International Journal of Computational Geometry & Applications

BOTTLENECK CONVEX SUBSETS: FINDING K LARGE CONVEX SETS IN A POINT SET

STEPHANE DUROCHER

Department of Computer Science,
University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada
durocher@cs.umanitoba.ca

J. MARK KEIL

Department of Computer Science,
University of Saskatchewan, Saskatoon, Canada, Saskatchewan S7N 5C9, Canada
keil@cs.usask.ca

SAEED MEHRABI

School of Computer Science,
Carleton University, Ottawa, Ontario K1S 5B6, Canada

DEBAJYOTI MONDAL

Department of Computer Science,
University of Saskatchewan, Saskatoon, Canada, Saskatchewan S7N 5C9, Canada
dmondal@cs.usask.ca

Received 24 February 2022
Revised 27 September 2022
Communicated by Chung-Shou Liao, Guest Editor

ABSTRACT

Chvátal and Klincsek (1980) gave an $O(n^3)$-time algorithm for the problem of finding a maximum-cardinality convex subset of an arbitrary given set P of n points in the plane. This paper examines a generalization of the problem, the *Bottleneck Convex Subsets* problem: given a set P of n points in the plane and a positive integer k, select k pairwise disjoint convex subsets of P such that the cardinality of the smallest subset is maximized. Equivalently, a solution maximizes the cardinality of k mutually disjoint convex subsets of P of equal cardinality. We give an algorithm that solves the problem exactly, with running time polynomial in n when k is fixed. We then show the problem to be NP-hard when k is an arbitrary input parameter, even for points in general position. Finally, we give a fixed-parameter tractable algorithm parameterized in terms of the number of points strictly interior to the convex hull.

Keywords: Convex set; NP-hard; FPT-algorithms.

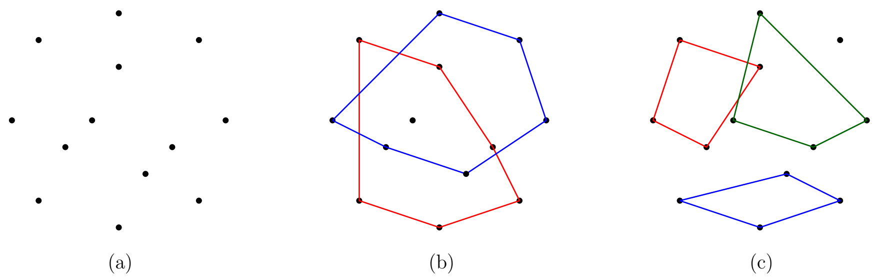

Fig. 1. (a) A point set P. (b) A solution to the Bottleneck Convex Subsets problem when $k = 2$. (c) A solution when $k = 3$.

1. Introduction

A set P of points in the plane is *convex* if for every $p \in P$ there exists a closed half-plane H^+ such that $H^+ \cap P = \{p\}$. Determining whether a given set P of n points in the plane is convex requires $\Theta(n \log n)$ time in the worst case, corresponding to the time required to determine whether the convex hull of P has n vertices on its boundary [19]. Chvátal and Klincsek [4] gave an $O(n^3)$-time and $O(n^2)$-space algorithm to find a maximum-cardinality convex subset of any given set P of n points in the plane. Later, Edelsbrunner and Guibas [8] improved the space complexity to $O(n)$. In this paper, we examine a generalization of the problems to multiple convex subsets of P. Given a set P of points in the plane and a positive integer k, we examine the problem of finding k convex and mutually disjoint subsets of P, such that the cardinality of the smallest set is maximized (e.g., see Figure 1). We define the problem formally, as follows.

BOTTLENECK CONVEX SUBSETS
Instance: A set P of n points in $\mathbb{R}^2$, and a positive integer k.
Problem: Select k sets $P_1, \ldots, P_k$ such that

- $\forall i \in \{1, \ldots, k\}$, $P_i \subseteq P$,
- $\forall i \in \{1, \ldots, k\}$, P_i is convex,
- $\forall \{i, j\} \subseteq \{1, \ldots, k\}$, $i \neq j \Rightarrow P_i \cap P_j = \varnothing$, and
- $\min\limits_{i \in \{1,\ldots,k\}} |P_i|$ is maximized.

Since every subset of a convex set of points remains convex, any k convex sets can be made to have equal cardinality by removing points from any set whose cardinality exceeds that of the smallest set. Therefore, an equivalent problem is to find k mutually disjoint convex subsets of P of equal cardinality, where the cardinality is maximized.

1.1. *Our contributions*

In this paper we examine the problem of finding k large convex subsets of a given point set with n points. Our contributions are as follows:

(1) We give a polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k. The algorithm constructs a directed acyclic graph G whose vertices correspond to distinct configurations of edges passing through vertical slabs between neighbouring points of P. A solution to the problem is found by identifying a node in G associated with a maximum-cardinality set that is reachable from the source node.
(2) Using a reduction from a restricted version of Numerical 3-Dimensional Matching, which is known to be NP-complete, we show that Bottleneck Convex Subsets is NP-hard when k is an arbitrary input parameter.
(3) We show that Bottleneck Convex Subsets is fixed-parameter tractable when parameterized by the number of points that are strictly interior to the convex hull of the given point set, i.e., the number of non-extreme points. Therefore, if the number of points interior to the convex hull is fixed, then for every k, Bottleneck Convex Subsets can be solved in polynomial time.

1.2. *Related work*

A *convex k-gon* is a convex set with k points. A convex *k-hole* within a set P is a convex k-gon on a subset of P whose convex hull is empty of any other points of P. A rich body of research examines convex k-holes in point sets [22]. By the Erdős-Szekeres theorem [12], every point set with n points in the Euclidean plan contains a convex k-gon for some $k \in \Omega(\log n)$. Urabe [23] showed that by repeatedly extracting such a convex $\Omega(\log n)$-gon, one can partition a point set into $O(n/\log n)$ convex subsets, each of size $O(\log n)$.

Given a set P of n points in the plane, there exist $O(n^3)$-time algorithms to compute a largest convex subset of P [4,8] and a largest empty convex subset of P [2]. Both problems are NP-hard in $\mathbb{R}^3$ [15]. In fact, finding a largest empty convex subset is W[1]-hard in $\mathbb{R}^3$ [15]. González-Aguilar et al. [16] have recently examined the problem of finding a largest convex set in the rectilinear setting.

The *convex cover number* of a point set P is the minimum number of disjoint convex sets that covers P. The *convex partition number* of a point set P is the minimum number of convex sets with disjoint convex hulls (in addition to their vertex sets being pairwise vertex disjoint) that covers P. Urabe [23] examined lower and upper bounds on the convex cover number and the convex partition number. He showed that the convex cover number of a set of n points in $\mathbb{R}^2$ is in $\Theta(n/\log n)$ and its convex partition number is bounded from above by $\lceil \frac{2n}{7} \rceil$. Furthermore, there exist point sets with convex partition number at least $\lceil \frac{n-1}{4} \rceil$.

Arkin et al. [1] proved that both finding the convex cover number and the convex partition number of a point set are NP-hard problems, and gave a polynomial-time $O(\log n)$-approximation algorithm for both problems. Although the Bottleneck Convex Subsets problem appears to be similar to the convex cover number problem as both problems attempt to find disjoint convex sets, the objective functions are different. Neither the NP-hardness proof nor the approximation result for convex cover number [1] readily extends to the Bottleneck Convex Subsets problem. Previous work has also considered partitioning a point set into empty convex sets, where the convex hulls of the sets do not contain any interior point. For the number of empty convex point sets, an upper bound of $\lceil \frac{9n}{34} \rceil$ and a lower bound of $\lceil \frac{n+1}{4} \rceil$ is known [5]. We refer the readers to [10,11] for related problems on finding convex sets with various optimization criteria.

Another related problem in this context is to partition a given point set using a minimum number of lines (Point-Line-Cover), which Megiddo and Tamir [21] showed to be NP-hard, and was subsquently shown to be APX-hard [3,20]. Point-Line-Cover is known to be fixed-parameter tractable when parameterized on the number of lines. Whether the minimum convex cover problem is fixed-parameter tractable remains an open problem [9]. Note that for any fixed k, one can decide whether the minimum convex cover number of a point set is at most k in polynomial time [1].

Previous work on the Ramsey-remainder problem provides insight into the Bottleneck Convex Subsets problem [13]. Given an integer i, the Ramsey-remainder is the smallest integer $rr(i)$ such that for every sufficiently large point set, all but at most $rr(i)$ points can be partitioned into convex sets of size at least i. Therefore, a Bottleneck Convex Subsets problem with sufficiently large n and with $k \leq \lfloor \frac{n-rr(k)}{k} \rfloor$ must have a solution where the size of the $(rr(k)+1)$th smallest convex set is at least k. Note that the Bottleneck Convex Subsets problem is straightforward to solve for the case when $k \geq n/3$, i.e., one needs to compute a balanced partition without worrying about the convexity of the sets. However, the case when $k = n/4$ already becomes nontrivial. Károlyi [18] derived a necessary and sufficient condition for a set of $4n$ points in general position to admit a partition into n convex quadrilaterals, and gave an $O(n \log n)$-time algorithm to decide whether such a partition exists.

2. A Polynomial-Time Algorithm for a Fixed k

Given a set P of n points in the plane and a fixed integer k, we describe an $O(kn^{5k+3})$-time algorithm that solves Bottleneck Convex Subsets for any fixed k.

Here we give an outline of the algorithm. The idea is to construct a directed acyclic graph G each of whose vertices corresponds to a vertical slab of the plane in a given state with respect to the selected subsets $P_1, \ldots, P_k$ of P, with an edge from one slab to the slab immediately to its right if the states of the two neighbouring slabs form a locally mutually compatible solution. The notion of compatibility depends on whether the union of the convex regions intersected by these two slabs is also convex. A feasible solution ($P_1, \ldots, P_k$ are mutually disjoint convex subsets of P) corresponds to a directed path starting at the root node in G, i.e., a sequence of consecutive compatible slabs. Among the feasible solutions, an optimal solution

($\min_{i\in\{1,\ldots,k\}} |P_i|$ is maximized) corresponds to a path that ends at a node for which the cardinality of the smallest set is maximized.

We now describe the details of the algorithm. Rotate P such that no two of its points lie on a common vertical line. Partition the plane into $n-1$ vertical slabs, $S_1,\ldots,S_{n-1}$, determined by the n vertical lines through points of P. Let L be the set of $\binom{n}{2}$ line segments whose endpoints are pairs of points in P. Within each slab, S_i, consider the set of line segments $L_i = \{l \cap S_i \mid l \in L\}$. A convex point set corresponds to the vertices of a convex polygon; in a feasible solution, j convex polygons intersect S_i for some $j \in \{0,\ldots,k\}$. Each of these polygons has a top segment and a bottom segment in L_i. There are at most $\binom{|L_i|}{2}$ possible choices of segments in L_i for the first polygon, $\binom{|L_i|-2}{2}$ for the second polygon, ..., and $\binom{|L_i|-2(j-1)}{2}$ for the jth polygon, giving $\prod_{x=0}^{j-1}\binom{|L_i|-2x}{2} \in O(|L_i|^{2j}) = O(n^{4j})$ possible combinations of edges in S_i for a given $j \in \{0,\ldots,k\}$.

We construct an unweighted directed acyclic graph G. Each vertex in $V(G)$ corresponds to a slab S_i, a $j \in \{0,\ldots,k\}$, and a top edge and a bottom edge for each of the j convex polygons that intersect S_i. Consequently, the number of vertices in G is $O(\sum_{i=1}^{n-1}\sum_{j=0}^{k} n^{4j}) = O(kn^{4k+1})$.

Furthermore, we create $(n/k)^k$ copies of each vertex associated with a slab S_i, each of which is assigned a distinct value $(\ell_1,\ldots,\ell_k) \in \mathbb{Z}^k$, where for each $j \in \{1,\ldots,k\}$, $\ell_j = |P_j \cap (S_1 \cup \cdots \cup S_i)|$, i.e., the number of points of P_j that lie in the first i slabs. We refer to $\ell = \min_{j\in\{1,\ldots,k\}} \ell_j$ as the *level* of the vertex. Each vertex at level ℓ in G corresponds to a slab S_i, such that the minimum cardinality of any polygon in $S_1 \cup \ldots \cup S_i$ (or partial polygon if it includes points to the right of S_i) is ℓ. Therefore, the resulting graph G has $O((n/k)^k kn^{4k+1}) \subseteq O(\frac{1}{k^{k-1}} \cdot n^{5k+1})$ vertices. See Figure 2.

Every slab has exactly one point of P on its left boundary and one on its right boundary. For each vertex v in G, let v_l and v_r denote these two points of P for the slab corresponding to v. We add an edge from vertex u to vertex v in G if they are *compatible*. See Figure 3. The vertices u and v are compatible if:

- u and v correspond to neighbouring slabs, u to S_i and v to S_{i+1}, for some i, and
- all top and bottom segments associated with u that do not pass through p_i continue in v, where $p_i = u_r = v_l$ is the point of P on the common boundary of S_i and S_{i+1}, and
- one of the four following conditions is met:

Case 1. either (a) one top associated with u ends at p_i and one top associated with v begins at p_i, forming a right turn at p_i, or (b) one bottom associated with u ends at p_i and one bottom associated with v begins at p_i, forming a left turn at p_i (all polygons in S_i continue in S_{i+1}; the number of edges in S_i is equal to that in S_{i+1});

Case 2. one top and one bottom associated with u end at p_i, (one polygon ends in S_i and all remaining polygons continue into S_{i+1});

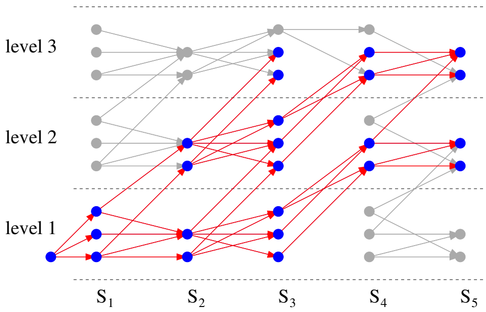

Fig. 2. Each slab S_i has various combinations of pairs of edges possible, each of which corresponds to a vertex in G, which is copied at levels 1 through n/k. Directed edges are added from a vertex associated with slab S_i to a vertex associated with a compatible slab S_{i+1}. The edge remains at the same level if the cardinality of the smallest set in $S_1 \cup \cdots \cup S_{i+1}$ remains unchanged; the level of S_{i+1} is one greater than the level of S_i if the cardinality of the smallest set in $S_1 \cup \cdots \cup S_{i+1}$ increases. Some vertices cannot be reached by any path from any source node at level 1 in slab S_1; these vertices and their out-edges are shaded gray. A feasible solution corresponds to a path rooted at a source node associated with the slab S_1 on level 1. An optimal solution ends at a sink node at the highest level among all feasible solutions.

Case 3. no top or bottom associated with u end at p_i, but one top and one bottom associated with v start at p_i (one polygon starts in S_{i+1} and all remaining polygons continue from S_i into S_{i+1}).

Case 4. all edges in u continue into v and no edge passes through $p_i = u_r = v_l$ (all polygons in S_i continue into S_{i+1}; the number of edges in S_i is equal to that in S_{i+1}).

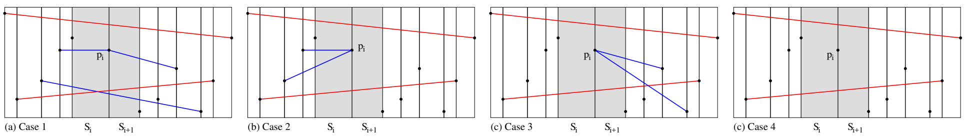

Fig. 3. The four cases in which we add an edge between the vertices u (associated with the slab S_i) and v (associated with the slab S_{i+1}) in G; i.e., u and v are *compatible*. In this example, $k = 2$, corresponding to two polygons, for which the edges through S_i and S_{i+1} are coloured blue and red, respectively. In Figure 3(a), p_i lies on the upper hull of the blue polygon, so the polygon makes a right turn at p_i, i.e., the angle below p_i must be convex. Figure 3(d), p_i is omitted from the selection.

For a given vertex u at most $n-2$ edges satisfy Case 1 (there are at most $n-2$ possible edges that continue from p_i to form a convex bend), at most one edge satisfies Case 2, at most $\binom{n-3}{2}$ edges satisfy Case 3, and at most one edge satisfies Case 4. Consequently, the number of edges in G is $O(n^2|V(G)|) \subseteq O(\frac{1}{k^{k-1}} \cdot n^{5k+3})$.

Any path from a source on level 1 to a highest-level node corresponds to an optimal solution, and can be found using breadth-first search in time proportional to the number of edges in G. The resulting worst-case running time is proportional to the number of vertices and edges in G: $O(|V(G)| + |E(G)|) = O(\frac{1}{k^{k-1}} \cdot n^{5k+3})$. In addition to storing a single in-neighbour from which a longest path reaches each node u, we can maintain a list of all of its in-neighbours that give a longest path, allowing the algorithm to reconstruct all distinct optimal solutions with the running time increased only by the output size.

The time for constructing the graph G is proportional to its number of edges. The combinations of $\binom{n}{2j}$ line segments in a slab S_i on level j can be enumerated and created in $O(1)$ time each, with $O(1)$ time per edge added if graph vertices are indexed according to their slab, their level, and the line segments they include. The level of each node in G is determined in $O(1)$ time per node by examining the level of any of its in-neighbours; the level increases by one in Cases 1 and 2 if the point p_i is added to the minimum-cardinality set and that set is the unique minimum.

Theorem 1. *Given a set P of n points in the plane, and a positive integer k, Bottleneck Convex Subsets can be solved exactly in $O(\frac{1}{k^{k-1}} \cdot n^{5k+3})$ time.*

Although we described Bottleneck Convex Subsets using a directed graph, one can think of a direct dynamic programming as follows. Consider the slabs from left to right. Assume that slabs $S_1, \ldots, S_i$ have been processed and we have at most $2k$ segments at S_i corresponding to the k convex sets. For each convex set, we need to consider $O(n^2)$ options for the top segment, $O(n^2)$ options for the bottom segment, and $O(n/k)$ options for its current size. Therefore, the number of entries in the dynamic programming table corresponding to S_i is $O(\frac{1}{k^k} n^{5k})$. While considering S_{i+1}, each convex subset for S_i can be extended to S_{i+1} in $O(n^2)$ ways. Hence, we need to spend $O(\frac{1}{k^k} n^{5k+2})$ time overall to fill the entries corresponding to S_{i+1}. Since there are n slabs, the overall time complexity becomes $O(\frac{1}{k^k} \cdot n^{5k+3})$.

Note that the graph model is sometimes beneficial over the dynamic programming because once the graph is computed, it allows us to employ various graph algorithms to readily find different types of outcomes. For example, let h be the highest-level such that a vertex of level h can be reached from a level 1 vertex. Then we can consider unit capacities for the edges and run a maximum flow algorithm with level 1 vertices as sources and the highest-level vertices as sinks to get a set of disjoint paths, which are likely to correspond to a diverse set of optimal solutions for Bottleneck Convex Subsets.

3. NP-Hardness

In this section we show that Bottleneck Convex Subsets is NP-hard. We first introduce some notation. Let $x(p)$ and $y(p)$ denote the $x-$ and y-coordinates of a point p, respectively. An angle $\angle pqr$ determined by points p, q and r is called a *y-monotone angle* if $y(p) > y(q) > y(r)$. A y-monotone angle is *left-facing* (resp. *right-facing*) if the point q lies interior to the left (resp., right) half-plane of the line through pr. If q lies on the line through pr, then we refer to $\angle pqr$ as a *straight angle*.

The idea of the hardness proof is as follows. We first prove that given a set of $3m$ points in the Euclidean plane, it is NP-hard to determine whether the points can be partitioned into m y-monotone angles, where none of them are right facing (Section 3.1). We then reduce this problem to Bottleneck Convex Subsets (Section 3.2).

3.1. *Covering points by straight or left-facing angles*

In this section we show that given a set of $3m$ points in the Euclidean plane, it is NP-hard to determine whether the points can be partitioned into m y-monotone angles, where none of them are right facing. In fact, we prove the problem to be NP-hard in a restricted setting, as follows:

> **ANGLE PARTITION**
> **Instance:** A set P of $3m$ points lying on three parallel horizontal lines ($y = 0, y = 1$ and $y = 2$) in the plane, where each line contains exactly m points.
> **Problem:** Partition P into at most m y-monotone angles, where none of them are right facing.

We reduce Distinct 3-Numerical Matching with Target Sums (DNMTS), which is known to be strongly NP-complete [17] .

> **DISTINCT NUMERICAL MATCHING WITH TARGET SUM**
> **Instance:** Three sets $A = \{a_1, \ldots, a_m\}, B = \{b_1, \ldots, b_m\}, C = \{c_1, \ldots, c_m\}$, each with m distinct positive integers, where $\sum_{i=1}^{m} a_i + \sum_{i=1}^{m} b_i = \sum_{i=1}^{m} c_i$.
> **Problem:** Decide whether there exist m triples (a_i, b_j, c_k), where $1 \le i, j, k \le m$, such that $a_i + b_j = c_k$ and no two triples share an element.

Theorem 2. *Angle Partition is NP-hard.*

Proof. Let $M = (X, Y, Z)$ be an instance of DNMTS, where each set A, B, C contains m positive integers. We now construct an instance Q of Angle Partition as follows: (I) For each $a \in A$, create a point at $(a, 0)$. (II) For each $b \in B$, create a point at $(b, 2)$. (III) For each $c \in C$, create a point at $(c/2, 1)$.

This completes the construction of the point set P of the Angle Partition instance Q (e.g., see Figure 4(a)). Since the numbers in A, B, C are distinct, no two points in P will coincide. Note that by definition, a y-monotone angle must contain one point from each of the lines $y = 0, y = 1$ and $y = 2$. Furthermore, every straight angle $\angle pqr$ will satisfy the equation $\frac{x(p)+x(r)}{2} = x(q)$. This transformation is inspired by a 3-SUM hardness proof for 'GeomBase' [14].

We now show that M has an affirmative solution if and only if P admits a partition into m y-monotone angles where none of them are right facing.

First consider that M has an affirmative answer, i.e., a set of m triples (a_i, b_j, c_k), where $1 \le i, j, k \le m$, such that $a_i + b_j = c_k$ and no two triples share an element. Therefore, we will have $\frac{(a_i+b_j)}{2} = \frac{c_k}{2}$. Hence we will find a straight line through $(a_i, 0), (b_k, 2), (c_j/2, 1)$. These lines will form m y-monotone straight angles (e.g., see

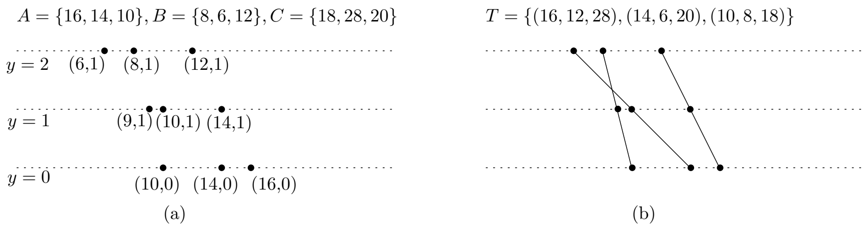

Fig. 4. (a) Construction of Q (points on three lines) from an instance $M = \{(16, 14, 10), (8, 6, 12), (18, 28, 28)\}$ of DNMTS. (b) A solution {(16,12,28),(14,6,20),(10,8,18)} for M and the corresponding angles of Q.

Figure 4(b)). Since none of these angles are right facing, this provides an affirmative solution for the instance Q.

Consider now the case when Q has an affirmative solution T, i.e., a partition of P into m y-monotone angles, where none of them are right facing. We first claim that (Step 1) all these m y-monotone angles must be straight angles and then (Step 2) show how to construct an affirmative solution for M.

Step 1: Suppose for a contradiction that the solution T contains one or more left-facing angles. For each left-facing angle $\angle rst$, where r, s, t are on lines $y = 0, y = 1$ and $y = 2$, respectively, we have $x(s) < \frac{x(r)+x(t)}{2}$. For each straight angle $\angle rst$, we have $x(s) = \frac{x(r)+x(t)}{2}$. Since we do not have any right-facing angle, the following inequality holds: $\sum_{\angle rst \in T} x(s) < \sum_{\angle rst \in T} \frac{x(r)}{2} + \sum_{\angle rst \in T} \frac{x(t)}{2}$. Since no two angles share a point, we have $\sum_{i=1}^{m}(c_i/2) < \sum_{i=1}^{m}(a_i/2) + \sum_{i=1}^{m}(b_i/2)$, which contradicts that M is an affirmative instance of DNMTS.

Step 2: We now transform the y-monotone straight angles of T into m triples for M. For each angle, $\angle rst$, where r, s, t are on lines $y = 0, y = 1$ and $y = 2$, we construct a triple $(x(r), x(t), 2x(s))$. Since $\angle rst$ is a straight angle, $x(r) + x(t) = 2x(s)$. Since no two angles share a point, the triples will be disjoint. □

3.2. *Bottleneck Convex Subsets is NP-hard*

In this section we reduce Angle Partition to Bottleneck Convex Subsets. Let P be an instance of Angle Partition, i.e., three lines $y = 0, y = 1$ and $y = 2$, each line containing m points. Without loss of generality we can assume that the coordinates of the points in P are positive; otherwise, it is straightforward to shift the point set to the right to construct an instance of Angle Partition that admits an affirmative answer if and only if P admits an affirmative answer. We now construct an instance H of Bottleneck Convex Subsets from P with $n = m(4m + 7)$ points and with $k = m$.

3.3. *Construction of H*

We first take a copy P' of the points of P and include those in H. Let Δ be a sufficiently large number (to be determined later). We now construct m *upper chains*. The ith upper chain U_i, where $1 \leq i \leq m$, is constructed following the step below (see Figure 5).

> *Construction of U_i:* Place two points at the coordinates $(i\Delta, \Delta^2 + 3)$ and $((i + 1)\Delta, 3)$. Let C be the curve determined by $f(x) = \Delta^2 + 3 - (x - i\Delta)^2$, which passes through these two points. Place $2m$ points $w_1, \ldots, w_{2m}$ uniformly on C between $(i\Delta, \Delta^2 + 3)$ and $((i + 1)\Delta, 3)$, i.e., $w_j = (i\Delta + \frac{j\Delta}{2m}, f(i\Delta + \frac{j\Delta}{2m}))$.

Each upper chain contains $(2m+2)$ points. We define the m lower chains symmetrically, where each lower chain V_i starts at $(i\Delta, -\Delta^2 - 1)$ and ends at $((i+1)\Delta, -1)$.

We now choose the parameter Δ. Let L be the set of lines with non-zero slopes passing through two points of P. Let t be the largest x-coordinate over all the intersection points created by L with lines $y = 0$, $y = 1$ and $y = 2$. We set Δ to be $(t + 2)^4$. This ensures that for any line ℓ with non-zero slope passing through two points of P, the upper and lower chains lie on the right half-plane of ℓ regardless of t. This concludes the construction of the Bottleneck Convex Subsets instance H, where $k = m$. Note that H has $3m + m(4m + 4) = m(4m + 7)$ points. In the best possible scenario, one may expect to cover all the points and have a partition into m disjoint convex subsets, where each set contains $(4m + 7)$ points.

We now have the following lemma.

Lemma 1. *Let W be a partition of the upper and lower chains into a set L of at most m disjoint convex sets. Then each convex set in L contains points from both an upper chain and a lower chain.*

Proof. Suppose for a contradiction that we have a convex set that contains points from either upper chains or lower chains, without loss of generality, from lower chains. Then we could delete all the points on the lower chain to obtain a convex set partition for the upper chains with fewer than m disjoint convex sets. To reach the contradiction, we now show that the upper chains cannot be covered with fewer than m disjoint convex sets.

Since an upper chain U contains $(2m+2)$ points, at least one convex set C must contain at least 3 or more points from this set. We assign C to U and repeat this process for the other upper chains. Let C_1 and C_2 be the convex sets assigned to the upper chains U_1 and U_2. We now show that C_1 and C_2 cannot be the same, which implies that there must be n disjoint convex sets.

Suppose for a contradiction that C_1 coincides with C_2. Then, by our assignment strategy, C_1 is guaranteed to contain three points from U_1 and three points from U_2. Since every three points of an upper chain form a right-facing y-monotone angle, we can define two right facing triangles A_1 and A_2 corresponding to U_1 and U_2. Without loss generality assume that U_1 appears to the left of U_2. Since A_1 is right

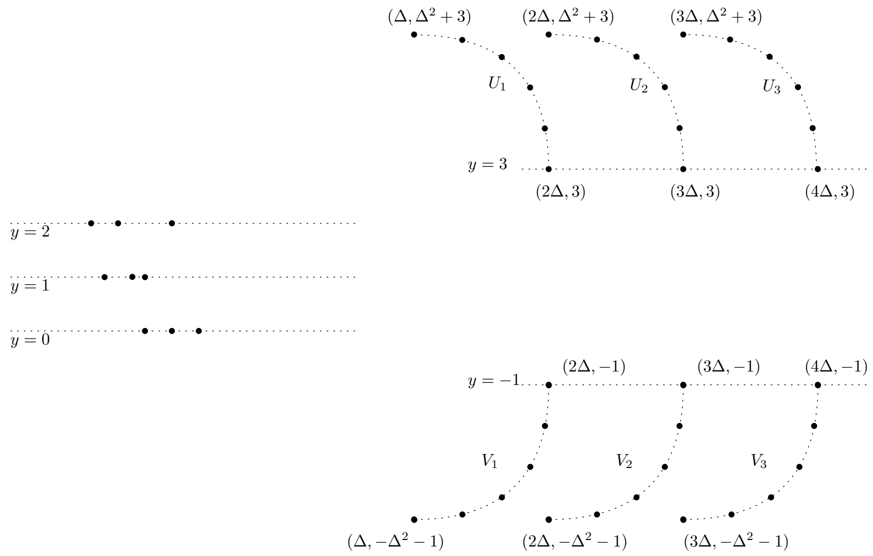

Fig. 5. Illustration for the construction of H. Note that this is only a schematic representation, which violates the property that all the chains are inside the wedge determined by the y-monotone angles.

facing, the line segment joining the two end points of A_1 cannot lie inside the convex hull of C_1, and hence C_1 cannot be a convex set. □

3.4. *Reduction*

We now show that the Angle Partition instance P admits an affirmative solution if and only if the Bottleneck Convex Subsets instance H admits $k(= m)$ disjoint convex sets with each set containing $(4m + 7)$ points.

Assume first that P admits an affirmative solution, i.e., P admits a set of m y-monotone angles such that none of these are right facing. By the construction of H, the corresponding point set P' must have such a partition into y-monotone angles. For each i from 1 to m, we now form a point set C_i that contains the ith y-monotone angle, the upper chain U_i and the lower chain V_i. Figure 6 illustrates such a scenario. By the construction of H, all the chains are inside the wedge determined by the y-monotone angle and hence C_i is a convex set with $(4m+7)$ points. Since the sets are disjoint, we obtain the required solution to the Bottleneck Convex Subsets instance.

Consider now that the points of H admits m disjoint convex sets with each set containing $(4m + 7)$ points. Since H contains $m(4m + 7)$ points, the convex sets form a partition of H. Let L be such a partition. We now show how to construct a solution for P using L. Let L' be a set of convex sets obtained by removing the points of P' from each convex set of L. By Lemma 1, each set of L' contains at

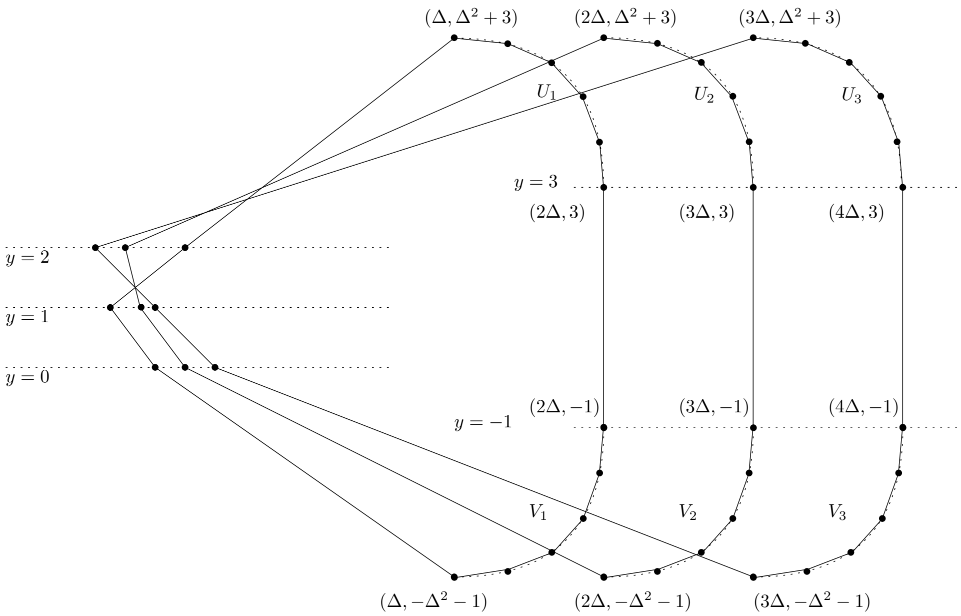

Fig. 6. A schematic representation for the construction of a convex partition for H from an angle partition of P.

least one point from the upper chains and one point from the lower chains. Since there are $3m$ points on P', to partition P' into m convex sets, we must need each convex set of L to contain a y-monotone angle with exactly one point from $y = 0$, one point from $y = 1$ and one point from $y = 2$. Since each convex set contains one point from an upper chain and one point from a lower chain, none of these y-monotone angles can be right facing. Hence we obtain a partition of P' into the required y-monotone angles, which implies a partition also for P. This completes the reduction. The following theorem summarizes the results.

Theorem 3. *The Bottleneck Convex Subsets problem is NP-hard.*

3.5. *NP-hardness for points in general position*

We now show that Bottleneck Convex Subsets remains NP-hard even for points in general position, i.e., when no three points lie on the same line.

We first consider Theorem 2. Given an instance $M = (X, Y, Z)$ of DNMTS, we showed in the proof of Theorem 2 that one can construct an instance of Angle Partition, i.e., a set P of $3m$ points on three parallel horizontal lines, such that M admits an affirmative solution if and only if P admits a partition into m y-monotone angles, where none of them are right facing. Here P contained collinear points (Figure 7(a)) and we showed if such a partition exists, then all m angles will be straight angles (Figure 7(b)). We now show how P can be perturbed to obtain a point set P' such that the following properties hold.

- Every right-facing y-monotone angle in P determines a right-facing angle in P'.
- Every left-facing y-monotone angle or straight angle in P determined by three points lying on three different horizontal lines determines a left-facing angle in P'.

Figure 7(c) illustrates an example of the point set P' obtained after the perturbation of P and Figure 7(d) illustrates a set of left-facing angles corresponding to the straight angles of Figure 7(b).

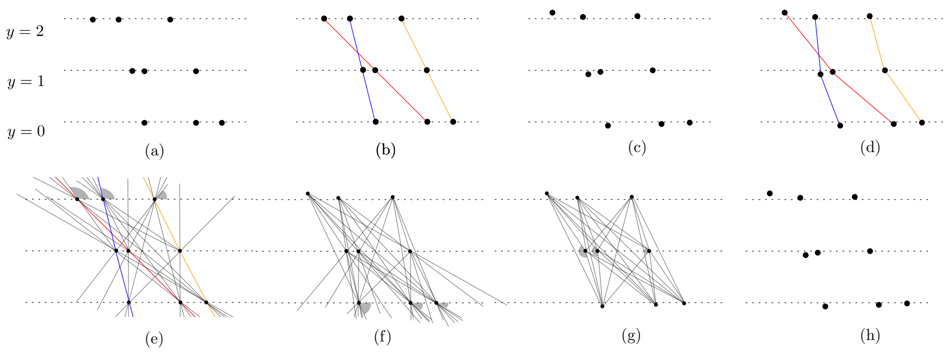

Fig. 7. Illustration for the perturbation of P to construct P'. (a)–(b) A point set P and its corresponding angle partition. (c)–(d) The perturbed point set P' and its corresponding angle partition. (e)–(g) Illustration for the perturbation of the points, where we first perturb the points on $y = 2$, then the points on $y = 0$ and finally the points on $y = 1$. (h) The point set P'.

.

Let $L(P)$ be the set of lines determined by pairs of points in P that lie on different horizontal lines. To perturb P, we will use this set L. For each point q on line $y = 2$, we consider a wedge W_q, as follows. One side of W_q is determined by the line $y = 2$. The other side is determined by the line of L that passes through q and makes the smallest anticlockwise angle with $y = 2$. Figure 7(e) illustrates these wedges in gray. We then perturb each point q inside its wedge W_q such that for every line $\ell \in L(P)$ and the corresponding line ℓ', the points that are on the open left (right) half-plane of ℓ, are included in the open left (right) half-plane of ℓ'. This ensures that the angles that were previously left or right facing in P, remain the same. However, the angles that were previously straight, now become left facing.

After the perturbation, only the lines $y = 0$ and $y = 1$ can contain collinear points. We next perturb the points on line $y = 0$ and finally, the points on line $y = 1$ using the same idea of creating a wedge for each point, as illustrated in Figure 7(f)–(h).

Consider now the reduction of Section 3.4. Here for each i from 1 to n, we perturb the points of the corresponding upper and lower chain by rotating the set $(U_i \cup V_i)$ by a small angle such that the circular ordering of the points around each point remains the same. Since the perturbation preserves all the convex and concave angles required for the reduction, we obtain the following theorem.

Theorem 4. *The Bottleneck Convex Subsets problem is NP-hard even for points in general position.*

4. Point Sets with Few Points inside the Convex Hull

In this section we show that the Bottleneck Convex Subsets problem is fixed-parameter tractable when parameterized by the number of points r inside the convex hull, i.e., these points do not lie on the convex-hull boundary.

Theorem 5. *Let P be a set of n points and let r be the number of points interior to the convex hull of P. Then one can solve the Bottleneck Convex Subsets problem on P in $f(r) \cdot n^{O(1)}$ time, i.e., the Bottleneck Convex Subsets problem is fixed-parameter tractable when parameterized by r.*

Proof. Let k be the number of disjoint convex sets that we need to construct. We guess the cardinality of the smallest convex set in an optimal solution and perform a binary search.

For a guess q, we use Algorithm 0 to check whether there exists k disjoint convex sets each with q points as follows.

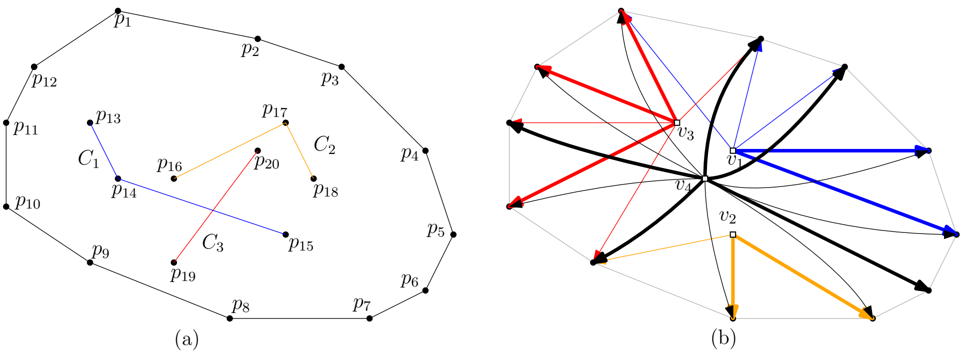

Fig. 8. Illustration for the Bottleneck Convex Subsets problem with eight points inside the convex hull of P. For the convex set corresponding to v_3, we have assigned the left halfplane of the line through p_{19} and p_{20}. The edges carrying the flow are shown in thick edges.

.

Assume that j of the k convex sets contain points from the interior. Since there are only r interior points, we must have $j \leq r$. We enumerate for each j from 0 to

Algorithm 1 Bottleneck Convex Subsets

Input: A point set P of n points and two positive integers k and q where $n \le kq$.
Output: k convex subsets $P_1, \ldots, P_k$ of P, each containing q points (if exists).
$r \leftarrow$ Number of interior points of P
for each set of $j \le r$ convex sets, each containing at most q interior points **do**
    $A \leftarrow$ A set of k vertices; j of them correspond to the convex sets $C_1, \ldots, C_j$
    $B \leftarrow$ A set of vertices of P on the convex hull
        ▷ Leverage the maximum flow algorithm
    $G \leftarrow$ A graph with vertex set $(A \cup B)$, where an edge (v, w) with $v \in A$ and $w \in B$ indicates that w together with the vertices of C_v form a convex set
    Set each vertex $v \in A$ to be a source with a production of $(q - |C_v|)$ units
    Set each vertex $w \in B$ to be a sink that can consume at most 1 unit
    if If the maximum flow is $\sum_{j=0}^{k}(q - |C_j|)$ units **then**
        Return $P_1, \ldots, P_k$, which are constructed using the k flow sources and their corresponding sinks
    else
        Return **Null**
    end if
end for

r, all possible sets of j convex sets, where each convex set contains at most q points from the interior of P. For each set of size $\ell \le r$, we also consider all possible convex orderings of the points such that each convex ordering determines a non-crossing convex path of length $(\ell - 1)$. Figure 8(a) illustrates such a set of $j = 3$ convex sets C_1, C_2, C_3 with a particular ordering of the points for each set. Therefore, we have $\sum_{j=0}^{k} j\binom{2^r}{j}$ possibilities to consider. Thus the number of elements in the enumeration is at most $\sum_{j=0}^{k} r\binom{2^r}{j}2^j \le \sum_{j=0}^{k} r2^{r^{j+1}} \le r2^{r^{k+2}}$.

The idea is to examine whether these j sets can be extended to contain q points each and to check whether the remaining points can be used to construct the remaining $(k - j)$ convex sets by modelling this with a maximum flow problem. We construct a bipartite graph G with vertex set $A \cup B$. The set A contains j vertices $v_1, \ldots, v_j$ corresponding to the sets $C_1, \ldots, C_j$ and $(k - j)$ additional vertices representing the remaining $(k - j)$ sets (which are currently empty) to be constructed. The set B consists of $(n - r)$ vertices, each corresponding to a distinct point on the convex hull of P. We add a directed edge from a vertex v in A to a vertex w in B if the point w together with the interior points corresponding to v form a convex set. For the case when the interior points corresponding to v form a straight line (e.g., C_3 in Figure 8(a)), we connect v to the points of B that lie on the halfplane assigned to v. Figure 8(b) illustrates the resulting graph.

We now consider a maximum flow on this graph where each vertex v_i in A has a production of $(q - |C_i|)$ units of flow and each sink can consume at most 1 unit of flow. A maximum flow of $\sum_{j=0}^{k}(q - |C_j|)$ units indicates that the guess q is feasible, and we continue the binary search by guessing a higher value. Otherwise, we search by guessing a lower value.

Hence the overall time complexity becomes $O(f(r) \cdot g(n) \log n)$, where $f(r) \in O(r2^{r^{k+2}})$, $g(n)$ is the time required for the maximum flow algorithm, and the $\log n$ term corresponds to the binary search. □

5. Discussion

We examined the Bottleneck Convex Subsets problem of selecting k mutually disjoint convex subsets of a given set of points P such that the cardinality of the smallest set is maximized. We gave an algorithm that solves Bottleneck Convex Subsets for small values of k, showed Bottleneck Convex Subsets is NP-hard for an arbitrary k, and proved Bottleneck Convex Subsets to be fixed parameter tractable when parameterized by the number of points interior to the convex hull. The problem is also solvable in polynomial time for specific large values of k. If $k > n/4$, then some subset has cardinality at most three; a solution is found trivially by arbitrarily partitioning P into k subsets of size $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$. If $k \in \{\lfloor n/5 \rfloor + 1, \ldots, n/4\}$ then some subset has cardinality at most four. As discussed in Section 1.2, Károlyi [18] characterized necessary and sufficient conditions for a set of n points in general position to admit a partition into $k = n/4$ convex quadrilaterals, and gave an $O(n \log n)$-time algorithm to decide whether such a partition exists; if no such partition exists, then some set must contain at most three points, which can be solved as described above. It remains open to determine whether Bottleneck Convex Subsets can be solved in polynomial time for all $k \in \Theta(n)$.

As a direction for future research, a natural question is to establish a good lower bound on the time required to solve these problems for small fixed values of k. In particular, is the $O(n^3)$-time algorithm of Chvátal and Klincsek [4] optimal for the case $k = 1$? Note that our algorithm has time $O(n^8)$ when $k = 1$. It would also be interesting to examine whether a fixed-parameter tractable algorithm exists for Bottleneck Convex Subsets when parameterized by k, and to find approximation algorithms for Bottleneck Convex Subsets when k is an arbitrary input parameter, with running time polynomial in n and k.

It would also be interesting to look into Bottleneck Convex Subsets where we enforce the convex hulls of the convex sets to be disjoint. The problem would relate to the problem of finding a convex point set embedding of a graph in a point set [7]. The input of a *convex point set embedding* consists of a planar point set P of n points and a planar graph G of n vertices. The goal is to determine whether G admits a planar straight line drawing where P determines the vertex locations and each bounded face in the drawing determines a convex polygon. Therefore, to test whether there are k partitioned convex sets, each with at least q points, we can set G to be a collection of k disjoint cycles, each of length q, and $(n - kq)$ isolated vertices.

Acknowledgement

The work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors dedicate the work in memory of Saeed Mehrabi. A preliminary set of results appeared in the Proceedings of the 27th International Computing and Combinatorics Conference (COCOON 2021)[6].

References

1. Esther M. Arkin, Sándor P. Fekete, Ferran Hurtado, Joseph S. B. Mitchell, Marc Noy, Vera Sacristán, and Saurabh Sethia. On the reflexivity of point sets. *Discrete and Computational Geometry*, 25:139–156, 2003.
2. David Avis and David Rappaport. Computing the largest empty convex subset of a set of points. In *Proc. Symposium on Computational Geometry (SoCG)*, pages 161–167, 1985.
3. Björn Brodén, Mikael Hammar, and Bengt J. Nilsson. Guarding lines and 2-link polygons is apx-hard. In *Proc. Canadian Conference on Computational Geometry (CCCG)*, pages 45–48, 2001.
4. V. Chvátal and G. Klincsek. Finding largest convex subsets. *Congressus Numerantium*, 29:453–460, 1980.
5. Ren Ding, Kiyoshi Hosono, Masatsugu Urabe, and Changqing Xu. Partitioning a planar point set into empty convex polygons. In *Proc. Discrete and Computational Geometry, Japanese Conference (JCDCG)*, volume 2866 of *LNCS*, pages 129–134, 2002.
6. Stephane Durocher, J. Mark Keil, Saeed Mehrabi, and Debajyoti Mondal. Bottleneck convex subsets: Finding k large convex sets in a point set. In *In Proc. of the 27th International Conference on Computing and Combinatorics*, volume 13025 of *LNCS*, pages 203–214. Springer, 2021.
7. Stephane Durocher and Debajyoti Mondal. On the hardness of point-set embeddability. In *Proc. International Workshop on Algorithms and Computation (WALCOM)*, volume 7157 of *LNCS*, pages 148–159. Springer, 2012.
8. Herbert Edelsbrunner and Leonidas J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38(1):165–194, 1989.
9. David Eppstein. *Forbidden Configurations in Discrete Geometry.* Cambridge University Press, 2018.
10. David Eppstein and Jeff Erickson. Iterated nearest neighbors and finding minimal polytopes. *Disc. Comp. Geom.*, 11:321–350, 1994.
11. David Eppstein, Mark H. Overmars, Günter Rote, and Gerhard J. Woeginger. Finding minimum area k-gons. *Disc. Comp. Geom.*, 7:45–58, 1992.
12. P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.
13. Paul Erdős, Zsolt Tuza, and Pavel Valtr. Ramsey-remainder. *Eur. J. Comb.*, 17(6):519–532, 1996.
14. Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comp. Geom.*, 45(4):140–152, 2012.
15. Panos Giannopoulos, Christian Knauer, and Daniel Werner. On the computational complexity of Erdős-Szekeres and related problems in R^3. In *Proc. European Symposium on Algorithms (ESA)*, pages 541–552, 2013.
16. Hernán González-Aguilar, David Orden, Pablo Pérez-Lantero, David Rappaport, Carlos Seara, Javier Tejel, and Jorge Urrutia. Maximum rectilinear convex subsets. In

Proc. Symposium on Fundamentals of Computation Theory (FCT), pages 274–291, 2019.
17. Heather Hulett, Todd G. Will, and Gerhard J. Woeginger. Multigraph realizations of degree sequences: Maximization is easy, minimization is hard. *Oper. Res. Lett.*, 36(5):594–596, 2008.
18. Gyula Károlyi. Ramsey-remainder for convex sets and the Erdős-szekeres theorem. *Discret. Appl. Math.*, 109(1-2):163–175, 2001.
19. D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, 15(1):287–299, 1986.
20. V. S. Anil Kumar, Sunil Arya, and H. Ramesh. Hardness of set cover with intersection 1. In *Proc. International Colloquium on Automata, Languages and Programming (ICALP)*, pages 624–635, 2000.
21. Nimrod Megiddo and Arie Tamir. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.*, 1(5):194–197, 1982.
22. Walter Morris and Valeriu Soltan. The Erdős-Szekeres problem on points in convex position - a survey. *Bulletin of The American Mathematical Society*, 37:437–459, 10 2000.
23. Masatsugu Urabe. On a partition into convex polygons. *Discret. Appl. Math.*, 64(2):179–191, 1996.