Solving Incidence and Tangency Constraints in 2D

Núria Mata
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Av. Diagonal 647, 8^a, 08028 Barcelona
e-mail: nmata@turing.upc.es

January 21, 1997

Abstract

This paper reports on solving geometric constraint satisfaction problems involving incidence and tangency constraints in 2D. A variational geometric constraint solver based on a constructive approach is used: the main goal is to keep the present set of rules as small as possible. Defining tangency conditions as distance and angle constraints allows solving fixed radius configurations. Non-fixed radius schemes are also characterized and a new set of constructive rules is proposed.

1 Introduction

In Geometric Constraint Satisfaction Problems (GCSP), modelling tasks are performed by giving dimensions and geometric especifications, or *constraints*, to rough sketches.

Several approaches to the geometric constraint satisfaction have been reported in the literature. Most constraint solvers translate constraint relations into a system of equations that are solved using iterative techniques, in *numerical constraint solvers*, or symbolic algebraic methods, in *symbolic constraint solvers*. Other approaches are based on *constraint propagation* on graphs representing systems of constraint equations. In *constructive solvers*, constraints are satisfied by placing subsets of well-constrained geometric elements in a finite number of construction steps. They are based on the fact that most configurations in an engineering drawing are solvable using a rather small set of tools like ruler, compass and protractor. Constructive solvers can be based either on rewriting rules [2, 13] or on constraint graph analysis [1, 12, 10].

Our group has built a variational geometric constraint solver based on a constructive approach [7]. It computes a solution in two phases. First, using rewriting rules, the solver builds a sequence of construction steps without the need of arranging the set of constraints in a predefined order. Then, the construction steps are carried out to generate an instance of the geometric object for the current dimension values.

Sketches are composed from geometric elements and constraints. At present, points and segments are the only geometric elements allowed in the solver. Constraints that can be defined on these objects include distance between two points, perpendicular distance between a point and a segment, and angle between two segments. The possible geometric elements that can be defined using tangency constraints are circles, arcs and segments. This paper describes how to extend the present sets of geometries and constraints, by including arcs, circles, incidence and 2D tangency constraints, while keeping the set of rules as small as possible.

In the next section, the constructive geometric constraint solver used in the present work is introduced. In section 3, we will discuss about constraints involving arcs and circles. In section 4, incidence and tangency constraints in 2D will be characterized: fixed and variable radii configurations will be introduced and a new set of constructive rules will be proposed. Finally, in section 5, our conclusions will focus on the convenience of extending the present set of rules in the geometric constraint solver.

2 The Constructive Geometric Constraint Solver

The geometric constraint solving system has two major components, the *analyser* and the *constructor*.

The analyser deals with the problem of determining symbolically whether or not a geometric sketch is solvable. It is based on a constructive technique which exhibits properties of both rule and graph constructive approaches. The solver is fed with a topologically correct sketch properly annotated with constraints. Then, if the set of constraints consistently defines the object, the analyser generates a sequence of constructive steps that determine each geometric element such that the constraints are satisfied.

The constructor responds to the problem of building an instance of the geometric object. The instantiation is carried out by applying the sequence of construction steps generated by the analyser to the actual parameters values. Whenever no numerical incompatibilities arise in the computation, an instance of the geometric object is generated.

The solver considers only well-constrained, two-dimensional sketches.

In the next sections, we discuss some issues concerning the data representation and the rules used.

2.1 Data representation in GCSP

A GCSP can be modelled using a *constraint graph* to represent both, geometric components and constraints defined on these elements. Each node represents the degrees of freedom of the underlying geometry in the graph. Edges constraint the possible movements of these geometries: the more edges in a graph, the more likely to be rigid.

A constraint graph is a simple, undirected and finite graph consisting of nodes representing geometries and pairwise edges corresponding to the equations between each two constraint geometries [12]. In GCSP, the primary interest is not in graphs, but rather in their concrete realizations in some Euclidean space. The *graph realization problem* is that of computing the relative locations of a set of vertices placed in Euclidean space, relying only upon some set of inter-vertex constraints.

By Laman's theorem [11], the relative positions of n given points are totally determined by $2n-3$ independent relations defined between them. In 2D Euclidean space, independent relations mean, using a graph approach, that a graph G with n geometric elements and $2n-3$ constraints is rigid if and only if no subgraph G' has more than $2n'-3$ edges, $n' \leq n$.

2.2 Rules

Rules are applied on subsets of points and constraints. These subsets are known as *constraint sets*, **CX** sets in short, and they are classified in CA, CD and CH sets.

An angle constraint set, **CA** set, is a pair of oriented segments which are mutually constrained by an angle.

A distance constraint set, **CD** set, is a set of points with mutually constrained distances.

A **CH** set is a point and a segment constrained by a perpendicular distance from the point to the segment.

A sketch is solved when all the points belong to the same CD set. Depending on the functionality of the rules, the following types are considered: creation rules, merging rules or construction rules.

- **Creation rules**
  Creation rules generate elementary **CD**, **CA** and **CH** sets as an interpretation of the dimensioning scheme defined by the user.
- **Merging rules**
  Merging rules allow to compute operations between constrained sets.
- **Construction rules**
  Models are built incrementally using locally solvable geometric constructions. **CD**, **CA** and **CH** sets are combined into larger **CD** sets if they pairwise share a single geometric element. Merging **CX** sets require rigid body motions.

All these rules are explained with detail in [8].

3 Constraints involving arcs and circles

2D tangency constraints are defined on arcs, circles and segments. Considering that arcs are partial visualizations of circles between the two arc endpoints, only the two following basic configurations need to be defined:

- 2D tangency constraints between a segment and a circle
- 2D tangency constraints between two circles

Besides tangency constraints, incidence on circles and constraints on the radius may also exist.

GCSP are solved when all points are positioned in a **CD** set. By Laman's theorem, $2n-3$ well-distributed constraints are enough to determine the relative position of n

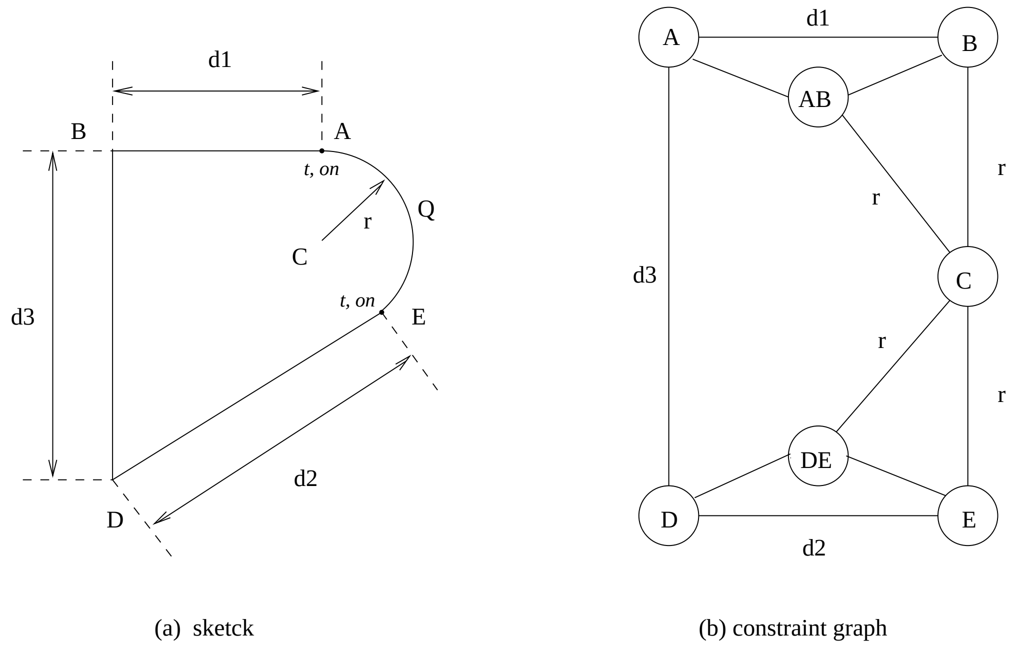


Figure 1: Solver representation of 2D tangency constraints in a fixed radii configuration.

given points. Using a kinematical analysis approach, the total number of allowed motions is the number of degrees of freedom, $2n$ (in two-dimensional space, a point has 2 traslational degrees of freedom), minus the rigid body motions (two traslational and one rotational rigid body motion on the plane).

When circles are included in the geometries set, a dimensionality problem arises. A point on a circle, $(x-c_x)^2+(y-c_y)^2 = r^2$, has 3 degrees of freedom: two degrees of freedom to position the center, $C = (c_x, c_y)$, and one more for the radius, r. Therefore, some care must be taken when determining the number of constraints that consistently define GCSP's when arcs and circles are involved. More general than Laman's theorem, although not sufficient, is Grübler's condition [3].

> The relative position of n given geometric elements in the two-dimensional Euclidean space is determined by $\sum_{i=1}^{n} d_i - 3$ well-distributed constraints, d_i the number of degrees of freedom of geometry i.

4 Characterization of incidence and tangency constraints in 2D

In this section, incidence and tangency constraints will be characterized in terms of constraint sets, **CD CA** and **CH** sets, considering both, fixed radii (section 4.1) and variable radii configurations (section 4.2).

4.1 Fixed radii configurations

In fixed radii configurations, points on a circle are determined once the position of the center point is known. The two remaining traslational degrees of freedom of the circle need to be constrained to consistently define a sketch.

With an appropiate representation and some preprocessing, we may restrict ourselves to points and segments, with pairwise distance and angle constraints. The circle placement problem is reduced to positioning the center point. Constraints on circles,

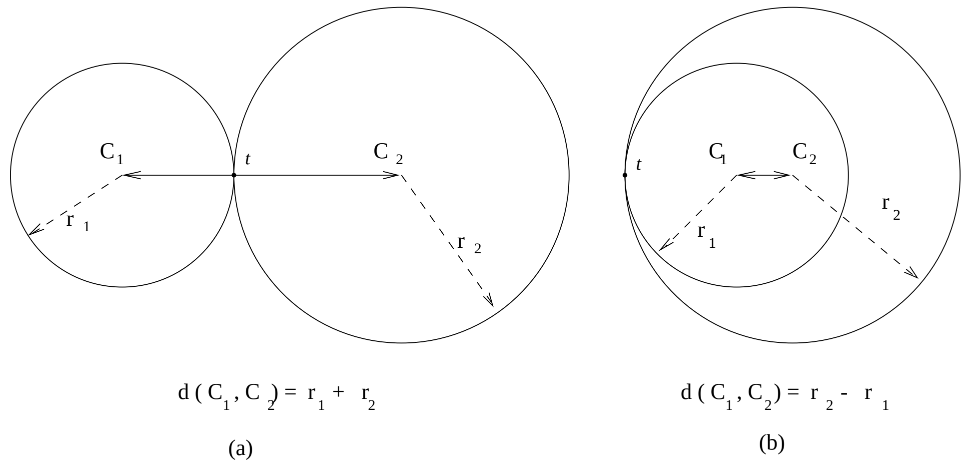


Figure 2: Solver representation of 2D tangency between to circles with known radius dimensions.

incidence and tangency constraints, are transformed into an equivalent representation in which only distance constraints appear. Let $Q(C, r)$ be the circle centered in C and given radius r. Three new creation rules are defined:

1. *point* P **on** *circle* $Q(C, r)$

   This constraint is translated into a distance constraint between the point P and the center C of circle $Q(C, r)$, $d(P, C) = r$. As shown in figure 1, A and E are points on[1] the circle Q and, therefore, incidence is translated into the following distance constraints: $d(C, A) = r$ and $d(C, B) = r$.

2. *segment* s **tangent to** *circle* $Q(C, r)$

   A tangency constraint between a segment s and a circle Q is expressed by a perpendicular distance constraint set between the center of the circle and the tangent segment. In the example shown in figure 1, Q is tangent[2] to segments $\overline{AB}$ and $\overline{DE}$. Tangency constraints are translated into $d(C, \overline{AB}) = r$ and $d(C, \overline{DE}) = r$. Since the distance from a point to a segment is the usual minimal distance, this constraint implicitly represent the usual information derived from tangency constraints between a segment and a circle: the radius and the segment are constrained by a right-angle through the tangency point.

3. *circle* $Q_1(C_1, r_1)$ **tangent to** *circle* $Q_2(C_2, r_2)$

   A tangency constraint between two circles, Q_1 and Q_2, is expressed by a distance constraint between the two center points, C_1 and C_2, equal to $\mid r_1 \pm r_2 \mid$. The two possible solutions are illustrated in figure 2.

4.2 Variable radii configurations

Circles can also be used even if values for the radii are not explicitly given. In this case, the solver not only has to determine the position of the center point, but also the value of the radius which satisfies the set of constraints on the circle. Circles with variable radius have three degrees of freedom. Therefore, at least three constraints on the circle need to be defined in order to cancel the two traslational degrees of freedom and a radial allowed motion.

Constraints on the circle may be referred either to the center point or to the circumference (incidence or 2D tangency constraints). Depending on whether the center point

[1] Incidence is represented by an *on* annotation in the sketch.
[2] In the sketch, tangency constraints are denoted by t.

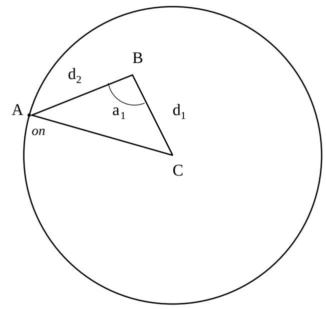


Figure 3: The radius of the circle is calculated as the distance between the center point, C, and point A on the circle.

can or cannot be positioned before the radius is determined, two diferent problems must be considered. If the center point can be positioned in a well-constrained subset of geometric elements, the additional constraint on the circle may allow to dimension the radius. Otherwise, the three constraints on the circle need to be simultaneously considered.

In section 4.2.1, the radius of the circle will be determined once the center point position is known. In section 4.2.2, both the center and the radius will be calculated from the set of constraints on the circle. Finally, in section 4.2.3, a new approach to solve variable radii configurations will be presented.

4.2.1 Computing the radius of the circle once the center point position is known

If the position of the center point can be determined independently from the radius of the circle, the radius dimension can be calculated using an incidence or a tangency constraint. This additional constraint on the circle will provide the value for the radius using one of the following rules:

1. *point* P **on** *circle* $Q(C, r)$

   If a point P is on a circle Q, and P and C belong to the same **CD** set, then the value for the radius r can be calculated by

   $$r = d(C, P)$$

   .

   In figure 3, points A, B and C are relatively positioned using two distance constraints (d_1 and d_2) and one angle constraint (a_1). The radius of the circle is determined by the distance between A and the center point, C, both belonging to the same **CD** set.

2. *segment* s **tangent to** *circle* $Q(C, r)$

   If a segment s and a circle $Q(C, r)$, s and C belonging to the same **CD** set, are related by a tangency constraint, then the value for the radius can be calculated as the perpendicular distance between the segment and the centre of the circle:

   $$r = d(C, s)$$

   In the example shown in figure 4, points A, B and C can be relatively positioned in a **CD** set. Then, the radius can be calculated as the distance between segment $\overline{AB}$ and the center point C. Once the radius is known, point D can be positioned in the sketch using the two remaining constraints: point D **on** Q and segment $\overline{BD}$ **tangent to** circle Q.

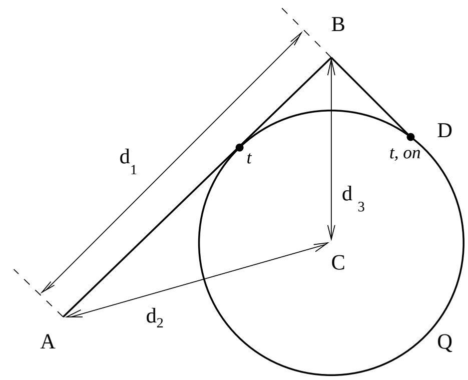


Figure 4: In this sketch, the value for the radii is calculated using a tangency constraint.

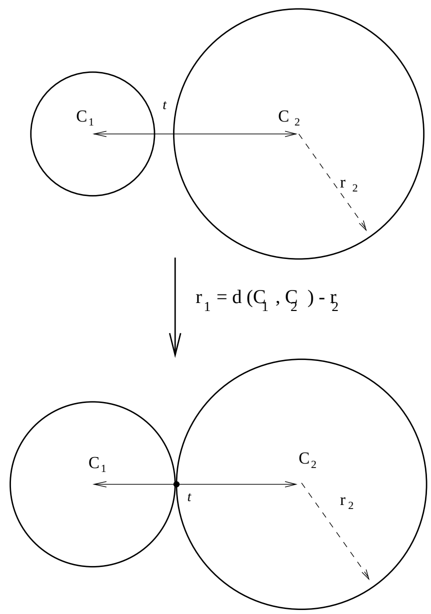


Figure 5: Computing the radius dimension r_1 from a tangency constraint between two circles.

3. *circle* Q_1 **tangent to** *circle* Q_2
   Given two circles $Q_1(C_1, r_1)$ and $Q_2(C_2, r_2)$ such that their center points, C_1 and C_2, belong to the same **CD** set, they will be well-constrained if and only if one of them has a radius dimension. If both circles have variable radii, the sketch is under-constrained.
   Considering r_1 to be the unknown radius, it can be calculated as follows:
   $$r_1 = \mid d(C_1, C_2) \mp r_2 \mid$$
   as illustrated in figure 5.

In our current approach, these three rules cannot be conceptually classified in any of the categories described in section 2.2. Computations on the radii are carried out and the dimension values obtained are later used as explicit constraints. In variable radii configurations, the relations derived from incidence and tangency constraints are symbolic, therefore, the valuation of the radii has to be performed before using creation rules to generate the elementary constraint sets involving the radius dimension. Since computing the value for the radii does not entail adding any geometric element to an existent **CD** set, this operation cannot be considered a construction rule, either. In fact, these three rules operate on elements belonging to the same **CD** set and not between constraint sets as it would be performed by merging rules.

At this stage, two different solutions have been considered: creating a new type of rules or using a geometric constraint solver based on a hybrid approach.

A new class of rules to deal with symbolic constraints -considering their functionality, we have named them *propagation rules*- can be defined. Some tangency and incidence constraints introduce symbolic dimensions that may be determined by computing geometric relations from a partially completed construction.

A solver based on a hybrid approach is a combination of several methods mentioned in section 1. A hybrid geometric constraint solver supporting symbolic constraints is described in [6]. This work reports on a technique to enhance constructive geometric constraint solvers with the capability of managing functional relationships between dimension variables. Essentially, it is a purely geometric constraint solver communicated bi-directionally with an equational solver.

In the next section, configurations using circles with variable center and radii will be analysed. Using the results obtained in sections 4.2.1 and 4.2.2, we will adopt the most suitable approach for variable radii configurations in section 4.2.3.

4.2.2 Computing the radius and the center point

In the previous sections, constraints on circles were transformed into an equivalent representation in which only distance between two points and perpendicular distance between a point and a segment appeared. In fixed radii configurations and in variable radii circles with known center position, the points of the circle have up to two degrees of freedom. Therefore, solving constraints on circles -expressed in terms of points, segments and distance constraints- can be done using the present set of rules defined on sets of points, distance, perpendicular distance and angle constraints.

In well-constrained sketches, when the radius dimension and the center point are unknown, all constraints on the circle must be expressed as a function of the radii. Incidence and tangency constraints must be translated into symbolic constraints using the relationships given in section 4.2.1:

- *point* P **on** *circle* $Q(C, r)$

$$d(C, P) = r$$

- *segment* s **tangent to** *circle* $Q(C, r)$

$$d(C, s) = r$$

- *circle* $Q_1(C_1, r_1)$ **tangent to** *circle* $Q_2(C_2, r_2)$

$$d(C_1, C_2) = \mid r_1 \pm r_2 \mid$$

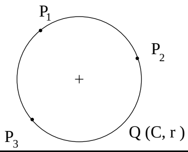
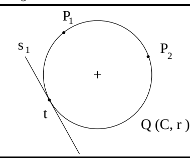
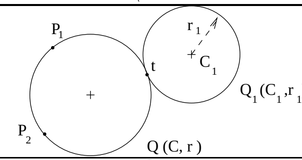
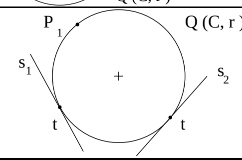
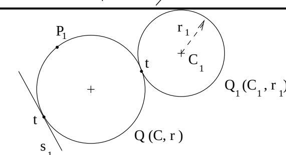
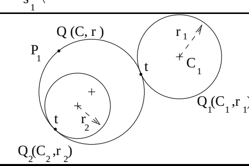
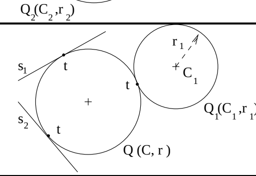
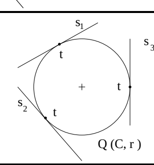
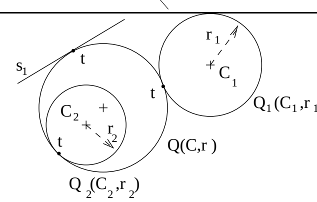
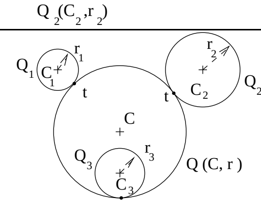
A combination of these three basic constraints gives ten different ways of defining circles using incidence and tangency constraints. The resulting configurations are summarized in table 1 and a more detailed information is provided below, as well as examples on each configuration.

1. **Circle defined by three incident points**

   Set of constraints:

   point P_1 **on** circle $Q(C, r)$ $\rightarrow$ $d(C, P_1) = r$
   point P_2 **on** circle $Q(C, r)$ $\rightarrow$ $d(C, P_2) = r$
   point P_3 **on** circle $Q(C, r)$ $\rightarrow$ $d(C, P_3) = r$

Table 1: Different ways to define a circle using incidence and tangency constraints.

Case	*points*	*segments*	*circles*	Configuration
1	P_1, P_2, P_3			
2	P_1, P_2	s_1		
3	P_1, P_2		Q_1	
4	P_1	s_1, s_2		
5	P_1	s_1	Q_1	
6	P_1		Q_1, Q_2	
7		s_1, s_2	Q_1	
8		s_1, s_2, s_3		
9		s_1	Q_1, Q_2	
10			Q_1, Q_2, Q_3	

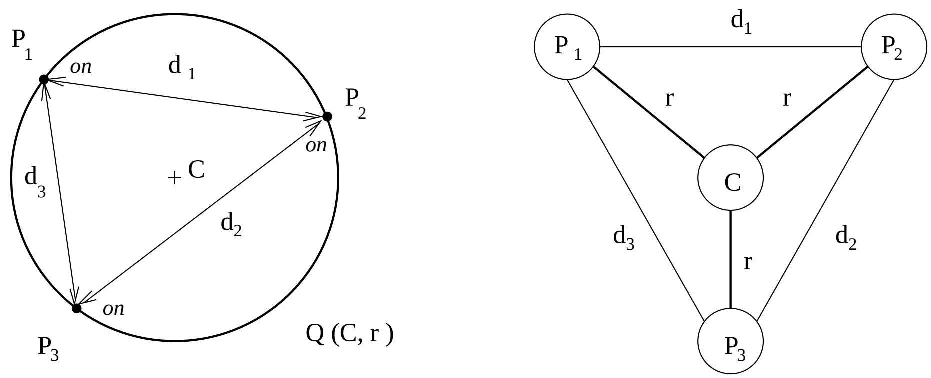


Figure 6: Circle defined by three points belonging to a **CD** set.

Three points, P_1, P_2 and P_3, relatively positioned on the plane (see figure 6), define a circle Q whose radius dimension and center point coordinates verify the following equations:

$$(P_{1x} - C_x)^2 + (P_{1y} - C_y)^2 = r^2$$
$$(P_{2x} - C_x)^2 + (P_{2y} - C_y)^2 = r^2$$
$$(P_{3x} - C_x)^2 + (P_{3y} - C_y)^2 = r^2$$

2. **Circle defined by two incident points and a tangency constraint between the circle and a segment**

   Set of constraints:

point P_1 **on** circle $Q(C, r)$	$\rightarrow$	$d(C, P_1) = r$
point P_2 **on** circle $Q(C, r)$	$\rightarrow$	$d(C, P_2) = r$
segment s_1 **tangent to** circle $Q(C, r)$	$\rightarrow$	$d(C, s_1) = r$

   As illustrated in figure 7, point P_2 is incident to the circle and segment s_{13} is tangent to Q in P_1. Considering segments expressed in a normal form,

   $$a \cdot x + b \cdot y + c = 0 \qquad a^2 + b^2 = 1$$

   the system of equations that has to be solved is:

   $$(P_{1x} - C_x)^2 + (P_{1y} - C_y)^2 = r^2$$
   $$(P_{2x} - C_x)^2 + (P_{2y} - C_y)^2 = r^2$$
   $$a_{13} \cdot C_x + b_{13} \cdot C_y + c_{13} = r$$

3. **Circle defined by two incident points and a tangency constraint with a circle**

   Set of constraints:

point P_1 **on** circle $Q(C, r)$	$\rightarrow$	$d(C, P_1) = r$
point P_2 **on** circle $Q(C, r)$	$\rightarrow$	$d(C, P_2) = r$
circle $Q_1(C_1, r_1)$ **tangent to** circle $Q(C, r)$	$\rightarrow$	$d(C, C_1) = \mid r \pm r_1 \mid$

   In figure 8, points P_1, P_2 and C_1 are relatively positioned by three distance constraints (d_1, d_2 and d_3). As circle Q_1 has an explicit value for the radius, points P_1, P_2 and circle Q_1 belong to a unique **CD** set. The position of the center point C and the value of the radii r can be calculated as follows:

   $$(P_{1x} - C_x)^2 + (P_{1y} - C_y)^2 = r^2$$
   $$(P_{2x} - C_x)^2 + (P_{2y} - C_y)^2 = r^2$$
   $$(C_{1x} - C_x)^2 + (C_{1y} - C_y)^2 = (r \pm r_1)^2$$

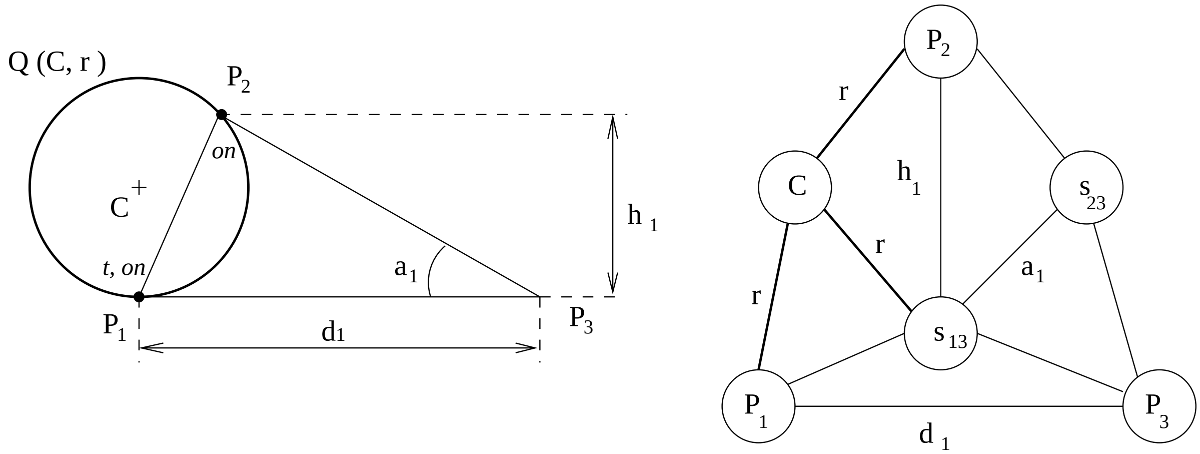


Figure 7: Circle defined by two incident points, P_1 and P_2, and a tangency constraint between the circle and segment s_{13}.

Besides diferent concrete realizations caused by reflexions [4], each tangency constraint between circles provide two diferent solutions (see figure 2). Depending on the relative position of the centers, the given circle can be contained within the solution circle or lie outside.

4. **Circle defined by an incident point and tangency constraints between the circle and two diferent segments**

   Set of constraints:

point P_1 **on** circle $Q(C,r)$	$\rightarrow$	$d(C,P_1)=r$
segment s_1 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_1)=r$
segment s_2 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_2)=r$

   As illustrated in figure 9, P_1, P_2 and P_3 are relatively positioned using distance and angle constraints. The circle tangent to segment s_1 in P_1 and tangent to segment s_2 in a priori unknown point is computed when the following system of equations has been solved:

$$
\begin{aligned}
(P_{1x}-C_x)^2+(P_{1y}-C_y)^2 &= r^2 \\
a_1\cdot C_x+b_1\cdot C_y+c_1 &= r \\
a_2\cdot C_x+b_2\cdot C_y+c_2 &= r
\end{aligned}
$$

5. **Circle defined by an incident point and two tangency constraints with a segment and a circle**

   Set of constraints:

point P_1 **on** circle $Q(C,r)$	$\rightarrow$	$d(C,P_1)=r$
segment s_1 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_1)=r$
circle $Q_1(C_1,r_1)$ **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,C_1)=\mid r\pm r_1\mid$

   In figure 10, once points P_1, P_2, P_3 and circle Q_1 have been placed using a constructive approach, circle Q can be computed considering that point P_1 is on its circumference, circle Q is tangent to the segment defined by P_2 and P_3, and Q is also tangent to circle Q_1, as set out below:

$$
\begin{aligned}
(P_{1x}-C_x)^2+(P_{1y}-C_y)^2 &= r^2 \\
a_{23}\cdot C_x+b_{23}\cdot C_y+c_{23} &= r \\
(C_{1x}-C_x)^2+(C_{1y}-C_y)^2 &= (r\pm r_1)^2
\end{aligned}
$$

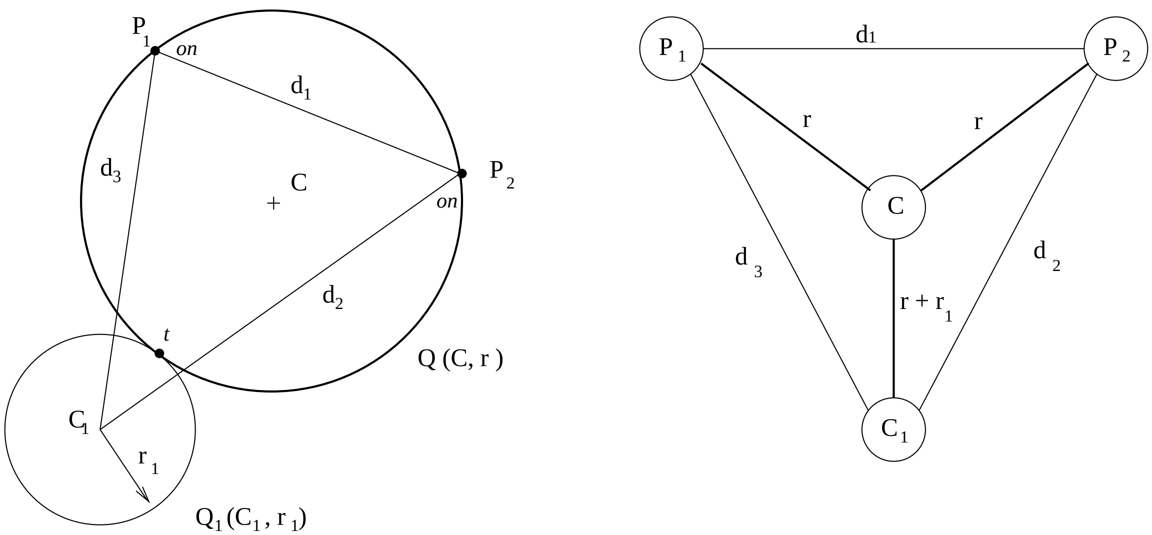


Figure 8: Circle defined by two incident points, P_1 and P_2, and a tangency constraint with circle Q_1.

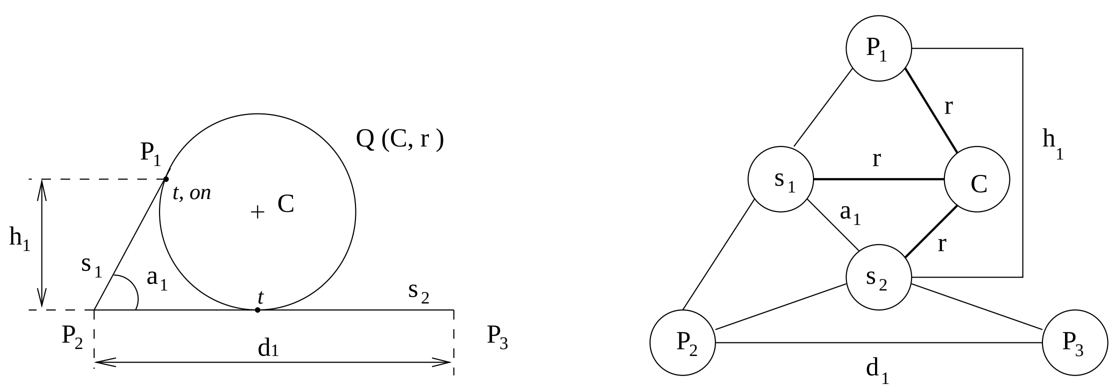


Figure 9: Circle tangent to segments s_1 in P_1 and s_2.

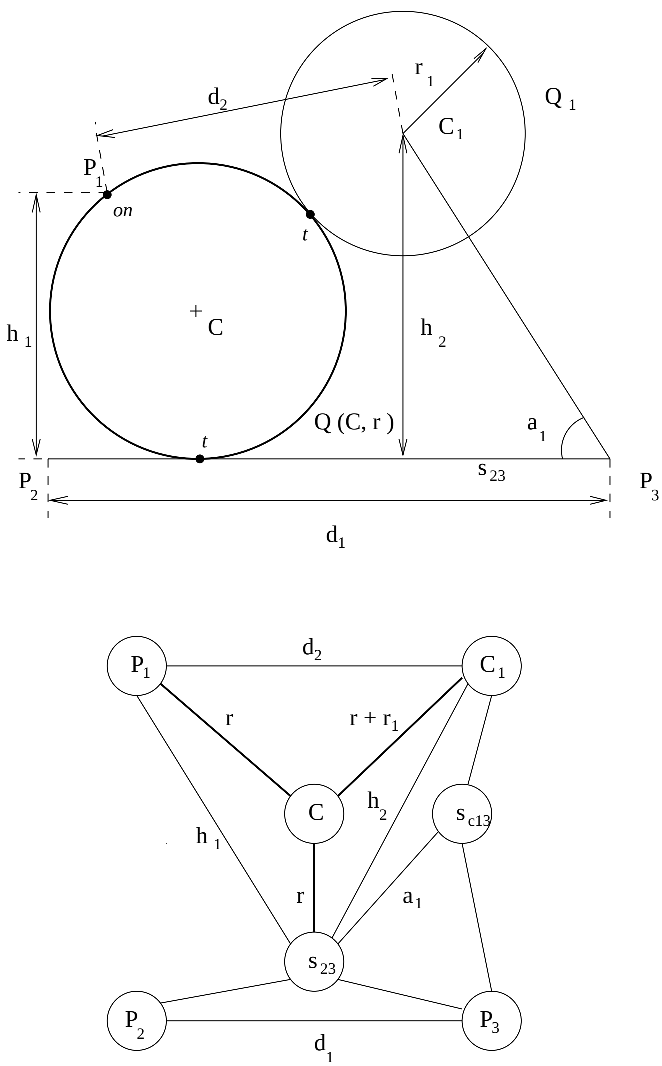


Figure 10: Circle incident to point P_1, tangent to segment s_{23} and tangent to circle Q_1.

6. **Circle defined by an incident point and two tangency constraints between circles**

   Set of constraints:

point P_1 **on** circle $Q(C,r)$	$\rightarrow$	$d(C,P_1) = r$
circle $Q_1(C_1,r_1)$ **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,C_1) = \mid r \pm r_1 \mid$
circle $Q_2(C_2,r_2)$ **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,C_2) = \mid r \pm r_2 \mid$

   Any real solution of the followin system of equations would verify a variable radii configuration with an incident point on the circumference and two tangency constraints between circles:

   $$(P_{1x} - C_x)^2 + (P_{1y} - C_y)^2 = r^2$$
   $$(C_{1x} - C_x)^2 + (C_{1y} - C_y)^2 = (r \pm r_1)^2$$
   $$(C_{2x} - C_x)^2 + (C_{2y} - C_y)^2 = (r \pm r_2)^2$$

   A concrete realization of two tangency constraints between circles is shown in figure 11. Since circles Q_1 and Q_2 both are lying outside the solution circle, the instantiation of the distances between the center points is $r + r_1$ and $r + r_2$, respectively.

7. **Circle defined by three tangency conditions on two segments and one circle**

   Set of constraints:

segment s_1 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_1) = r$
segment s_2 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_2) = r$
circle $Q_1(C_1,r_1)$ **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,C_1) = \mid r \pm r_1 \mid$

   In figure 12, a circle Q tangent to segments s_{23} and s_{34} and to circle Q_1 is defined. The solution circle is obtained as follows:

   $$a_{23} \cdot C_x + b_{23} \cdot C_y + c_{23} = r$$
   $$a_{34} \cdot C_x + b_{34} \cdot C_y + c_{34} = r$$
   $$(C_{1x} - C_x)^2 + (C_{1y} - C_y)^2 = (r \pm r_1)^2$$

   In the example shown in figure 12, the solution circle lies outside the tangent circle. In this configuration, the distance between the center points is $d(C, C_1) = r + r_1$.

8. **Circle defined by tangency constraints between the circle and three segments**

   Set of constraints:

segment s_1 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_1) = r$
segment s_2 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_2) = r$
segment s_3 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_3) = r$

   The following system of equations translates the constraints between circle Q and segments s_1, s_2 and s_3 (see figure 13). The center position and the value for the radii are computed by solving this system of linear equations.

   $$a_1 \cdot C_x + b_1 \cdot C_y + c_1 = r$$
   $$a_2 \cdot C_x + b_2 \cdot C_y + c_2 = r$$
   $$a_3 \cdot C_x + b_3 \cdot C_y + c_3 = r$$

9. **Circle defined by one tangency constraint on a segment and two tangency constraints between circles**

   Set of constraints:

segment s_1 **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,s_1) = r$
circle $Q_1(C_1,r_1)$ **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,C_1) = \mid r \pm r_1 \mid$
circle $Q_2(C_2,r_2)$ **tangent to** circle $Q(C,r)$	$\rightarrow$	$d(C,C_2) = \mid r \pm r_2 \mid$

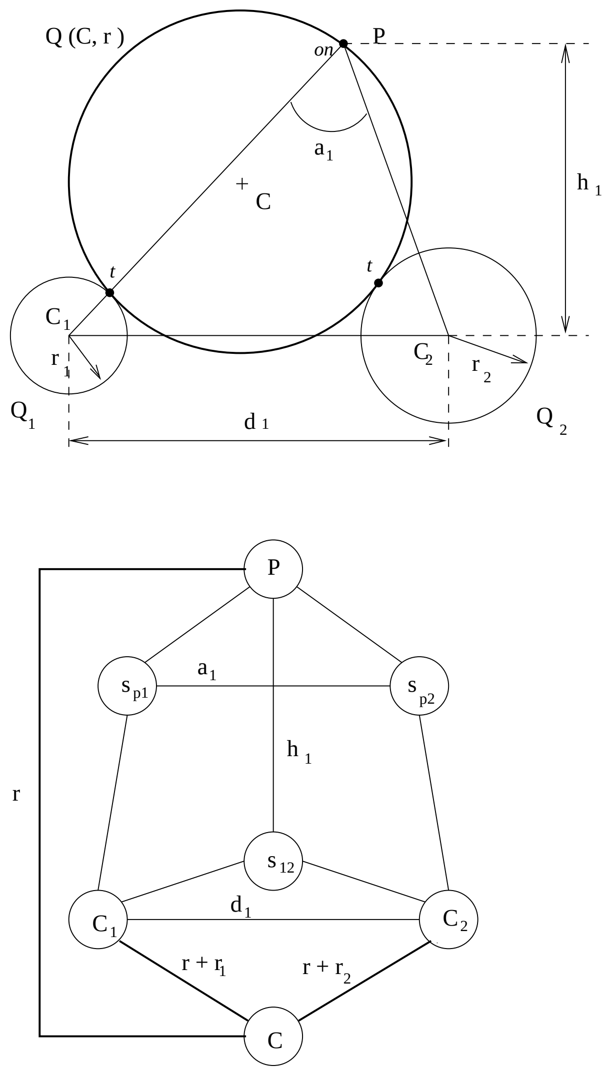


Figure 11: Circle incident to point P and tangent to circles Q_1 and Q_2.

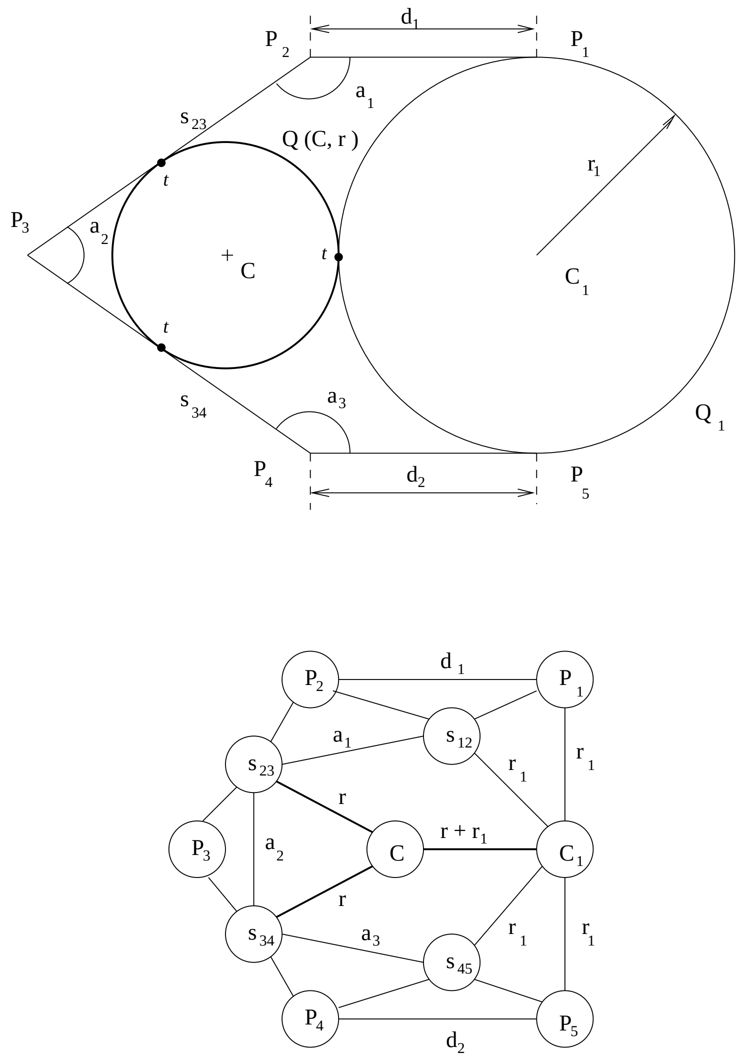


Figure 12: Circle tangent to segment s_{23}, segment s_{34} and circle Q_1.

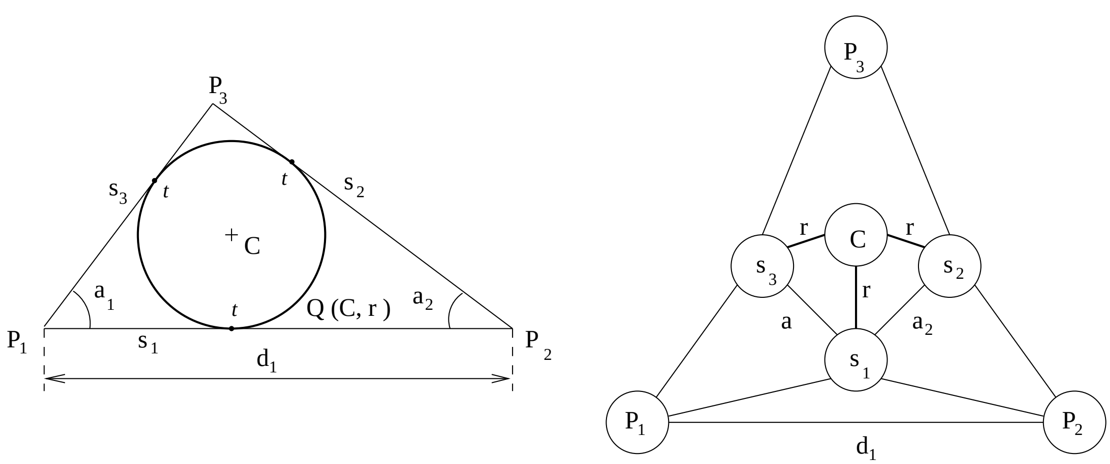


Figure 13: Circle tangent to three segments: s_1, s_2 and s_3.

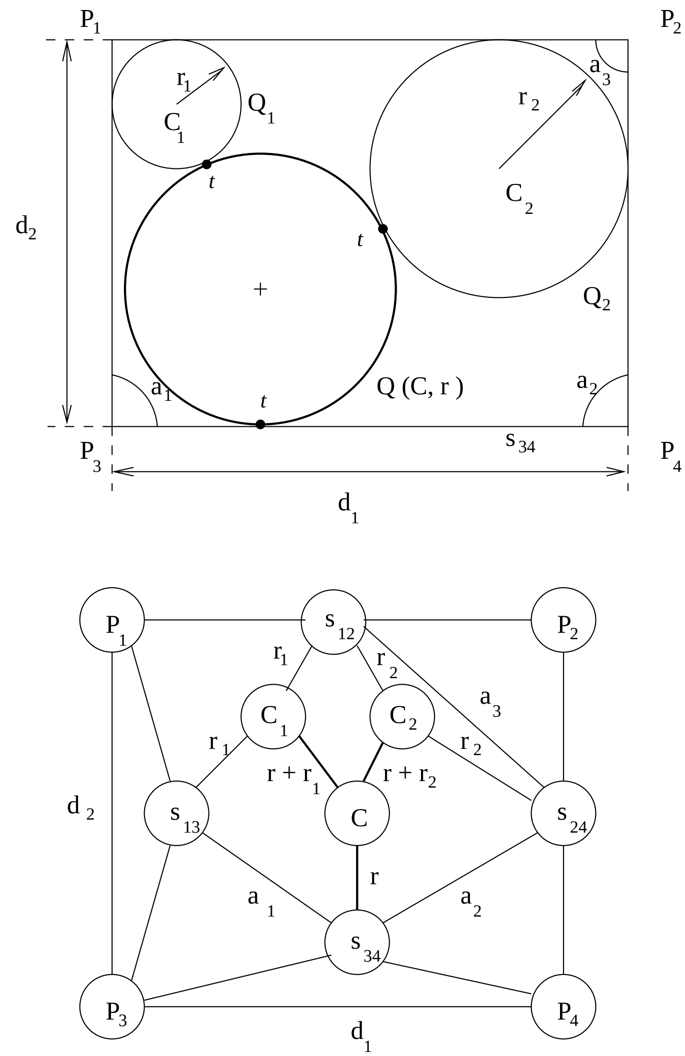


Figure 14: Circle tangent to segment s_34 and tangent to circles Q_1 and Q_2.

The sketch shown in figure 14 is an example of how the value for the radius can be calculated and the center point positioned, while satisfying the set of dimensional and 2D tangency constraints defined on circle Q. Once P_1, P_2, P_3, P_4, C_1 and C_2 have relatively been placed, the construction that needs to be solved is:

$$a \cdot C_x + b \cdot C_y + c = r$$
$$(C_{1x} - C_x)^2 + (C_{1y} - C_y)^2 = (r \pm r_1)^2$$
$$(C_{2x} - C_x)^2 + (C_{2y} - C_y)^2 = (r \pm r_2)^2$$

In this sketch, four possible solutions verify the whole set of constraints, but only one of them would capture the user intent. The problem to choose the desired solution among all possible constructions is a complex task in GCSP [5].

10. **Circle defined by three tangency constraints between circles**

Set of constraints:

circle $Q_1(C_1, r_1)$ **tangent to** circle $Q(C, r)$ $\rightarrow$ $d(C, C_1) = \mid r \pm r_1 \mid$
circle $Q_2(C_2, r_2)$ **tangent to** circle $Q(C, r)$ $\rightarrow$ $d(C, C_2) = \mid r \pm r_2 \mid$
circle $Q_3(C_3, r_3)$ **tangent to** circle $Q(C, r)$ $\rightarrow$ $d(C, C_3) = \mid r \pm r_3 \mid$

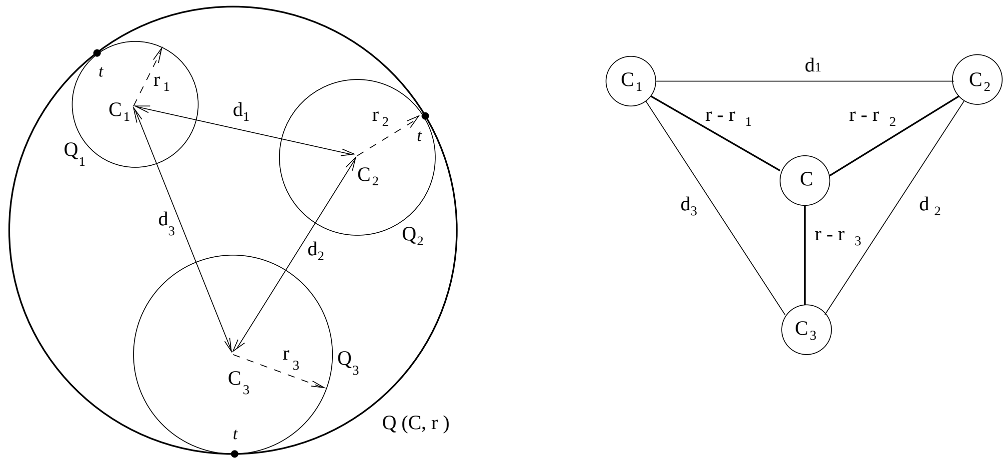


Figure 15: Circle tangent to three circles: Q_1, Q_2 and Q_3.

A circle tangent to three given circles is known as the Problem of Apollonius. In fact, this is the general problem which includes the 9 configurations described above. A point is a circle with null dimension radius. A segment is a circle with a value for the radius equal to infinite.

There are, in general, eight possible solutions depending on the relative position of C with respect to the already placed center points C_1, C_2 and C_3, [9]. These solutions are obtained from the following system of equations:

$$
\begin{aligned}
(C_{1x} - C_x)^2 + (C_{1y} - C_y)^2 &= (r \pm r_1)^2 \\
(C_{2x} - C_x)^2 + (C_{2y} - C_y)^2 &= (r \pm r_2)^2 \\
(C_{3x} - C_x)^2 + (C_{3y} - C_y)^2 &= (r \pm r_3)^2
\end{aligned}
$$

In figure 15, the three given circles are contained within the solution circle.

In a rule-based approach, a specific rule has to be defined for each of the 10 basic configurations given above. In section 4.2.1, using the information given by the position of the center point, three more rules needed to be defined. As a whole, 13 propagation rules need to be considered to solve any configuration involving arcs and circles with variable radii.

On the other hand, any circle defined by three symbolic constraints can be translated into a system of three quadratic equations, with the coordinates of the center point and the value of the radius as the unknowns. The hybrid solver in [6], can switch to numeric methods to compute the position of the center point and the radius dimension. Then it switches back to the constructive method to solve the remaining constraints.

4.2.3 Solving variable radii configurations using propagation rules or a hybrid approach

Adding propagation rules to the solver or using the geometric constraint solver based on a hybrid approach need to be evaluated taking into account the following considerations:

- the number of propagation rules that must be defined
- the scope of the solver
- the correctness of the analyzer using symbolic constraints

Thirteen propagation rules need to be defined to solve any variable radii circle configuration. Since the solver already has 18 construction rules, that means to nearly double the number of rules in the geometric constraint solver.

The hybrid solver described in [6] considers 2D geometric constraint problems involving constraints with fixed value as well as constraints with symbolic value. The technique used deals with two sets of data: the geometric constraint data, represented by a set of clusters, and the symbolic equation data, described by a bigraph.

The constructor builds an instance of the solution by executing a sequence of construction steps generated by the analyser. Since all values needed must be available when a construction step is carried out, the analyser classifies each symbolic constraint according to the way by which its value will be computed. A symbolic constraint is *computable* when its value is to be found by solving a subset of constraint equations. A symbolic constraint is *propagatable* when its value can be derived from geometric elements already placed with respect to each other. When a constraint can be both computable and propagatable, it is considered to be propagatable by the analyser.

The relations derived from incidence and 2D tangency constraints in variable radii circles become propagatable or computable depending on whether the center point position is known or not. As it has been introduced in section 4.2.1, when the center point position is known, all geometric elements involved in the valuation of the radius belong to the same **CD** set. In this configuration, the dimension of the radius is a propagatable constraint. Otherwise, the radius is a computable constraint because its dimension has to be valuated along with the coordinates of the center point (see configurations defined in section 4.2.2).

Propagation rules, of course, enlarge the scope of the solver, but further extensions would require including even more rules. On the contrary, using a suitable translation of new geometries and constraints, these may be included to the hybrid solver without much effort. Besides, the hybrid solver supports other kind of relations such as engineering constraints.

The correctness of the analyser for the currently available set of rules has been established in [7]. Adding propagation rules to the constructive solver would imply proving termination again. The correctness of the analyser in the hybrid solver has been shown in [6].

All these considerations lead us to choose the geometric constraint solver based on a hybrid approach to solve variable radii configurations involving arcs or circles on the plane.

5 Conclusions

This work has reported on the use of circles in geometric constraint satisfaction problems. Incidence and 2D tangency constraints on circles either with fixed or variable radii have been studied.

Considering circles in 2D geometric constraint solving entails three new geometric constraints: point incident to a circle, straigth segment tangent to a circle, and tangency between two circles.

We have shown that geometric objects including circles with fixed radius can be solved without extending the set of construction rules available in our rule-based solver. This has been achieved by including three new creation rules which translate the new geometric constraints into the already existing point to point distance constraint and point to segment distance constraint.

Some constraint driven variational CAD systems cannot solve models where circles have variable radius. We have characterized variable radii configurations and two diferent approaches have been presented: adding a new set of rules to the solver, *propagations rules*, or using a hybrid solver. The hybrid solver extends the capabilities of the constructive solver to constraints on variable radii circles without increasing the present set of construction rules. The correctness of the analyser has been established in previous works.

Acknowledgements

I wish to thank Prof. Robert Joan-Arinyo and Antoni Soto for their advice and fruitful collaboration on the development of this work. This research has been supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT), under the project "Diseño mediante restricciones geométricas" (TIC95-0630-C05-04).

References

[1] W. Bouma, I. Fudos, C. Hoffmann, J. Cai, and R. Paige. Geometric constraint solver. *Computer Aided Design*, 27(6):487–501, June 1995.

[2] B.D. Brüderlin. Using geometric rewrite rules for solving geometric problems symbolically. In *Theoretical Computer Science 116*, pages 291–303. Elsevier Science Publishers B.V., 1993.

[3] E.A. Dijksman. *Motion Geometry of Mechanisms.* Cambridge University Press, 1976.

[4] B. Hendrickson. Conditions for unique graph realizations. *SIAM J.Comput.*, 21(1):65–84, 1992.

[5] C.H. Hoffmann and R. Joan-Arinyo. *Geometric Modeling for Product Realization*, chapter Erep - An Editable, High-Level Representation for Geometric Design and Analysis. Elsevier Science Publishers B.V. (North-Holland), 1993.

[6] C.M. Hoffmann and R. Joan-Arinyo. Symbolic constraints in constructive geometric constraint solving. *Journal of Symbolic Computation*, 1997. To appear.

[7] R. Joan-Arinyo and A. Soto. A rule-constructive geometric constraint solver. Technical Report LSI-95-25-R, Department LiSI, Universitat Politècnica de Catalunya, 1995.

[8] R. Joan-Arinyo and A. Soto. A set of rules for a constructive geometric constraint solver. Technical Report LSI-95-19-R, Department LiSI, Universitat Politècnica de Catalunya, 1995.

[9] R. Rajagopalan Kavasseri. Variable radius circle computations in geometric constraint solving. Master's thesis, Purdue University, August 1996.

[10] G.A. Kramer. A geometric constraint engine. *Artificial Intelligence*, 58(1-3):327–360, 1992.

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

[12] J.C. Owen. Algebraic solution for geometry from dimensional constraints. *ACM*, pages 397–407, 1991.

[13] A. Verroust, F. Schonek, and D. Roller. Rule-oriented method for parameterized computer-aided design. *Computer Aided Design*, 24(10):531–540, October 1992.