

NP-COMPLETENESS OF SLOPE-CONSTRAINED DRAWING OF COMPLETE GRAPHS

Cédric Pilatte*†

ABSTRACT. We prove the NP-completeness of the following problem. Given a set S of n slopes and an integer $k \geq 1$, is it possible to draw a complete graph on k vertices in the plane using only slopes from S ? Equivalently, does there exist a set K of k points in general position such that the slope of every segment between two points of K is in S ? We then present a polynomial-time algorithm for this question when $n \leq 2k - c$, conditional on a conjecture of R.E. Jamison. For $n = k$, an algorithm in $\mathcal{O}(n^4)$ was proposed by Wade and Chu. For this case, our algorithm is linear and does not rely on Jamison’s conjecture.

Keywords – Computational Complexity, Discrete Geometry.

1 Introduction

A *straight-line drawing* of an undirected graph G is a representation of G in the plane using distinct points for the vertices of G and line segments for the edges. The segments are allowed to intersect, but not to overlap, meaning that no segment may pass through a non-incident vertex. The slope of a line l is denoted by $\text{slp}(l) \in \mathbb{R} \cup \{\infty\}$. If A is a set of points, we write $\text{slp}(A)$ for the set of all slopes determined by A , i.e.

$$\text{slp}(A) = \{\text{slp}(A_1A_2) \mid A_1, A_2 \in A, A_1 \neq A_2\}.$$

The number of slopes used in a straight-line drawing is the number of distinct slopes of the segments in the drawing. In 1994, Wade and Chu [26] introduced the *slope number* of a graph G , which is the smallest number n for which there exists a straight-line drawing of G using n slopes. This notion has been the subject of extensive research. It was proven independently by Pach, Pálvölgyi [22] and Barát, Matoušek, Wood [1] that graphs of maximum degree five may have arbitrarily large slope number. In the opposite direction, Mukkamala and Pálvölgyi [20] showed that graphs of maximum degree three have slope number at most four, generalizing results in [6, 17, 21]. Whether graphs of maximum degree four have bounded slope number is still an open problem. Computing the slope number of a graph is difficult in general: it is NP-complete to determine whether a graph has slope number two [8]. See also [3, 9, 11, 16] for the study of the *planar* slope number and related algorithmic questions.

Let us consider the case of the complete graph K_k on k vertices. Let $K \subset \mathbb{R}^2$ be the set of points corresponding to the vertices in a straight-line drawing of K_k . From the

*Department of Mathematics, University of Mons (UMONS), Place du Parc 20, 7000 Mons, Belgium.

†Dep. Math. and Applications, École Normale Supérieure (ENS), Rue d’Ulm 45, 75005 Paris, France.

definitions, we know that K is in general position and the set of slopes used in the drawing is exactly $\text{slp}(K)$. As in [14], we will use the adjective *simple* instead of *in general position*, for brevity. It is easily seen that a simple set of k points determines at least k slopes [14,26]. On the other hand, a straight-line drawing of K_k with k slopes may be obtained by considering the vertices of a regular k -gon. The slope number of K_k is thus exactly k .

The slope number of a graph G provides only partial information about the possible sets of slopes of straight-line drawings of G . Two questions arise naturally:

1. What can be said about the straight-line drawings of a graph G that use only a certain number of slopes?
2. Given a set S of slopes, does there exist a straight-line drawing of G using only slopes from S ?

We focus on the case where G is a complete graph. For this case, both questions can be rephrased by replacing *straight-line drawings* by *simple sets of points*. The case of complete graphs is already difficult and sheds some light on the general situation. As we explain below, the first question is still unanswered, in almost all cases, while the second is already NP-complete when restricted to complete graphs (Theorem 3.10).

Regarding the first question, we have already seen that regular k -gons are examples of simple sets of k points that use only k slopes. As affine transformations preserve parallelism, the image of a regular k -gon under an invertible affine transformation is also a simple set of k points with k slopes (a set obtained this way is called an *affinely-regular k -gon*).¹ Jamison [14] proved that these are the only possibilities, thereby classifying all straight-line drawings of K_k with exactly k slopes. In the same paper, he conjectured a much more general statement.

Conjecture (Jamison). For some constant c_1 , the following holds. If $n \leq 2k - c_1$, every simple set of k points forming (exactly) n slopes is contained in an affinely-regular n -gon.

The case $n = k$ corresponds to Jamison's result, and the case $n = k + 1$ has been proven recently [23]. The conjecture is still open for $n = k + 2$ and beyond.

The aim of this paper is to investigate the second question: the algorithmic problem of deciding whether a complete graph admits a straight-line drawing that uses only slopes from a given set.

Definition 1.1. The *slope-constrained complete graph drawing problem* (SCGD for short), is the following decision problem.

- Input.**
- A set S of n slopes;
 - A natural number k .
- Output.**
- YES if there exists a simple set K of k points in \mathbb{R}^2 such that $\text{slp}(K) \subseteq S$;
 - NO otherwise.

As a simple set of k points determines at least k slopes, the problem is only interesting when $n \geq k$. Wade and Chu [26] gave an algorithm with time complexity $\mathcal{O}(n^4)$ for the

¹By *regular n -gon* or *affinely-regular n -gon*, we actually mean the set of vertices of the corresponding polygon.

restricted version of the SCGD problem, where the number of points is equal to the number of slopes, i.e. $k = n$. They asked how to solve the problem when the set of slopes contains more than k slopes.

In this article, we also consider variants of the SCGD problem where the input is required to satisfy an inequality of the type “ $n \leq f(k)$ ”, for several functions f . As we will see, the complexity of the restricted SCGD problem is highly dependent on the choice of f , if $P \neq NP$. Our results are the following (also summarized in Figure 1):

- **Section 3:** The SCGD problem is NP-complete (Theorem 3.10) in the case where there is no restriction on n . A careful examination of the proof shows that the SCGD problem remains NP-complete when restricted to $n \leq ck^2$ (for some $c > 0$, which may be chosen to be 2). The key ingredient is a notion of *slope-generic sets* (Definition 3.5).
- **Section 4:** The SCGD problem becomes polynomial when the number of slopes is not too large compared to the number of points. More precisely, assuming Jamison’s conjecture, there is an $\mathcal{O}(n(n-k+1)^4)$ time algorithm for the SCGD problem restricted to $n \leq 2k - c_1$, where c_1 is the constant appearing in the conjecture. We also give a randomized variant of the algorithm which runs in $\mathcal{O}(n)$ time and gives the correct output with high probability (one-sided Monte-Carlo algorithm).

As mentioned earlier, Jamison’s conjecture has been proven for $n = k$ and $n = k + 1$. Consequently, our algorithm is correct unconditionally when restricted to $n \leq k + 1$. Moreover, in this case, it is linear, which is easily seen to be optimal. In particular, it improves the $\mathcal{O}(n^4)$ algorithm of Wade and Chu [26] (which applies to the case $n = k$ only).

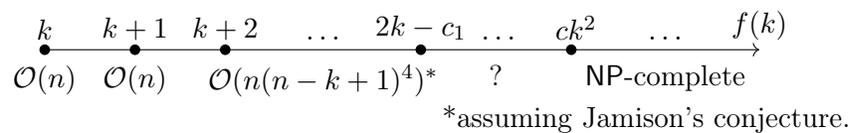


Figure 1: Complexity of the SCGD problem restricted to $n \leq f(k)$ for different functions f , where n is the number of slopes and k the number of points.

The question of the complexity of the SCGD problem in the intermediate case, denoted by a question mark in figure Figure 1, is still unanswered.

2 Notations and terminology

Recall that a set of points A is *simple* if no three points of A are collinear. We will say that A has *distinct slopes* if it is simple and $\text{slp}(A_1A_2) \neq \text{slp}(A_3A_4)$ for every four distinct points A_i of A .

We use the term *list* for an ordered sequence. By abuse of notation, if $A = (A_i)_{i \in I}$ is a list, we continue to write A for the underlying set $\{A_i \mid i \in I\}$. The notions defined for sets of points thus apply to lists of points by ignoring the order structure.

A set of slopes is naturally endowed with a cyclic order, induced from the one on $\mathbb{R} \cup \{\infty\}$ (see [10, §7.2]). If S is a set of slopes and $T = (s_1, \dots, s_n)$ is a list of slopes, we say the slopes of T are *consecutive slopes* of S if T is an interval of the cyclically ordered set S . This means that s_1, s_n are the two endpoints of the interval and that s_1, \dots, s_n are all the intermediate slopes, in the correct cyclic order.

Let $\text{Aff}(2, \mathbb{R})$ be the group of invertible affine transformations of the plane. A slope can be identified with a point at infinity. There is a natural action of $\text{Aff}(2, \mathbb{R})$ on the line at infinity. Therefore, if $\phi \in \text{Aff}(2, \mathbb{R})$, it makes sense to write $\phi(s)$ when s is a slope. Let \mathbb{H} denote the subgroup of $\text{Aff}(2, \mathbb{R})$ of translations and homotheties. These are precisely the affine transformations that map every line l to a line parallel to l . In other words, \mathbb{H} is the pointwise stabilizer of the line at infinity (see for instance [19, Chapter 5]).

Definition 2.1. If A, B are two point sets in the plane, we write $A \leftrightarrow B$ if there is a $\phi \in \mathbb{H}$ with $\phi(A) \subseteq B$. If $\phi(A) = B$, we write $A \sim B$ and say that A and B are *homothetic*.

Remark 2.2. Note that $A \sim B$ implies $\text{slp}(A) = \text{slp}(B)$. The converse is false (see e.g. Remark 3.4).

Remark 2.3. If two simple lists $E = (E_1, \dots, E_n)$ and $F = (F_1, \dots, F_n)$ of $n \geq 2$ distinct points satisfy $\text{slp}(E_i E_j) = \text{slp}(F_i F_j)$ for all $i \neq j$, there is a unique $\phi \in \mathbb{H}$ such that $\phi(E_i) = F_i$ for all i . In particular, $E \sim F$. Note that this assumption is stronger than just $\text{slp}(E) = \text{slp}(F)$.

3 NP-completeness of the SCGD problem

In Section 3.1, we informally present some ideas leading to the introduction of the notion of *slope-generic sets*. The actual definitions are given in Section 3.2, where we show the first properties of slope-generic sets. In Section 3.3, we prove that the SCGD problem is NP-complete, assuming that slope-generic sets can be constructed in polynomial time. This last assertion is proved in Section 3.4.

3.1 Motivation

A sufficiently general set of n points in the plane determines $\binom{n}{2}$ distinct slopes. Suppose now that we wish to go the other way, and define a point set by specifying its set of slopes. First of all, we can only hope to define a point set modulo \mathbb{H} , since $\text{slp}(A) = \text{slp}(B)$ whenever A and B are homothetic. There is a more fundamental problem: the slopes are not independent.² As soon as we consider four distinct points, the six slopes they determine are related by a certain polynomial equation (cf. Lemma 3.12). When we consider a point set A of size n , we get a polynomial equation for every subset of four points of A . Hence, most choices of $\binom{n}{2}$ slopes do not determine a set of n points modulo \mathbb{H} .

²Three real parameters are needed to specify an element of \mathbb{H} . Therefore, $2n - 3$ real parameters are needed to identify a set of n points modulo \mathbb{H} . For $n \geq 4$, this is less than the number of slopes, $\binom{n}{2}$ (it is equal for $n \in \{2, 3\}$).

We would like to capture the notion of a point set A having a “generic” (i.e. “ordinary”) set of slopes $\text{slp}(A)$. For many purposes, a set of points can be considered “generic” when it is simple (i.e. in general position). In the context of slopes, this is not a sufficiently restrictive condition. For example, a regular n -gon cannot be considered “generic”, since it has the very special property of forming only n different slopes. Having distinct slope is the first (but not the only) condition that we will impose on a point set to have a “generic” set of slopes.

Our notion should be designed in such a way that any “sufficiently random” point set A should have a “generic” set of slopes $\text{slp}(A)$. However “random” A may be, the elements of $\text{slp}(A)$ will always satisfy a system of polynomial equations given by Lemma 3.12. The intuitive idea is that the slopes of a “generic” point set A should not satisfy more equations of the same form.

There is another way to decide whether the slopes of a point set A should be considered “generic”. The basic idea is as follows: if A is “generic”, the knowledge of $\text{slp}(A)$ should be enough to recover the original set A , modulo \mathbb{H} . In other words, for any point set K with slope set $\text{slp}(K) = \text{slp}(A)$, we should have $K \sim A$. As such, this statement is not entirely correct (we give a refined heuristic below). Moreover, to ensure that $\text{slp}(A)$ is really “generic”, we will need to impose a similar condition *for many subsets T of $\text{slp}(A)$* , instead of for $T = \text{slp}(A)$ only. A more elaborate version of the previous idea could be as follows.

Definition 3.1. A point set A with distinct slopes is *k -ordinary* if every subset $T \subseteq \text{slp}(A)$ of $\binom{k}{2}$ slopes has the following property: whenever K is a set of k points with $\text{slp}(K) = T$, we have $K \leftrightarrow A$.

We cannot hope to say anything interesting about A by considering the subsets of three slopes of $\text{slp}(A)$.³ Hence, from now on, we will only be interested in the case $k \geq 4$. The intuition one should keep in mind is that any “sufficiently random” set A is k -ordinary for all $k > 4$. For $k = 4$, where the situation is more subtle, Definition 3.1 has to be modified to correspond to the behaviour of “sufficiently random” point sets.

Let us explain the difference between the cases $k > 4$ and $k = 4$. Let $k \geq 4$, let A be a “sufficiently random” point set and let $T \subseteq \text{slp}(A)$ be a set of $\binom{k}{2}$ slopes. There are two possibilities to consider.

1. The first possibility is that one cannot find a set K of k points with $\text{slp}(K) = T$. For this T , the property in Definition 3.1 is vacuously true. This situation is represented in Figure 2. The slopes $s \in T$ must originate from at least $k + 1$ points of A (otherwise, they would form the slope set of a subset of k points of A , as in Figure 3). For this reason, these slopes are “independent”, or “unrelated”, in some appropriate sense.

³Given any set T of three distinct slopes, there exists a triangle K with $\text{slp}(K) = T$, and the triangle is unique modulo \mathbb{H} . This is true whether $T \subseteq \text{slp}(A)$ or not. Thus, each of the $\binom{|\text{slp}(A)|}{3}$ choices of three slopes of A gives a different set K , but only $\binom{|A|}{3}$ of those K will have $K \leftrightarrow A$. Therefore, no set of more than three points is 3-ordinary.

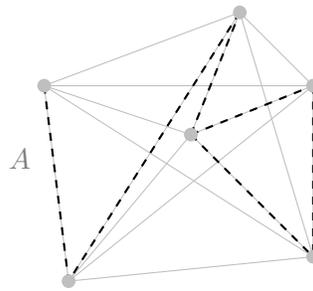


Figure 2: No set of k points in the plane has slope set T (T is in dashed).

2. The other alternative is when we can write T as $T = \text{slp}(K)$ for some set K of k points. Therefore, the slopes in T are “related”. If A is “generic”, we expect to have $T = \text{slp}(B)$ for some subset B of k points of A (as in the left part of Figure 3, in order to avoid the situation of Figure 2).
 - Assume that $k > 4$. As B is “sufficiently random”, we expect B to be the only point set modulo \mathbb{H} with slope set T . Thus, generically, we have $K \sim B$, so $K \hookrightarrow A$.
 - Suppose that $k = 4$. Perhaps surprisingly, it turns out that we do not necessarily have $K \sim B$. Nevertheless, we can always construct from K a point set K^* (see Definition 3.2), for which $\text{slp}(K^*) = \text{slp}(K)$ and yet $K^* \not\sim K$ in general. In Figure 3, we have $K \not\sim B$, but $K^* \sim B$. This will be true more generally: if T is the set of slopes of a “generic” point set K , we predict every set of four points with slope set T to be homothetic to either K or K^* . Therefore, we should have either $K \hookrightarrow A$ (if $K \sim B$) or $K^* \hookrightarrow A$ (if $K^* \sim B$).

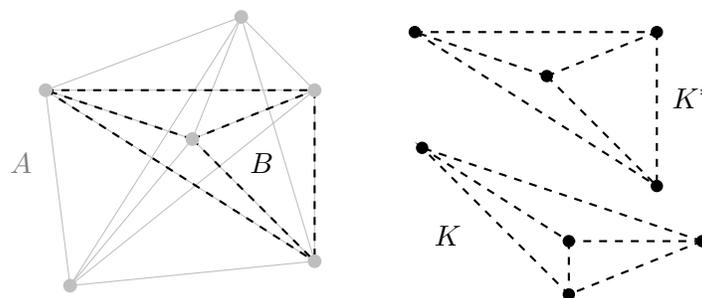


Figure 3: T (in dashed) is determined by a subset B of four points of A .

The intuitive reasoning in the case $k = 4$ leads to the definition of *slope-generic sets* (Definition 3.5). We will see in Lemma 3.7 that slope-generic sets are automatically k -ordinary for all $k \geq 5$. Slope-generic sets will demonstrate their usefulness in the proof of our main theorem (Theorem 3.10). Over the course of Section 3.4, we will relate this purely geometric point of view with the previous algebraic considerations (cf. the proof of Lemma 3.15 and Remark 3.17).

3.2 Slope-generic sets

We start by defining the dual of a list of four points, which is another list of four points that determines the exact same slopes, while not being related to the first list by an affine transformation. We denote by \mathfrak{S}_n the n th symmetric group and by $l(X; PQ)$ the parallel to PQ through X .

Definition 3.2. Let $E = (E_1, E_2, E_3, E_4)$ be a simple list of four points. We define a new list F of four points by setting $F_1 := E_2E_4 \cap l(E_1; E_2E_3)$, $F_2 := E_1E_3 \cap l(E_2; E_1E_4)$, $F_3 := E_2$ and $F_4 := E_1$ (see Figure 4). The intersections exist as E is simple. We call F the *dual* of E and write $F = E^*$.

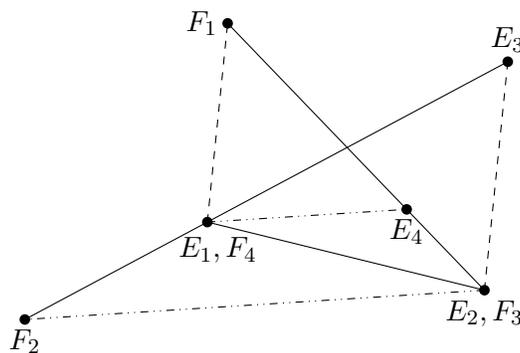


Figure 4: Construction of the dual.

Lemma 3.3. Let E be a simple list of four points and let $F = E^*$. Then $\text{slp}(F) = \text{slp}(E)$. More precisely, $F_{\sigma(1)}F_{\sigma(2)} \parallel E_{\sigma(3)}E_{\sigma(4)}$ for every permutation $\sigma \in \mathfrak{S}_4$.

Proof. Of the parallelism conditions that need to be verified, all but one follow directly from the definition. For example, F_1F_4 and E_2E_3 are parallel since, by definition, F_1 lies on the parallel to E_2E_3 through F_4 .

The only non-trivial fact is that $F_1F_2 \parallel E_3E_4$. If $E_1E_4 \parallel E_2E_3$, we have $F_1 = E_4$, $F_2 = E_3$ and there is nothing to prove. Otherwise, this is an application of Pappus's hexagon theorem with the collinear triples $(F_1, E_4, E_2 = F_3)$ and $(E_3, F_2, E_1 = F_4)$. \square

Remark 3.4. • If E is a simple list of four points, $\text{slp}(E^*) = \text{slp}(E)$, but $E^* \not\sim E$ in general (see Figures 4 and 5).

- It is nonetheless true that $(E^*)^* \sim E$, as can be observed in Figure 5. To prove it, note that, for every permutation $\sigma \in \mathfrak{S}_4$, we have $E_{\sigma(1)}^{**}E_{\sigma(2)}^{**} \parallel E_{\sigma(3)}^*E_{\sigma(4)}^* \parallel E_{\sigma(1)}E_{\sigma(2)}$ by Lemma 3.3. By Remark 2.3, we conclude that $(E^*)^* \sim E$.
- Changing the order of the points of E in Definition 3.2 does not change the point set E^* modulo \mathbb{H} . In other words, $(E_{\sigma(1)}, \dots, E_{\sigma(4)})^* \sim (E_1, \dots, E_4)^*$ for any permutation $\sigma \in \mathfrak{S}_4$. This again follows from Lemma 3.3 and Remark 2.3.

We now define slope-generic sets.

Definition 3.5. A set $A \subset \mathbb{R}^2$ is called *slope-generic* if it has distinct slopes and satisfies the following property: for every simple list E of four points, $\text{slp}(E) \subseteq \text{slp}(A)$ implies $E \hookrightarrow A$ or $E^* \hookrightarrow A$.

Example 3.6. • The set $A = \{(-2, 2), (-1, 0), (0, 0), (0, 1)\}$ is not slope-generic. Indeed, for $E = ((1, -1), (-1, 0), (0, 0), (0, 1))$, we have $\text{slp}(E) = \text{slp}(A)$. However, $E \not\hookrightarrow A$ and $E^* \not\hookrightarrow A$, since A is in convex position, unlike E and E^* (see Figure 5).

- We will give examples of slope-generic sets in Lemma 3.15. A careful inspection of the proof of Lemma 3.15 reveals that any set of “sufficiently random” points is slope-generic (see Remark 3.17).

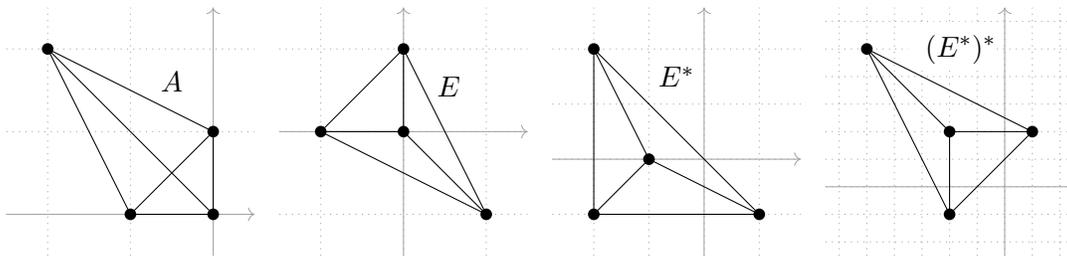


Figure 5: Examples of point sets with the same set of slopes.

Lemma 3.7. Let A be slope-generic. Then A has the following property: for every simple set $K \subset \mathbb{R}^2$ of at least five points, $\text{slp}(K) \subseteq \text{slp}(A)$ implies $K \hookrightarrow A$.

Proof. Let K be simple with $|K| \geq 5$ and $\text{slp}(K) \subseteq \text{slp}(A)$.

We start by proving that, for every subset $E \subset K$ of four points, we have $E \hookrightarrow A$. By contradiction, there is a quadruple $E = (E_1, E_2, E_3, E_4)$ in K such that $E \not\hookrightarrow A$. As A is slope-generic, $E^* \hookrightarrow A$. Thus, there exist four points $A_i \in A$ and a map $\phi \in \mathbb{H}$ such that $\phi(E_i^*) = A_i$ for $i \in \{1, \dots, 4\}$, where $(E_1^*, E_2^*, E_3^*, E_4^*) = E^*$.

Let $P \in K \setminus E$ and $E' = (P, E_2, E_3, E_4)$. We also have $E' \hookrightarrow A$ or $E'^* \hookrightarrow A$, so there is a subset $B \subseteq A$ of size 4 which is homothetic to E' or E'^* . Let $\psi \in \mathbb{H}$ be the map corresponding to $(E'$ or $E'^*) \hookrightarrow B$. Notice that

$$\text{slp}(A_1 A_2) \stackrel{\phi}{=} \text{slp}(E_1^* E_2^*) = \text{slp}(E_3 E_4) \in \text{slp}(E') = \text{slp}(E'^*) \stackrel{\psi}{=} \text{slp}(B).$$

As we assumed that A has distinct slopes, this implies that $A_1, A_2 \in B$. The same argument with $\text{slp}(A_1 A_3)$ and $\text{slp}(A_1 A_4)$ shows that all the A_i 's are in B , so $B = \{A_1, A_2, A_3, A_4\}$.

To summarize, we have $(E'$ or $E'^*) \stackrel{\psi}{\hookrightarrow} B = \{A_1, \dots, A_4\} \stackrel{\phi}{\hookrightarrow} E^*$. In particular, we deduce that $\text{slp}(E') = \text{slp}(B) = \text{slp}(E)$. Therefore,

$$\{\text{slp}(PE_2), \text{slp}(PE_3), \text{slp}(PE_4)\} = \{\text{slp}(E_1 E_2), \text{slp}(E_1 E_3), \text{slp}(E_1 E_4)\},$$

because all the slopes of B are distinct. We can repeat the preceding argument with

(P, E_1, E_3, E_4) , (P, E_1, E_2, E_4) and (P, E_1, E_2, E_3) in place of E' and deduce the equalities

$$\begin{cases} \{\text{slp}(PE_1), \text{slp}(PE_3), \text{slp}(PE_4)\} = \{\text{slp}(E_2E_1), \text{slp}(E_2E_3), \text{slp}(E_2E_4)\} \\ \{\text{slp}(PE_1), \text{slp}(PE_2), \text{slp}(PE_4)\} = \{\text{slp}(E_3E_1), \text{slp}(E_3E_2), \text{slp}(E_3E_4)\} \\ \{\text{slp}(PE_1), \text{slp}(PE_2), \text{slp}(PE_3)\} = \{\text{slp}(E_4E_1), \text{slp}(E_4E_2), \text{slp}(E_4E_3)\}. \end{cases}$$

Taking the union of the left-hand sides and right-hand sides yields

$$\{\text{slp}(PE_1), \text{slp}(PE_2), \text{slp}(PE_3), \text{slp}(PE_4)\} = \text{slp}(E),$$

which is a contradiction since $|\text{slp}(E)| = |\text{slp}(B)| = 6$.

For every $E \subset K$ with $|E| = 4$, we have proven that there exists a transformation $\phi_E \in \mathbb{H}$ such that $\phi_E(E) \subseteq A$. We will now prove that all ϕ_E are equal, which concludes the proof of the lemma. It is sufficient to prove that $\phi_E = \phi_{E'}$ whenever E, E' are two subsets of four elements of K with $|E \cap E'| = 3$. Let $E \cap E' = \{E_1, E_2, E_3\}$. For $1 \leq i < j \leq 3$, we have

$$\text{slp}(\phi_E(E_i)\phi_E(E_j)) = \text{slp}(E_iE_j) = \text{slp}(\phi_{E'}(E_i)\phi_{E'}(E_j)).$$

Since A has distinct slopes, it must be the case that $\{\phi_E(E_i), \phi_E(E_j)\} = \{\phi_{E'}(E_i), \phi_{E'}(E_j)\}$. It follows that the restrictions $\phi_E|_{E \cap E'}$ and $\phi_{E'}|_{E \cap E'}$ are equal. An affine transformation is uniquely determined by its action on three non-collinear points, so $\phi_E = \phi_{E'}$ as claimed. \square

Lemma 3.8. *There exists an algorithm to compute a slope-generic set of size n in time polynomial in n . Moreover, the coordinates of the constructed points are integers with polynomially many digits, and every slope determined by the set is an integer.*

We postpone the proof of this lemma to Section 3.4 and concentrate on the main theorem.

3.3 Proof of NP-completeness

Let us recall the problem that will be shown to be NP-complete, as well as some variants which we mentioned in the introduction.

Definition 3.9. The *slope-constrained complete graph drawing problem* (SCGD for short) [restricted to “ $n \leq f(k)$ ”], is the following decision problem.

- Input.**
- A set S of n slopes;
 - A natural number $k \leq n \lfloor \leq f(k) \rfloor$.
- Output.**
- YES if there exists a simple set K of k points in \mathbb{R}^2 such that $\text{slp}(K) \subseteq S$;
 - NO otherwise.

The condition $k \leq n$ is not restrictive as every simple set of k points determines at least k slopes.

For the model of computation, we will work with the common logarithmic-cost integer RAM model [2]. The size of the input (S, k) is measured by the number of bits used to represent k and the slopes of S (k may be ignored since $k \leq |S|$).

This model of computation does not allow the manipulation of arbitrary real numbers. This is not an issue: to compare the computational complexity of two problems, it is necessary to use similar models of computation for both. If we were to use, say, the real RAM model, we would not be able to talk about NP-complete problems in the usual sense.

In order to study the complexity class of the SCGD problem, we need to specify what inputs are allowed. Since the input slopes $s \in S$ must be representable in our model of computation, the two most natural choices might be to only consider rational slopes, or integer slopes. We will assume that the slopes are integers. This is not a problem: the more we restrict the possible inputs (keeping the same model of computation), the easier the problem becomes. Since the SCGD problem is already NP-complete when restricted to integer slopes, it will still be NP-hard (and thus NP-complete, by the same proof) for rational slopes.⁴

Theorem 3.10. *The SCGD problem is NP-complete.*

Proof. We begin by showing that the SCGD problem is in NP. Suppose that there exists a simple set $K = \{K_1, \dots, K_k\}$ of k points with $\text{slp}(K) \subseteq S$.⁵ Let $s_{i,j} := \text{slp}(K_i K_j) \in S$, for $1 \leq i < j \leq k$. Consider the system of linear equations $Y_j - Y_i = s_{i,j}(X_j - X_i)$. A non-trivial solution corresponds to an instance of a simple set of k points with slopes contained in S . Let the witness be the list of triples $(i, j, s_{i,j})$, for $1 \leq i < j \leq k$. It is polynomial in the size of the input (S, k) . If the witness is given, verifying that the corresponding system of linear equations has a non-trivial solution is also polynomial in the size of the input. Finally, it is a polynomial time check to verify that $s_{i,j} \in S$ and that $s_{i,j} \neq s_{j,k}$ for all distinct i, j, k (the latter condition ensuring that any non-trivial solution of the system yields a simple point set).

We now prove that SCGD is NP-hard, by showing that CLIQUE⁶ can be polynomially reduced to the SCGD problem.

Let $G = (V, E)$ be a finite graph and let k be a positive integer. If $k \leq 4$, solving the clique problem with input (G, k) takes polynomial time. Therefore, we only consider the case $k \geq 5$. The idea is to consider an embedding of V into a slope-generic set. We construct a slope-generic set A of size $|V|$ in polynomial time using the algorithm of Lemma 3.8. Fix a bijection $f : V \rightarrow A$. Let

$$S = \{\text{slp}(f(v)f(w)) \mid vw \text{ is an edge in } G\} \subset \mathbb{Z}.$$

We execute the hypothetical SCGD algorithm with input (S, k) . We claim that the output of the algorithm (YES or NO) is exactly the answer to the CLIQUE problem with input (G, k) .

⁴Or for any other reasonable choice containing the integers, representable within the logarithmic-cost integer RAM model.

⁵We may not directly use K as a witness, as the coordinates of the points of K could be arbitrary real numbers. Since the slopes are integers, it is true that there always exists another choice of K whose points have integer coordinates. However, we would also need to explain that K can be chosen to be representable with polynomially many bits. Instead, we choose a more indirect witness already containing all the necessary information.

⁶The CLIQUE decision problem is the following: given a graph G and a positive integer k , decide whether G contains a clique of size k . It is one of the first problems that was shown to be NP-complete [15].

- If the output is **NO**, there could not have been a k -clique in G . Indeed, by contraposition: let H be a k -clique in G . Then $f(H)$ is a simple set of k points in the plane having slopes in S .
- If the output is **YES**, there is a simple set K in the plane of size k with $\text{slp}(K) \subseteq S \subseteq \text{slp}(A)$. As A is slope-generic, $K \hookrightarrow A$ by Lemma 3.7. Thus, there is no loss of generality in assuming that $K \subseteq A$. The proof of the claim is completed by showing that $f^{-1}(K)$ is a k -clique of G . Let $A_1, A_2 \in K$. We know that $\text{slp}(A_1 A_2) \in S$, which means $\text{slp}(A_1 A_2) = \text{slp}(f(v)f(w))$ for some edge vw in G . As A has distinct slopes, we deduce that $\{A_1, A_2\} = \{f(v), f(w)\}$, which implies that $f^{-1}(A_1)f^{-1}(A_2)$ is an edge of G .

To conclude, we check that the reduction is polynomial-time (in the size of G).

- By Lemma 3.8, the set A can be computed in polynomial time (with respect to $|V|$). The coordinates of the points of A have polynomially many digits, so the computation time of each $\text{slp}(f(v)f(w))$ is polynomial in $|V|$.⁷ Thus, the computation of S is polynomial-time in the size of G .
- The size of the input (S, k) is polynomial in the size of G , which concludes the proof. \square

Remark 3.11. It is well-known that the **HALFCLIQUE**⁸ problem is NP-complete [25, Chapter 7]. We can apply the same proof with $k = \lceil |V|/2 \rceil$ to get a reduction from **HALFCLIQUE** to the **SCGD** problem. With the notation from the proof, we have $|S| \leq \binom{|V|}{2} \leq 2k^2$. Therefore, the **SCGD** problem restricted to $n \leq 2k^2$ is also NP-complete. No attempt has been made here to reduce the constant in the inequality.

3.4 Construction of slope-generic sets

This section is devoted to the proof of Lemma 3.8. The following lemma gives a condition for six real numbers to constitute the slope set of a set of four points.

Lemma 3.12. *Let $(m_{i,j})_{1 \leq i < j \leq 4}$ be six real numbers. Assume that there exist four distinct points E_1, \dots, E_4 in the plane such that $\text{slp}(E_i E_j) = m_{i,j}$ for all $1 \leq i < j \leq 4$. Then*

$$Q(m_{1,2}, m_{1,3}, m_{1,4}, m_{2,3}, m_{2,4}, m_{3,4}) = 0,$$

where Q is the polynomial

$$Q(z_1, \dots, z_6) := (z_3 - z_5)(z_6 - z_2)(z_4 - z_1) - (z_2 - z_4)(z_5 - z_1)(z_6 - z_3).$$

Proof. We can suppose that $E_1 = (0, 0)$, translating the four points if necessary. Consider the linear system given by the six equations

$$y_j - y_i - m_{i,j}(x_j - x_i) = 0, \quad 1 \leq i < j \leq 4,$$

⁷By construction (see the proof of Lemma 3.8 in Section 3.4), the slope of the line $f(v)f(w)$ is an integer, which is just the sum of the x -coordinates of $f(v)$ and $f(w)$.

⁸The **HALFCLIQUE** problem is the task of deciding, given a graph G as input, whether G contains a clique of size $\lceil n/2 \rceil$, where n is the number of vertices of G .

where the six unknowns are $x_2, x_3, x_4, y_2, y_3, y_4$ and we fixed $x_1 = y_1 = 0$. It admits the trivial solution where all variables are zero. By assumption, there is another solution, given by $(x_i, y_i) = E_i$ for $2 \leq i \leq 4$. Hence, the determinant of the system vanishes. This determinant computes to

$$Q(m_{1,2}, m_{1,3}, m_{1,4}, m_{2,3}, m_{2,4}, m_{3,4}),$$

concluding the proof of the lemma. \square

We will also use the existence of integer sequences with polynomial growth avoiding certain additive configurations.

Definition 3.13 (Generalized Sidon Sequences). A strictly increasing sequence \mathcal{C} of positive integers is a B_h -sequence if there is no integer $n \geq 1$ which can be expressed as the sum of exactly h (non-necessarily distinct) elements of \mathcal{C} in two different ways.

Lemma 3.14. *Let $h \geq 2$ be fixed. There is a strictly increasing B_h -sequence $(c_i)_{i \in \mathbb{N}}$ and an algorithm (the “classic greedy algorithm”) such that*

1. *the sequence has polynomial growth, more precisely: $c_n = \mathcal{O}(n^{2h-1})$;*
2. *the algorithm computes c_1, \dots, c_n in polynomial time (with respect to n).⁹*

For a treatment of a more general case, we refer the reader to [4] (the proof of Lemma 3.14 can be found in the introduction). Lemma 3.15 is the last step before the proof of Lemma 3.8.

Lemma 3.15. *The set $A := \{(50^{c_i}, 50^{2c_i}) \mid i \geq 1\}$ is a slope-generic set for any B_3 -sequence $\mathcal{C} = (c_i)_{i \geq 1}$.*

Let us make some comments before starting the proof. In the proof of Theorem 3.10, we needed to be able to exhibit slope-generic sets of arbitrary size. The intuition given in Section 3.1 was that most sets are slope-generic. However, to give an explicit example, one must verify the condition in Definition 3.5 for a concrete set A . This is not an easy task: we have to check the condition $(E \hookrightarrow A \text{ or } E^* \hookrightarrow A)$ for every possible simple list E of four points with $\text{slp}(E) \subseteq \text{slp}(A)$. For a finite set A , this is a finite computation (considering E modulo \mathbb{H}). Here, we give a family of infinite slope-generic sets. The points of A are chosen on the parabola $y = x^2$ in order for both the slopes and the points to be integers with very simple expressions (the same idea was used in [13, Theorem 8.3]).

Proof of Lemma 3.15. Let $A_i = (50^{c_i}, 50^{2c_i})$ for $i \in \mathbb{N}$. As $\text{slp}(A_i A_j) = 50^{c_i} + 50^{c_j}$ for $i \neq j$, it is clear that A has distinct slopes.

Suppose that $E := \{E_1, \dots, E_4\}$ is a simple set of four points with $\text{slp}(E) \subseteq \text{slp}(A)$. We have to show that $E \hookrightarrow A$ or $E^* \hookrightarrow A$. We let $m_{i,j} := \text{slp}(E_i E_j)$ for $i < j$. As $m_{1,2} \in \text{slp}(A)$, there are two integers $x_1 \neq x_2$ in the sequence \mathcal{C} such that $m_{1,2} = 50^{x_1} + 50^{x_2}$. Similarly, $m_{1,3} = 50^{x_3} + 50^{x_4}$, and so on, until $m_{3,4} = 50^{x_{11}} + 50^{x_{12}}$, for some elements of \mathcal{C} with $x_{2i-1} \neq x_{2i}$ (see Figure 6).

⁹The degree of the polynomial depends on h .

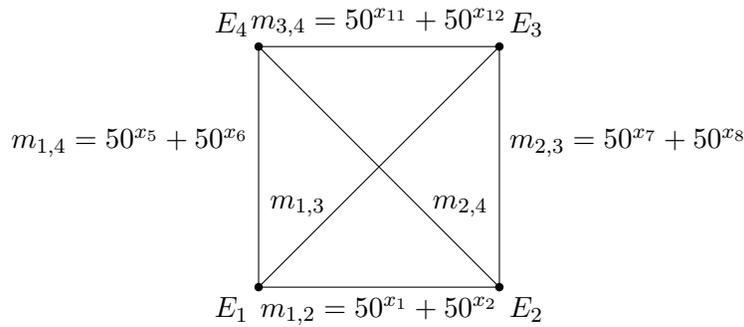


Figure 6: Schematic representation of the set E with its six slopes $m_{i,j}$.

Since E is simple, two lines $E_iE_{j_1}$ and $E_iE_{j_2}$ passing through the same point E_i cannot have the same slope if $j_1 \neq j_2$. For example, this tells us that $m_{1,2} \neq m_{2,3}$, i.e. $50^{x_1} + 50^{x_2} \neq 50^{x_7} + 50^{x_8}$, which is equivalent to $\{x_1, x_2\} \neq \{x_7, x_8\}$.

Putting all the constraints together, we have¹⁰

$$\begin{cases} \text{For } 1 \leq i \leq 6, x_{2i-1} \neq x_{2i}; \\ \text{For } 1 \leq i, j \leq 6 \text{ and } j \notin \{i, 7-i\}, \{x_{2i-1}, x_{2i}\} \neq \{x_{2j-1}, x_{2j}\}. \end{cases} \quad (1)$$

By Lemma 3.12 with the slopes $m_{i,j}$, we know that

$$Q(50^{x_1} + 50^{x_2}, \dots, 50^{x_{11}} + 50^{x_{12}}) = 0. \quad (2)$$

Unless many x_i 's are actually equal, an equality as (2) is unlikely to hold. The reason is that the polynomial Q (of relatively small degree and coefficients) cannot vanish when evaluated at integers of completely different orders of magnitude.

The goal is to use the constraints (1) and (2) to prove that the x_i 's can take only four distinct values and to know for which indices i, j we have $x_i = x_j$.¹¹ We will see that there are exactly two possibilities, depicted in Figure 7 (here, y_1, \dots, y_4 are distinct integers and each x_i is equal to one of the y_j 's).

Let $r = |\{x_1, \dots, x_{12}\}|$ be the number of distinct x_i 's. We will show that (2) does not only hold as an equality between integers, but also holds “formally” (or “symbolically”). To state this precisely, we use the language of polynomial rings. Let $\mathbb{Z}[T_1, \dots, T_r]$ be the polynomial ring in r indeterminates. We choose a bijection $\mathcal{J} : \{x_1, \dots, x_{12}\} \rightarrow \{T_1, \dots, T_r\}$.

Claim. $Q(\mathcal{J}(x_1) + \mathcal{J}(x_2), \dots, \mathcal{J}(x_{11}) + \mathcal{J}(x_{12}))$ is the zero polynomial in $\mathbb{Z}[T_1, \dots, T_r]$.

Essentially, the claim means that, if we replace each x_i by a formal indeterminate, consistently (meaning that, if $x_i = x_j$, we replace x_i and x_j by the same indeterminate), (2) still holds. To be precise, it is 50^{x_i} that we replace with some indeterminate in $\{T_1, \dots, T_r\}$.

¹⁰The case $j = 7 - i$ corresponds to two lines E_iE_j and E_kE_l with $\{i, j\} \cap \{k, l\} = \emptyset$, which could be parallel a priori. For instance, we do not know (yet) that $m_{1,2} \neq m_{3,4}$.

¹¹We can compare this with Figure 2 from Section 3.1. If the integers x_i were to take more than four different values, it would mean that $\text{slp}(E)$ (which corresponds to T on Figure 2) is not the slope set of a subset B of four points of A .

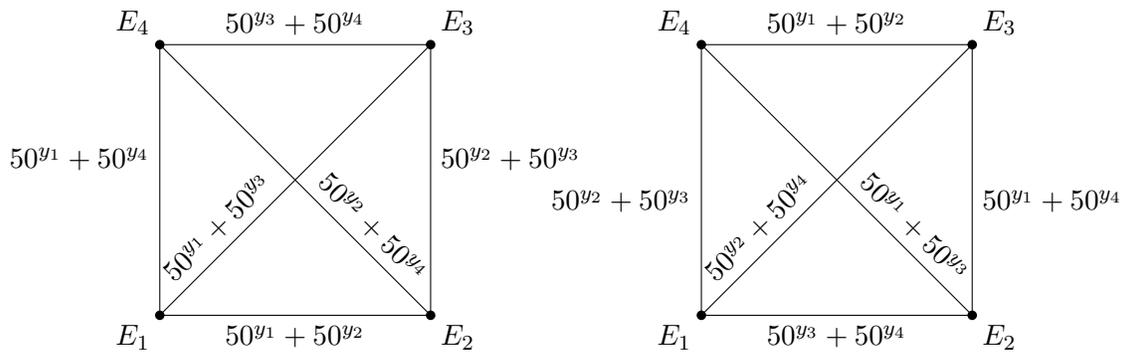


Figure 7: The two possibilities for the slopes of E , given the constraints (1) and (2).

Proof of claim. If we expand the left-hand side of (2), simplify, and move half of the terms to the right-hand side, we get an equation of the form

$$\sum_{i=1}^{48} 50^{x_{L_1(i)} + x_{L_2(i)} + x_{L_3(i)}} = \sum_{i=1}^{48} 50^{x_{R_1(i)} + x_{R_2(i)} + x_{R_3(i)}} \tag{3}$$

for some known maps $L_k, R_k : \{1, 2, \dots, 48\} \rightarrow \{1, 2, \dots, 12\}$, $k = 1, 2, 3$ (these maps are just obtained by replacing the polynomial Q by its exact expression, given in Lemma 3.12). Since the exponents are natural numbers and each sum contains less than fifty terms, equation (3) is equivalent to

$$\exists \sigma \in \mathfrak{S}_{48}, \forall i \in [1, 48], \quad x_{L_1(i)} + x_{L_2(i)} + x_{L_3(i)} = x_{R_1(\sigma(i))} + x_{R_2(\sigma(i))} + x_{R_3(\sigma(i))}. \tag{4}$$

Using the fact that $\{x_1, \dots, x_{12}\}$ is part of the B_3 -sequence \mathcal{C} , we can rewrite (4) as

$$\exists \sigma \in \mathfrak{S}_{48}, \forall i \in [1, 48], \exists \tau \in \mathfrak{S}_3, \forall k \in [1, 3], \quad x_{L_k(i)} = x_{R_{\tau(k)}(\sigma(i))}. \tag{5}$$

To shorten notation, we write X_i instead of $\mathcal{J}(x_i)$.¹² Thus, each X_i is an indeterminate and, since \mathcal{J} is a bijection, $X_i = X_j$ if and only if $x_i = x_j$. In particular, (1) becomes

$$\begin{cases} \text{For } 1 \leq i \leq 6, X_{2i-1} \neq X_{2i}; \\ \text{For all } i, j \text{ with } j \notin \{i, 7-i\}, \{X_{2i-1}, X_{2i}\} \neq \{X_{2j-1}, X_{2j}\}. \end{cases} \tag{1'}$$

Applying the bijection, (5) translates to

$$\exists \sigma \in \mathfrak{S}_{48}, \forall i \in [1, 48], \exists \tau \in \mathfrak{S}_3, \forall k \in [1, 3], \quad X_{L_k(i)} = X_{R_{\tau(k)}(\sigma(i))}. \tag{5'}$$

Just like above, this equation can be rewritten as

$$\exists \sigma \in \mathfrak{S}_{48}, \forall i \in [1, 48], \quad X_{L_1(i)} X_{L_2(i)} X_{L_3(i)} = X_{R_1(\sigma(i))} X_{R_2(\sigma(i))} X_{R_3(\sigma(i))},$$

then as

$$\sum_{i=1}^{48} X_{L_1(i)} X_{L_2(i)} X_{L_3(i)} = \sum_{i=1}^{48} X_{R_1(i)} X_{R_2(i)} X_{R_3(i)},$$

¹²Note that X_i is not a new indeterminate, it is just a notation for the indeterminate T_j which is associated to x_i .

and finally (by definition of the functions L_k, R_k) as

$$Q(X_1 + X_2, \dots, X_{11} + X_{12}) = 0, \tag{2'}$$

which proves the claim. □

We will now see what the constraints (1') and (2') imply about the X_i 's. We will use the fact that (1') and (2') are (in)equalities in the more convenient ring $\mathbb{Z}[T_1, \dots, T_r]$. By definition of Q (cf. Lemma 3.12), the equality (2') is equivalent to

$$(Z_3 - Z_5)(Z_6 - Z_2)(Z_4 - Z_1) = (Z_2 - Z_4)(Z_5 - Z_1)(Z_6 - Z_3), \tag{6}$$

where we used the notations $(Z_1, Z_2, \dots, Z_6) := (X_1 + X_2, X_3 + X_4, \dots, X_{11} + X_{12})$ for the equation to fit on a single line.

Let us look more closely at the factors of (6). They are all of the form $Z_i - Z_j$ for some pairs (i, j) with $j \notin \{i, 7 - i\}$. As Z_i is just an abbreviation for $X_{2i-1} + X_{2i}$, we have $Z_i - Z_j = X_{2i-1} + X_{2i} - X_{2j-1} - X_{2j}$. By (1'), we know that $X_{2i-1} \neq X_{2i}$, $X_{2j-1} \neq X_{2j}$ and $\{X_{2i-1}, X_{2i}\} \neq \{X_{2j-1}, X_{2j}\}$, since $j \notin \{i, 7 - i\}$. Remembering that each of $X_{2i-1}, X_{2i}, X_{2j-1}, X_{2j}$ is an element of $\{T_1, \dots, T_r\}$, we observe that $Z_i - Z_j$ must be a (non-zero) homogeneous polynomial with content 1 and (total) degree 1.¹³ Thus, for these pairs (i, j) , the polynomial $Z_i - Z_j$ is irreducible, hence prime, in the unique factorization domain $\mathbb{Z}[T_1, \dots, T_r]$. We have just proved that each side of (6) is a product of prime elements. By unique factorisation in $\mathbb{Z}[T_1, \dots, T_r]$, we can thus identify each factor on one side with one of the factors on the other side (up to a sign).

From the single equation (6), we thus deduce three equations (for every choice of signs and every permutation of the factors of the right-hand side). For example, if we choose all the signs to be + and we don't permute the factors, the three equations are

$$\begin{cases} Z_3 - Z_5 = Z_2 - Z_4 \\ Z_6 - Z_2 = Z_5 - Z_1 \\ Z_4 - Z_1 = Z_6 - Z_3. \end{cases} \tag{7}$$

Let us continue this example, and come back to the general case afterwards. After replacing the Z_i 's by their definition and noticing that the last equation in (7) is redundant, we get

$$X_1 + X_2 + X_{11} + X_{12} = X_3 + X_4 + X_9 + X_{10} = X_5 + X_6 + X_7 + X_8. \tag{8}$$

Since every X_i is an indeterminate T_j in $\mathbb{Z}[T_1, \dots, T_r]$, the only way (8) can hold is if X_3, X_4, X_9, X_{10} and X_5, X_6, X_7, X_8 are permutations of X_1, X_2, X_{11}, X_{12} . Not every pair of permutations is allowed: the constraints (1') must still be satisfied. Two ways that (1') and (8) can simultaneously be verified are as follows.

$$(i) \begin{cases} X_1 = X_3 = X_5 \\ X_2 = X_9 = X_7 \\ X_{11} = X_4 = X_8 \\ X_{12} = X_{10} = X_6 \\ X_1, X_2, X_{11}, X_{12} \text{ pairwise distinct} \end{cases} \quad (ii) \begin{cases} X_{11} = X_9 = X_7 \\ X_{12} = X_3 = X_5 \\ X_1 = X_{10} = X_6 \\ X_2 = X_4 = X_8 \\ X_{11}, X_{12}, X_1, X_2 \text{ pairwise distinct} \end{cases} \tag{9}$$

¹³The content of a polynomial over \mathbb{Z} is the greatest common divisor of its coefficients.

We now return to the general case. We use a computer program to check all the possibilities (the source code can be found in Appendix A). With the help of the program, we conclude the following.

1. There is only one way to choose three signs and a permutation of the factors that does not lead to a contradiction (it is the choice made in (7)).
2. There are 2^7 possibilities in total. Up to the symmetries $X_{2i-1} \leftrightarrow X_{2i}$ (there are 2^6 combinations of such transpositions), there are actually only two possibilities, given by (i) and (ii) from (9).

We can now prove that A is slope-generic. Recall that $X_i = X_j$ if and only if $x_i = x_j$. Without loss of generality (performing exchanges $X_{2i-1} \leftrightarrow X_{2i}$ and $x_{2i-1} \leftrightarrow x_{2i}$ if necessary), we may thus assume to be in one of the following two cases.

$$(i) \begin{cases} y_1 := x_1 = x_3 = x_5 \\ y_2 := x_2 = x_9 = x_7 \\ y_3 := x_{11} = x_4 = x_8 \\ y_4 := x_{12} = x_{10} = x_6 \\ y_1, y_2, y_3, y_4 \text{ pairwise distinct} \end{cases} \quad (ii) \begin{cases} y_1 := x_{11} = x_9 = x_7 \\ y_2 := x_{12} = x_3 = x_5 \\ y_3 := x_1 = x_{10} = x_6 \\ y_4 := x_2 = x_4 = x_8 \\ y_1, y_2, y_3, y_4 \text{ pairwise distinct} \end{cases}$$

In each case, we define the distinct points $B_i := (50^{y_i}, 50^{2y_i})$, for $1 \leq i \leq 4$. The subset $B = \{B_1, B_2, B_3, B_4\}$ of A is our candidate to verify the slope-genericity of A with respect to E .

- In the case (i), we have $E \sim B$. To see this, note that the quadruples (E_1, E_2, E_3, E_4) and (B_1, B_2, B_3, B_4) are exactly in the configuration of Remark 2.3. This is an immediate verification: for example, one has $\text{slp}(E_2 E_4) = 50^{x_9} + 50^{x_{10}} = 50^{y_2} + 50^{y_4} = \text{slp}(B_2 B_4)$. Hence, $E \hookrightarrow A$.
- In the case (ii), write $E^* = (F_1, \dots, F_4)$. This time, (F_1, F_2, F_3, F_4) and (B_1, B_2, B_3, B_4) are in the configuration of Remark 2.3, so $E^* \sim B$ and $E^* \hookrightarrow A$.

Remark 3.16. By exploiting symmetry, it is possible to work out all the cases by hand, instead of using a computer program.

Remark 3.17. In the previous subsections, we said that one could think of a slope-generic set as any ‘‘sufficiently random’’ set of points. We now give a more concrete explanation of this intuition, in the light of the proof of Lemma 3.15 (supposing that we do not require the slopes to be integers anymore). Let A be a set of points, and let E be a simple set of four points such that $\text{slp}(E) = \text{slp}(A)$. Thus, every slope $m_{i,j}$ of E can be written as $m_{i,j} = \frac{y_{i,j,2} - y_{i,j,1}}{x_{i,j,2} - x_{i,j,1}}$ for some points $(x_{i,j,k}, y_{i,j,k})$ of A . By Lemma 3.12, we have

$$Q \left(\frac{y_{1,1,2} - y_{1,1,1}}{x_{1,1,2} - x_{1,1,1}}, \frac{y_{1,2,2} - y_{1,2,1}}{x_{1,2,2} - x_{1,2,1}}, \dots, \frac{y_{3,4,2} - y_{3,4,1}}{x_{3,4,2} - x_{3,4,1}} \right) = 0. \quad (10)$$

If A is ‘‘random enough’’, equation (10) cannot hold by an ‘‘arithmetic coincidence’’. Following the proof of Lemma 3.15, this means that (10) still holds true when we replace the $x_{i,j,k}$ ’s and $y_{i,j,k}$ ’s by formal variables, in a ‘‘consistent’’ way. Once we have the formal equality, it is

conceivable that the same type of arguments as in the second half of the proof can yield the desired result, i.e. that $E \hookrightarrow A$ or $E^* \hookrightarrow A$. Since this is not needed for our main theorem, we will not give more details.

We can now prove Lemma 3.8.

Lemma 3.8. *There exists an algorithm to compute a slope-generic set of size n in time polynomial in n . Moreover, the coordinates of the constructed points are integers with polynomially many digits, and every slope determined by the set is an integer.*

Proof. First, a B_3 -sequence $c_1 < \dots < c_n$ is calculated in polynomial time using the greedy algorithm of Lemma 3.14. Then, one computes $A_i = (50^{c_i}, 50^{2c_i})$ for $1 \leq i \leq n$, and returns $\{A_1, \dots, A_n\}$. Because $c_i = \mathcal{O}(n^{2 \cdot 3^{-1}})$, the computation of the powers 50^{c_i} and 50^{2c_i} is indeed polynomial in n . \square

Remark 3.18. The property that the sequence (c_i) grows polynomially is crucial in the logarithmic-cost model of computation. In the uniform-cost model, we could just have chosen the B_3 -sequence $c_i = 4^i$. The number 50^{4^n} can be computed in linear time by repeated squaring, even though this number has exponentially many digits. This is the reason why the uniform-cost RAM model (with multiplication) is not considered to be a reasonable model of computation (see [12, pp. 177-178] and [2, §2.2.2]).

4 Algorithms for the restricted SCGD problem

In this section, we present two polynomial algorithms for the SCGD problem when the number of slopes $n = |S|$ is not much larger than k . The first one (Proposition 4.6) is deterministic, the other one (Proposition 4.8) is probabilistic.

4.1 Affinely-regular polygons

We give two equivalent definitions and some elementary properties of affinely-regular polygons that can be found in [5]. For a survey of affinely-regular polygons over an arbitrary field, we refer the reader to [7].

Definition 4.1. Let $n \geq 3$. An affinely-regular n -gon is a finite set of points P satisfying one of the following equivalent properties:

- P is the image of a regular n -gon under some $\psi \in \text{Aff}(2, \mathbb{R})$;
- $P = \{\phi^i(P_0) \mid i \in \mathbb{Z}\}$ for some $\phi \in \text{Aff}(2, \mathbb{R})$ of order n and some $P_0 \in \mathbb{R}^2$.

Fact 4.2. Let P be an affinely-regular polygon with vertices P_0, \dots, P_{n-1} , in cyclic order (say counterclockwise). Let ϕ be the unique affine transformation such that $\phi(P_i) = P_{i+1}$ for $i = 0, 1, 2$. Then, considering the indices modulo n , we have

1. ϕ has order n and $\phi^i(P_0) = P_i$ for all i ;
2. $\text{slp}(P_{i-k}P_{j+k}) = \text{slp}(P_iP_j)$ for all i, j, k with $i \neq j$;

3. If $s_0 := \text{slp}(P_{n-1}P_1)$ and $s_i := \text{slp}(P_0P_i)$ for $1 \leq i \leq n-1$, then the slopes of P are precisely s_0, s_1, \dots, s_{n-1} , in this (cyclic) order;
4. The sequence of boundary slopes $(\text{slp}(P_0P_1), \text{slp}(P_1P_2), \dots, \text{slp}(P_{n-1}P_0))$ is

$$\begin{cases} (s_1, s_3, s_5, \dots, s_{n-2}, s_0, s_2, \dots, s_{n-1}) & \text{if } n \text{ is odd;} \\ (s_1, s_3, s_5, \dots, s_{n-1}, s_1, s_3, \dots, s_{n-1}) & \text{if } n \text{ is even.} \end{cases}$$

We also recall Jamison's conjecture on affinely-regular polygons.

Conjecture (Jamison). For some constant c_1 , the following holds. If $n \leq 2k - c_1$, every simple set of k points forming (exactly) n slopes is contained in an affinely-regular n -gon.

4.2 Model of computation

We want to give an algorithm for the SCGD problem when Jamison's conjecture applies, i.e. when $n \leq 2k - c_1$. Assuming the conjecture, the sets K that satisfy the SCGD problem are subsets of affinely-regular polygons. However, affinely-regular n -gons have irrational slopes as soon as $n \neq 3, 4, 6$ (because $\cos(2\pi/n)$ has degree $\phi(n)/2$ over \mathbb{Q} , as was proven by D. H. Lehmer [18]). The problem is thus trivial if the slopes given as inputs are integers, as in Section 3.

We will therefore allow the slopes to be arbitrary real numbers,¹⁴ and adopt the *real RAM model* described in [24]: the primitive arithmetic operations $+, -, \cdot, /$ and comparisons on real numbers are available at unit time cost.

4.3 Deterministic algorithm

We start by solving a problem similar to the restricted SCGD problem when four points of the set K are already given as inputs.

Lemma 4.3. *There is a deterministic algorithm with time complexity $\mathcal{O}(n)$ for the following problem.*

- Input.**
- A sorted list S of n slopes;
 - A natural number $k \leq n$;
 - A simple list (P_0, \dots, P_3) of four points.

Problem. *Does there exist a point set K satisfying the following conditions?*

- (i) K forms an affinely-regular polygon;
- (ii) P_0, \dots, P_3 are consecutive points of K (in that order);
- (iii) $k \leq |K| \leq n$;
- (iv) $\text{slp}(K) \subseteq S$.

Proof. Let ϕ be the unique affine transformation that maps P_j to P_{j+1} , for $j = 0, 1, 2$. By Fact 4.2, if there exists a set K satisfying (i), (ii) and (iii), ϕ must have order $|K| \in [k, n]$ (and K is the orbit of P_0 under ϕ). This explains the first steps of the algorithm.

¹⁴An alternative would be to restrict to algebraic numbers, as explained in Yap [27].

- 1: Compute ϕ (as a 3×3 matrix). Compute ϕ^j , $1 \leq j \leq n$.
- 2: If the order of ϕ is in the interval $[k, n]$, call it d . Otherwise, return NO.

Suppose now that ϕ has order $d \in [k, n]$. Let $K := \{\phi^j(P_0) \mid 0 \leq j < d\}$. By Definition 4.1 and Fact 4.2, K is the unique affinely-regular polygon satisfying (i), (ii) and (iii). This means that we only need to check whether K satisfies $\text{slp}(K) \subseteq S$.

- 3: We compute the slopes of K . Let $s_0 = \text{slp}(P_1\phi^{d-1}(P_0))$ and $s_j = \text{slp}(P_0\phi^j(P_0))$, for $1 \leq j \leq d-1$. By Fact 4.2, the slopes of K are exactly s_0, \dots, s_{d-1} , in this order.
- 4: Return YES if $\text{slp}(K) \subseteq S$, and NO otherwise. It is possible to check the inclusion in linear time since both sides are sorted. \square

Now, we suppose that we already know four consecutive slopes determined by K .

Lemma 4.4. *There is a deterministic $\mathcal{O}(n)$ algorithm for the following problem.*

- Input.**
- A sorted list S of n slopes;
 - A natural number $k \leq n$;
 - A list (s_0, \dots, s_3) of four distinct slopes.

Problem. *Does there exist a point set K satisfying the following conditions?*

- (I) K forms an affinely-regular polygon;
- (II) s_0, \dots, s_3 are four consecutive slopes of K (in that order);
- (III) $k \leq |K| \leq n$;
- (IV) $\text{slp}(K) \subseteq S$.

Proof. We reduce the task to Lemma 4.3. We first give the two steps of the reduction and then provide the explanations.

- 1: Compute four distinct points P_0, \dots, P_3 (resp. $\tilde{P}_0, \dots, \tilde{P}_3$) satisfying the equalities (11) (resp. (12)) below. Such points exist as s_0, \dots, s_3 are distinct.

$$\text{slp}(P_0P_1) = s_0, \text{slp}(P_0P_2) = s_1, \text{slp}(P_1P_2) = s_2 = \text{slp}(P_0P_3), \text{ and } \text{slp}(P_1P_3) = s_3 \quad (11)$$

$$\text{slp}(\tilde{P}_0\tilde{P}_2) = s_0, \text{slp}(\tilde{P}_1\tilde{P}_2) = s_1 = \text{slp}(\tilde{P}_0\tilde{P}_3), \text{slp}(\tilde{P}_1\tilde{P}_3) = s_2, \text{ and } \text{slp}(\tilde{P}_2\tilde{P}_3) = s_3 \quad (12)$$

- 2: For $\overline{P} \in \{(P_0, \dots, P_3), (\tilde{P}_0, \dots, \tilde{P}_3)\}$, run the algorithm of Lemma 4.3 with S , k and \overline{P} as inputs. Return YES if one of the two outputs is YES, and NO if both are NO.

Let us explain why this algorithm is correct.

- Suppose that there exists an affinely-regular d -gon K , of which s_0, \dots, s_3 are consecutive slopes. Then, by Fact 4.2, at least one of s_0 and s_1 is a boundary slope of K . We have the following alternative (see Figure 8):

- (a) If s_0 is a boundary slope of K , there are four consecutive vertices P'_0, \dots, P'_3 of K such that

$$\text{slp}(P'_0P'_1) = s_0, \text{slp}(P'_0P'_2) = s_1, \text{slp}(P'_1P'_2) = s_2 = \text{slp}(P'_0P'_3), \text{ and } \text{slp}(P'_1P'_3) = s_3. \quad (11')$$

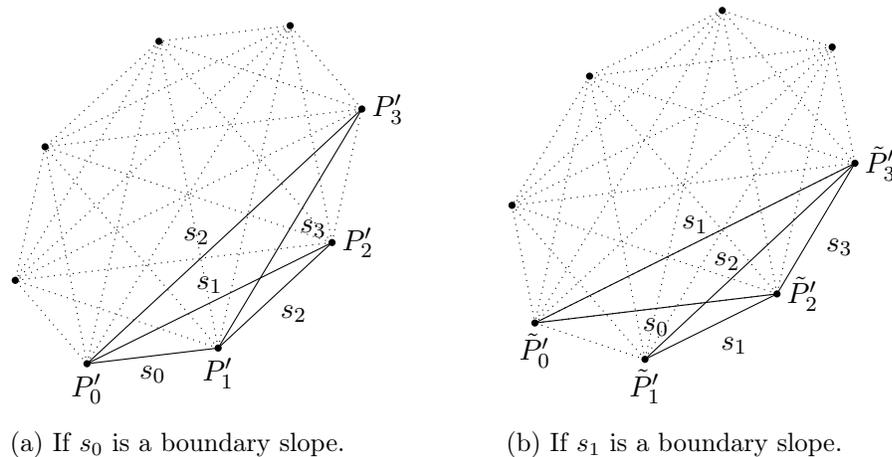


Figure 8: Construction of four consecutive points of K in the proof of Lemma 4.4.

(b) If s_1 is a boundary slope of K , there are four consecutive vertices $\tilde{P}'_0, \dots, \tilde{P}'_3$ of K such that

$$\text{slp}(\tilde{P}'_0\tilde{P}'_1) = s_0, \text{slp}(\tilde{P}'_1\tilde{P}'_2) = s_1 = \text{slp}(\tilde{P}'_0\tilde{P}'_3), \text{slp}(\tilde{P}'_1\tilde{P}'_3) = s_2, \text{and } \text{slp}(\tilde{P}'_2\tilde{P}'_3) = s_3. \quad (12')$$

The conditions (11') uniquely determine the distinct points $\tilde{P}'_0, \dots, \tilde{P}'_3$, up to an element of \mathbb{H} (the group of homotheties and translations). Therefore, in the case (a), we may assume that $\tilde{P}'_0, \dots, \tilde{P}'_3$ are precisely the points constructed in the first step of the algorithm (by applying an element of \mathbb{H} to K if necessary). The same is true with $\tilde{P}'_0, \dots, \tilde{P}'_3$ in the case (b). So we are in the setting of Lemma 4.3.

- Conversely, it is clear that a set K satisfying properties (i) through (iv) of Lemma 4.3 for one of those two quadruples will also satisfy properties (I) through (IV). \square

Remark 4.5. In fact, it is not necessary to consider case (b), because it is possible to prove the following. If s_0 is a slope of an affinely-regular polygon \tilde{K} , there is another affinely-regular polygon K which has s_0 as a boundary slope and such that $\text{slp}(\tilde{K}) = \text{slp}(K)$.

We can now give the claimed algorithm. The problem here is that we do not know a priori which slopes of S will be used by K .

Proposition 4.6. *Assuming Jamison’s conjecture, there is an $\mathcal{O}((n - k + 1)^4n)$ time deterministic algorithm for the SCGD problem restricted to $n \leq 2k - c_1$:*

- Input.**
- A sorted list S of n slopes
 - A natural number k such that $n \leq 2k - c_1$

Problem. *Does there exist a simple set K of k points with $\text{slp}(K) \subseteq S$?*

Proof. If $n < k$, return NO, as every simple set of k points has at least k slopes. By Jamison’s conjecture, the problem is equivalent to: “does there exist an affinely-regular m -gon P with $\text{slp}(P) \subseteq S$ for some $m \geq k$?”. Let $r := n - k$.

1: Let T be the list of the first $r + 4$ slopes of S .

Suppose that there exists an affinely-regular m -gon P with $\text{slp}(P) \subseteq S$ for some $m \geq k$. We have $|\text{slp}(P) \cap T| \geq 4$, as otherwise $|\text{slp}(S)| \geq |\text{slp}(P) \cup T| \geq m + (r + 4) - 3 > n$. So, there exist four slopes of T which are also slopes of P . Let s_0, \dots, s_3 be the first four occurrences of slopes of P in the list T , in this order. By construction, the slopes T were chosen to be consecutive slopes of S . As $\text{slp}(P) \subseteq S$, this implies that s_0, \dots, s_3 must be consecutive slopes of P .

2: For every subsequence of four slopes of T , execute the algorithm of Lemma 4.4 (with the same S and k). If the output is YES for at least one subsequence of four slopes of T , return YES. Otherwise, return NO.

As the algorithm in Lemma 4.4 has runtime $\mathcal{O}(n)$, the time complexity of the full algorithm is $\mathcal{O}\left(\binom{r+4}{4}n\right)$. \square

Remark 4.7. • If we restrict the inputs to have $n \leq k + M$ for some fixed M , we get an algorithm in $\mathcal{O}(n)$. This is the optimal complexity since all the slopes have to be taken into account in the worst case.

- Jamison's conjecture was proven in the cases $n = k$ (by Jamison in his original paper [14]) and $n = k + 1$ (recently, by Pilatte [23]). In those cases, the correctness of our algorithm is independent of any assumption.
- For $n = k$, Wade and Chu presented an algorithm in $\mathcal{O}(n^4)$ for this problem (see [26]). We have thus reduced the complexity to $\mathcal{O}(n)$. For $n = k + 1$, no polynomial algorithm had been proposed before.

4.4 Monte-Carlo algorithm

The previous algorithm has runtime $\mathcal{O}((n - k + 1)^4 n)$, which is $\mathcal{O}(n^5)$ in the worst case. We can improve it to $\mathcal{O}(n)$ if we are willing to use a probabilistic algorithm.

Proposition 4.8. *Assuming Jamison's conjecture, there is a one-sided error Monte-Carlo algorithm with running time $\mathcal{O}(n)$ for the decision problem described in Proposition 4.6.*

Proof. The idea is the same as in the proof of Proposition 4.6: we will find quadruples of four consecutive slopes in S and apply Lemma 4.4 with them.

- 1:** Pick one slope t_1 of S uniformly at random.
 - 2:** Select the slopes t_2, \dots, t_{12} of S such that $T = (t_1, \dots, t_{12})$ is a list of consecutive slopes in S .
 - 3:** For each subsequence of four slopes of T , use Lemma 4.4. If the output is YES for at least one subsequence, return YES. Otherwise, return NO.
- If this algorithm outputs YES, the existence of a valid set K is guaranteed by Lemma 4.4, so the output is correct.

- What is left is to show that the probability of incorrectly outputting NO is bounded away from 1. Suppose that the correct answer is YES. Equivalently, by Jamison's conjecture, there is an affinely-regular polygon P of with $\text{slp}(P) \subseteq S$ and $|P| \geq k$. Let X be the random variable defined by $X = |T \cap \text{slp}(P)|$ (S and P are fixed and T is random). The algorithm outputs YES whenever $X \geq 4$. As

$$|\text{slp}(P)| \geq |P| \geq k \geq \frac{n + c_1}{2} \geq \frac{n}{2} = \frac{|S|}{2},$$

we have

$$\mathbb{E}(X) = \sum_{1 \leq i \leq 12} \mathbb{P}[t_i \in \text{slp}(P)] = 12 \cdot \frac{|\text{slp}(P)|}{|S|} \geq 6.$$

Hence $\mathbb{P}[\text{output is NO}] \leq \mathbb{P}[X < 4] = \mathbb{P}[12 - X \geq 9] \leq 2/3$, by Markov's inequality, which completes the proof. \square

Acknowledgements

I am very grateful to Christian Michaux for his lasting support and valuable advice, and to Pierre Vandenhove for the fruitful discussions that we had about slope-generic sets.

References

- [1] János Barát, Jiří Matoušek, and David R Wood. Bounded-degree graphs have arbitrarily large geometric thickness. *The Electronic Journal of Combinatorics*, 13(1):3, 2006.
- [2] P Van Emde Boas. Machine models and simulations. In *Handbook of Theoretical Computer Science, volume A*, pages 1–66. 2014.
- [3] G. Brückner, N. D. Krisam, and T. Mchedlidze. Level-planar drawings with few slopes. In *International Symposium on Graph Drawing and Network Visualization*, pages 559–572. Springer, 2019.
- [4] Javier Cilleruelo. A greedy algorithm for $B_h[g]$ sequences. *arXiv preprint arXiv:1601.00928*, 2016.
- [5] H. S. M. Coxeter. Affinely regular polygons. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 34, pages 38–58. Springer, 1969.
- [6] Max Engelstein. Drawing graphs with few slopes. *Intel Competition for high school students, New York*, 2005.
- [7] J. Chris Fisher and Robert E. Jamison. Properties of affinely regular polygons. *Geometriae Dedicata*, 69(3):241–259, 1998.
- [8] M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, F. T. Leighton, A. Symvonis, E. Welzl, and G. J. Woeginger. Drawing graphs in the plane with high resolution. *SIAM Journal on Computing*, 22(5):1035–1052, 1993.

- [9] Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001.
- [10] A. Heyting. *Axiomatic projective geometry, 2nd ed.* North-Holland Publishing Company, 1980.
- [11] Udo Hoffmann. On the complexity of the planar slope number problem. *J. Graph Algorithms Appl.*, 21(2):183–193, 2017.
- [12] Juraĵ Hromkoviĉ. *Theoretical computer science: introduction to Automata, computability, complexity, algorithmics, randomization, communication, and cryptography.* Springer, 2003.
- [13] Robert E. Jamison. A survey of the slope problem. *Annals of the New York Academy of Sciences*, 440(1):34–51, 1985.
- [14] Robert E. Jamison. Few slopes without collinearity. *Discrete mathematics*, 60:199–206, 1986.
- [15] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [16] Balázs Keszegh, János Pach, and Dömötör Pálvölgyi. Drawing planar graphs of bounded degree with few slopes. *SIAM Journal on Discrete Mathematics*, 27(2):1171–1183, 2013.
- [17] Balázs Keszegh, János Pach, Dömötör Pálvölgyi, and Géza Tóth. Drawing cubic graphs with at most five slopes. *Computational Geometry*, 40(2):138–147, 2008.
- [18] Derik H. Lehmer. A note on trigonometric algebraic numbers. *The American Mathematical Monthly*, 40(3):165–166, 1933.
- [19] Bruce E. Meserve. *Fundamental concepts of geometry.* Dover Publications, 1983.
- [20] Padmini Mukkamala and Dömötör Pálvölgyi. Drawing cubic graphs with the four basic slopes. In *International Symposium on Graph Drawing*, pages 254–265. Springer, 2011.
- [21] Padmini Mukkamala and Mario Szegedy. Geometric representation of cubic graphs with four directions. *Computational Geometry*, 42(9):842–851, 2009.
- [22] János Pach and Dömötör Pálvölgyi. Bounded-degree graphs can have arbitrarily large slope numbers. *The Electronic Journal of Combinatorics*, 13(1):N1, 2006.
- [23] Cédric Pilatte. On the sets of n points forming $n + 1$ directions. *The Electronic Journal of Combinatorics*, 27(1):1.24, 2020.
- [24] Franco P. Preparata and Michael I. Shamos. *Computational geometry: an introduction.* Springer, 2012.
- [25] Michael Sipser. *Introduction to the Theory of Computation, 2nd ed.* Thomson Course Technology, 2006.

- [26] G. A. Wade and J.-H. Chu. Drawability of complete graphs using a minimal slope set. *The Computer Journal*, 37(2):139–142, 1994.
- [27] Chee-Keng Yap. Towards exact geometric computation. *Computational Geometry*, 7(1-2):3–23, 1997.

A Source code for Lemma 3.15

This is the source code of the Python program mentioned in the proof of Lemma 3.15. It runs in Python 3.7.7. It uses the Sympy library (see <https://www.sympy.org>) for symbolic computations. It is also available on arXiv (<https://arxiv.org/src/2001.04671v4/anc>) as a Python file.

```

from time import sleep
import itertools as it
from sympy import * # Symbolic computations

def parse_expression(expression):
    """
    Input: a linear combination of the X_i's, e.g. 2*X_1-X_3+X_4.
    Output: two lists of terms with coefficients, one with the positive coefficients
           and one with negative coefficients.e.g. [(2, X_1), (1, X_4)], [(-1, X_3)].
    """
    list_terms = Add.make_args(expression)
    coeff_terms = [Mul.make_args(term) for term in list_terms]
    for i, term in enumerate(coeff_terms):
        if len(term) == 1: # No coefficient -> the coefficient is 1
            coeff_terms[i] = (1, term[0])
    positive_terms = [t for t in coeff_terms if t[0] > 0]
    negative_terms = [t for t in coeff_terms if t[0] < 0]
    return positive_terms, negative_terms

def substitute_all(old_var, new_var, equations, remove_trivial = False):
    """
    Substitutes the occurrences of old_var by new_var in all equations.
    If 'remove_trivial' is True, all expressions which are zero after substitution are discarded.
    """
    new_equations = [eq.subs(old_var, new_var) for eq in equations]
    return [new_eq for new_eq in new_equations if new_eq != 0 or not remove_trivial]

def solve_recurs(equations, inequations, substitutions = []):
    """
    Input: - 'equations' and 'inequations', two lists of expressions (each expression
           is a linear combinations of the X_i's)
           - 'substitutions', a list of pairs of variables (X_i, X_j), indicating that X_i has
           previously been substituted with X_j
    Output: A list of substitutions

    Finds all partitions on the set of variables {X_0, ..., X_11} with the property that:
    - for every expression in 'equations', the reduced expression is zero
    - for every expression in 'inequations', the reduced expression is nonzero.

    By reduced expression, we mean the following. If X_{i_1}, ..., X_{i_k} is a set of representatives
    for the partition, and if 'e' is an expression, the reduced version of 'e' is the expression
    obtained by
    1) substituting in 'e' every variable (an element of {X_0, ..., X_11}) by the
       variable X_{i_k} that is in the same class of the partition;
    2) simplifying the expression as much as possible.
    """

```

```

The partition is represented as a sequence of substitutions. The partition corresponding
to a list of substitutions is the partition with the fewest number of classes with the
following property: for every substitution (X_i, X_j), X_i and X_j are in the same class.
'''
ans_substitutions = []
if len(equations) == 0:
    return [substitutions]
positive_terms, negative_terms = parse_expression(equations[0])
old_var = positive_terms[0][1] # old_var is in the first equation with a positive coefficient
for coef, new_var in negative_terms: # old_var must cancel out with some other variable new_var
    # that appears in the first equation with a negative coefficient
    new_inequations = substitute_all(old_var, new_var, inequations)
    for nonzero in new_inequations:
        if nonzero == 0: # The substitution old_var <- new_var leads to a contradiction
            break
    else: # Perform the substitution and make a recursive call
        new_equations = substitute_all(old_var, new_var, equations, remove_trivial = True)
        sub = solve_rekurs(new_equations, new_inequations, substitutions+[old_var, new_var])
        if sub is not None:
            ans_substitutions.extend(sub)
return ans_substitutions

def pretty_print_sub(substitutions):
    '''Prints the given list of substitutions as a sequence of equalities.'''
    partition = []
    sub_to_str = lambda t: (str(t[0]).ljust(4), str(t[1]).ljust(4))
    str_substitutions = map(sub_to_str, substitutions)
    for old, new in str_substitutions:
        for partition_class in partition: # We search for the class of old and new in the partition
            if old in partition_class or new in partition_class:
                partition_class.update({old, new})
                break
        else: # Create a new class in the partition with old and new
            partition.append({old, new})
    for partition_class in partition:
        print(" = ".join(partition_class))

if __name__ == "__main__":
    # Define the variables X_i, the Z_i's and the factors appearing in the proof
    X = [None] + list(symbols('X_1:13')) # We do not use X[0] to match the notations of the proof
    Z = [None] + [X[2*i-1] + X[2*i] for i in range(1, 7)]
    factors_LHS = [Z[3]-Z[5], Z[6]-Z[2], Z[4]-Z[1]]
    factors_RHS = [Z[2]-Z[4], Z[5]-Z[1], Z[6]-Z[3]]

    nonzero_expressions = [] # List of conditions of the form 'expr != 0' satisfied by the X_i's
    for i in range(1, 7):
        nonzero_expressions.append(X[2*i-1] - X[2*i])
        nonzero_expressions.extend([Z[i] - Z[j] for j in range(1, i) if i + j != 7])

    cnt_print = 0
    max_num_sol_to_print = 10
    all_solutions = []
    for signs in it.product([+1, -1], repeat = 3): # Choose 3 signs
        if prod(signs) == 1:
            for permuted_factors_RHS in it.permutations(factors_RHS):
                # Each factor on the LHS must equal a factor on the RHS, up to a sign
                equations = [factors_LHS[i] - signs[i]*permuted_factors_RHS[i] for i in range(3)]

                # Printing parameters
                cnt_print += 1
                print("Case", cnt_print, "out of 24.")
                print("-> The signs are", " ".join([str(sign).rjust(2, "+") for sign in signs]))
                print("-> Permutation", (cnt_print-1)%6+1, "out of 6.")

```

```
print("The system of equations is:")
for eq in equations:
    print(str(eq)+" = 0")
print("Computing the solutions...")

answer = solve_rekurs(equations, nonzero_expressions)
all_solutions.extend(answer)

# Printing solutions
print("...there are", len(answer), "solutions.")
if len(answer) > 0:
    num_sol_to_print = min(len(answer), max_num_sol_to_print)
    print("\nFor example, here are", num_sol_to_print, "solutions:\n")
    for i in range(num_sol_to_print):
        pretty_print_sub(answer[i])
        print()
    print("There are", len(answer)-num_sol_to_print, "more solutions.\n")
print()
sleep(1)
```