply enumerate over all pairs of possibilities of A_1 and A_2 and choose the best valid one according to the greedy choice (Definition 5.9, see also Figures 6, 37 and 39). By Lemma 5.7, any partition we get here can be extended to a partition meeting the tripod legs. Conversely, any constructable partition with a tripod supported by D_1, D_2 and D_3, is also a partition of $P \setminus \Delta$ (as no star center would be inside the pseudo-triangle Δ; so we can apply Lemma 2.1 to carve out this part). Hence we do not loose or gain anything by restricting ourselves to finding optimal partitions restricted by the pseudo-diagonal, instead of restricted by the (so far unknown) tripod.

Full algorithm. The pseudo-code can be found in Algorithm 2, consisting of two mutually recursive functions `TripodGreedyChoice()` and `SolveSubregion()`. The function `TripodGreedyChoice()` will find the tripod point of the greedy tripod, and `SolveSubregion(`D_1, D_2`)` will optimally solve the sub-polygon enclosed by the diagonal D_1-D_2. To obtain the optimal partition for the full polygon, we can just call `SolveSubregion()` on some edge of P. The correctness follows from the above discussion.

Running time. We now analyze the running time of Algorithm 2, and again we apply memoization to not recompute the same subpolygons multiple times. We will see that the total running time is $O(\text{poly}(n))$—or, in fact, it takes $O(n^{105})$ arithmetic operations or $O(n^{107})$ time. According to Corollary 3.7 and Corollary 4.3, every star center or steiner point can be encoded by $O(n)$ corners of P, therefore each arithmetic operation can be done in $O(n^2)$ time.

— `TripodGreedyChoice()` will be called at most $O(n^3)$ times, since there are only $O(n^3)$ choices for 3-tuples of corners D_1, D_2, D_3 of P. Constructing the pseudo-diagonals can

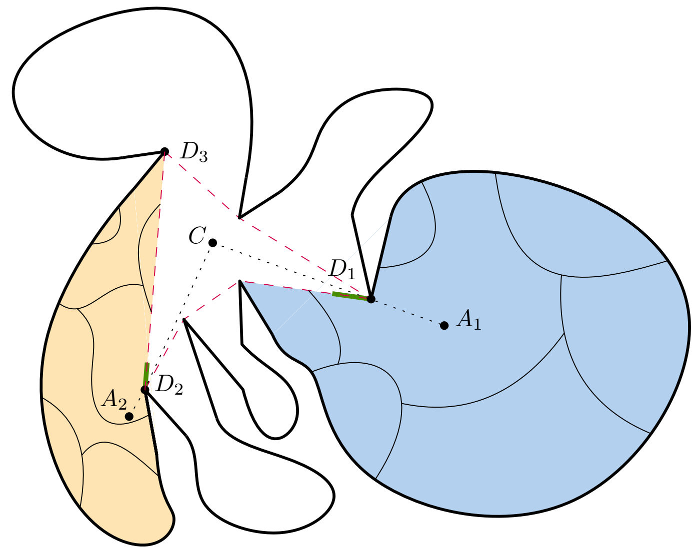

Figure 39. An illustration of how `TripodGreedyChoice`(D_1, D_2, D_3) works. First the pseudo-triangle Δ (dashed in red) is computed. Then it calls `SolveSubregion`() on the yellow and blue subpolygons of $P \setminus \Delta$, to recursively partition these optimally. Moreover, using Observation 5.5, we find all potential positions (in greedy-constructable optimal solutions) for star centers A_1 and A_2, whose piece contains a prefix (marked green in the figure) of the pseudo-diagonals adjacent to D_1 and D_2 (respectively). The pair which makes for the *least restrictive* (Definition 5.9) tripod point is chosen, and this point C is returned. Curved parts indicated that details have been omitted.

be done in $O(n)$ time. There are at most $O(n^6)$ possible positions for star centers, so only $O(n^{12})$ possible combinations for the greedy choice. For each of these combinations, we might need to go through the full polygon to see that the legs of the tripod does not intersect the polygon. Hence the computation inside `TripodGreedyChoice()` need in total $O(n^3 \cdot n^{12} \cdot n) = O(n^{16})$ time over the run of the full algorithm.

— `SolveSubregion()` will be called at most $O(n^2)$ times, since there are only $O(n^2)$ choices for pairs of corners D_1, D_2 of P. Enumerating valid tripod-positions can be done in $O(n^4)$ time ($O(n^3)$ many possible 3-tuples of corners, and each can be checked in $O(n)$ time by going through the polygon and constructing the pseudo-triangle to see if a valid tripod can be formed there). We then construct the set $S^{\text{(CENTERS)}}$ of size at most $O(n^6)$. Calling the dynamic programming algorithm on this set takes $O(n^{103})$ arithmetic operations and $O(n^{105})$ time (see Theorem 5.2 and Remark 5.3). Followed from Observation 5.5, in the same time, we can find all possible positions of star centers covering the start of the D_1-D_2 segment: indeed, we call the dynamic programming algorithm `SolveSeparator`(D_1, A, X) (where $X \neq D_1$ is the closest point to D_1 in $S^{\text{(BORDER)}}$ on the D_1-D_2 segment) for all possible star centers $A \in S^{\text{(CENTERS)}}$, to see which ones of these give a partition of minimum size. Note that between these calls to `SolveSeparator`, we do not need to reset the dp-cache, so in total this takes only $O(n^{103})$ arithmetic operations and $O(n^{105})$ time. Hence, in total, over

function TripodGreedyChoice(D_1, D_2, D_3):

▹ Returns the greedy choice tripod point of the tripod supported by corners D_1, D_2, D_3 of P. See Figure 39.

▹ Suppose, w.l.o.g. (other cases are similar), that we are like in Figure 6: that is the tripod should be oriented towards D_3 and the root is in the face fenced of by the D_2-D_3 pseudo-diagonal.

Construct the pseudo-diagonal D_1-D_2, say it goes through points $D_1 = X_1, X_2, ..., X_k = D_2$

Calculate the optimal number of star pieces to cover the sub-polygon defined by separator X_1-X_2, by calling SolveSubregion (X_1, X_2)

Additionally, this finds all possible positions, in greedy constructable optimal solutions, of star centers that see a small part of the segment X_1-X_2 next to D_1

Do the same for diagonal D_2-D_3

Look at all combinations of star centers and return only the single tripod-center that makes for the greedy choice (if any).

function SolveSubregion(D_1, D_2):

▹ Requires that D_1, D_2 is a diagonal of P. Call the sub-polygon on the right side (when looking from D_1 to D_2) of the diagonal P'.

▹ `SolveSubregion` (D_1, D_2) will optimally partition P' into a minimum number of star-shaped pieces. Moreover, it will consider all possible positions, in greedy constructable solutions, for the star center that sees a small part of the $D_1 D_2$ segment next to D_1.

Enumerate all valid positions of tripods (i.e. 3-tuples of concave corners) inside P', and call TripodGreedyChoice() on these.

Let L be the set of lines that are either (i) the line through two corners of P, or (ii) the line through some tripod-center and the corresponding concave corner of P.

Let $S^{(\text{centers})}$ be the set of intersection points of pairs of lines from L.

Call the dynamic programming algorithm on P' and $S^{(\text{centers})}$ to find an optimal solution.

By Observation 5.5 we can additionally find all possible positions of the star center seeing a prefix of the D_1-D_2 pseudo-diagonal from D_1 to D_2.

Algorithm 2. Minimum Star Partition Algorithm.

the full run of the algorithm, we will spend $O(n^2 \cdot n^{103}) = O(n^{105})$ arithmetic operations and $O(n^2 \cdot n^{105}) = O(n^{107})$ time in `SolveSubregion`.

The above discussion concludes the proof of Theorem 1.1.

References

[1] **Mikkel Abrahamsen**. Covering polygons is even harder. *62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 375–386, 2021. DOI (3)

[2] **Mikkel Abrahamsen**, **Anna Adamaszek**, and **Tillmann Miltzow**. The art gallery problem is ∃ℝ-complete. *J. ACM*, 69(1):4:1–4:70, 2022. DOI. Preliminary version at STOC 2018 (3)

[3] **Mikkel Abrahamsen**, **Joakim Blikstad**, **André Nusser**, and **Hanwen Zhang**. Minimum star partitions of simple polygons in polynomial time. *56th Annual ACM Symposium on Theory of Computing (STOC 2024)*, pages 904–910. ACM, 2024. (1)

[4] **Hugo A. Akitaya**, **Greg Aloupis**, **Jeff Erickson**, and **Csaba D. Tóth**. Recognizing weakly simple polygons. *Discret. Comput. Geom.* 58(4):785–821, 2017. DOI (13)

[5] **Esther M. Arkin**, **Rathish Das**, **Jie Gao**, **Mayank Goswami**, **Joseph S. B. Mitchell**, **Valentin Polishchuk**, and **Csaba D. Tóth**. Cutting polygons into small pieces with chords: laser-based localization. *28th Annual European Symposium on Algorithms (ESA 2020)*, 7:1–7:23, 2020. DOI (3)

[6] **Takao Asano**, **Tetsuo Asano**, and **Hiroshi Imai**. Partitioning a polygonal region into trapezoids. *J. ACM*, 33(2):290–312, 1986. DOI. Preliminary version af FOCS 1983. (3, 4)

[7] **Takao Asano**, **Tetsuo Asano**, and **Ron Y. Pinter**. Polygon triangulation: efficiency and minimality. *J. Algorithms*, 7(2):221–231, 1986. DOI (3, 12)

[8] **David Avis and Godfried T. Toussaint**. An efficient algorithm for decomposing a polygon into star-shaped polygons. *Pattern Recognit.* 13(6):395–398, 1981. DOI (2)

[9] **Chandrajit L. Bajaj and Tamal K. Dey**. Convex decomposition of polyhedra and robustness. *SIAM J. Comput.* 21(2):339–364, 1992. DOI (12)

[10] **Avisekh Banerjee**, **Hsi-Yung Feng**, and **Evgueni V. Bordatchev**. Process planning for floor machining of $2\frac{1}{2}$D pockets based on a morphed spiral tool path pattern. *Computers & Industrial Engineering*, 63(4):971–979, 2012. DOI (5)

[11] **Michael B. Bieterman and Donald R. Sandstrom**. A curvilinear tool-path method for pocket machining. *J. Manuf. Sci. Eng.* 125(4):709–715, 2003. DOI (5)

[12] **Maike Buchin and Leonie Selbach**. Decomposing polygons into fat components. *33rd Canadian Conference on Computational Geometry (CCCG 2021)*, pages 175–184, 2021. (3)

[13] **Hsien-Chih Chang**, **Jeff Erickson**, and **Chao Xu**. Detecting weakly simple polygons. *Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*, pages 1655–1670, 2015. DOI (13)

[14] **Bernard Chazelle**. A theorem on polygon cutting with applications. *23rd Annual Symposium on Foundations of Computer Science (FOCS 1982)*, pages 339–349, 1982. DOI (3)

[15] **Bernard Chazelle**. Approximation and decomposition of shapes. **Jacob T. Schwartz and Chee-Keng Yap**, editors, *Algorithmic and Geometric Aspects of Robotics*. Volume 1, Advances in Robotics, pages 145–185. 1987. (3, 11)

[16] **Bernard Chazelle**. Computational geometry and convexity. PhD thesis, Yale University, 1980. (2, 4)

[17] **Bernard Chazelle**. Convex partitions of polyhedra: A lower bound and worst-case optimal algorithm. *SIAM J. Comput.* 13(3):488–507, 1984. DOI (12)

[18] **Bernard Chazelle**. Triangulating a simple polygon in linear time. *Discret. Comput. Geom.* 6:485–524, 1991. DOI (3, 12)

[19] **Bernard Chazelle and David P. Dobkin**. Optimal convex decompositions. **Godfried T. Toussaint**, editor, *Computational Geometry*. Volume 2, Machine Intelligence and Pattern Recognition, pages 63–133. 1985. DOI. Preliminary version at STOC 1979 (3, 4, 6)

[20] **Bernard Chazelle and Leonidas Palios**. Decomposing the boundary of a nonconvex polyhedron. *Algorithmica*, 17(3):245–265, 1997. DOI (12)

[21] **Bernard Chazelle and Leonidas Palios**. Decomposition algorithms in geometry. **Chandrajit L. Bajaj**, editor, *Algebraic Geometry and its applications*, pages 419–447. 1994. DOI (3)

[22] **Vasek Chvátal**. A combinatorial theorem in plane geometry. *J. Comb. Theory, Ser. B*, 18(1):39–41, 1975. DOI (2)

[23] **Mirela Damian**. Exact and approximation algorithms for computing optimal fat decompositions. *Comput. Geom.* 28(1):19–27, 2004. DOI (3)

[24] **Mirela Damian and Sriram V. Pemmaraju**. Computing optimal diameter-bounded polygon partitions. *Algorithmica*, 40(1):1–14, 2004. DOI (3)

[25] **Mirela Damian-Iordache and Joseph O'Rourke**. Partitioning regular polygons into circular pieces I: convex partitions. *Proceedings of the 15th Canadian Conference on Computational Geometry (CCCG 2003)*, pages 43–46, 2003. URL (3)

[26] **Andrew Dobson and Kostas E Bekris**. Sparse roadmap spanners for asymptotically near-optimal motion planning. *The International Journal of Robotics Research*, 33(1):18–47, 2014. (6, 12)

[27] **Michal Etzion and Ari Rappoport**. On compatible star decompositions of simple polygons. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):87–95, 1997. DOI (6)

[28] **Steven N. Evans**. Probability and real trees, number 1920 in Lecture notes in mathematics. Springer, 2008. DOI (61)

[29] **Hou-Yuan F. Feng and Theodosios Pavlidis**. Decomposition of polygons into simpler components: feature generation for syntactic pattern recognition. *IEEE Trans. Computers*, 24(6):636–650, 1975. DOI (3, 12)

[30] **Steve Fisk**. A short proof of chvátal's watchman theorem. *J. Comb. Theory, Ser. B*, 24(3):374, 1978. DOI (2)

[31] **M. R. Garey**, **David S. Johnson**, **Franco P. Preparata**, **and Robert Endre Tarjan**. Triangulating a simple polygon. *Inf. Process. Lett.* 7(4):175–179, 1978. DOI (3)

[32] **Stefan Hertel and Kurt Mehlhorn**. Fast triangulation of the plane with respect to simple polygons. *Inf. Control.* 64(1-3):52–76, 1985. DOI (3)

[33] **J. Mark Keil**. Decomposing a polygon into simpler components. *SIAM J. Comput.* 14(4):799–817, 1985. DOI (2, 3, 5, 6, 12)

[34] **J. Mark Keil**. Polygon decomposition. **Jörg-Rüdiger Sack and Jorge Urrutia**, editors, *Handbook of computational geometry*, chapter 11, pages 491–518. 1999. DOI (3)

[35] **J. Mark Keil and Jörg-R. Sack**. Minimum decompositions of polygonal objects. **Godfried T. Toussaint**, editor, *Computational Geometry*. Volume 2, Machine Intelligence and Pattern Recognition, pages 197–216. 1985. DOI (3)

[36] **J. Mark Keil and Jack Snoeyink**. On the time bound for convex decomposition of simple polygons. *Int. J. Comput. Geom. Appl.* 12(3):181–192, 2002. DOI (3)

[37] **Jyh-Ming Lien**. Approximate Star-Shaped Decomposition of Point Set Data. *Eurographics Symposium on Point-Based Graphics (PBG 2007)*, 2007. DOI (6, 12)

[38] **Jyh-Ming Lien and Yanyan Lu**. Planning motion in point-represented contact spaces using approximate star-shaped decomposition. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pages 5420–5425, 2009. DOI (6, 12)

[39] **Robin Liu and Simeon C. Ntafos**. On decomposing polygons into uniformly monotone parts. *Inf. Process. Lett.* 27(2):85–89, 1988. DOI (3)

[40] **Robon Liu and Simeon C. Ntafos**. On partitioning rectilinear polygons into star-shaped polygons. *Algorithmica*, 6(6):771–800, 1991. DOI (3, 4)

[41] **Anna Lubiw**. Decomposing polygonal regions into convex quadrilaterals. *First Annual Symposium on Computational Geometry (SoCG 1985)*, pages 97–106, 1985. DOI (3)

[42] **Matthias Müller-Hannemann and Karsten Weihe**. Minimum strictly convex quadrangulations of convex polygons. *Thirteenth Annual Symposium on Computational Geometry (SoCG 1997)*, pages 193–202, 1997. DOI (3)

[43] **Joseph O'Rourke**. Art Gallery Theorems and Algorithms. Oxford University Press, 1987. (2, 3)

[44] **Joseph O'Rourke and Kenneth J. Supowit**. Some np-hard polygon decomposition problems. *IEEE Trans. Inf. Theory*, 29(2):181–189, 1983. DOI (2)

[45] **Joseph O'Rourke**, **Subash Suri**, **and Csaba D. Tóth**. Polygons. **J. E. Goodman and J. O'Rourke**, editors, *Handbook of discrete and computational geometry*, chapter 30, pages 787–810. third edition edition, 2018. DOI (3)

[46] **Divyangkumar D. Patel and Devdas I. Lalwani**. Effect of algorithm parameters in development of spiral tool path for machining of 2.5D star-shaped pockets. *International Journal of Computer Aided Engineering and Technology*, 11(6):727–746, 2019. DOI (5)

[47] **Bruce J. Schachter**. Decomposition of polygons into convex sets. *IEEE Trans. Computers*, 27(11):1078–1082, 1978. DOI (3)

[48] **Michal Shapira and Ari Rappoport**. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, 1995. DOI (6)

[49] **Thomas C. Shermer**. Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399, 1992. DOI (2, 3)

[50] **Godfried T. Toussaint**. Computational geometric problems in pattern recognition. *Pattern Recognition Theory and Applications: Proceedings of the NATO Advanced Study Institute*, pages 73–91, 1982. DOI (2)

[51] **Marc J. van Kreveld**. On fat partitioning, fat covering and the union size of polygons. *Comput. Geom.* 9(4):197–210, 1998. DOI (3)

[52] **Gokul Varadhan and Dinesh Manocha**. Star-shaped roadmaps – a deterministic sampling approach for complete motion planning. *Robotics: Science and Systems (RSS 2005)*, pages 25–32, 2005. DOI (5, 12)

[53] **Jiazhi Xia**, **Ying He**, **Shuchu Han**, **Chi-Wing Fu**, **Feng Luo**, **and Xianfeng Gu**. Parameterization of star-shaped volumes using Green's functions. *Advances in Geometric Modeling and Processing (GMP 2010)*, pages 219–235, 2010. DOI (6)

[54] **Wuyi Yu and Xin Li**. Computing 3D shape guarding and star decomposition. *Computer Graphics Forum*, 30(7):2087–2096, 2011. DOI (6)

[55] **Liangjun Zhang**, **Young J. Kim**, **Gokul Varadhan**, **and Dinesh Manocha**. Fast C-obstacle query computation for motion planning. *IEEE International Conference on Robotics and Automation (ICRA 2006)*, pages 3035–3040, 2006. DOI (6, 12)

A. Existence of Coordinate and Area Maximum Partitions

PROOF OF LEMMA 2.3. Recall that the *Hausdorff distance* between two compact sets $A, B \subset \mathbb{R}^2$ is defined as $d_H(A, B) := \inf\{r \geq 0 \mid A \subset B \oplus D(r) \text{ and } B \subset A \oplus D(r)\}$, where $\oplus$ is the Minkowski sum and $D(r)$ is the disk of radius r centered at the origin. We will use the fact that $(M(\mathbb{R}^2), d_H)$ is a compact metric space, where $M(\mathbb{R}^2)$ denotes the set of all non-empty closed subsets of $\mathbb{R}^2$; see for instance [28, Theorem 4.5].

Let $c^* = \sup c(Q)$, where the supremum is taken over all optimal star partitions Q of P. We claim there exists a star partition which realizes the coordinate vector c^*. Consider a sequence of optimal star partitions $(Q_i)_{i\in\mathbb{N}}$ so that $c(Q_i)$ converges to c^* as $i \longrightarrow \infty$. Let $c(Q_i) = \langle A_{1i}, \ldots, A_{ki}\rangle$ and let the pieces of Q_i be $Q_{1i}, \ldots, Q_{ki}$ so that the maximum star center of Q_{ji} is A_{ji}. Be passing to a subsequence, we can assume that the sequence $(Q_{1i})_{i\in\mathbb{N}}$ converges to a compact set Q_1^* with respect to Hausdorff distance. Similarly, we can assume that $(Q_{ji})_{i\in\mathbb{N}}$ converges to Q_j^* for each $j \in \{1, \ldots, k\}$. Let $Q^* = \{Q_1^*, \ldots, Q_k^*\}$ and $c^* = \langle A_1^*, \ldots, A_k^*\rangle$. We claim that Q^* is a star partition of P where A_j^* is the star center of Q_j^*.

To see that Q_j^* is star-shaped with star center A_j^*, we first observe that $A_j^* \in Q_j^*$. Otherwise, the star center A_{ji} would not be in Q_{ji} for sufficiently large values of i. Suppose now that A_j^* is not a star center of Q_j^*. This means that there is a point $B \in Q_j^*$ so that the line segment A_j^*B is not in Q_j^*. For each $i \in \mathbb{N}$, we can choose $B_i \in Q_{ji}$ so that $B_i \longrightarrow B$ for $i \longrightarrow \infty$. Since $A_{ji} \longrightarrow A_j^*$, $B_i \longrightarrow B$ and A_j^*B is not in Q_j^*, it follows that $A_{ji}B_i$ is not in Q_{ji} for sufficiently high values of i, which contradicts that A_{ji} is a star center of Q_{ji}. In a similar way, we can argue that the sets in Q^* are interior-disjoint; otherwise, the sets in Q_i could not be interior-disjoint for sufficiently high values of i. Likewise, we get that $\bigcup Q^* = P$, and we conclude that Q^* is a coordinate maximum optimal star partition of P. ■

PROOF SKETCH OF LEMMA 2.4. The proof is analogous to that of Lemma 2.3. Without loss of generality, we can assume that $d = (1, 0)$, so that for any set of points $A'_i, \ldots, A'_k$, we have

$$\langle A'_i \cdot d, A'_i \cdot d^\perp, A'_{i+1} \cdot d, A'_{i+1} \cdot d^\perp, \ldots, A'_k \cdot d, A'_k \cdot d^\perp\rangle = \langle A'_i, A'_{i+1}, \ldots, A'_k\rangle.$$

Define the supremum $\langle A_i^*, A_{i+1}^*, \ldots, A_k^*\rangle = \sup\langle A'_i, A'_{i+1}, \ldots, A'_k\rangle$ over all partitions with star centers $A'_i, A'_{i+1}, \ldots, A'_k$ in the region F and the rest fixed at the points $A_1, \ldots, A_{i-1}$. We then consider a convergent sequence of star partitions with fixed star centers $A_1, \ldots, A_{i-1}$ and the rest converging to $A_i^*, \ldots, A_k^*$, and it follows that the limit of the pieces constitute a partition realizing the supremum. ■

PROOF OF LEMMA 2.5. The proof is similar to that of Lemma 2.3. Namely, let $a^* = \sup a(Q)$, where the supremum is taken over all star partitions Q of P with star centers $\mathcal{A}$. We then consider a sequence of partitions $(Q_i)_{i\in\mathbb{N}}$ with star centers $\mathcal{A}$ so that $a(Q_i) \longrightarrow a^*$ as $i \longrightarrow \infty$.

Let the pieces of Q_i be $Q_{1i}, \dots, Q_{ki}$ so that A_j is a star center of Q_{ji}. By passing to a subsequence, we can assume that $(Q_{ji})_{i\in\mathbb{N}}$ converges to a compact set Q_j^* for each $j \in \{1, \dots, k\}$.

As in the proof of Lemma 2.3, we can conclude that $Q^* = \{Q_1^*, \dots, Q_k^*\}$ is a star partition of P with star centers $\mathcal{A}$ and that $a(Q^*) = a^*$, so Q^* is area maximum. ■

B. Structural Theorem

In this section, we give an elementary and independent proof of a structural theorem about optimal star partitions, which is not used in the rest of the paper. A point on the boundary of the polygon P is *canonical* if it is a corner of P or the endpoint of an extension of an edge of P; see Figure 40 (left).

THEOREM B.1. *Let k be minimum such that there exists a star partition of P consisting of k polygons and assume $k \geq 2$. There exists a star partition $Q_1, \dots, Q_k$ of P, where each piece Q_i has the following properties:*

1. *∂Q_i contains a concave corner of P, and*
2. *for each connected component of the shared boundary $\partial P \cap \partial Q_i$, both endpoints are canonical.*

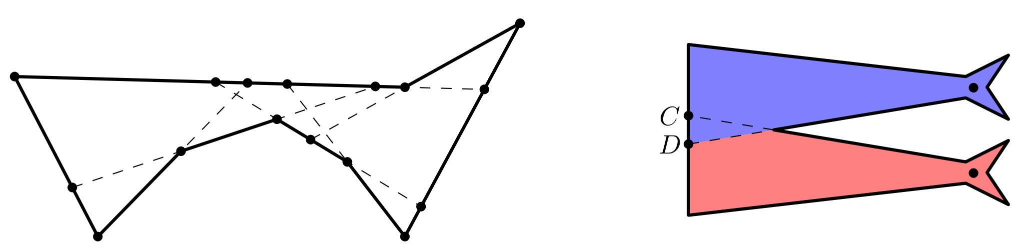

Figure 40. Left: A polygon with the extensions of the edges and the canonical points shown. Right: Using only canonical points as corners of pieces on the boundary of P, one of the points C and D must be used.

Before going into the proof, let us first make a few remarks. Note that there are at most $3n$ canonical points, since each edge of P creates at most 2 canonical points that are not corners of P. The algorithm described in this paper considers $O(n^{13})$ Steiner points on the boundary of P, so it might be possible to use property 2 to obtain a faster algorithm. However, in the proof of the theorem, we change the partition in order to only use canonical Steiner points on the boundary. It does therefore not follow immediately that our algorithm can also be modified to find the resulting partition, but we are confident that the result can be used to create a faster algorithm.

We find it somewhat surprising that canonical Steiner points suffice on the boundary of P, since, as shown in Figure 1, it is sometimes necessary to use Steiner points in the interior of P

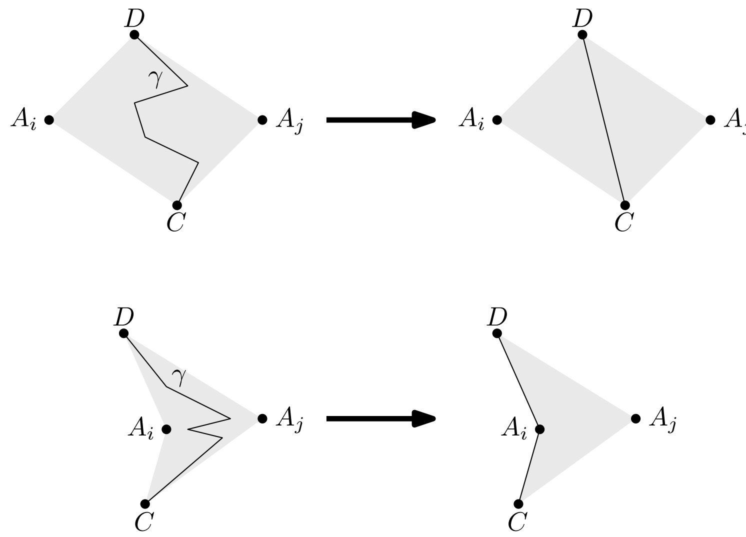

Figure 41. The process simplifying the shared boundary between Q_i and Q_j in order to satisfy Claim B.2. The gray region is the quadrilateral F.

of degree $\Omega(n)$, whereas the canonical points have degree at most 1 (as defined in Section 1). Figure 40 (right) proves that it is necessary to have least some Steiner points on the boundary of P: In any partition into two star-shaped pieces, we must have a Steiner point on the segment CD.

PROOF. Let $Q_1, \ldots, Q_k$ be a minimum star partition of P. Let us fix a star center A_i in each piece Q_i. If $\partial Q_i \cap \partial Q_j \neq \emptyset$ and $i \neq j$, we say that Q_i and Q_j are *neighbours*. The shared boundary $\partial Q_i \cap \partial Q_j$ of two neighbours Q_i and Q_j is a collection of open polygonal curves. An *interior point* of an open curve is a point on the curve which is not an endpoint.

CLAIM B.2. *Let γ be an open curve in the shared boundary $\partial Q_i \cap \partial Q_j$ of two neighbours Q_i and Q_j. We can assume that γ is either a line segment or two line segments that have one of the star centers A_i and A_j as a common endpoint. If a star center of one piece is a corner of γ, the corner is convex with respect to that piece and concave with respect to the other piece.*

Proof. Let the endpoints of γ be C and D; see Figure 41. Since the segments A_iC and A_iD are in Q_i, and A_jC and A_jD are in Q_j, we have a well-defined quadrilateral $F = A_iCA_jD$. If A_i and A_j are both convex corners of F, we replace γ by the segment CD, which must be a diagonal of F. Otherwise, consider without loss of generality the case that A_i is a concave corner of F. We then replace γ by $CA_i \cup A_iD$. In either case, the modification clearly leaves Q_i and Q_j star-shaped, and we are left with a shared boundary of the claimed type. ◆

Property 1. In order to prove that there is a partition satisfying Property 1, suppose that ∂Q_i does not contain a concave corner of P. We show how to modify the partition so that the property is eventually satisfied. In essence, we expand the piece Q_i until Property 1 is eventually satisfied.

CLAIM B.3. *We can assume that each concave corner D of Q_i is a concave corner of P or a star center A_j of a neighbour Q_j of Q_i.*

Proof. We describe a way to expand Q_i so that we eliminate concave corners of Q_i which are neither corners of P nor star centers of neighbours (note that if a concave corner of Q_i touches P, then it touches P at a concave corner). Let CD and DE be maximal segments on the boundary of Q_i such that D is a concave corner of Q_i and no corner of P and no star center of a neighbour is an interior point of $CD \cup DE$. It follows by Claim B.2 that this point D can be assumed to lie on the boundary of two polygons Q_j and Q_l whose boundaries share a line segment DF. Informally, we now move D towards F. This will expand Q_i and shrink Q_j and Q_l and possibly also other neighbours of Q_i whose boundaries contain segments or points on $CD \cup DE$. To be precise, define D' to be the first point on DF from D such that one of the following cases holds: (i) one of the segments CD' or $D'E$ contains a corner of P, (ii) D' is the intersection of DF and CE (if it exists), (iii) one of the segments CD' or $D'E$ contains a star center A_m, $m \neq i$, (iv) $D' = F$. The cases are shown in Figure 42. We then assign the quadrilateral $CD'ED$ to Q_i, which will increase Q_i, decrease the pieces intersected by $CD' \cup D'E$ and, by Lemma 2.1, all the involved pieces remain star-shaped. We repeat this operation, and it remains to argue that the process eventually terminates.

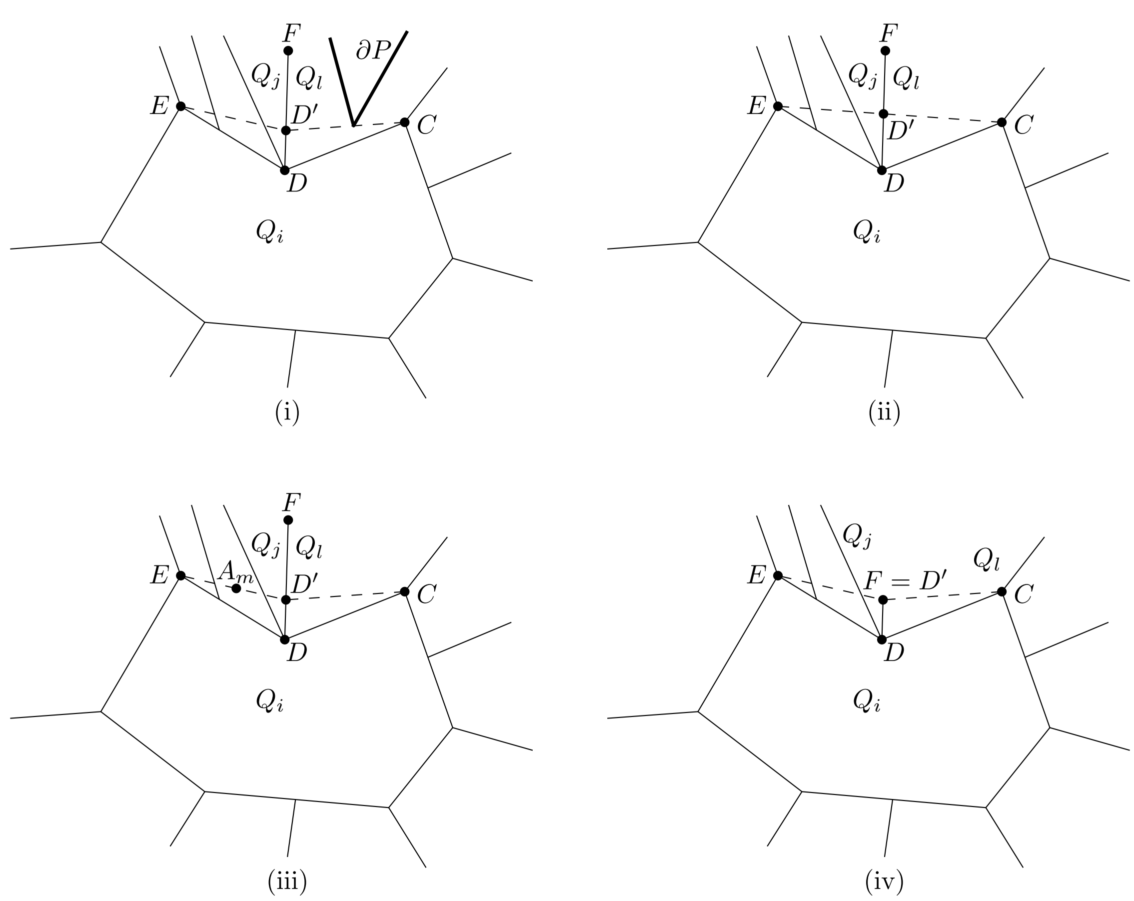

Figure 42. The process of eliminating concave corners of Q_i that are not star centers of neighbouring pieces.

In case (i), Q_i now touches P at a corner which clearly must be concave. In case (ii), we have eliminated a concave corner of Q_i. In case (iii), we have increased the number of star centers on the boundary of Q_i, which can happen at most $k-1$ times. In case (iv), we eliminated a segment DF of Q_j and Q_l, which decreases the total number of segments of the pieces. We conclude that the operation can be repeated at most a finite number of times and the process therefore eventually terminates. ◆

A *star neighbour* of Q_i is a neighbour Q_j whose star center A_j is on ∂Q_i. Recall that we assume ∂Q_i does not contain a concave corner of P. Now, additionally assume that Q_i is non-convex, i.e., Q_i has a concave corner. As ∂Q_i does not contain a concave corner of P, we can assume that this concave corner is the star center of a star neighbour by Claim B.3. To obtain a contradiction with the minimality of the start partition, we show that Q_i can be subsumed by the star neighbours. To this end, we consider a triangulation of Q_i. The diagonals of the triangulation that have an endpoint at a concave corner of Q_i partition Q_i into convex polygons $R_1, \ldots, R_m$, as illustrated in Figure 43 (left), where solid diagonals have an endpoint at a concave corner. We assign a polygon R_p to Q_j if A_j is on the boundary of R_p and R_p has not already been

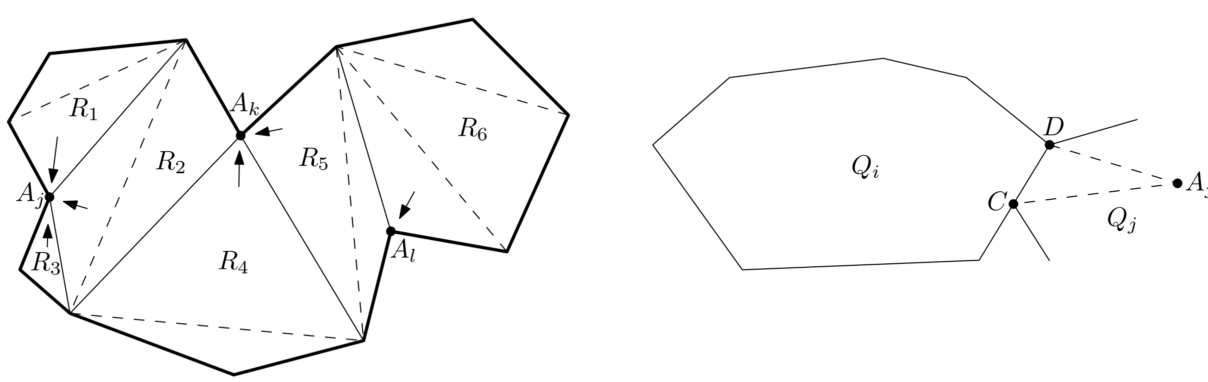

Figure 43. Left: We reassign the piece Q_i to the pieces of the concave star neighbours. Right: We expand the convex piece Q_i by adding the triangle A_jCD.

assigned to another star neighbour, as shown by the arrows in the figure. Thus, the considered star partition did not consist of a minimum number of polygons, which is a contradiction. Consequently, Q_i must be convex.

As we assume $k \geq 2$, there has to be a piece Q_j that is a neighbour of Q_i so that the common boundary $\partial Q_i \cap \partial Q_j$ contains a segment CD. We expand Q_i and shrink Q_j by adding the triangle A_jCD to Q_i, as shown in Figure 43 (right), which keeps Q_i as well as Q_j star-shaped. This can introduce concave corners C and D on Q_i, but we can proceed as in the proof of Claim B.3 and expand Q_i until we hit a concave corner of P or obtain that all concave corners are star centers. If we do not hit a concave corner of P, we can again argue that Q_i can be subsumed by Q_j and potentially other star neighbours, contradicting the minimality of the star partition.

Note that if a concave corner of P appears on the boundary of a piece before the changes described above (including in the proof of Claim B.3), then the corner also appears on the piece afterwards. We conclude that we can expand each polygon Q_i that does not contain a concave corner of P until it eventually does. In the end, we obtain a minimum star partition with Property 1.

Property 2. We now prove that we can obtain Property 2. Consider an edge CD of P. For each piece Q_i, it holds by the optimality of the partition that the intersection $CD \cap \partial Q_i$ is either empty or a single segment EF (which may be a single point); see Figure 44. We call such a segment EF a *shared* segment, and we say that a shared segment is *canonical* if both endpoints are canonical. We define the *canonical prefix* of CD to be all the shared segments from C to (and excluding) the first non-canonical shared segment. We show that we can modify the partition such that the number of segments in the canonical prefix increases. It therefore holds that the process must stop, so that all shared segments on CD have canonical endpoints in the end. The process does not change whether shared segments on other edges of P are canonical, so repeating the process for all edges yields a star partition with Property 2.

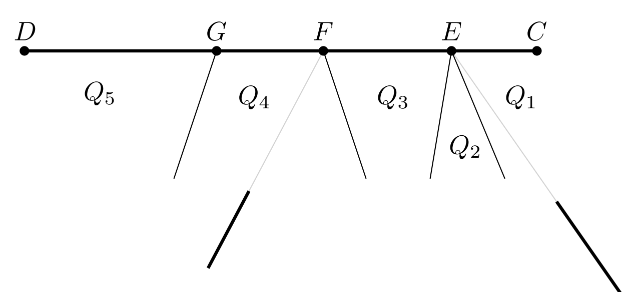

Figure 44. The fat segments are edges of P and the thin black segments indicate the boundaries of pieces in the interior of P. The corners C and D as well as the interior points E and F are canonical, but G is not. There are three segments in the canonical prefix of CD, namely the intersections of CD with each of $\partial Q_1, \partial Q_2, \partial Q_3$.

Consider the first non-canonical segment EF, where E is canonical but F is not. Let F' be the first point on FD from F such that either (i) F' is canonical, (ii) A_iF' contains another star center A_j, or (iii) A_iF' contains a corner G of P. Since the endpoint D is canonical, the point F' is well-defined. The cases are illustrated in Figures 45 and 47 and Figure 46.

In case (i), we assign the triangle A_iFF' to Q_i, which according to Lemma 2.1 keeps all pieces star-shaped as the triangle does not contain any star centers of other pieces than Q_i. We have then increased the number of segments in the canonical prefix.

In case (ii), we assign the triangle A_iEF' to the piece Q_j. The shared segment $s = CD \cap \partial Q_j$ of the piece Q_j now starts at the point E. The other endpoint of s is either F' (as in Figure 45) or a later point. In the latter case, it is possible that s is canonical, and we have increased the number of segments in the canonical prefix. If s is not canonical, we repeat the process of repairing the non-canonical endpoint of s. Since the star center A_j must be closer to the edge CD than A_i, we encounter case (ii) less than k times before we end in case (i) or (iii).

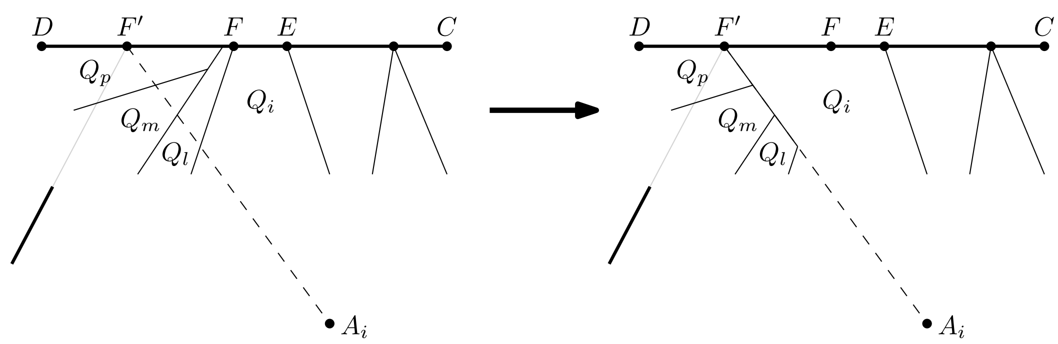

Figure 45. Case (i). We expand Q_i with the triangle A_iFF'.

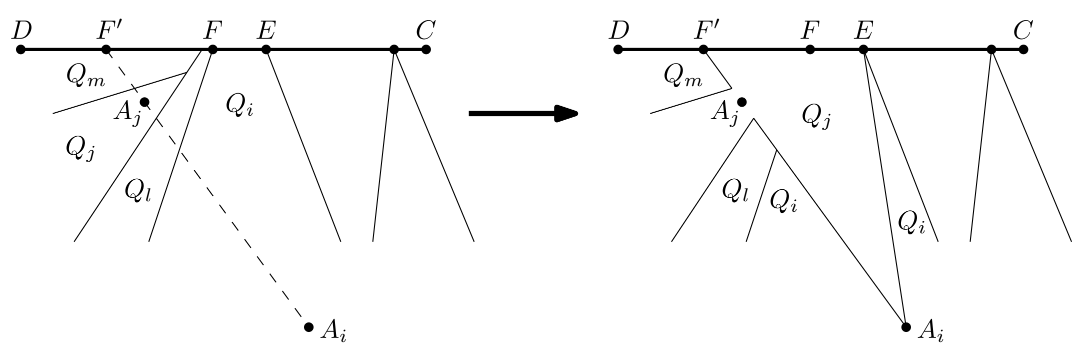

Figure 46. Case (ii). We expand Q_j with the triangle A_iFF'.

It remains to consider case (iii), namely that A_iF' contains a corner G of P. We know that $G \neq D$, since otherwise, we would have been in case (i). Let F'' be the first canonical point on the segment $F'E$ from F', which is well-defined as E is canonical. Furthermore, we know that F'' is on the segment EF, since otherwise, we should have been in case (i). Let $Q' = \{Q'_1, \ldots, Q'_{f'}\} \setminus \{Q_i\}$ be the set of pieces intersected by the interior of GF' in order from G and excluding Q_i.

In a first step, we assign the triangle $A_iF''F'$ to the piece Q_i. This can reduce pieces intersecting A_iF', but Lemma 2.1 ensures that they remain star-shaped. However, the point F' need not be canonical. In a second step, we therefore remove the triangle $\Delta = GF''F'$ from A_i and instead distribute Δ among the pieces Q', as follows. For each $j = 1, \ldots, f' - 1$, we consider the segment s on the shared boundary of Q'_j and Q'_{j+1} with an endpoint on GF'. We then extend s into Δ until we reach one of the other segments GF'' or $F''F'$ bounding Δ, or we meet an extension that was already added for a smaller value of j. Since F'' was chosen as the first canonical point on $F'E$, this results in a star partition of P. Furthermore, the shared segment of Q_i on CD has now become the segment EF'' (which might just be a single point), which is canonical, so we have increased the number of segments in the canonical prefix. Recall that we consider the number of segments in the prefix here and not the geometric length of the prefix; the geometric length of the prefix indeed can decrease.

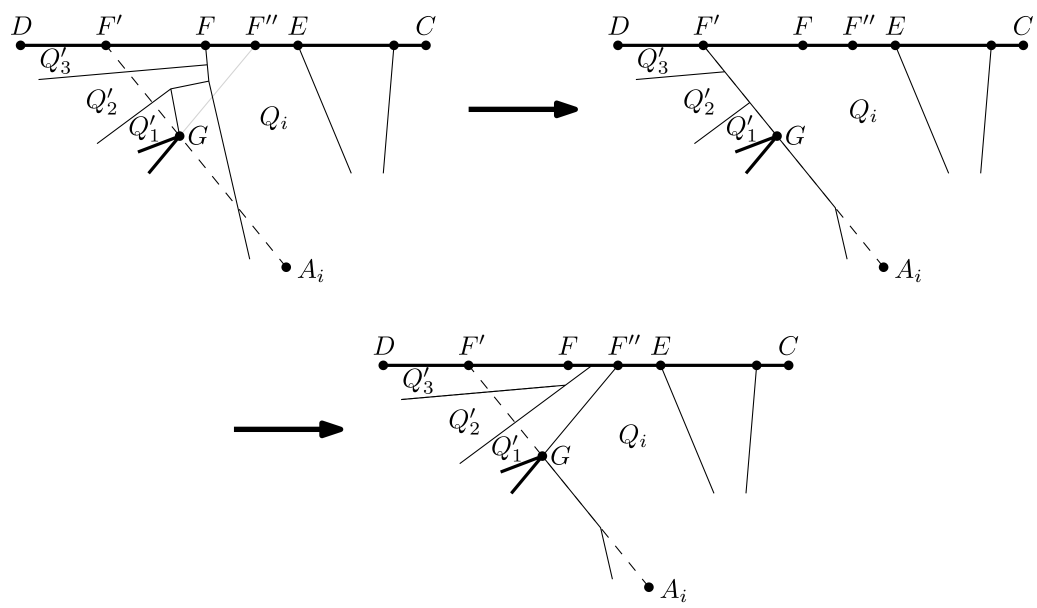

Figure 47. Case (iii). In two steps, we distribute the triangle $A_iF''F'$ among Q_i and the pieces Q_1'', Q_2'', Q_3'' intersected by GF'.

Finally, note that none of the above modifications of the star partition cause a piece Q to become non-adjacent to a concave corner H of P if Q was adjacent to H before. More precisely, only the modifications in case (iii) changes the neighbourhood of a corner G of P. However, as there are no star centers in the triangle A_iEF', the modifications will not remove G from the boundary of any piece. ∎

This work is licensed under the Creative Commons Attribution 4.0 International License.
http://creativecommons.org/licenses/by/4.0/
© Mikkel Abrahamsen, Joakim Blikstad, André Nusser, Hanwen Zhang.