

9-1998

## NP-Hardness in Geometric Construction Problems with One Interval Parameter

Nuria Mata

Vladik Kreinovich

*The University of Texas at El Paso*, [vladik@utep.edu](mailto:vladik@utep.edu)

Follow this and additional works at: [https://scholarworks.utep.edu/cs\\_techrep](https://scholarworks.utep.edu/cs_techrep)



Part of the [Computer Engineering Commons](#)

Comments:

Technical Report: UTEP-CS-98-29

Published as *Universitat Politecnica de Catalunya, Departament de Llenguatges i Sistemes Informatics Barcelona, Spain*, Technical Report No. LSI-98-55-R, November 1998; final version appeared in: *Proceedings of the Workshop on Applications of Interval Analysis to Systems and Control with special emphasis on recent advances in Modal Interval Analysis MISC'99*, Girona, Spain, February 24-26, 1999, pp. 85-98.

---

### Recommended Citation

Mata, Nuria and Kreinovich, Vladik, "NP-Hardness in Geometric Construction Problems with One Interval Parameter" (1998). *Departmental Technical Reports (CS)*. 450.

[https://scholarworks.utep.edu/cs\\_techrep/450](https://scholarworks.utep.edu/cs_techrep/450)

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of ScholarWorks@UTEP. For more information, please contact [lweber@utep.edu](mailto:lweber@utep.edu).

# NP-Hardness In Geometric Construction Problems With One Interval Parameter

Núria Mata

Departament de Llenguatges i Sistemes Informàtics  
Universitat Politècnica de Catalunya  
Av. Diagonal 647, 8<sup>a</sup>, 08028 Barcelona, Spain  
e-mail: nmata@lsi.upc.es

Vladik Kreinovich

Computer Science Department  
The University of Texas at El Paso  
El Paso, Texas 79968-0518, U.S.A.  
e-mail: vladik@cs.utep.edu

**Keywords:** geometric construction problems, interval arithmetic, under-constrained problems.

## 1 Introduction

**CAD/CAM: how to transform the geometric parameters of a technical drawing into coordinates of the corresponding points.** One of the main objectives of Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM) is to design and manufacture gadgets based on their specifications, typically described by *technical drawings*. Most *planar* technical drawings consist of circular and linear segments, i.e., of arcs and (closed) intervals. Typically, the user describes the geometric data such as the lengths of different linear segments, the radii of the arcs, the angles between different linear segments, etc.

To perform the corresponding manufacturing task, we can use general-purpose precise machines and tools. These machines and tools require that we supply them with *coordinates* of the corresponding points (segment endpoints and circle centers). Thus, we arrive at the problem of transforming the original numerical information contained in a technical drawing into coordinates of the corresponding points.

If we know the coordinates of all the points, then, of course, we can determine all the distances, radii, and angles in terms of these coordinates: e.g., a distance between the two points  $P_1 = (x_1, y_1)$  and  $P_2 = (x_2, y_2)$  is equal to  $d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ . Thus, each known numerical parameter of the technical drawing can be viewed as an *equation* on the unknown parameters on its points. For example, if we know that the distance between the points  $P_1$  and  $P_2$  (with unknown coordinates  $x_1, y_1, x_2$ , and  $y_2$ ) is equal to  $d_{12}$ , we get an equation

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

with four unknowns  $x_1, y_1, x_2$ , and  $y_2$ . If we collect all equations corresponding to all known numerical parameters of the technical drawing, we get a system of equations for determining the (unknown) coordinates of all the points.

**Geometric constructions help in solving this transformation problem.** From the purely algebraic viewpoint, each equation is a non-linear polynomial equation (or it can even be more

complicated). For realistically complex technical drawings, with many points, we have a large system of non-linear polynomial equations with many variables. In *general*, such algebraic systems are very difficult to solve (and for technical drawings, the corresponding problem is indeed difficult to solve). However, most real-life technical drawings belong to a naturally describable class [1, 7, 8] for which we can *explicitly construct* all the desired points, one by one, by means of *geometric construction* (i.e., a construction which uses only ruler and compass).

Let us formulate the notion of a geometric (= ruler-and-compass) construction in precise terms (see, e.g., [3, 4, 13]). Suppose that we have a finite collection of points on a plane. Starting with these points, we perform some geometric constructions step-by-step. On each step, we can do one of the following elementary constructions:

- given two different points  $P$  and  $Q$ , we can construct a *ray* (infinite semi-line) which starts at  $P$  and goes into the direction of  $Q$  (this construction is done by a *ruler*);
- given a point  $P$  and two other points  $Q$  and  $R$ , we can construct a circle for whom the point  $P$  is a center and whose radius is equal to the distance between  $Q$  and  $R$  (this construction is performed by a *compass*);
- given two different lines (two different rays, two different circles, or a ray and a circle), we can construct their intersection points.

All these constructions are described by simple explicit formulas (or, in case of an intersection, simple and easy-to-solve systems of equations). Thus, if we can find a geometric construction for constructing a point, we can then follow this construction step-by-step and compute the coordinates of the corresponding points.

To apply this approach to the problem of interpreting technical drawings, we must describe the original information (lengths, etc.) in geometric terms. This can be easily done if we select a starting point  $O$ , a ray  $r$  starting at  $O$ , and represent every real number  $d$  (length, etc.), by a point  $P_d$  which is located on this ray  $r$  at a distance  $d$  from the starting point  $O$ .

This idea has indeed led to successful transformation of technical drawings into sequences of geometric constructions; a working tool and examples of its use are given in [1, 7, 8].

**In many practical CAD/CAM problems, we have interval uncertainty.** This tool solves the above problem perfectly well; so, if we know the *exact* values of all the parameters of the technical drawing, we can efficiently transform this drawing into a geometric construction, which, in its turn, enables us to compute the coordinates of all the points and thus, prepare the drawing for its CAD/CAM use.

In many practical situations, we do know the exact values of the desired geometric parameters (distances etc.), or at least the inaccuracy which we allow is so small that for all practical purposes, we can assume that these parameters are known exactly. In some practical situations, however, we do not know the exact values of some of these geometric parameters, we only know *approximate* values of these parameters. For example, if we know that the distance between the points is  $2.0 \pm 0.5$ , this means that this distance is allowed to take any value from  $2.0 - 0.5 = 1.5$  and  $2.0 + 0.5 = 2.5$ , i.e., any value from the *interval*  $[1.5, 2.5]$ . If we only know the geometric parameters of the original drawing with interval uncertainty, then, of course, we can not determine the *exact* coordinates of the corresponding points, we can only compute the *intervals* for the corresponding parameters. Thus, we encounter the problem of transforming the interval-valued drawing into *intervals* of possible values of coordinates.

**For simple (single-step) geometric constructions, interval estimations can also be done by geometric means.** In the simplest situation, when the geometric construction consists of a single geometric step, we can get the explicit description of the corresponding coordinate intervals; moreover, it turns out that if we represent the endpoints  $\underline{d}$  and  $\overline{d}$  of each original interval  $\mathbf{d} = [\underline{d}, \overline{d}]$

by points on the standard ray  $r$ , then for such one-step constructions, we can compute the bounds of the corresponding intervals by using only ruler and compass [6].

**For more complex (multi-step) geometric constructions, known methods overestimate the resulting intervals.** For more realistic multi-step constructions, we can repeat the same procedure for each geometric step (in the style of *interval mathematics*), and get intervals which *contain* the desired coordinate intervals. The problem with this approach is that while for one-step constructions, we get the *exact* endpoints of the corresponding intervals, for multi-step constructions, we often get an *overestimation* (i.e., an interval which does contain the desired one but is much wider than the desired interval).

**The problems.** Thus, we arrive at the following *natural question*:

Is it possible, for multi-step geometric constructions, to construct the endpoints of the corresponding coordinate intervals by using only ruler and compass?

If this is possible, then, since (as we have mentioned) ruler-and-compass constructions can be easily simulated on a computer, we would get a feasible (fast) algorithm for computing the endpoints of the corresponding coordinate intervals. If, however, for some cases, the ruler-and-compass construction of interval endpoints is impossible, then we have a next natural question:

Is it possible to compute the endpoints of the corresponding coordinate intervals in feasible time?

In asking these two questions, we implicitly assumed that there are some values from the given intervals for which the construction is possible. When input data are known exactly, the question of whether the construction is possible at all is easy to check: we apply the construction step-by-step and check if it is possible (and sometimes it is not: e.g., circles have no intersection, or the two rays which were supposed to be different are, in fact, identical, etc.). When we only know the *intervals* of possible geometric parameters, then the question of whether the construction is possible at all (and if yes, for which exactly values of the parameters) becomes non-trivial. It is therefore desirable to analyze whether this construction possibility problem can be solved in feasible time or by a geometric construction.

**What we are planning to do.** The more parameters are known with interval uncertainty, the more complex the resulting problem. In this paper, we will show that even if we have only one interval parameter, the above problems are, in general, NP-hard (i.e., crudely speaking, computationally infeasible), and unsolvable by ruler-and-compass constructions.

Before we proceed to exact formulations and proofs, we will briefly remind the readers what NP-hard means.

## 2 What is NP-hard? A brief and informal reminder

**What is “feasible”?** In theory of computation, it is well known that not all algorithms are feasible (see, e.g., [2, 10, 11, 12]), whether an algorithm is feasible or not depends on how many computational steps it needs.

For example, if for some input  $x$  of length  $\text{len}(x) = n$ , an algorithm requires  $2^n$  computational steps, then for an input of a reasonable length  $n \approx 300$ , we would need  $2^{300}$  computational steps. Even if we use a hypothetical computer for which each step takes the smallest physically possible time (the time during which light passes through the smallest known elementary particle), we would still need more computational steps than can be performed during the (approximately 20 billion years) lifetime of our Universe.

A similar estimate can be obtained for an arbitrary algorithm whose running time  $t(n)$  on inputs of length  $n$  grows at least as an exponential function, i.e., for which, for some  $c > 0$ ,  $t(n) \geq \exp(c \cdot n)$  for all  $n$ . As a result, such algorithms (called *exponential-time*) are usually considered *not feasible*.

The fact that an algorithm is not feasible, does not mean that it can never be applied: it simply means that there are cases when its running time will be too large for this algorithm to be practical; for other inputs, this algorithm can be quite useful.

On the other hand, if the running time grows only as a polynomial of  $n$  (i.e., if an algorithm is *polynomial-time*), then the algorithm is usually quite feasible.

As a result of the above two examples, we arrive at the following idea: An algorithm  $\mathcal{U}$  is called *feasible* if and only if it is *polynomial-time*, i.e., if and only if there exists a polynomial  $P(n)$  such that for every input  $x$  of length  $\text{len}(x)$ , the computational time  $t_{\mathcal{U}}(x)$  of the algorithm  $\mathcal{U}$  on the input  $x$  is bounded by  $P(\text{len}(x))$ :  $t_{\mathcal{U}}(x) \leq P(\text{len}(x))$ .

In most practical cases, this idea *adequately* describes our intuitive notion of feasibility: *polynomial-time* algorithms are usually *feasible*, and *non-polynomial-time* algorithms are usually *not feasible*.

Although in *most* cases, the above idea adequately describes the intuitive notion of feasibility, the reader should be warned that this idea is *not perfect*: in some (very rare) cases, it does not work (see, e.g., [2, 10, 11, 12]):

- Some algorithms are polynomial-time but not feasible: e.g., if the running time of an algorithm is  $10^{300} \cdot n$ , this algorithm is polynomial-time, but, clearly, not feasible
- Vice versa, there exist algorithms whose computation time grows, say, as  $\exp(0.000 \dots 01 \cdot \text{len}(x))$ . Legally speaking, such algorithms are exponential time and thus, not feasible, but for all practical purposes, they are quite feasible.

It is therefore desirable to look for a *better* formalization of feasibility but as of now, “polynomial-time” is the best known description of feasibility.

**What is “tractable” and what is “intractable”?** At first glance, now, that we have a definition of a feasible algorithm, we can describe which problems are tractable and which problems are intractable: If there exists a polynomial-time algorithm that solves all instances of a problem, this problem is tractable, otherwise, it is intractable.

In some cases, this ideal solution is possible, and we either have an explicit polynomial-time algorithm, or we have a proof that no polynomial-time algorithm is possible. Unfortunately, in many cases, we do not know whether a polynomial-time algorithm exists or not. This does not mean, however, that the situation is hopeless: instead of the missing *ideal* information about intractability, we have another information that is almost as good:

Namely, for some cases, we do not know whether the problem can be solved in polynomial time or not, but we do know that this problem is as hard as practical problems can get: if we can solve *this* problem easily, then we would have an algorithm that solves *all* problems easily, and the existence of such universal solves-everything-fast algorithm is very doubtful. Such problems are called *NP-hard* (for exact definitions, see, e.g., [10]). In view of the above explanations, we can see why NP-hard problems are also called *intractable*.

### 3 Definitions and the main results

Now, we are ready for precise definitions and results.

**Definition 1.**

- By a *simple line*, we will mean a ray or a circle.
- By a *geometric object*, we mean a point or a simple line.

**Definition 2.** Let a finite sequence of geometric objects  $\{Z_1, \dots, Z_n\}$  be given. By an elementary step of a geometric construction, we mean one of the following instructions:

1. construct a ray which starts at a point  $Z_i$  and goes into the direction of a point  $Z_j$ ; this ray will be denoted by  $Z_iZ_j$ ;
2. construct a circle with a center in a point  $Z_i$  whose radius is equal to the distance between the points  $Z_j$  and  $Z_k$ ;
3. construct all points from the intersection between the two simple lines  $Z_i$  and  $Z_j$ .

We will say that an elementary step is *applicable* to the given set of objects if, correspondingly:

1. the points  $Z_i$  and  $Z_j$  are different ( $Z_i \neq Z_j$ );
2. the points  $Z_j$  and  $Z_k$  are different ( $Z_j \neq Z_k$ );
3. the simple lines  $Z_i$  and  $Z_j$  are different and have at least one common point ( $Z_i \neq Z_j$  and  $Z_i \cap Z_j \neq \emptyset$ ).

**Definition 3.** Let  $n$  be a positive integer. By a geometric construction with  $n$  input points  $Z_1 = P_1, \dots, Z_n = P_n$ , we mean a finite sequence of elementary steps:

- the first step is applied to a sequence  $\{Z_1, \dots, Z_n\}$ ; the result of applying this step will be denoted by  $Z_{n+1}$ ;
- the second step is applied to a sequence  $\{Z_1, \dots, Z_n, Z_{n+1}\}$ ; the result of applying this step will be denoted by  $Z_{n+2}$ ;
- ...
- the final ( $s$ -th) step is applied to a sequence  $\{Z_1, \dots, Z_n, \dots, Z_{n+(s-1)}\}$ ; the result of applying this step will be denoted by  $Z_{n+s}$ .

We say that a geometric construction is *applicable* to a sequence of points if for this sequence of points  $\{P_1, \dots, P_n\}$ , each elementary step of the construction is applicable.

*Example.* Let us give a simple CAD/CAM-motivated example of a geometric construction. Namely, we want to construct a triangle with known sides  $a$ ,  $b$ , and  $c$ . As we mentioned earlier, these sides are given as points on a ray. In other words, the input consists of four points: a point of origin  $P_1$ , and three points  $P_2$ ,  $P_3$ , and  $P_4$  on the same ray which represent the given distances  $d(P_1, P_2) = a$ ,  $d(P_1, P_3) = b$ , and  $d(P_1, P_4) = c$ .

In our construction, we start with the points  $Z_1 = P_1$ ,  $Z_2 = P_2$ ,  $Z_3 = P_3$ , and  $Z_4 = P_4$ . Based on these four points, we already have one side of the desired triangle:  $Z_1Z_2$  with the length  $a$ . To construct the desired triangle, we can perform the following sequence of elementary steps:

- first, we construct a circle  $Z_5$  with a center in  $Z_1$ , and whose radius is equal to the distance  $d(Z_1, Z_3) = b$ ;
- second, we construct a circle  $Z_6$  with a center in  $Z_2$ , and whose radius is equal to the distance  $d(Z_2, Z_4) = c$ ;
- finally, we construct the third point  $Z_7$  of the desired triangle as the intersection of the circles  $Z_5$  and  $Z_6$ .

Due to our construction, we have  $d(P_1, P_2) = a$ ,  $d(P_1, Z_7) = b$ , and  $d(P_2, Z_7) = c$ , i.e.,  $P_1P_2Z_7$  is the desired triangle.

*Comment.* As we have mentioned before, in geometric construction problems stemming from CAD/CAM, we usually start with the points  $P_1, \dots, P_n$  which all belong to one and the same ray:

- the point  $P_1$  is the starting point of this ray;
- the point  $P_2$  represents a unit distance (i.e.,  $d(P_1, P_2) = 1$ ); and
- all other points represent the distances on a technical drawing.

Therefore, we naturally arrive at the following definition:

**Definition 4.** Let  $n$  be a positive integer.

- By an *exactly known input data*, we mean a sequence of  $n$  points  $P_1, \dots, P_n$  which all belong to the same ray  $P_1P_2$  and for which  $d(P_1, P_2) = 1$ .
- By an *interval input data with one interval parameter*, we mean a sequence of  $n - 1$  points  $P_1, \dots, P_{n-1}$  which are all located on the ray  $P_1P_2$  (with  $d(P_1, P_2) = 1$ ), and two extra points  $\underline{P}_n$  and  $\overline{P}_n$  on the same ray  $P_1P_2$  (such that the point  $\underline{P}_n$  precedes  $\overline{P}_n$  on this ray). This interval input data will be denoted by  $\{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\}$ .
- We say that a construction  $\mathcal{C}$  (with  $n$  input points) is *applicable* to the interval input data  $\{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\}$  if there exists a point  $P_n$  from the segment  $\underline{P}_n\overline{P}_n$  for which this construction is applicable to the sequence  $\{P_1, \dots, P_n\}$ .

The first problem we face is how to check whether a construction is applicable to a given data. It is known that there exists an algorithm (originally proposed by Tarski; for details, see, e.g., [10]) which solves all problems from elementary geometry. In particular, one can show that this algorithm solves our checking problem as well. The problem with this solution is that Tarski's algorithm requires exponentially long time and is, therefore, infeasible. The question is, therefore: is there a *feasible* algorithm for this checking? The answer to this question is negative:

**Theorem 1.** *The problem of checking whether a given geometric construction is applicable to a given interval input data with one interval parameter, is NP-hard.*

*Comment.* For readers' convenience, all the proofs are placed in the special proofs section.

If the construction is applicable, then the next question is: how to describe the set of all points from the segment  $\underline{P}_n\overline{P}_n$  for which it is applicable? In particular, if this set is itself a segment, then it is natural to ask whether the endpoints of this segment can be constructed by using a geometric construction. The answer to this question is also negative:

**Definition 5.** We say that a point  $P$  is *constructible* from the points  $P_1, \dots, P_n$  if there is a geometric construction which starts with the points  $P_1, \dots, P_n$ , and which constructs (among other objects) the point  $P$ .

**Theorem 2.** *There exists a geometric construction  $\mathcal{C}$  and an interval input data  $\{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\}$  for which the following two statements hold:*

- the set  $\mathcal{I}$  of all points  $P_n \in [\underline{P}_n, \overline{P}_n]$  for which the construction  $\mathcal{C}$  is applicable

$$\mathcal{I} = \{P_n \mid \mathcal{C} \text{ is applicable to } \{P_1, \dots, P_{n-1}, P_n\}\}$$

*is an interval;*

- the endpoints of this interval  $\mathcal{I}$  are not constructible from the input points  $P_1, \dots, P_{n-1}, \underline{P}_n, \overline{P}_n$ .

Our ultimate goal is, given a geometric construction and an interval input data to which this construction is applicable, to compute the (endpoints of the) interval of possible locations of each constructed point  $P_m$ . This problem is the easiest to formulate if this point also belongs to the same ray  $P_1P_2$  as the input points. Since even for this simplest case, we will get negative results (that even this particular case is difficult to solve), this will show that the more general problem is complicated as well.

**Definition 6.** By a *geometric construction problem with one interval parameter*, we mean a pair  $\langle \mathcal{C}, \{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\} \rangle$  consisting of a geometric construction  $\mathcal{C}$  with  $n$  inputs, and an interval input data with one interval parameter  $\{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\}$  which satisfies the following two properties:

- the given construction  $\mathcal{C}$  is applicable to the given data (i.e., there is a point  $P_n \in [\underline{P}_n, \overline{P}_n]$  for which the construction  $\mathcal{C}$  is applicable to the sequence  $\{P_1, \dots, P_n\}$ ), and
- for each point  $P_n \in [\underline{P}_n, \overline{P}_n]$  for which the construction  $\mathcal{C}$  is applicable to the sequence  $\{P_1, \dots, P_n\}$ , the final object  $Z_{n+s}$  of the construction  $\mathcal{C}$  is a point located on the ray  $P_1P_2$ .

The objective of this problem is to compute the set  $\mathcal{S}$  of possible locations of the point  $Z_{n+s}$ . This set will be called a *solution set*.

*Comment.* We will prove two negative results:

- that even in the simplest case, when this set  $\mathcal{S}$  is an interval, we still cannot always construct this set (or, to be more precise, its endpoints) by using only a ruler and a compass; and
- that even a problem of checking whether a given interval  $\mathcal{I}$  has common points with this set  $\mathcal{S}$  is NP-hard.

**Theorem 3.** *There exists a geometric construction problem  $\langle \mathcal{C}, \{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\} \rangle$  with one interval parameter for which the following three statements hold:*

- for every point  $P_n \in [\underline{P}_n, \overline{P}_n]$ , the construction  $\mathcal{C}$  is applicable to the sequence  $\{P_1, \dots, P_n\}$ ;
- the set of all points  $Z_{n+s}$  obtained by using all possible  $P_n \in [\underline{P}_n, \overline{P}_n]$  is an interval, and
- the endpoints of this interval are not constructible from the input points  $P_1, \dots, P_{n-1}, \underline{P}_n, \overline{P}_n$ .

**Theorem 4.** *The following problem is NP-hard: given a geometric construction problem  $\langle \mathcal{C}, \{P_1, \dots, P_{n-1}, [\underline{P}_n, \overline{P}_n]\} \rangle$  with one interval parameter and an interval  $\mathcal{I}$  on the ray  $P_1P_2$ , check whether this interval has common points with the solution set, i.e., whether  $\mathcal{I} \cap \mathcal{S} \neq \emptyset$ .*

## 4 Proofs

### 4.1 Proof of Theorem 1

Let us show that the problem of checking whether a given geometric construction is applicable to a given interval input data with one interval parameter, is NP-hard.

To prove NP-hardness, we will reduce a problem which is known to be NP-hard to this problem. In other words, we will show that for every instance of the known NP-hard problem, there exists a particular case of our geometric problem whose solution will lead to a solution to the original instance. This would imply that our geometric problem is also NP-hard.

*Comment.* The exact definitions and arguments about NP-hardness can be taken, e.g., from [10]; however, it is worth mentioning that the main idea of the reduction proofs can be easily explained in informal terms: If we had a feasible algorithm  $G$  for solving our geometric problem, then we



would be able to solve all the instances of the known NP-hard problem really fast by reducing them to the corresponding instance of our problem and applying the (hypothetic) algorithm  $G$ . Since the known problem is known to be hard, such a fast solution is hardly possible, and therefore, it is unlikely that we would be able to find a feasible algorithm  $G$  for solving all instances of our geometric problem. Thus, our geometric problem is also hard-to-solve (NP-hard).

As the known NP-hard problem, we take a *partition problem*: given  $m$  positive integers  $s_1, \dots, s_m$ , check whether there exist values  $x_1, \dots, x_m \in \{-1, 1\}$  for which  $s_1 \cdot x_1 + \dots + s_m \cdot x_m = 0$ .

In our reduction, we will represent this problem in geometric terms. Before we start the reduction, let us mention that in the partition problem, the values  $x_i$  can be both positive and negative, while in the geometric construction, it is easier to represent *positive* numbers (as distances). Thus, before the reduction, we will re-formulate the partition problem in such a way that it involves only positive numbers: Given  $m$  positive integers  $s_1, \dots, s_m$ , check whether there exist values  $y_1, \dots, y_m \in \{1, 3\}$  for which  $s_1 \cdot y_1 + \dots + s_m \cdot y_m = 2m$ . This new problem is clearly equivalent to the old one; indeed:

- if we have values  $x_i \in \{-1, 1\}$  for which  $s_1 \cdot x_1 + \dots + s_m \cdot x_m = 0$ , then we can take  $y_i = x_i + 2$  and guarantee that  $s_1 \cdot y_1 + \dots + s_m \cdot y_m = 2m$  and  $y_i \in \{1, 3\}$ ;
- vice versa, if we have  $s_1 \cdot y_1 + \dots + s_m \cdot y_m = 2m$  for some  $y_i \in \{1, 3\}$ , then we can take  $x_i = y_i - 2$  and guarantee that  $s_1 \cdot x_1 + \dots + s_m \cdot x_m = 0$  for some  $x_i \in \{-1, 1\}$ .

For each instance of this new problem, we will design a construction  $\mathcal{C}$  and an interval input data for which  $\mathcal{C}$  is applicable to the given data if and only if the original instance has a solution. As the interval input data, we take  $\{P_1, P_2, [\underline{P}_3, \overline{P}_3]\}$ , where  $d(P_1, P_2) = 1$ ,  $\underline{P}_3 = P_1$ , and  $\overline{P}_3 = P_2$ . Let us denote by  $y$  the distance  $d(P_1, P_3)$  between the origin  $P_1$  and the (unknown) point  $P_3 \in [\underline{P}_3, \overline{P}_3]$ . Then,  $y \in [0, 1]$ .

Based on this value  $y$ , we want to construct the values  $y_1, \dots, y_m \in \{1, 3\}$ . The main idea of our construction is as follows:

- We start with an interval  $[0, 1]$ ; we divide this interval into two halves ( $[0, 1/2]$  and  $[1/2, 1]$ ) and check where  $y$  is with respect to this division:
  - if the value  $y$  is strictly in the middle of the original interval, the construction is not applicable, and we stop;
  - if the value  $y$  is in the left half, we take  $y_1 = 1$ ;
  - if the value  $y$  is in the right half, we take  $y_1 = 3$ .

In both cases, we get a new interval of half-size which contains  $y$ .

- Again, we divide this new interval into two halves and check where  $y$  is with respect to this division:
  - if the value  $y$  is strictly in the middle of the new interval, the construction is not applicable, and we stop;
  - if the value  $y$  is in the left half, we take  $y_1 = 1$ ;
  - if the value  $y$  is in the right half, we take  $y_2 = 3$ .

In both cases, we get a new interval of half-size which contains  $y$ .

- Based on this new interval, we find  $y_3, y_4$ , etc., until we have found all  $m$  values  $y_i$ .

It is easy to check that for every sequence of values  $y_i \in \{1, 3\}$ , there exists a value  $y \in [0, 1]$  which leads to exactly this sequence: Indeed, if, e.g., we want to represent a sequence  $y_1 = 1, y_2 = 3, y_3 = 1$ , etc., we do the following:

- First, we divide the original interval in half and choose the left half  $[0, 1/2]$  of the original interval (left, because we want  $y_1 = 1$ ). The interval  $(0, 1/2)$  has width  $2^{-1}$ , and for all values  $y$  from this open interval, the above construction leads to  $y_1 = 1$ .
- Then, we divide the resulting half-interval  $[0, 1/2]$  in half, and choose the right half  $[1/4, 1/2]$  (right, because we want  $y_2 = 3$ ). The interval  $(1/4, 1/2)$  has width  $2^{-2}$ , and for all values  $y$  from this open interval, the above construction leads to  $y_1 = 1$  and  $y_2 = 3$ .
- Then, we divide the resulting quarter-interval  $[1/4, 1/2]$  into two equal halves, and choose the left half  $[1/4, 3/8]$  (left, because we want  $y_3 = 1$ ). The interval  $(1/4, 3/8)$  has width  $2^{-3}$ , and for all values  $y$  from this open interval, the above construction leads to  $y_1 = 1$ ,  $y_2 = 3$ , and  $y_3 = 1$ .

...

At the end, we get an open interval of width  $2^{-m}$ , all values  $y$  from which lead to the given sequence  $y_1, \dots, y_m$ .

Let us show how this construction of  $y_i$  can be performed by using ruler and compass. First, let us extend the ray  $P_1P_2$  to the other side of  $P_1$ . This can be easily done if we construct a ray  $P_2P_1$  starting at  $P_2$  and going in the direction of  $P_1$ . Now, we can construct points on both sides of  $P_1$ , i.e., we can treat the union of these two rays as a true coordinate axis.

We start with the point  $P_3$  which is located between  $P_1$  and  $P_2$ . We want to construct a point  $Y_1$  whose distance from  $P_1$  is equal to exactly  $y_1$ , i.e., it is equal to either 1 or 3 depending on whether  $y = d(P_1, P_3)$  belongs to the first or to the second half of the interval  $[P_1, P_2]$ . To construct this interval, we first construct two auxiliary points:

- the midpoint  $P_{0.5}$  of the interval  $[P_1, P_2]$ , and
- an auxiliary point  $P_{-1.5}$  which is located on the ray  $P_2P_1$  at a distance 2.5 from  $P_2$  (i.e., at a distance 1.5 beyond the origin  $P_1$ ).

The distance between these two auxiliary points is  $d(P_{-1.5}, P_{0.5}) = 1.5 + 0.5 = 2$ .

Then, we perform the following construction:

- first, we construct a ray starting at the central point  $P_{0.5}$  and going in the direction of the point  $P_3$ ; this is possible if  $P_3 \neq P_{0.5}$ , i.e., if the point  $P_3$  is not exactly in the middle of the interval  $[P_1, P_2]$ ;
- on the ray  $P_{0.5}P_3$ , we select a point  $D$  at a distance 1 from  $P_{0.5}$ ;
  - if  $y > 1/2$ , then the point  $D$  is to the right of  $P_{0.5}$ , and therefore, its distance from  $P_{-1.5}$  is  $2 + 1 = 3$ ;
  - if  $y < 1/2$ , then the point  $D$  is to the left of  $P_{0.5}$ , and therefore, its distance from  $P_{-1.5}$  is  $2 - 1 = 1$ ;
- finally, we build a point  $Y_1$  on the ray  $P_1P_2$  for which  $d(P_1, Y_1) = d(P_{-1.5}, D)$ ;

The point  $Y_1$  is at a distance  $y_1$  from  $P_1$ , where  $y_1 = 1$  or  $y_1 = 3$  depending on whether  $P_3$  belongs to the left or to the right half of the interval  $[P_1, P_2]$  (i.e., whether  $y < 1/2$  or  $y > 1/2$ ).

Now, we want to construct  $Y_2$ . Since we have already invested quite some effort in constructing  $Y_1$ , and the construction  $Y_2$  is similar, we would like to re-use the previous construction instead of starting everything from scratch. The only difference between constructing  $Y_1$  and constructing  $Y_2$  is that for  $Y_1$ , we compare the point  $P_3$  with the midpoint of the original interval  $[0, 1]$ , while to find  $Y_2$ , we must compare the point  $P_3$  with the midpoint of a smaller half-interval  $([0, 1/2]$  or  $[1/2, 1])$ , depending on whether  $y < 1/2$  or  $y > 1/2$ ). Therefore, to reduce the problem of constructing  $Y_2$

to the problem of constructing  $Y_1$ , we can linearly map this half-interval onto the original interval  $[P_1, P_2]$ ; then, the point  $P_3$  maps into a new point  $P_3^{(1)}$ , whose relation to the midpoint of the interval  $[P_1, P_2]$  is exactly the same as the relation of the original point  $P_3$  to the midpoint of the half-interval. Therefore, to find  $Y_2$ , all we have to do is compare this new point  $P_3^{(1)}$  with the midpoint of the interval  $[P_1, P_2]$  and construct a point  $Y_2$  by using exactly the same steps that we used when we constructed  $Y_1$ .

How can we perform this linear transformation  $t(x) = a \cdot x + b$ ? In terms of the coordinates on the ray, this transformation takes the following form:

- If  $y < 1/2$ , i.e., if  $y_1 = 1$ , then this transformation should transform the half-interval  $[0, 1/2]$  onto  $[0, 1]$ . In other words, we must have  $t(0) = 0$  and  $t(1/2) = 1$ . The only linear function which satisfies these two conditions is  $t(x) = 2x$ .
- If  $y > 1/2$ , i.e., if  $y_1 = 3$ , then this transformation should transform the half-interval  $[1/2, 1]$  onto  $[0, 1]$ . In other words, we must have  $t(1/2) = 0$  and  $t(1) = 1$ . The only linear function which satisfies these two conditions is  $t(x) = 2x - 1$ .

We can combine these two cases into a single formula  $t(x) = 2x - (y_1 - 1)/2$ . We need to apply this transformation to a point  $P_3$  at a coordinate  $y$ , and get a new point  $P_3^{(1)}$  at a coordinate  $y^{(1)} = 2y - (y_1 - 1)/2$ .

It is known that if we have segments of lengths  $a$  and  $b$ , then, by using geometric (i.e., ruler-and-compass) constructions, we can construct segments of lengths  $a + b$ ,  $|a - b|$ ,  $a \cdot b$ , and  $a/b$  [3, 4, 13]. Both values  $y$  and  $y_1$  are represented as lengths; therefore, by implementing the arithmetic operations step-by-step, we can get the get a geometric construction for  $y^{(1)}$  (i.e., for  $P_3^{(1)}$ ).

Applying the above construction to the new point  $P_3^{(1)}$ , we get a point  $Y_2$ . Similarly, we can map the resulting half-interval into a new interval, apply the same procedure, and get  $Y_3$ , etc. After we repeat the same construction  $m$  times, we get all  $m$  intervals  $Y_1, \dots, Y_m$  for which  $y_i = d(P_1, Y_i) \in \{1, 3\}$ .

An arbitrary integer  $s_i$  can be represented as  $1 + 1 + \dots + 1$  ( $s_i$  times) and can therefore be geometrically constructed; using the fact that multiplication and addition are constructible, we can thus construct a point  $P$  on the ray  $P_1P_2$  whose distance from  $P_1$  is equal to  $d(P_1, P) = l = s_1 \cdot y_1 + \dots + s_m \cdot y_m$ .

We can also construct another point  $P_{4m}$  on the same ray, at a distance  $4m$ . Then, on the final two steps of our construction, we construct the two auxiliary points  $R$  and  $S$  for which  $d(P_1, R) = d(R, P) = d(P, S) = d(S, P_{4m}) = 2m$ .

To be more precise: to construct  $R$ , we construct two circles of radius  $m$  with centers in  $P_1$  and  $P$ , and takes their intersection as  $R$ . The point  $S$  is constructed in a similar manner.

This construction will only be possible if  $d(P_1, P) \leq 2m$  and  $d(P, P_{4m}) \leq 2m$ . Since  $d(P, P_{4m}) = 4m - d(P_1, P)$ , the first inequality leads to  $d(P, P_{4m}) \geq 4m - 2m = 2m$ . Thus, the whole construction is possible only if  $d(P_1, P) \leq 2m$  and  $d(P_1, P) \geq 2m$ , i.e., only if  $d(P, P_1) = 2m$ . But we know that  $d(P, P_1) = s_1 \cdot y_1 + \dots + s_m \cdot y_m$ . Thus, the geometric construction is possible for some  $P_3 \in [\underline{P}_3, \overline{P}_3]$  if and only if there exists values  $y_i \in \{1, 3\}$  for which  $s_1 \cdot y_1 + \dots + s_m \cdot y_m = 2m$ , i.e., if and only if the original instance of the partition problem has a solution.

The reduction is proven, and therefore, our original geometric problem is indeed NP-hard. The theorem is proven.

## 4.2 Proof of Theorem 2

It is known that if we have segments of lengths  $a$  and  $b$ , then, by using geometric (i.e., ruler-and-compass) constructions, we can construct segments of lengths  $a + b$ ,  $|a - b|$ , and  $a \cdot b$  [3, 4, 13]. For

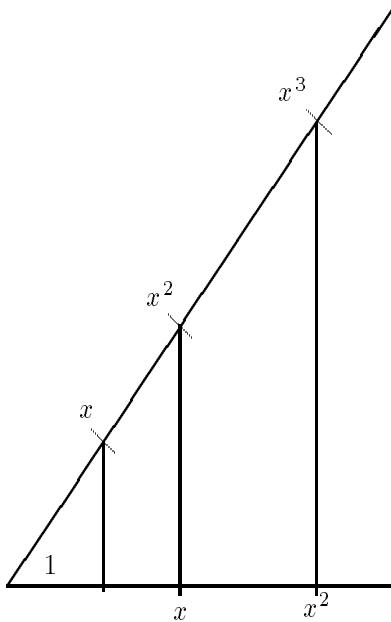


Figure 1: Geometric construction for length  $l$  to be  $2m$ .

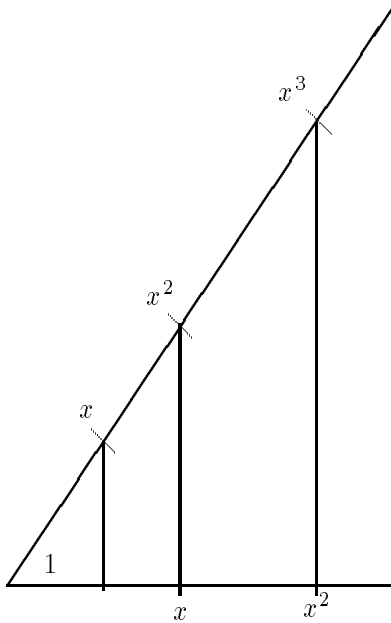


Figure 2:

example, if we have two segments of length 1 and  $x$ , then the construction of a segment of length  $x^3$  is outlined on Fig. 2.

We will use this construction to prove Theorem 2. Namely, as  $\mathcal{C}$ , we will take the following construction with three inputs:

- We start with three points  $P_1$ ,  $P_2$ , and  $P_3$  on the same ray  $P_1P_2$ , for which  $d(P_1, P_2) = 1$ . In this proof, we will denote the distance  $d(P_1, P_3)$  by  $x$ .
- At first, we use the construction outlined in Fig. 2 to construct the point  $Z$  on the ray  $P_1P_2$  for which  $d(P_1, Z) = x^3$ .
- Then, we construct a point  $T$  which is at distance 1 from both  $P_1$  and  $Z$  (this point can be constructed as an intersection of two circles of radius  $1 = d(P_1, P_2)$  with centers correspondingly in  $P_1$  and in  $Z$ ).

The first part of this construction (constructing the point  $Z$  for which  $d(P_1, Z) = x^3$ ) is always applicable. The second part is applicable only when the circles have a non-empty intersection, i.e., when there exists a point  $T$  from which  $d(P_1, T) = d(Z, T) = 1$ . Due to triangle inequality, the existence of such a point is equivalent to  $d(P_1, Z) \leq 2$ . Thus, the whole construction is possible if and only if  $x^3 \leq 2$ , i.e., if  $x \leq \sqrt[3]{2}$ .

Let us take, as an interval input data, the points  $P_1$  and  $P_2$  for which  $d(P_1, P_2) = 1$ , and the interval  $[\underline{P}_3, \overline{P}_3]$  in which the point  $\underline{P}_3 = P_2$  is located at a distance 1 from  $P_1$ , and the point  $\overline{P}_3$  is located on the same ray at a distance 2 from the origin  $P_1$ .

We will show that the above construction  $\mathcal{C}$  and the above interval input data  $\{P_1, P_2, [\underline{P}_3, \overline{P}_3]\}$  satisfy both statements of Theorem 2 (and thus, Theorem 2 holds). Indeed, let us first show that the set of all points  $P_3 \in [\underline{P}_3, \overline{P}_3]$  for which the construction  $\mathcal{C}$  is applicable is an interval  $\mathcal{I}$ . Indeed, since the construction  $\mathcal{C}$  is possible only when the distance  $d(P_1, P_3)$  does not exceed  $\sqrt[3]{2}$ , the points  $P_3 \in [\underline{P}_3, \overline{P}_3]$  for which this construction is possible form an interval  $\mathcal{I} = [P_2, Q]$  in which the left endpoint is located at a distance 1 from the origin  $P_1$ , and the right endpoint  $Q$  is located at a distance  $\sqrt[3]{2}$  from the origin  $P_1$ .

To complete this proof of the theorem, we must show that the endpoints of this interval  $\mathcal{I}$  are not constructible from the input points  $P_1, P_2, \underline{P}_3, \overline{P}_3$ . The impossibility of constructing a distance  $\sqrt[3]{2}$  is a well-known fact in ruler-and-compass construction theory [3, 4, 13]; it is actually the solution to one of the three classical problems with which this theory originally started: is it possible to double a cube, i.e., to construct a line segment of a cube with volume equal to 2. One of the first results of ruler-and-compass theory was that such a construction is impossible. The theorem is thus proven.

### 4.3 Proof of Theorem 3

*Comment.* This proof use general ideas from the paper [9], in which algebraic properties of interval estimates are analyzed. Thus, the readers who want to understand where our formulas and constructions came from, are welcome to read this paper.

As we have mentioned in the proof of Theorem 2, if we have segments of lengths  $a$  and  $b$ , then, by using geometric (i.e., ruler-and-compass) constructions, we can construct segments of lengths  $a + b$ ,  $|a - b|$ ,  $a \cdot b$ , and  $a/b$ . Since every polynomial  $P(a)$  with rational coefficients can be represented as a composition of addition, subtraction, and multiplication, we can, therefore, combine the corresponding geometric constructions and construct a segment of length  $P(a)$  (provided, of course, that the number  $P(a)$  is non-negative).

We will use this possibility to prove Theorem 3. Namely, as  $\mathcal{C}$ , we will take the geometric construction with three input points  $P_1$ ,  $P_2$ , and  $P_3$  on the same ray  $P_1P_2$  (for which  $d(P_1, P_2) = 1$ ); this construction results in a point  $Z$  on the same ray for which  $d(P_1, Z) = 1 + 2 \cdot d(P_1, P_3) - d^4(P_1, P_3)/4$ . If we denote the original distance  $d(P_1, P_3)$  by  $x$ , then we get  $d(P_1, Z) = 1 + 2x - x^4/4$ .

As in Theorem 2, we will take, as an interval input data, the points  $P_1$  and  $P_2$  for which  $d(P_1, P_2) = 1$ , and the interval  $[\underline{P}_3, \overline{P}_3]$  in which the point  $\underline{P}_3 = P_2$  is located at a distance 1 from  $P_1$ , and the point  $\overline{P}_3$  is located on the same ray at a distance 2 from the origin  $P_1$ .

We will show that the above construction  $\mathcal{C}$  and the above interval input data  $\{P_1, P_2, [\underline{P}_3, \overline{P}_3]\}$  satisfy all three statements of Theorem 3 (and thus, Theorem 3 holds).

First of all, let us show that for every point  $P_3 \in [\underline{P}_3, \overline{P}_3]$ , the construction  $\mathcal{C}$  is applicable to the sequence  $\{P_1, P_2, P_3\}$ . Indeed, the construction  $\mathcal{C}$  is applicable whenever  $1 + 2x - x^4/4 \geq 0$ . So, to check its applicability for all the points  $P_3 \in [\underline{P}_3, \overline{P}_3]$  (i.e., for all the values  $x \in [1, 2]$ ), we must check that  $1 + 2x - x^4/4 \geq 0$  for all such  $x$ . According to well-known results from calculus, the set  $\mathcal{S}$  of all the values of a continuous (and differentiable) function  $f(x) = 1 + 2x - x^4/4$  over  $x \in [1, 2]$  is an interval; the endpoints of this interval can be obtained if we take the values of the function at the endpoints of the interval  $[1, 2]$  and in all the points where the derivative  $df/dx = 0$ : the smallest of these values is the lower endpoint, and the largest of these values is the upper endpoint. For our function,  $f(1) = 2.75$  and  $f(2) = 1$ . The derivative  $df/dx = 2 - x^3$  is equal to 0 only for  $x = \sqrt[3]{2}$ ; for this  $x$ ,  $f(x) = 1 + 2x - x^4/4 = 1 + 2x - x \cdot x^3/4 = 1 + 2x - x/2 = 1 + (3/2) \cdot x = 1 + (3/2) \cdot \sqrt[3]{2}$ . By comparing these three values, we can conclude that the interval of possible values of  $f(x)$  is  $\mathcal{S} = [1, 1 + (3/2) \cdot \sqrt[3]{2}]$ . All points from this interval are positive and therefore, the construction  $\mathcal{S}$  is indeed always applicable.

While proving the first statement, we also proved that the set of all points  $Z$  obtained by using all possible points  $P_3 \in [\underline{P}_3, \overline{P}_3]$  is an interval.

To complete the proof of Theorem 3, it is sufficient to show that the endpoints of this interval are not constructible from the original input points  $P_1, P_2 = \underline{P}_3, \overline{P}_3$ . Indeed, the upper endpoint of this interval is located at a distance  $d = 1 + (3/2) \cdot \sqrt[3]{2}$  from the origin  $P_1$ . If we could construct this point, then, since we can construct the difference and the product, we would be able to also construct the interval of length  $(d - 1) \cdot (2/3) = \sqrt[3]{2}$ , which (as we have mentioned in the Proof of Theorem 2) is impossible. Thus, it is also impossible to construct the endpoints of the interval of possible points  $Z$ . The theorem is proven.

#### 4.4 Proof of Theorem 4

The proof of this theorem is similar to the proof of Theorem 1: namely, similarly to that proof, we will reduce the partition problem to the geometric problem NP-hardness of which we want to prove.

Let us start with an instance of the partition problem, i.e., with the positive integers  $s_1, \dots, s_m$  and with the problem to check whether there exist values  $x_i \in \{-1, 1\}$  for which  $s_1 \cdot x_1 + \dots + s_m \cdot x_m = 0$ . If we denote  $s_{m+1} = (1/2) \cdot (s_1 + \dots + s_m)$ , then we can formulate a new equation:  $s_1 \cdot x_1 + \dots + s_m \cdot x_m + s_{m+1} \cdot x_{m+1} = s_{m+1}$ , with  $x_i \in \{-1, 1\}$ . This equation always has a solution: namely, we can take  $x_1 = \dots = x_m = 1$  and  $x_{m+1} = -1$ . However, the solution with  $x_{m+1} = 1$  is only possible when  $s_1 \cdot x_1 + \dots + s_m \cdot x_m = 0$ , i.e., when the original instance of the partition problem has a solution.

To reduce the new problem to a geometric construction, we first (as we did in the proof of Theorem 1) reformulate this new problem in terms of the new variables  $y_i = x_i + 2$ , as the problem of checking whether there exists a solution of the equation  $s_1 \cdot y_1 + \dots + s_m \cdot y_m + s_{m+1} \cdot y_{m+1} = s_{m+1} + 2 \cdot (m + 1)$ , for which  $y_i \in \{1, 3\}$  for all  $i$ , and  $y_{m+1} = 3$ .

Then, we make the same construction as in the proof of Theorem 1, except that now, from  $y$ , we extract not  $m$  but  $m + 1$  variables  $y_i \in \{1, 3\}$ . After we perform the two final steps of the geometric construction from the proof of Theorem 1 (i.e., after we construct the points  $R$  and  $S$ ), we return the point  $Y_{m+1}$  (for which  $d(P_1, Y_{m+1}) = y_{m+1}$ ) as the final result of our construction.

Since for  $y_{m+1} = 3$ , the equation is solvable, the resulting construction  $\mathcal{C}$  is indeed applicable to the given interval data  $\{P_1, P_2, [P_1, P_2]\}$ , and whenever it is applicable, the resulting final object  $Z_{n+s}$  is indeed a point on the ray  $P_1P_2$ . Therefore, according to Definition 6, the pair

$\langle C, \{P_1, P_2, [P_1, P_2]\} \rangle$  is indeed a geometric construction problem with one interval parameter.

For this problem, the solution set is equal to one of the following two sets:

- If the original instance of the partition problem does not have a solution, then  $\mathcal{S} = \{3\}$ .
- If the original instance of the partition problem has a solution, then  $\mathcal{S} = \{1, 3\}$ .

Thus, for the interval  $\mathcal{I} = [P_1, P_2]$  ( $= [0, 1]$ ), the intersection  $\mathcal{I} \cap \mathcal{S} \neq \emptyset$  if and only if the original instance of the partition problem has a solution. Thus, we have completed the desired reduction, and therefore, our geometric problem is NP-hard. The theorem is proven.

## Conclusions

In many practical CAD/CAM problems, we must solve *geometric construction problems*, i.e., find coordinates of different points based on distances and other parameters of a technical drawing. This problem can be often solved by using ruler and compass.

In some cases, we do not know the *exact* values of some parameters, we only know the *intervals* of possible values of these parameters. In this case, we are interested in finding the *intervals* of possible values of the corresponding coordinates.

There exist methods for finding these intervals, but these methods sometimes overestimate: they produce intervals which contain the desired coordinate intervals, but which are much wider than these coordinate intervals. It was originally hoped that *sharp* (non-overestimating) interval estimates are possible at least for the cases when we have only one interval parameter (and all other parameters are known precisely). In this paper, we have shown that even for this simplest case, the problem of computing sharp interval estimates is computationally intractable (NP-hard). This means that for feasible algorithms, overestimation is inevitable.

## Acknowledgments

This research has been supported by the Comisión Interministerial de Ciencia y Tecnología (CI-CYT), under the project TIC95-0630-C05-04. N. Mata is also partially supported by a PhD. fellowship of the Associació d'Enginyers Industrials de Catalunya.

V. Kreinovich was also partially supported by NASA under cooperative agreement NCC5-209, by NSF grant No. DUE-9750858, by United Space Alliance, grant No. NAS 9-20000 (PWO C0C67713A6), by Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518, and by the National Security Agency under Grant No. MDA904-98-1-0564.

## References

- [1] I. Fudos and C.M. Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics*, 16(2):179–216, April 1997.
- [2] M. E. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.
- [3] D.J.H. Garling. *A course in Galois theory*. Cambridge University Press, 1986.
- [4] C.H. Hadlock. *Field Theory and its Classical Problems*, volume 19 of *The Carus Mathematical Monographs*. The Mathematical Association of America, 1978.

- [5] R. Joan-Arinyo. Triangles, ruler and compass. Technical Report LSI-95-6-R, Department LiSI, Universitat Politècnica de Catalunya, 1995.
- [6] R. Joan-Arinyo and N. Mata, Basic Geometric Operations in Ruler-and-Compass Constraint Solvers using Interval Arithmetic, *International Conference on Interval Methods and their Application in Global Optimization (INTERVAL'98), April 20–23, Nanjing, China, Extended Abstracts*, 1998, pp. 6–8; a detailed version has appeared as a Technical Report LSI-98-18-R, Department LiSI, Universitat Politècnica de Catalunya, April 1998, 52 pp.
- [7] R. Joan-Arinyo and A. Soto. A ruler-and-compass geometric constraint solver. In *Product Modeling for Computer Integrated Design and Manufacture*, pages 384 – 393. Chapman and Hall, London, May, 18-23 1997.
- [8] R. Joan-Arinyo and A. Soto-Riera. A correct rule-based geometric constraint solver. *Computers & Graphics*, 21(5), 1997.
- [9] V. Kreinovich, Interval rational = algebraic, *ACM SIGNUM Newsletter*, 1995, Vol. 30, No. 4, pp. 2–13.
- [10] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. *Computational Complexity and Feasibility of Data Processing and Interval Computations*, volume 10 of *Applied Optimization*. Kluwer Academic Publishers, 1998.
- [11] H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, Inc., New Jersey, 1981.
- [12] J. C. Martin, *Introduction to languages and the theory of computation*, McGraw-Hill, N.Y., 1991.
- [13] P. Morandi. *Field and Galois Theory*, volume 167 of *Graduate Texts in Mathematics*. Springer-Verlag, 1996.