A Data Structure for Solving Geometric Construction Problems with Interval Parameters

R. Joan-Arinyo and N. Mata
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Av. Diagonal 647, 8ª, 08028 Barcelona
e-mail: [robert, nmata]@lsi.upc.es

April 4, 2000

Abstract

In advanced computer-aided design systems, an object is defined by a finite collection of points, linear segments, and circular segments, and a finite set of geometric constraints relating these points and segments. Given a set of values for the constraint parameters, the geometric problem consists of computing the coordinates of the points of the object. Traditionally, constraint parameters have been considered to be exact values. In many applications, however, it is more realistic to consider that some dimensions of the object may have tolerances and, therefore, that the corresponding constraint parameters are intervals. In this situation, the geometric problem becomes indeed more difficult to solve because the resulting coordinates of the points are also intervals. In this paper, we show how constructible geometric problems can be solved when the constraint parameters are intervals.

Keywords: CAD, geometric constraint solving, interval computations.

1 Introduction

One of the main objectives of Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM) is to transform the sketch of an object given by the designer into explicit manufacturing instructions. The main difficulty in this transformation is that the sketch describes the shape of the object in terms of geometric constraints (distances between points, angle between lines, tangency relations, etc.); while the information required in the manufacturing process is the exact coordinates of the points, and the exact (coordinate) equations of the corresponding lines. This leads to the problem of obtaining coordinates of points from the geometric input data in the sketch.

We consider the simplest case of 2D objects which consist of linear segments and circular segments with constant radii. From the mathematical point of view, these linear and circular segments are represented by a set of points: a linear segment is represented by the start- and the endpoint; a circular segment is represented by the centerpoint of the circle, and the endpoints of the segment on the circle. We call endpoints of segments, and the
centerpoints of circles, the points of the object. The values the coordinates of these points can simultaneously take are restricted by the constraints of the geometric problem. Given a solvable geometric problem and a set of values assigned to the constraint parameters, our goal is to compute the coordinates of all the points of the object.

In this work, however, we are not concerned with the problem of solvability of the geometric problem, namely, the problem of determining whether or not the geometric constraints correctly define the object. That is a central problem of many other research works [7, 17, 19]. Our concern here is, given a well-defined geometric problem and a set of intervals assigned to the constraint parameters, how we can solve the numerical problem of computing the interval coordinates of the points of the object.

In principle, each geometric constraint describes an equation relating coordinates of the points of the object. For instance, if we know the distance \(d\) between the points \(P_1 = (x_1, y_1)\) and \(P_2 = (x_2, y_2)\), we thus have the equation \((x_1 - x_2)^2 + (y_1 - y_2)^2 = d^2\). For the set of geometric constraints, therefore, we get a system of equations that, once solved, gives all the coordinates of the points. The main problem with this approach is that we get very large systems of non-linear equations with multiple solutions which are, in general, difficult to solve. Instead of using time-consuming methods to solve such general systems, it is possible to develop specific techniques which take into account the special features of the domain of the problem. Since our problem is geometric, it is natural to look for geometric methods of solving it. For a reasonably large class of geometric constraint problems, such a solution is provided by constructive geometric constraint solving systems [6, 16, 18].

Constructive geometric constraint solving systems transform the original constraints into the coordinates of the points by using only a certain set of tools; for instance, ruler, compass and protractor. As a result, they provide a sequence of basic construction steps (an algorithm) that enables to compute, one-by-one, all the points of the object. We denote the realization of this geometric algorithm as the geometric construction problem.

In this work, we are given a geometric problem defined by a sequence of basic construction steps, and an assignment of interval values to the geometric constraint parameters. Our goal is to compute the range of the coordinates of the points for this set of interval parameters. To do so, first we describe our model of geometric problem and we establish how a sequence of basic construction steps represents a unique solution object. Second, we show how the sequence of basic construction steps is transformed into a structured list of basic geometric functions. Third, we show how to compute a tight enclosure for the range of each geometric function over the interval input parameters.

The paper is organized as follows. Section 2 introduces the basic concepts of constructible geometric problems. In Section 3, we formulate the geometric construction problem with interval parameters. Section 4 shows how the range of the coordinates of the points is computed. We present some numerical results in Section 5, and we give our conclusions in Section 6.

2 Constructible geometric problems

A geometric object is defined by a collection of points, linear segments, and circular segments with constant radii, and a set of geometric relations between these points and segments, which are the constraints of the problem. The types of geometric constraints we
Figure 1: Geometric problem defined by constraints.

consider are, among others, distance between two points, distance from a point to a line, angle between two rays (infinite semi-lines), point on a segment, and segment tangent to a circle.

From the mathematical point of view, linear and circular segments are represented by sets of points: a linear segment is represented by its endpoints, and a circular segment is represented by the centerpoint of the circle, and the two endpoints of the segment on the circle. All considered geometric constraints can be transformed into the following relations [12]: distance between two points, distance from a point to the line defined by a linear segment, and angle between two rays, each ray also defined by a linear segment.

A geometric problem is then defined by a finite set of points \( \{P_1, \ldots, P_n\} \), and a finite set of geometric constraint parameters \( \mathcal{C} = \{c_1, \ldots, c_m\} \). Each parameter \( c_i \) is either a distance or an angle. We assume that the set of geometric constraints correctly define the object under design, and that the object is generically well-constrained [7, 11].

For instance, the sketch of Figure 1 defines a geometric problem with points \( \{P_1, P_2, P_3, P_4, P_5\} \). The initial set of constraints on these points is transformed in such a way that the constraint parameters are only distances and angles. The geometric constraints point on a circle are translated into distance relations. Namely,

\[
\begin{align*}
on(P_3, circle(P_4, d_3)) & \rightarrow distance(P_3, P_4) = d_3 \\
on(P_5, circle(P_4, d_3)) & \rightarrow distance(P_5, P_4) = d_3
\end{align*}
\]

Analogously, the constraints line tangent to a circle are transformed into distances from a point to a line. Namely,

\[
\begin{align*}
tangent(line(P_2, P_3), circle(P_4, d_3)) & \rightarrow distance(P_4, line(P_2, P_3)) = d_3 \\
tangent(line(P_1, P_5), circle(P_4, d_3)) & \rightarrow distance(P_4, line(P_1, P_5)) = d_3
\end{align*}
\]

For the example of Figure 1, the set of constraint parameters is therefore the set \( \mathcal{C} = \{d_1, d_2, d_3, a_1\} \).

Then, solving the geometric problem of Figure 1 consists of computing the coordinates of the set of points \( \{P_1, P_2, P_3, P_4, P_5\} \) from the geometric input data \( \{d_1, d_2, d_3, a_1\} \). The linear and circular segments in the initial sketch can then easily be obtained from the coordinates of the points of the object.
2.1 Basic geometric constructions

We assume that a constructive geometric constraint solving system provides us with a construction plan for the object of our geometric problem. That is, we assume that we have a sequence of construction steps that enables to place, one-by-one, all the points of the object using a finite collection of basic geometric constructions.

For a reasonably large class of geometric problems in CAD/CAM, it is enough to take these steps to be constructible with ruler, compass, and protractor. Such a sequence of basic construction steps can be obtained, for instance, using the geometric constraint solving systems in [3, 5, 10]. For further details on constructive solving techniques, see also references [4, 6, 18, 16, 20].

Using a ruler, we can construct a line between two given points; using a compass, we can construct a circle centered at a given point with a given radius; using a protractor, we can add or subtract given angles. Given two different lines or circles, we can construct new points by intersecting these lines and circles.

The sequence of basic construction steps we consider has the following properties. We assume that we have two initial points $P_1$ and $P_2$ for which their distance is known. The point $P_1$ is taken to be the origin of the coordinate system; the positive $x$-axis is then selected in the direction from $P_1$ to $P_2$. A right-handed Cartesian coordinate system is now fixed. We take $P_1$ and $P_2$ to be two initial constructible points. Furthermore, we assume that a set of constraint parameters $C = \{c_1, \ldots, c_m\}$ is given. An arbitrary point $P$ is constructible according to the following definition.

**Definition 2.1** A point $P \in \mathbb{R}^2$ is constructible with respect to $P_1, P_2,$ and $C$, if there exists a finite sequence of points in the plane, $P_1, P_2, \ldots, P_n = P$, with the following property. Let $\mathcal{X}_k$ be the set of points $\{P_1, P_2, \ldots, P_k\}$ with $k \in \{2, \ldots, n\}$.

For each $k \in \{3, \ldots, n\}$, $P_k$ is either

(i) the intersection of two distinct $\mathcal{X}_{k-1}$-constructible lines, or

(ii) a point of intersection of a $\mathcal{X}_{k-1}$-constructible line and a $\mathcal{X}_{k-1}$-constructible circle, or

(iii) a point of intersection of two non-concentric $\mathcal{X}_{k-1}$-constructible circles.

A $\mathcal{X}_k$-constructible circle and a $\mathcal{X}_k$-constructible line are defined as follow:

**Definition 2.2** A circle is $\mathcal{X}_k$-constructible if its centerpoint is in $\mathcal{X}_k$, and its radius is $\mathcal{X}_k$-constructible.

**Definition 2.3** A radius is $\mathcal{X}_k$-constructible if it is either the distance between two points in $\mathcal{X}_k$, or a given distance constraint parameter in $C$.

**Definition 2.4** A line is $\mathcal{X}_k$-constructible if it can be constructed through a point in $\mathcal{X}_k$ at an $\mathcal{X}_k$-constructible angle.

**Definition 2.5** An angle is $\mathcal{X}_k$-constructible if it is either the direction defined by two points in $\mathcal{X}_k$ with respect to the $x$-axis (given by $P_1$ and $P_2$), or an angle constraint parameter in $C$, or the sum or the difference of two such angles.
Figure 2: Selection criterion for point $P_k$ in the intersection between a line and a circle.

Let an object $\mathcal{X}$ be defined by a given set of points $\{P_1, \ldots, P_n\}$.

**Definition 2.6** The object $\mathcal{X}$ is constructible if and only if all the points $P_1, \ldots, P_n$ are constructible.

Note that intersection operations involving circles may lead to more than one constructible point. For a given object, however, we are interested in only one of these points, preferably, the one the designer intended to describe in the sketch. In order to characterize this point, and to be able to reconstruct it, we define the following point selection criteria for the intersection of a line and a circle, and for the intersection of two circles.

### 2.2 Point selection criteria

Let $U_1$ be either a line or a circle, and let $U_2$ be a circle. For an intersection point $P_k$, let $P_k = \text{intersect}(U_1, U_2, s_k)$ denote an intersection operation involving, at least, one circle. The variable $s_k$ is an integer parameter in the set $\{-1, 0, 1\}$ that we define in the following way.

We consider first the intersection of a line $U_1$ and a circle $U_2$. Let $P_i$ be a point on the line $U_1$, and let $P_j$ be the centerpoint of the circle $U_2$. We define $\vec{r}$ as the vector from $P_j$ to $P_k$, and $\vec{s}$ as the unit vector of the ray from $P_i$ to $P_k$. The parameter $s_k$ is then the sign of the dot product $\vec{r} \cdot \vec{s}$. If $\vec{r} \cdot \vec{s} > 0$, then $s_k = +1$; if $\vec{r} \cdot \vec{s} = 0$, then $s_k = 0$; and if $\vec{r} \cdot \vec{s} < 0$, then $s_k = -1$.

The geometric meaning of $s_k$ is illustrated in Figure 2. Let $Q_j$ be the projection of $P_j$ on the ray. Looking from $P_j$ to $Q_j$, if $s_k = +1$, then the point $P_k$ is the most distant intersection point from $P_i$; if $s_k = -1$, then $P_k$ is the closest intersection point to $P_i$. We use $s_k = 0$ to denote that the line $U_1$ is necessarily tangent to the circle and, then, $P_k$ is exactly $Q_j$.

In the intersection of two circles, the parameter $s_k$ is defined as follows. Let $P_i$ and $P_j$ be the centerpoints of the circles $U_1$ and $U_2$. Let $\vec{r}$ denote the vector from $P_i$ to $P_j$, and let $\vec{s}$ denote the vector from $P_i$ to $P_k$. Then, $s_k$ takes the sign of the cross product $\vec{r} \times \vec{s}$. If $\vec{r} \times \vec{s} > 0$, then $s_k = +1$; if $\vec{r} \times \vec{s} = 0$, then $s_k = 0$; and if $\vec{r} \times \vec{s} < 0$, then $s_k = -1$.

In the intersection of two circles, the geometric meaning of $s_k$ is illustrated in Figure 3. Looking from $P_i$ in the direction of $\vec{r}$, $s_k = +1$ means that the point $P_k$ is on the left of
the line defined by $\vec{r}$; if $s_k = -1$, the point $P_k$ is on the right of the line. If $s_k = 0$, we denote that $P_k$ is the point where the two circles are tangent and, therefore, $P_k$ is on the line.

For the geometric problem of Figure 1, the object defined by the set of points $\{P_1, \ldots, P_5\}$ is constructible, and a possible construction plan for this object (the construction plan given by the geometric constraint solving system developed in [18]) is the following

$P_1 := \text{point}(0,0)$
$P_2 := \text{point}(d_1,0)$
$\alpha_1 := \text{direction}(P_1, P_2)$
$\alpha_2 := \text{adif}(\alpha_1, a_1)$
$P_3 := \text{rc}(\text{ray}(P_2, \alpha_2), \text{circle}(P_2, d_2), s_p_3)$
$\alpha_3 := \text{direction}(P_2, P_3)$
$\alpha_4 := \text{asum}(\alpha_3, \frac{\pi}{2})$
$Q_1 := \text{lc}(\text{line}(P_2, \alpha_4), \text{circle}(P_2, d_3), s_{q_1})$
$P_4 := \text{lc}(\text{line}(Q_1, \alpha_3), \text{circle}(P_3, a_3), s_{p_4})$
$Q_2 := \text{midpoint}(P_1, P_4)$
$\tau_1 := \text{distance}(P_1, Q_2)$
$P_5 := \text{cc}(\text{circle}(P_4, d_3), \text{circle}(Q_2, \tau_1), s_{p_5})$
where \textit{asum} and \textit{adif} denote the addition and subtraction of angles, and \textit{rc}, \textit{lc}, and \textit{cc}, the intersection of a ray and a circle, a line and a circle, and two circles, respectively. The operation \textit{midpoint} is a composite geometric operation in which the point equidistant to the endpoints of the segment is constructible, and the constant angle $\pi/2$ is also constructible. For the complete set of basic geometric constructions, refer to [8].

With respect to the selection criteria for the intersection points $P_3, Q_1, P_3$ and $P_5$, the integer parameters $s_{p_3}, s_{q_1}, s_{p_4}$ and $s_{p_5}$ take values

$$s_{p_3} = s_{q_1} = 1, \quad s_{p_4} = 0, \quad \text{and} \quad s_{p_5} = -1$$

in the sketch of Figure 1.

### 2.3 The set of characteristic points

Note that the points of the object $\{P_1, \ldots, P_5\}$ are only a subset of the sequence of constructible points $\{P_1, P_2, P_3, Q_1, P_4, Q_2, P_5\}$. As illustrated in Figure 4a, placing the point $P_4$ needs the auxiliary point $Q_1$ to be constructible with ruler and compass; in Figure 4b we show that we can only construct the point $P_5$ if the midpoint $Q_2$ has already been constructed. This usually happens in composite geometric constructions such as line tangent to a circle, or line parallel to another line, [8]. Composite geometric constructions lead, in a natural way, to the definition of \textit{set of characteristic points} given below.

**Definition 2.7** Let $P_1, \ldots, P_n$ be the finite sequence of points required to construct all the points of an object. The set of points $\{P_1, \ldots, P_n\}$ is denoted the set of characteristic points of the object.

### 2.4 The construction plan

In order to construct all the points of the object, we not only have to construct the set of characteristic points, but also additional dimensional parameters such as radius and angles. Therefore, for the geometric object $\mathcal{X}$, a general ruler-and-compass construction plan has the form:

$$e_1 = \omega_1(c_1, \ldots, c_m)$$

$$e_2 = \omega_2(e_1, c_1, \ldots, c_m)$$

$$\vdots$$

$$e_k = \omega_k(e_1, \ldots, e_{k-1}, c_1, \ldots, c_m)$$

$$\vdots$$

$$e_{\rho} = \omega_{\rho}(e_1, \ldots, e_{\rho-1}, c_1, \ldots, c_m)$$

Here, by $e_k$ we denote geometric data. If $\mathcal{C} = \{c_1, \ldots, c_m\}$ is the set of constraint parameters, $e_k$ is either a point, a radius, or an angle, constructible with respect to $e_1, \ldots, e_{k-1}$ and $\mathcal{C}$.

Note that $e_1$ and $e_2$ are always the two first constructible points with $e_1 = P_1 = (0, 0)$, and $e_2 = P_2 = (d, 0)$, with $d$ a known distance (a given constraint parameter or the unit distance), [18]; and the geometric element $e_{\rho}$ is always the last constructible point of the object.
3 Formulation of the geometric construction problem with interval parameters

Let $\mathcal{X} = \{P_1, \ldots, P_n\}$ be the set of characteristic points of the object under design. Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be the set of parameters of the given geometric constraints that correctly define the ruler-and-compass constructible object. Let $\Omega(\mathcal{X}, \mathcal{C})$ then denote a sequence of basic construction steps $\{\omega_1, \ldots, \omega_p\}$. Each construction step $\omega_i$ computes geometric data of the type point - the coordinates of a point in $\mathcal{X}$, distance or angle, [8].

Let $\mathcal{D}_i$ denote the set of all possible values that can be assigned to the variable $c_i \in \mathcal{C}$. Since geometric constraints can only be distances and angles, we have that $\mathcal{D}_i = [0, \infty]$ when $c_i$ is a distance constraint, and $\mathcal{D}_i = [0, 2\pi]$ when $c_i$ is an angle.

Let $\mathcal{V} = \{v_1, \ldots, v_m\}$ be the set of intervals assigned to the constraint parameters of the geometric construction problem.

An interval is a closed and bounded subset of the real numbers

$$v := \overline{[v, \bar{v}]} := \{v \in \mathbb{R} \ | \ v \leq v \leq \bar{v}\}$$

where $v$ and $\bar{v}$ denote the lower and the upper bounds of the interval $v$, respectively. By an assignment

$$c_i := v_i \quad \text{with} \quad v_i \subseteq \mathcal{D}_i, \quad \forall i = \{1, \ldots, m\}$$

we mean that $c_i$ may take any value in the interval $v_i$, $\forall i = \{1, \ldots, m\}$.

We can formally define the geometric construction problem with interval parameters as follows.

**Definition 3.1** Given a sequence of basic construction steps $\Omega(\mathcal{X}, \mathcal{C})$ for an object $\mathcal{X}$, and given an assignment of interval values $\mathcal{V}$ to the constraint parameters $\mathcal{C}$, the geometric construction problem consists of computing the intervals of values for the coordinates of the points in $\mathcal{X}$ for which $\Omega(\mathcal{X}, \mathcal{V})$ is realizable.

4 Computing the range of the coordinates of the points

Computing the intervals of values for the coordinates of the characteristic points means computing the range of the coordinate functions over the interval input data. However, the sequence of basic construction steps $\Omega(\mathcal{X}, \mathcal{C})$ does not provide us explicit functions for the coordinates. Instead, the coordinates are implicitly expressed as the intersection of two constructible objects such as rays, lines, or circles.

In order to get the explicit coordinate functions, we associate to each basic geometric operation $\omega_i$ a set of functions. If $\omega_i$ constructs a point $P$, then we get two geometric functions: one for the $x$, and one for the $y$, coordinates of the point. If $\omega_i$ calculates a distance or an angle from already constructed points, then we get only one function, which is exactly the geometric operation $\omega_i$.

Using this approach, the sequence of basic construction steps

$$\Omega(\mathcal{X}, \mathcal{C}) = \{\omega_1, \ldots, \omega_p\}$$

8
<table>
<thead>
<tr>
<th>Basic Geometric Operation</th>
<th>No.</th>
<th>code</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1 = \text{point}(0,0)$</td>
<td>1</td>
<td>10</td>
<td>zero</td>
<td>zero</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_2 = \text{point}(d_1,0)$</td>
<td>2</td>
<td>10</td>
<td>$d_1$</td>
<td>zero</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$a_1 = \text{direction}(P_1, P_2)$</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$a_2 = \text{adj}(a_1, a_1)$</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>$a_1$</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_3 = \text{rc}(\text{ray}(P_2, a_2), \text{circle}(P_2, d_2), 1)$</td>
<td>5</td>
<td>22</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>$d_2$</td>
<td>plus 1</td>
</tr>
<tr>
<td>$a_3 = \text{direction}(P_3, P_3)$</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$a_4 = \text{asum}(a_3, \pi/2)$</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>$\pi/2$</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$Q_1 = \text{lc}(\text{line}(P_2, a_4), \text{circle}(P_2, d_3), 1)$</td>
<td>8</td>
<td>20</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>$d_3$</td>
<td>plus 1</td>
</tr>
<tr>
<td>$P_4 = \text{lc}(\text{line}(Q_1, a_3), \text{circle}(P_3, d_3), 0)$</td>
<td>9</td>
<td>20</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>$d_3$</td>
<td>zero</td>
</tr>
<tr>
<td>$Q_2 = \text{midpoint}(P_1, P_1)$</td>
<td>10</td>
<td>12</td>
<td>1</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$r_1 = \text{distance}(P_1, Q_1)$</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_5 = \text{cc}(\text{circle}(P_4, d_3), \text{circle}(Q_2, r_1), -1)$</td>
<td>12</td>
<td>24</td>
<td>9</td>
<td>$a_3$</td>
<td>10</td>
<td>11</td>
<td>minus 1</td>
</tr>
</tbody>
</table>

Table 1: Code list of geometric operations for the construction plan in Section 2.2.

is transformed into a sequence of functions

$$\mathcal{F} = \{f_1(c_1, \ldots, c_m), \ldots, f_\xi(c_1, \ldots, c_m)\},$$

where $f_k : \mathbb{R}^m \rightarrow \mathbb{R}$, with $k = \{1, \ldots, \xi\}$, determines either a coordinate of a point in $\mathcal{X}$ or a dimensional parameter (distance or angle).

Note that for an object $\mathcal{X} = \{P_1, \ldots, P_n\}$, the number of basic construction steps is $\rho \geq n$, and the number of basic geometric functions is $\xi = \rho + n$ (in the plane, we simply need to consider as many additional functions to $\rho$ as the number of characteristic points).

Once $\Omega(\mathcal{X}, \mathcal{C})$ has been transformed into the corresponding sequence of geometric functions $\mathcal{F}$, then we are able to compute the range of each function $f_k$, with $k = \{1, \ldots, \xi\}$, for an assignment of interval parameters $\mathcal{V}$.

### 4.1 Transforming $\Omega(\mathcal{X}, \mathcal{C})$ into $\mathcal{F}$

Given a sequence of basic construction steps $\Omega(\mathcal{X}, \mathcal{C})$, the data structure we have used to represent this sequence is a code list of geometric operations based on the codification described in [8]. With each geometric operation $\omega_i$, we associate an operation number (the index $i$), an operation code, and an ordered list of 5 parameters $a_1, \ldots, a_5$.

A basic construction step involves either two geometric elements (points, rays, lines, or circles), or two dimensional parameters (two distances or two angles). In the first situation, the parameter $a_1$ is the operation number that constructs the coordinates of the characteristic point of the first geometric element; the parameter $a_2$, if exists, is the dimensional parameter for the first geometric element. In an analogous way, $a_3$ and $a_4$ are the characteristic point and the dimensional parameter for the second geometric element. The parameter $a_5$, when needed, is the sign for the selection criteria we have defined in Section 2.2. In the second situation, $a_2$ and $a_4$ are simply the first and the second dimensional parameters.

We can then find three types of parameters in the code list of geometric operations:

1. an element in the set of constraint parameters $\mathcal{C} = \{c_1, \ldots, c_m\}$; or
<table>
<thead>
<tr>
<th>Basic Geometric Operation</th>
<th>No.</th>
<th>code</th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
<th>$v_6$</th>
<th>$v_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1 = \text{point}(0,0)$</td>
<td>1</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>zero</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>11</td>
<td>-</td>
<td>-</td>
<td>zero</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_2 = \text{point}(d_1,0)$</td>
<td>3</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>$d_1$</td>
<td>-</td>
<td>zero</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>11</td>
<td>-</td>
<td>-</td>
<td>$d_1$</td>
<td>-</td>
<td>zero</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_1 = \text{direction}(P_1, P_2)$</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>3</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_2 = \text{adj}(\alpha_1, a_1)$</td>
<td>6</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>$a_1$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_3 = \text{ray}(P_2, \alpha_2)$</td>
<td>7</td>
<td>22</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>$d_2$</td>
<td>plus 1</td>
</tr>
<tr>
<td>$\text{circle}(P_3, d_2, 1)$</td>
<td>8</td>
<td>23</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>$d_2$</td>
<td>plus 1</td>
</tr>
<tr>
<td>$\alpha_3 = \text{direction}(P_2, P_3)$</td>
<td>9</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>-</td>
<td>7</td>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_4 = \text{asum}(\alpha_3, \pi/2)$</td>
<td>10</td>
<td>3</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>$\pi/2$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$Q_1 = \text{line}(P_2, \alpha_4)$</td>
<td>11</td>
<td>20</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>4</td>
<td>$d_3$</td>
<td>plus 1</td>
</tr>
<tr>
<td>$\text{circle}(P_3, d_3, 1)$</td>
<td>12</td>
<td>21</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>4</td>
<td>$d_3$</td>
<td>plus 1</td>
</tr>
<tr>
<td>$P_4 = \text{line}(Q_1, \alpha_3)$</td>
<td>13</td>
<td>20</td>
<td>11</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>$d_3$</td>
<td>zero</td>
</tr>
<tr>
<td>$\text{circle}(P_3, d_3, 0)$</td>
<td>14</td>
<td>21</td>
<td>11</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>$d_3$</td>
<td>zero</td>
</tr>
<tr>
<td>$Q_2 = \text{midpoint}(P_1, P_4)$</td>
<td>15</td>
<td>12</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>13</td>
<td>14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>13</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>13</td>
<td>14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$r_1 = \text{distance}(P_1, Q_2)$</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>15</td>
<td>16</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_5 = \text{cc}(\text{circle}(P_4, d_3),$</td>
<td>18</td>
<td>24</td>
<td>13</td>
<td>14</td>
<td>$d_3$</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>minus 1</td>
</tr>
<tr>
<td>$\text{circle}(Q_2, r_1), -1)$</td>
<td>19</td>
<td>25</td>
<td>13</td>
<td>14</td>
<td>$d_3$</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>minus 1</td>
</tr>
</tbody>
</table>

Table 2: Corresponding code list of geometric functions for the construction plan in Section 2.2.

2. a constant value; or

3. an index $j$ referring to a previously evaluated geometric operation $\omega_j$, with $j < i$.

An example of a code list for the construction plan of Section 2.2 is given in Table 1. For instance, for the point $P_4$, the code we have assigned to the intersection operation $\text{lc}$ in [8] is 20. The parameters of the line are the point $Q_1$ (constructed in operation 8), and the angle $\alpha_3$ (constructed in operation 6); the parameters of the circle are the centerpoint $P_3$ (operation 5), and the constraint parameter $d_3$. The geometric constraint solving system has assigned the sign factor 0 to this intersection operation, meaning that the line and the circle have to be necessarily tangent.

The task of transforming $\Omega(X, C)$ into $\mathcal{F} = \{f_1, \ldots, f_\xi\}$ then consists of translating the code list of geometric operations into a new code list of geometric functions.

If the geometric operation $\omega_i$ constructs a point, we add two new functions in the code list: the function $f_k$ that computes the $x$ coordinate of the point, and the function $f_{k+1}$ for the $y$ coordinate. If $\omega_i$ constructs a dimensional parameter, the geometric operation $\omega_i$ is exactly the new function $f_k$ in the list.

We associate to each function $f_k$, $k \in \{1, \ldots, \xi\}$ a function number (the index $k$), a function code, and an ordered list of 7 parameters $v_1, \ldots, v_7$. These parameters are obtained from $a_1, \ldots, a_5$ in the following way.

The parameters $v_1$ and $v_2$ are respectively the function numbers for the $x$ and $y$ coordinates of the point in the operation number $a_1$. The parameter $v_3$ is the function number of the dimensional parameter $a_2$. The parameters $v_4, v_5$ and $v_6$ are obtained in an analogous way.
from $a_3$ and $a_4$. Finally, $v_7$ is $a_5$. Note that the $v_k$’s always refer to scalar quantities, while the $a_i$’s can also refer to 2-dimensional data such as points.

The parameters in the code list of geometric functions are of the type:

1. an element in the set of constraint parameter $C = \{c_1, \ldots, c_m\}$; or
2. constant value; or
3. index $j$ referring to a previously evaluated geometric function $f_j$, with $j < k$.

In Table 2, we show an example of code list for the sequence of basic geometric functions of the construction plan in Section 2.2. We consider, for instance, the functions for the coordinates of the point $P_3$. The code assigned to the $x$ coordinates in the intersection of a line and a circle in [8] is 20, and the code for the $y$ coordinate is 21. The parameters of both functions are the following: the $x$ coordinate of point $Q_1$ (calculated in function number 11), the $y$ coordinate of point $Q_1$ (calculated in function 12), the angle $a_3$ (calculated in 9), the $x$ coordinate of point $P_3$ (function 7), the $y$ coordinate of $P_3$ (function 8), the constraint parameter $d_3$, and the constant sign factor 0.

Once we know how to transform the sequence of geometric steps $\Omega(\mathcal{X}, \mathcal{C})$ into the corresponding sequence of geometric functions $\mathcal{F}$, we have each function $f_k$ defined in terms of $c_1, \ldots, c_m$ and $f_1, \ldots, f_{k-1}$. By recursively evaluating the functions $f_1, \ldots, f_{k-1}$, the function $f_k$ depends only on the input parameters $C = \{c_1, \ldots, c_m\}$. Now, we want to compute the range of each geometric function $f_k$, with $k \in \{1, \ldots, \xi\}$, for an assignment of interval values to the constraint parameters.

4.2 The need of computing tight enclosures for the range of the coordinates

Ranges of functions are usually evaluated using interval analysis, [1, 14]. The problem is that the interval extension of a real function can only give bounds on the range of the function, an enclosure for the range, and these bounds frequently overestimate the exact range. In the geometric construction problem, overestimated enclosures lead to unfeasible solutions for the coordinates of the points, and this is, of course, not desirable in our application.

Furthermore, in our problem, overestimation can lead to an even more serious problem: using interval arithmetic, it may be impossible to compute an enclosure for the range of a function, even when the range actually exists. Let us explain how this can happen.

In the construction plan $\Omega(\mathcal{X}, \mathcal{C})$, a new point $P_1$ of the object is constructed from already constructed points in $\mathcal{X}_{i-1}$. Analogously, in the sequence of geometric functions $\mathcal{F}$, already computed functions $f_1, \ldots, f_{k-1}$ are the input parameters for the functions $f_k, \ldots, f_1$.

When the enclosures for the functions $f_1, \ldots, f_{k-1}$ are overestimated, the interval input parameters for the function $f_i$, with $i \in \{k, \ldots, \xi\}$, may not be a subset in the domain $f_i$. In this situation, the range of the function cannot be computed.

For instance, in the example of Figure 1, we consider the following assignment of values to the constraint parameters $d_1, d_2, d_3$ and $a_1$:

$$d_1 = [17.5, 18.5], \quad d_2 = [19.5, 20.5], \quad d_3 = 5, \text{ and } a_1 = 4.5$$
<table>
<thead>
<tr>
<th>Point coordinate</th>
<th>Enclosure</th>
<th>Computed enclosures</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1.x$</td>
<td>$ia, range$</td>
<td>[0.00000000000, 0.00000000000]</td>
</tr>
<tr>
<td>$P_1.y$</td>
<td>$ia, range$</td>
<td>[0.00000000000, 0.00000000000]</td>
</tr>
<tr>
<td>$P_2.x$</td>
<td>$ia, range$</td>
<td>[17.50000000000, 18.50000000000]</td>
</tr>
<tr>
<td>$P_2.y$</td>
<td>$ia, range$</td>
<td>[0.00000000000, 0.00000000000]</td>
</tr>
<tr>
<td>$P_3.x$</td>
<td>$ia$</td>
<td>[13.1342512426, 14.4390848782]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[13.1786861116, 14.3894819112]</td>
</tr>
<tr>
<td>$P_3.y$</td>
<td>$ia$</td>
<td>[18.83: 8118552, 20.2454266548]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[19.0618372944, 20.0393674122]</td>
</tr>
<tr>
<td>$Q_1.x$</td>
<td>$ia$</td>
<td>[11.5750421394, 14.6587421121]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[12.6123494116, 13.6123494117]</td>
</tr>
<tr>
<td>$Q_1.y$</td>
<td>$ia$</td>
<td>[-1.68441562194, -0.584991017315]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[-1.05397899716, -1.05397899714]</td>
</tr>
<tr>
<td>$P_4.x$</td>
<td>$ia$</td>
<td>[5.35314979317, 12.0533309063]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[8.29103552333, 9.50183132280]</td>
</tr>
<tr>
<td>$P_4.y$</td>
<td>$ia$</td>
<td>[15.4236356975, 21.3006110186]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[18.0078582973, 18.9853844150]</td>
</tr>
<tr>
<td>$Q_2.x$</td>
<td>$ia$</td>
<td>[2.67657489658, 6.02666545312]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[4.14551776166, 4.75091566140]</td>
</tr>
<tr>
<td>$Q_2.y$</td>
<td>$ia$</td>
<td>[7.71181784877, 10.6503055093]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[9.00392914865, 9.49269420750]</td>
</tr>
<tr>
<td>$P_5.x$</td>
<td>$ia$</td>
<td>[3.36141910047, 4.64208611010]</td>
</tr>
<tr>
<td></td>
<td>$range$</td>
<td>[18.9388804473, 20.0584618519]</td>
</tr>
</tbody>
</table>

Table 3: Enclosures for the coordinates of the characteristic points of $X$

First, we transform the construction plan in Section 2.3 into a sequence of geometric functions. Second, we evaluate these functions using interval arithmetic. The results we get are summarized in Table 3. There, $ia$ denotes the interval enclosures of the range with interval arithmetic, and $range$ denotes the exact range (computed via elementary trigonometry in the example). The points $P_1$ and $P_2$ are the two first constructible points, and the enclosures for their coordinates are exactly the range. For the points $P_3, Q_1, P_4$ and $Q_2$, we get enclosures for the coordinates with growing overestimation. When we try to compute the coordinates of the point $P_5$ from these previously computed enclosures, intermediate interval parameters are partially out of the domain of the coordinate functions. However, for the given assignment of values to the input parameters $d_1, d_2, d_3$ and $a_1$, the range of the coordinates of the point $P_5$ can definitely be computed.

4.3 Computing tight enclosures for the basic geometric functions

A method for computing tight enclosures for the range of a sequence of geometric functions $\mathcal{F}(\mathcal{C})$ over an interval vector $V = (v_1, \ldots, v_m)$ is the interval method presented in [9].

Computing the range of a function $f \in \mathcal{F}$ restricted to the interval vector $V$ consists of computing the set

$$f(V) = \{f(V) \mid V \in V\} = [y_{min}, y_{max}]$$

12
with \( y_{\text{min}} := \min \{ f(V) \mid V \in \mathcal{V} \} \), and \( y_{\text{max}} := \max \{ f(V) \mid V \in \mathcal{V} \} \).

The problem of computing \( f(V) \) can therefore be seen as the problem of computing the minimum and the maximum values of \( f \) in the interval \( V \), \([2, 15]\). First, one computes an enclosure for the minimum value of the function in the given interval \( V \) (the lower endpoint of the range)

\[
y_{\text{min}} = [\underline{y}_{\text{min}}, \overline{y}_{\text{min}}] \quad \text{so that} \quad y_{\text{min}} \in y_{\text{min}}
\]

Second, one computes an enclosure for the maximum value of the function in the interval \( V \) (the upper endpoint of the range)

\[
y_{\text{max}} = [\underline{y}_{\text{max}}, \overline{y}_{\text{max}}] \quad \text{so that} \quad y_{\text{max}} \in y_{\text{max}}
\]

Finally, one encloses the range \( f(V) \) in the interval \( y = [\underline{y}_{\text{min}}, \overline{y}_{\text{max}}] \).

The algorithm ConstructPlan developed in \([9]\) computes guaranteed bounds for the endpoints of the range with arbitrarily small overestimation \( \varepsilon_X \), to be chosen by the user. It combines two different approaches: a gradient-based method, and an optimization method. First, we use the information contained in the gradient to compute enclosures for the endpoints of the range. If these enclosures are not tight, we apply an optimization method in order to get the enclosures at a desired accuracy.

5 Results

In Table 4, we show the numerical results obtained for the example of Section 4.2. We consider the assignment

\[
d_1 = [17.5, 18.5], \ d_2 = [19.5, 20.5], \ d_3 = 5, \ \text{and} \ a_1 = 4.5
\]

to the constraint parameters of Figure 1. For each coordinate of the points in \( \mathcal{X} \), we compute

1. the enclosure for the range using interval arithmetic (denoted by \( ia \) in Table 4);

2. the enclosure for the range using the interval propagation algorithm ConstructPlan from \([9]\) at a desired accuracy \( \varepsilon_X = 10^{-1} \) (\( ip \) in the table); and

3. the exact range (\( \text{range} \)).

As we saw in Section 4.2, for this assignment of values to the constraint parameters, the interval arithmetic is not able to compute an enclosure for the coordinates of the point \( P_5 \). Using the algorithm ConstructPlan, however, we can compute the enclosures for the range of the coordinates, at least, at the desired accuracy \( \varepsilon_X \). We want to emphasize that, for most of the coordinates, the precision we get is definitely much better than \( \varepsilon_X \).

6 Conclusions

In this paper we have described how to deal with constructible geometric problems with interval parameters.
<table>
<thead>
<tr>
<th>Point coordinate</th>
<th>Enclosure</th>
<th>Computed enclosures</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[0.00000000000, 0.00000000000]$</td>
</tr>
<tr>
<td>$P_1.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[0.00000000000, 0.00000000000]$</td>
</tr>
<tr>
<td>$P_2.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[17.50000000000, 18.50000000000]$</td>
</tr>
<tr>
<td>$P_2.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[0.00000000000, 0.00000000000]$</td>
</tr>
<tr>
<td>$P_3.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[18.318118652, 20.2454266548]$</td>
</tr>
<tr>
<td>$P_3.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[19.0618372944, 20.0393674122]$</td>
</tr>
<tr>
<td>$Q_1.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[11.579041394, 14.658742112]$</td>
</tr>
<tr>
<td>$Q_1.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[12.6015448717, 13.6243183772]$</td>
</tr>
<tr>
<td>$P_4.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[5.35314979317, 12.0533300963]$</td>
</tr>
<tr>
<td>$P_4.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[8.2910352329, 9.50183132283]$</td>
</tr>
<tr>
<td>$Q_2.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[2.67657509658, 6.0266545312]$</td>
</tr>
<tr>
<td>$Q_2.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[4.14551776164, 4.75091566142]$</td>
</tr>
<tr>
<td>$P_5.x$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[7.71181784877, 10.650355093]$</td>
</tr>
<tr>
<td>$P_5.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[9.00392914862, 9.49269420753]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$[9.00392914865, 9.49269420750]$</td>
</tr>
<tr>
<td>$P_5.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[3.30724318495, 4.69826351007]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$[3.36149104047, 4.64208611010]$</td>
</tr>
<tr>
<td>$P_5.y$</td>
<td>$ia$, $ip$, $range$</td>
<td>$[18.8957016297, 20.0995401782]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$[18.9388804473, 20.0584618519]$</td>
</tr>
</tbody>
</table>

Table 4: Enclosures for the range of the coordinates of the points in $X$ for the input parameters: $d_1 = [17.5, 18.5]$, $d_2 = [19.5, 20.5]$, $d_3 = 5$, and $a_1 = 4.5$.  

14
Given a construction plan for a geometric object, and an assignment of interval values to the constraint parameters, solving the geometric construction problem consists of computing the range of the coordinates of the characteristic points that define the object.

Computing the range of the coordinates requires solving two problems: getting the functions of the coordinates from the sequence of basic construction steps, and computing tight enclosures for the range of these functions.

In this work, we have precisely defined the geometric construction problem, we have characterized the construction plans we solve, and we have given the data structure to represent construction plans in this form.

Given a construction plan and an assignment of interval values to the constraint parameters, we compute the range of the coordinates using the algorithm ConstructPlan, [9]. First, the algorithm transforms the sequence of basic geometric operations into the corresponding sequence of basic geometric functions. Second, it computes a tight enclosure for the range of each geometric function. The results obtained show how good enclosures for the range can be computed using the algorithm ConstructPlan.

Acknowledgements

This research has been supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT), under the project TIC98-0586-C03-03. N. Mata is also partially supported by a PhD. fellowship of the Associació d’Enginyers Industrials de Catalunya.

References


