Geometric Constraint Problems and Solution Instances

M. Hidalgo, R. Joan-Arinyo, A. Soto-Riera
Grup d'Informàtica a l'Enginyeria
Universitat Politècnica de Catalunya
Av. Diagonal 647, 8a, 08028 Barcelona
[mhidalgo,robert,tonis]@lsi.upc.edu

Abstract

Geometric constraint solving is a growing field devoted to solve geometric problems defined by relationships, called constraints, established between the geometric elements. In this work we show that what characterizes a geometric constraint problem is the set of geometric elements on which the problem is defined. If the problem is wellconstrained, a given solution instance to the geometric constraint problem admits different representations defined by measuring geometric relationships in the solution instance.

Keywords: Abstract Geometric Constraint Problem, Problem Instance, Solution Instance, Realization.

1 Introduction

In two-dimensional constraint-based geometric design, the designer creates a rough sketch of an object made out of simple geometric elements like points, lines, circles and arcs of circle. Then the intended exact shape is specified by annotating the sketch with constraints like distance between two points, distance from a point to a line, angle between two lines, line-circle tangency and so on. A geometric constraint solver then checks whether the set of geometric constraints coherently defines the object and, if so, determines the position of the geometric elements. The designer can now modify the values of constraints or ask the geometric constraint solver for alternative solutions that also satisfy the constraints.

Since in general geometric constraint solvers are not complete, given a geometric problem based on constraints, the set of solution instances that can be determined depends on the specific solving technique used. However, once the set of geometric objects in the problem has been fixed, it would be interesting to know whether different placements for the geometry are solutions to the same problem.

In this report we present a way of building different representations for a solution instance by properly taking measures. We deliberately avoid considering any specific geometric constraint solving technique and focus on the general concept of solution realization. That is, we focus on an actual placement of the geometric elements without considering how the placement has actually been figured out.

The outline of the paper is as follows. In Section 2 we recall declarative definitions for several basic concepts relevant in geometric constraint solving. Geometric constraint problems and their categorization are defined in Section 3. In Section 4 we identify a first order logic formula to represent solutions to geometric constraint problems. The main result on geometry and realizations is presented in Section 5. Finally, in Section 6 we offer a brief discussion.

2 Notation and Concepts

Following Joan-Arinyo [7] and Vila [9], in this section we recall concepts and notational conventions that will be used later on. We assume that a constraint-based design is made of geometric elements like point, lines, circles and arcs of circle. The intended shape is defined by means of constraints like distance between two points, distance from a point to a line, angle between two lines, line-circle tangency and so on.

In what follows, the symbols to represent geometric elements will be taken from the set

$$\mathcal{L}_G = \{p_1, l_1, c_1, p_2, l_2, c_2, \ldots, p_n, l_n, c_n, \ldots\}$$

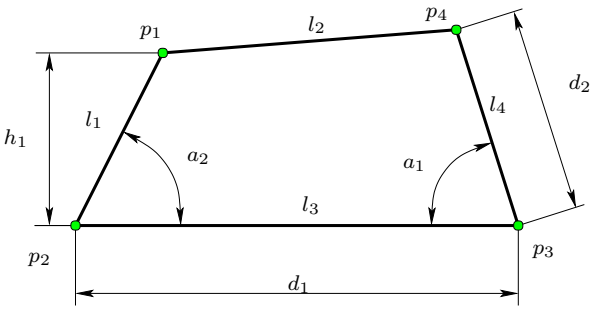
where p_i denotes a point, l_i a straight line and c_i a circle. We assume that the number of different available symbols is unlimited.

Constraints will be represented by predicates relating geometric elements or geometric elements plus a symbolic value called *parameter*. For example,

$$\begin{aligned}\mathcal{L}_R = \{&onPL(p, l),\\ &distPP(p_i, p_j, d),\\ &distPL(p_i, l_j, h),\\ &angleLL(l_i, l_j, a), \ldots\}\end{aligned}$$

Predicate names are self explanatory. The predicate $onPL(p, l)$ specifies that point p must lie on line l, $distPP(p_i, p_j, d)$ specifies a point-point distance, $distPL(p_i, l_j, h)$ defines the perpendicular distance from a point to a straight line and, $angleLL(l_i, l_j, a)$ denotes the angle between two straight lines. The number and syntax of available constraints are fixed. Symbols d, h and a are parameters. The symbols to represent parameters will be taken from the set

$$\mathcal{L}_P = \{d_1, h_1, a_1, d_2, h_2, a_2, \ldots, d_n, h_n, a_n, \ldots\}$$



$onPL(p_1, l_1)$	$onPL(p_1, l_2)$
$onPL(p_2, l_1)$	$onPL(p_2, l_3)$
$onPL(p_3, l_3)$	$onPL(p_3, l_4)$
$onPL(p_4, l_2)$	$onPL(p_4, l_4)$
$distPP(p_2, p_3, d_1)$	$distPP(p_3, p_4, d_2)$
$distPL(p_1, l_3, h_1)$	$angleLL(l_3, l_1, a_2)$
$angleLL(l_3, l_4, a_1)$	

Figure 1: Geometric problem defined by constraints.

d_i denoting a distance between two points, h_i a distance between a point and a line and a_i an angle between two lines. Figure 1 shows an example of a constraint-based design and the set of constraints defined between the geometric elements.

Given a set of symbols S and a set of values V, a *textual substitution* α is a total mapping from S to V. Let W be a set of predicates and α a textual substitution, we note by $\alpha.W$ the set of predicates obtained by replacing every occurrence of any symbol $s \in S$ found in W by $\alpha(s) \in V$.

Example 2.1 Let $S = \{a_1, h_1\}$ be a set of symbols and $V = \mathbb{R}$. Let α a textual substitution from S to V defined as

$$\alpha(a_1) = 0.57, \quad \alpha(h_1) = 4.0$$

and let W be a set of predicates in $\mathcal{L}_R$ with

$$\begin{aligned} W \quad = \quad & \{onPL(p_1, l_1), angleLL(l_1, l_3, a_1), \\ & distPL(p_1, l_3, h_1)\}. \end{aligned}$$

Then $\alpha.W$ is

$$\begin{aligned} \alpha.W \quad = \quad & \{onPL(p_1, l_1), angleLL(l_1, l_3, 0.57), \\ & distPL(p_1, l_3, 4.0)\}. \end{aligned}$$

$\diamond$

In this paper we will also apply textual substitutions to first order logic formulae and other syntactical descriptions.

3 Geometric Constraint Problems

We define and describe declaratively the concepts of abstract geometric constraint problem and of instance of a geometric constraint problem. Abstract entities are exclusively defined in terms of symbols like those in the sets $\mathcal{L}_G$, $\mathcal{L}_P$ and $\mathcal{L}_I$. Instance entities are abstract entities where some of the symbols occurring in them have been replaced by values.

3.1 Abstract Problem

An *abstract geometric constraint problem*, or *abstract problem* in short, is a tuple $A = \langle G, C, P \rangle$ where G is a set of symbols in $\mathcal{L}_G$ denoting geometric elements, C is a set of constraints taken from $\mathcal{L}_R$ and defined between elements of G, and P is the set of parameters taken from $\mathcal{L}_P$.

Example 3.1 Consider the sketch with annotated dimension lines shown in Figure 1. It can be seen as an abstract problem $A = \langle G, C, P \rangle$ where the set of geometric elements is

$$G = \{p_1, p_2, p_3, p_4, l_1, l_2, l_3, l_4\} ,$$

C is the set of constraints listed in Figure 1 and, the set of parameters is

$$P = \{d_1, d_2, a_1, a_2, h_1\} .$$

$\diamond$

A convenient way to fully describe an abstract problem is the algorithm-like notation. In this notation, the abstract problem in Example 3.1 can be expressed as

gcp A
  param
    d_1, d_2, a_1, a_2, h_1 : **real**
  endparam
  geom
    p_1, p_2, p_3, p_4 : **point**
    l_1, l_2, l_3, l_4 : **line**
  endgeom

$onPL(p_1, l_1)$
$onPL(p_1, l_2)$
$onPL(p_2, l_1)$
$onPL(p_2, l_3)$
$onPL(p_3, l_3)$
$onPL(p_3, l_4)$
$onPL(p_4, l_4)$
$onPL(p_4, l_2)$
$distPP(p_2, p_3, d_1)$
$distPP(p_3, p_4, d_2)$
$distPL(p_1, l_3, h_1)$
$angleLL(l_3, l_1, a_2)$
$angleLL(l_3, l_4, a_1)$
endgcp

Note that an abstract problem defines a family of geometric constraint solving problems parameterized by the set P.

3.2 Problem Instance

A *parameters assignment* is a textual substitution α from a set of parameters P to $\mathbb{R}$.

Let $A = \langle G, C, P \rangle$ be an abstract problem and α be a parameters assignment from P. We say that $\alpha.A = \langle G, \alpha.C, P \rangle$ is a *problem instance* of A. Note that given an abstract problem, each different parameters assignment defines a different problem instance.

Example 3.2 Consider the abstract problem $A = \langle G, C, P \rangle$ described in the Example 3.1. An example of parameters assigment α is

$$\begin{aligned} \alpha(a_1) &= -1.222 \\ \alpha(a_2) &= 1.0472 \\ \alpha(h_1) &= 160.0 \\ \alpha(d_1) &= 290.0 \\ \alpha(d_2) &= 130.0 \end{aligned}$$

A description for the problem instance $\alpha.A$ is

gcp $\alpha.A$
  param
    d_1, d_2, a_1, a_2, h_1 : **real**

endparam
geom
p_1, p_2, p_3, p_4 : **point**
l_1, l_2, l_3, l_4 : **line**
endgeom
$onPL(p_1, l_1)$
$onPL(p_1, l_2)$
$onPL(p_2, l_1)$
$onPL(p_2, l_3)$
$onPL(p_3, l_3)$
$onPL(p_3, l_4)$
$onPL(p_4, l_4)$
$onPL(p_4, l_2)$
$distPP(p_2, p_3, 290.0)$
$distPP(p_3, p_4, 130.0)$
$distPL(p_1, l_3, 160.0)$
$angleLL(l_3, l_1, 1.0472)$
$angleLL(l_3, l_4, -1.222)$
endgcp

◇

Problem instances are no longer parameterized because the parameters have been replaced by the corresponding actual values.

Figure 2 shows a picture for the problem instance $\alpha.A$ given in Example 3.2. Now parameters are no longer symbolic but actual values defined by the assignment α. This picture is a declarative description of the geometric elements and constraints and, thus the actual geometry is irrelevant. For example, the actual values of h_1 and d_2 in the figure do not match the values defined by $\alpha(h_1)$ and $\alpha(d_2)$.

Notice that abstract problems precisely describe a set of geometric elements and the constraints that must fulfill, but they do not define how to place the geometric elements to satisfy the constraints.

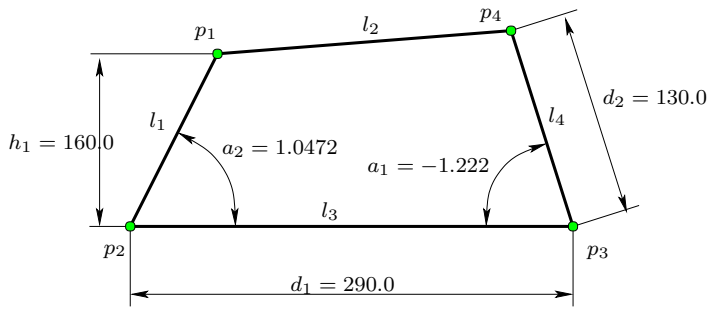


Figure 2: Problem instance.

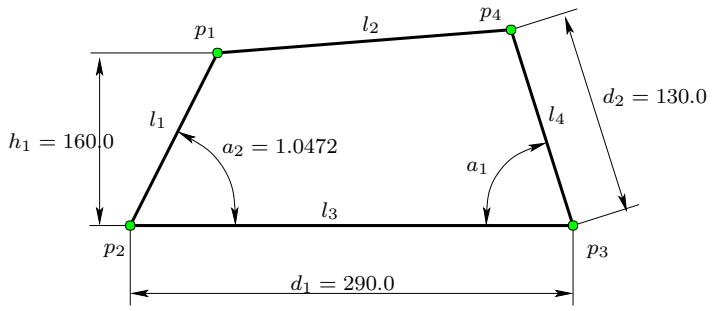


Figure 3: Problem instance with one degree of freedom, a_1.

3.3 Problem Instance with One Degree of Freedom

A *partial parameters assignment* is a textual substitution α from a subset of parameters $P' \subset P$ to $\mathbb{R}$. That is, a partial parameters assignment, or *partial assignment* in short, is an assignment such that at least one parameter in P is not assigned a specific value.

Let $A = \langle G, C, P \rangle$ be an abstract problem and α be a parameters assignment from P such that just one parameter in P has not been assigned a specific value. We say that $\alpha.A = \langle G, \alpha.C, P \rangle$ is a *problem instance* of A with *one degree of freedom*.

Figure 3 shows an instance of the problem in Figure 2 where angle a_1 has not been assigned a value. a_i is the problem degree of freedom. Problems with one degree of freedom will play a central role in this work.

3.4 Problem Categorization

One of the main goals of the geometric constraint solving community focuses on solutions to a constraint problem that are determined up to a global coordinate system, that is, where solutions are congruent under the rigid-body transformations of translation and rotation. We call a configuration of geometric objects in Euclidean space *rigid* when all objects are fixed with respect to each other up to translation and rotation.

An intuitive way to introduce rigidity comes from considering the number of solutions that a geometric constraint problem has. There are three categories: A problem is *structurally under constrained* if there are infinitely many solutions that are not congruent under rigid transformation, *structurally well-constrained*, if there are finitely many solutions modulo rigid transformation, and *structurally over constrained* if the deletion of one or more constraints results in a well-constrained problem. A constraint problem naturally corresponds to a set of (usually nonlinear) algebraic equations.

Defined in this way, the concept of rigidity appears to be simple but it is not quite in

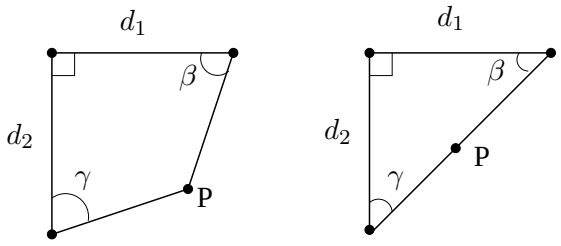


Figure 4: Left: General configuration. Right: Degenerate configuration for $\gamma+\beta = 90^\circ$.

accord with the intuition about rigidity. The categories so defined only refer to the problem's structure and do not account for other issues such as inconsistencies that could originate from specific values assigned to the constraints. Clearly a problem that is structurally well-constrained could actually be underconstrained for specific values of the constraints.

For example, consider the *structurally* well constrained problem given in Figure 4, see Fudos and Hoffmann, [2]. Point P is properly placed whenever $\gamma + \beta \neq 90^\circ$ and the problem is well-constrained. But if $\gamma + \beta = 90^\circ$, then the placement for point P is undetermined and, therefore, the problem is no longer well constrained.

Different formal definitions of rigidity have been explored in the literature. See, for example, the work by Henneberg, [4], and Laman, [8], or the more recent works by Graver *et al.* [3], Fudos and Hoffmann, [2], Hoffmann *et al.*, [6], and Whitley, [10, 11].

4 Solution to a Geometric Constraint Problem

Many attempts to provide general, powerful and efficient methods for solving problems of geometric constraints have been reported in the literature. For an extensive review in geometric constraint solving techniques refer to Durand [1] and to Hoffmann and Joan-Arinyo [5]. The work reported here does not depend on any specific geometric constraint solving technique. Therefore, we will not go further in this subject.

Existing geometric constraint solving techniques have been developed under the assumption that problems are wellconstrained. As defined in Section 3.4, in well-constrained problems the number of constraints and their placement on the geometric elements define a problem with a finite number of solutions for nondegenerate configurations. In what follows we only consider wellconstrained geometric constraint problems.

4.1 The First Order Formula

For our pourposes, it is convenient to interpret geometric constraint problems as first order logic formulae.

Let $A = \langle G, C, P \rangle$ be an abstract geometric constraint problem with the set of constraints $C = \{c_1, c_2, \ldots, c_m\}$. Then the geometric constraint problem A can be expressed as the first order logic formula,

$$\Sigma(A) \equiv \bigwedge_{i=1}^{m} c_i$$

where the geometric elements of G and the parameters of P occurring in Σ are interpreted as free variables.

Example 4.1 The first order formula of the abstract problem A given in Example 3.1 is

$$\begin{aligned}
\Sigma(A) \quad \equiv \quad & \mathit{onPL}(p_1, l_1) \wedge \mathit{onPL}(p_1, l_2) \wedge \\
& \mathit{onPL}(p_2, l_1) \wedge \mathit{onPL}(p_2, l_3) \wedge \\
& \mathit{onPL}(p_3, l_3) \wedge \mathit{onPL}(p_3, l_4) \wedge \\
& \mathit{onPL}(p_4, l_2) \wedge \mathit{onPL}(p_4, l_4) \wedge \\
& \mathit{distPP}(p_2, p_3, d_1) \wedge \\
& \mathit{distPP}(p_3, p_4, d_2) \wedge \\
& \mathit{distPL}(p_1, l_3, h_1) \wedge \\
& \mathit{angleLL}(l_3, l_1, a_2) \wedge \\
& \mathit{angleLL}(l_3, l_4, a_1)
\end{aligned}$$

$\diamond$

Note that, as defined, a textual substitution α can be applied to both an abstract problem and to a first order logic formula. Therefore, the relation $\Sigma(\alpha.A) = \alpha.\Sigma(A)$ is well defined. Let α be a parameters assignment for P and $\alpha.A$ the corresponding instance problem. Then the first order formula $\Sigma(\alpha.A)$ expresses the instance problem. See Example 4.2.

Example 4.2 The first order formula for the instance problem A given in Exam-

ple 3.1 is

$$\begin{aligned}\Sigma(\alpha.A) \equiv\ & \textit{onPL}(p_1, l_1) \wedge \textit{onPL}(p_1, l_2) \wedge \\ & \textit{onPL}(p_2, l_1) \wedge \textit{onPL}(p_2, l_3) \wedge \\ & \textit{onPL}(p_3, l_3) \wedge \textit{onPL}(p_3, l_4) \wedge \\ & \textit{onPL}(p_4, l_2) \wedge \textit{onPL}(p_4, l_4) \wedge \\ & \textit{distPP}(p_2, p_3, 290.0) \wedge \\ & \textit{distPP}(p_3, p_4, 130.0) \wedge \\ & \textit{distPL}(p_1, l_3, 160.0) \wedge \\ & \textit{angleLL}(l_3, l_1, 1.0472) \wedge \\ & \textit{angleLL}(l_3, l_4, -1.222)\end{aligned}$$

◇

4.2 Solution Instances

First we define the concept of *realization*. Let $A = \langle G, C, P\rangle$ be an abstract geometric constraint problem. A *geometry assignment* or *realization*, $\rho.A$, is a textual substitution that embeds in $\mathbb{R}^2$ the geometric elements in P by assigning an actual geometry to each element in the set of geometric symbols G. That is, $\rho.A = \langle \rho.G, C, P\rangle$. Example 4.3 illustrates this concept.

Example 4.3 Let (x, y) denote a point in $\mathbb{R}^2$ and (a, b, c) be the coefficients of the straight line $ax + by + c = 0$ with $a^2 + b^2 = 1$. Then an example of realization for the abstract problem A given in Example 3.1 is

$$\begin{aligned}\rho(p_1) &= (92.38, 160)\\ \rho(p_2) &= (0, 0)\\ \rho(p_3) &= (290, 0)\\ \rho(p_4) &= (245.54, 122.16)\\ \rho(l_1) &= (-0.87, 0.5, 0)\\ \rho(l_2) &= (-0.24, -0.97, 177.48)\\ \rho(l_3) &= (0, -1, 0)\\ \rho(l_4) &= (0.94, 0.34, -272.51)\end{aligned}$$

◇

It is easy to see that the realtionships $\rho.\Sigma(A) = \Sigma(\rho.A)$ and $\rho.(\alpha.(\Sigma(A))) = \Sigma(\rho.\alpha.A))$ are welldefined. Moreover, α and ρ commute. Deriving the first order

formula Σ resulting from applying the realization ρ defined in Example 4.3 to the abstract formula given in Example 4.1 is routine matter.

Next we dfine the concept of *solution instance*. Let $A = \langle G, C, P\rangle$ be an abstract problem on geometric constraints, let $\alpha.A$ be a problem instance and let $\rho.(\alpha.A) = \langle G, \rho.(\alpha.C), P\rangle$ be a realization of $\alpha.A$. We say that $\rho.(\alpha.A)$, in short $\rho.\alpha.A$, is a *solution instance* to the problem instance $\alpha.A$ if and only if all the constraints in C hold for the realization $\rho.\alpha.A$.

Example 4.4 Solution instance for the instance problem A given in Example 3.1 and realization given in Example 4.3.

$$
\begin{aligned}
\Sigma(\rho.\alpha.A) \quad \equiv \quad & \textit{onPL}((92.38, 160), (-0.87, 0.5, 0)) \wedge \\
& \textit{onPL}((92.38, 160), (0, -1, 0)) \wedge \\
& \textit{onPL}(0, 0), (-0.87, 0.5, 0)) \wedge \\
& \textit{onPL}((0, 0), (0.94, 0.34, -272.51)) \wedge \\
& \textit{onPL}((290, 0), (0, -1, 0)) \wedge \\
& \textit{onPL}((290, 0), (0.94, 0.34, -272.51)) \wedge \\
& \textit{onPL}((245, .54, 122.16), (-0.24, -0.97, 177.48)) \wedge \\
& \textit{onPL}((245.54, 122.16), (0.94, 0.34, -272.51))) \wedge \\
& \textit{distPP}((0, 0), (290.0), 290.0) \wedge \\
& \textit{distPP}((290, 0), (245.54), 122.16), 130.0) \wedge \\
& \textit{distPL}((92.38, 160), (0, -1, 0), 160) \wedge \\
& \textit{angleLL}((0, -1, 0), (-0.87, 0.5, 0), 1.0472) \wedge \\
& \textit{angleLL}((0, -1, 0), (0.94, 0.34, -272.51), -1.222)
\end{aligned}
$$

$\diamond$

Note that the set of solution instances for the instance problem $\alpha.A$ is the set of realizations $\rho.A$ for which $\Sigma(\rho.\alpha.A)$ holds.

5 Geometries and Realizations

In this section we show how solution instances to different abstract problems defined on the same set of geometric symbols are related once the realization ρ has been fixed. We start by stating a trivial lemma.

Lemma 5.1 *Let $A = \langle G, C, P\rangle$ be an abstract wellconstrained geometric constraint problem and let $\alpha.A = \langle G, \alpha.C, P\rangle$ be a problem instance. Let $\rho.\alpha.A$ be a realization of problem A. Then for any pair of symbols $g_i, g_j \in G$ any relationship between them can be measured in $\rho.\alpha.A$.*

Proof
Since the problem is wellconstrained, just consider that any realization $\rho.\alpha.A$ places all the geometric elements in G in a common reference. □

Next we state and prove the theorem that relates solution instances for different abstract problems on the same set of geometric symbols for a given realization ρ.

Theorem 5.2 *Let $A_1 = \langle G, C_1, P_1 \rangle$ and $A_2 = \langle G, C_2, P_2 \rangle$ be two wellconstrained abstract geometric constraint problems defined on the same set of geometric elements G. Let $\rho.\alpha_1.A_1$ be a solution instance for the problem instance $\alpha_1.A_1$. Let α_2 be an assigment such that replaces each symbol $p_i \in P_2$ with the corresponding measure taken in $\rho.\alpha_1.A_1$. Then the realization $\rho.\alpha_2.A_2$ is a solution instance to the problem instance $\alpha_2.A_2$.*

Proof
The actual solution instance $\rho.\alpha_1.A_1$ is a realization for the wellconstrained problem instance $\alpha_1.A_1$. Thus Lemma 5.1 allows measuring actual values for the constraints parameters in P_2. Since the realization ρ is preserved, $\rho.\alpha_2.A_2$ is a solution instance to the wellconstrained problem instance $\alpha_2.A_2$. □

Example 5.1 Consider the constraint problems $A_1 = \langle G, C_1, P_1 \rangle$ and $A_2 = \langle G, C_2, P_2 \rangle$ be two abstract geometric constraint problems with points

$$G = \{A, B, C, D, E\}$$

and respective distance constraints

$$C_1 = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$$
$$C_2 = \{d_1, d_2, d'_3, d_4, d_5, d_6, d_7\}$$

Assume that Figure 5 left is the realization $\rho.\alpha_1.A_1$. If we define α_2 by replacing in α_1 the constraint d_3 with d'_3 and assigning to d'_3 the value resulting from measuring the distance between points A and D in the realization of A_1, then $\rho.\alpha_2.A_2$ is also a realization for the problem A_2. See Figure 5 right.

◇

Finally, we have

Corollary 5.3 *Let $A_\lambda = \langle G, C_\lambda, P_\lambda \rangle$ and $A_\mu = \langle G, C_\mu, P_\mu \rangle$ be two wellconstrained geometric constraint problems, with one degree of freeedom, defined on the same set of geometric elements such that $C_\lambda - \{\lambda\} = C_\mu - \{\mu\}$. Then the problem instances $\alpha_\lambda.A_\lambda$ and $\alpha_\mu.A_\mu$ have the same set of solution instances.*

Proof
For each value assigned to the degree of freedom of one of the problems, apply Theorem 5.2. □

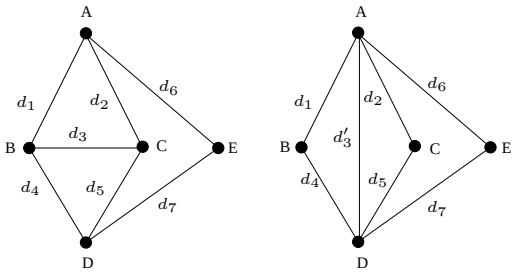


Figure 5: Left) Realization $\rho.\alpha_1.A_1$ Right) Realization $\rho.\alpha_2.A_2$ where constraint d'_3 has been assigned a value measured in $\rho.\alpha_1.A_1$.

6 Discussion

We have shown that given two wellconstrained abstract geometric problems defined by constraints on the same set of geometric elements, once we know one realization for one solution instance, we can derive different parameters assignments that result in different representations of the same solution instance just by measuring properies in the given solution instance.

This is a formal statement of the fact that given a finite set of geometric elements, we can place them with respect to each other in the twodimensional space at our will without changing the nature of the placing problem. That is, what characterizes a geometric constraint problem is not the set of constraints but the set of geometric elements.

Note that the geometric constraint solving method that may be used to solve the problem, does not play any role in this discussion.

Acknowledgements

This research has been partially funded by the Spanish Ministerio de Educación y Ciencia and by FEDER under grant TIN2007-67474-C03-01.

References

[1] C. Durand. *Symbolic and Numerical Techniques for Constraint Solving*. PhD thesis, Purdue University, Department of Computer Sciences, December 1998.

[2] 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.

[3] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*. American Mathematical Society, 1993.

[4] L. Henneberg. *Die Graphische Statik de Starren Systeme*. Leipzig, 1911.

[5] C.M. Hoffmann and R. Joan-Arinyo. A brief on constraint solving. *Computer-Aided Design and Applications*, 2(5):655–663, 2005.

[6] C.M. Hoffmann, A. Lomonosov, and M. Sitharam. Finding solvable subsets of constraint graphs. In *Principles and Practice of Constraint Programming*, pages 463–477, Schloss Hagenberg, Austria, October 29 - November 1 1977.

[7] R. Joan-Arinyo, A. Soto-Riera, S. Vila-Marta, and J. Vilaplana. Declarative characterization of a general architecture for constructive geometric constraint solvers. In D. Plemenos, editor, *The Fifth International Conference on Computer Graphics and Artificial Intelligence*, pages 63–76, Limoges, France, 14-15 May 2002. Université de Limoges.

[8] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4(4):331–340, October 1970.

[9] S. Vila. *Contribution to Geometric Constraint Solving in Cooperative Engineering*. PhD thesis, Departament Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 2003.

[10] W. Whiteley. Applications of the geometry of rigid structures. In Henry Crapo, editor, *Computer Aided Geometric Reasoning*, pages 219–254. INRIA, 1987.

[11] W. Whiteley. Rigidity and scene analysis. In J.E. Goodman and J. O'Rourke, editors, *Handbook for discrete and computational geometry*, pages 893–916. CRC Press LLC, 1998.