Geometric Constraint Solving with Conics and Linkages*

Xiao-Shan Gao, Kun Jiang, and Chang-Cai Zhu
Institute of Systems Science, AMSS, Academia Sinica, Beijing 100080
(xgao,kjiang,czhu)@mmrc.iss.ac.cn

Abstract

Most geometric construction methods of geometric constraint solving systems use line and circle (ruler and compass) as basic drawing tools. In this paper, by introducing conics and linkages, we provide a set of complete drawing tools for the construction approach of geometric constraint solving. Using these tools, we may enlarge the drawing scope of the construction approach and still keep the elegant style of geometric solutions to geometric constraint solving. As applications, we obtain pure geometric solutions to three sets of well-known constraint problems: the ten Apollonian drawing problems, the thirteen cases of a smallest tri-connected constraint graph, and constraint problems with distance constraints only.

Keyword. Geometric constraint solving, parametric CAD, construction methods, conics, linkages, Apollonian drawing problems, tri-connected constraint graph.

1 Introduction

Geometric constraint solving (GCS) is the central topic in much of the current work of developing intelligent and parametric CAD systems. It also has applications in chemical molecular modeling, linkage design, computer vision and computer aided instruction [8]. GCS algorithms accept the declarative description of geometric diagrams or engineering drawings as input and output a drawing procedure for the diagram. There are four main approaches to GCS: the graph analysis approach [19, 20, 26, 29], the rule-based approach

* This work is supported in part by Chinese NSF under an outstanding youth grant (NO. 69725002) and by a National Key Basic Research Project of China (NO. G1998030600).

[2, 9, 14, 27, 31, 30], the numerical computation approach [18, 28, 25], and the symbolic computation approach [4, 10, 24]. In practice, combinations of these approaches are often used to obtain the best result. The work in [15] is such a successful attempt.

The graph analysis and the rule-based approaches are also called geometric construction approach. In this approach, a pre-treatment is carried out to transform the constraint problem into a constructive form that is easy to draw. In most cases, we construct the diagram sequentially with ruler and compass. This approach actually follows the tradition of engineering drawings, where ruler and compass are used as basic drawing tools. The construction approaches are the first choices for most GCS systems because they may provide complete, fast and stable solutions. But, the drawing scope of ruler and compass are restrictive. It is well known that using ruler and compass alone, we can draw diagrams that can be described by a sequence of triangular equations (see Section 2.1) of degree less than or equal to two. As shown in [10], we may find very simple design problems that cannot be solved with ruler and compass. The aim of this paper is to introduce new drawing tools to enlarge the drawing scope of the construction approach and still keep its elegant solution style.

We will consider two kinds of tools: conics and linkages. We prove that the diagrams that can be drawn with conics are those that can be described by a sequence of triangular equations of degree less than or equal to four. Therefore, the drawing scope by introducing conics is essentially larger than using lines and circles only. Since equations with degree ≤ 4 can be solved explicitly by radicals, the computation procedure is still complete and efficient as the case of ruler and compass construction. Furthermore, for some diagrams that can be drawn with ruler and compass, we may give much simpler construction procedures by using conics. As an example, simple solutions are obtained for the ten *Apollonian drawing problems*.

Linkages are one of the most basic kinds of mechanisms. Here, we will use them as drawing tools. Based on a result of Kempe, we prove that linkages as tools are complete in certain sense. We also give an algorithm to find linkages in a constrained diagram. As a consequence, we prove that all constraint problems involving distance constraints only can be solved sequentially with linkages.

To solve the equations raised from linkage constructions, besides the often used numerical and symbolic computation methods, we introduce a *geometric method* which is to use the linkages to generate loci and then to find the intersection of these loci by searching the points on the loci. This geometric method is based on *dynamic generation* of geometric loci which are widely used in *dynamic geometry* software [12].

As an application, we show that a simplest constraint graph which is beyond the scope of Hoffmann and Owen's triangle decomposition methods can be transformed to pure geometric constructive form if linkages are allowed as construction tools. The linkages used in the construction are three kinds of *four-bar linkages*. One of the four-bar linkage is the traditional one. The other two kinds of four-bar linkages are newly introduced.

The rest of this paper is organized as follows. In Section 2, we show how to solve constraint problems with conics. In Section 3, we introduce linkages as a drawing tool. In Section 4, we present the solution to the smallest tri-connected constraint graph. In Section 5, we present the conclusion.

2 Geometric Constraint Solving with Conics

2.1 Constructive Sequence with Drawing Tools

We will introduce several basic concepts for drawing geometric diagrams with general drawing tools. In this paper, a *2D drawing tool* is a mechanism with one degree of freedom. The *locus of a drawing tool* is the trace of a geometric object in the mechanism, usually a point or a line. When the trace object is a point, the locus is a plane curve. A diagram can be drawn with a set of drawing tools sequentially if the geometric objects in the diagram can be listed in an order

$$P_1, P_2, \cdots, P_m \tag{2.1}$$

such that each P_i is introduced by the following basic constructions using objects already drawn $P_1, \cdots, P_{i-1}$:

1. POINT(P): take a free point P in the plane.
2. ON(P, t): take a semi-free point P on the locus of a drawing tool t.
3. INTERSECTION(P, t_1, t_2): take the intersection P of two loci of two drawing tools t_1 and t_2.
4. We also need the definitions for all geometric objects. For instance, CIR(O, r) is the circle with center O and radius r.

Since geometric objects such as lines and circles can be treated as functions of points, we assume without loss of generality that the geometric objects P_i in (2.1) are points. Let C_i be the construction introducing point P_i. We say that the diagram can be drawn with the following *construction sequence*

$$C_1, C_2, \cdots, C_m. \tag{2.2}$$

After introducing coordinates for each point, we may obtain a set of equations from the construction sequence (2.2). If P is introduced by POINT(P), P is a free point. No equation is introduced. If P is introduced by ON(P, t), we need one algebraic equation:

$$f(u,x) = 0 \tag{2.3}$$

to describe P, where $f(u,x) = 0$ is the equation for the locus of t. Note that u is a parameter and x is a dependent variable. Hence $f(u,x) = 0$ will result infinite solutions. The construction INTERSECTION(P, t_1, t_2) will produce two equations:

$$\begin{cases} f_1(x_1,x_2) = 0 \\ f_2(x_1,x_2) = 0 \end{cases} \tag{2.4}$$

where $P = (x_1, x_2)$ and $f_1(x_1,x_2) = 0$ and $f_2(x_1,x_2) = 0$ are the equations for the loci of t_1 and t_2. Since each point is introduced by the three basic constructions, the algebraic equations of (2.2) are as follows by changing the names of the variables appropriately.

$$\begin{cases} f_{1,1}(u_1,\cdots,u_q,x_1,\cdots,x_{n_1}) = 0 \\ \quad\cdots \\ f_{1,n_1}(u_1,\cdots,u_q,x_1,\cdots,x_{n_1}) = 0 \\ f_{2,1}(u_1,\cdots,u_q,x_1,\cdots,x_{n_2}) = 0 \\ \quad\cdots \\ f_{2,n_2}(u_1,\cdots,u_q,x_1,\cdots,x_{n_2}) = 0 \\ \quad\cdots \\ f_{m,n_m}(u_1,\cdots,u_q,x_1,\cdots,x_{n_m}) = 0 \end{cases} \tag{2.5}$$

where $f_{k,1},\cdots,f_{k,n_k}$ are the equations for construction C_k in (2.2), the u_i are the free parameters, and the x_i are the dependent variables. We have $n_i < n_{i+1}$. This means that the equations are naturally divided into blocks consisting of at most two equations. In order to solve (2.5), a natural way is to *triangularize* it [5, 32]. Because of the special form, we may trianglarize (2.5) easily, say, using resultant computation. Let the triangular equations be

$$TS = \begin{cases} g_1(u_1,\cdots,u_q,x_1) = 0 \\ g_2(u_1,\cdots,u_q,x_1,x_2) = 0 \\ \quad\vdots \\ g_p(u_1,\cdots,u_q,x_1,\cdots,x_p) = 0 \end{cases} \tag{2.6}$$

where $p = n_m$. (2.6) is called the *triangular equation system* of the related constraint problem. It is readily seen that besides some special cases, all solutions of the diagram can be obtained from (2.6) by solving several univariate-equations.

2.2 Scope of Construction with Conics

Now suppose that the drawing tools are conics. Then for each $f_{i,j}$ in (2.5), degree$(f_{i,j}) \le 2$. Since $n_i \le 2$, by the *Bezout Theorem*, degree$(g_i) \le 4, (1 \le i \le p)$ for g_i in (2.6).

We have proved that a construction sequence of lines, circles and conic will lead to a triangular set of equations with degree ≤ 4. Now, we will prove that triangular equations with degree ≤ 4 can be generalized by a construction sequence of lines, circles and conics. Note that we need only to solve univariate equations with degree ≤ 4 in the process of solving (2.6). Hence we need only to prove that the root of any univariate equation with degree ≤ 4 can be constructed with lines, circles and conics. The cases of equations with degree less than three are trivial. We only consider the cases of equations with degrees three and four.

Let a cubic univariate equation be

$$p(x) = x^3 + ax^2 + bx + c$$

where a, b, c are numerical coefficients. We can construct two conics $f_1(x, y) = 0$ and $f_2(x, y) = 0$, where,

$$f_1(x, y) = y - x^2, f_2(x, y) = (x + c_1)(y + c_2) + f.$$

Eliminating y from the two equations, we have

$$\begin{aligned} f_2(x, y) &= (x + c_1)(x^2 + c_2) + f \\ &= x^3 + c_1 x^2 + c_2 x + f + c_1 c_2. \end{aligned}$$

Comparing the coefficients of this cubic equation with that of $p(x)$, we have $a = c_1, b = c_2, c = f + c_1 c_2$. Clearly, the roots of $x^3 + ax^2 + bx + c = 0$ can be constructed by conics: $y = x^2$ and $(x + a)(y + b) + c - ab = 0$.

The conics to construct the roots of the following quartic univariate equation

$$p(x) = x^4 + ax^3 + bx^2 + cx + d$$

are of the form

$$\begin{aligned} f_1(x, y) &= x^2 + dy^2 + ax + cy + b \\ f_2(x, y) &= xy - 1. \end{aligned}$$

We have proved the following result.

Theorem 2.1 *A diagram can be drawn with lines, circles and conics sequentially if and only if the coordinates of the points in the diagram can be expressed as roots of a sequence of triangular algebraic equations with degree less than or equal to four.*

From the above theorem, we have two consequences:

1. The drawing scope of conics is essentially larger than using ruler and compass alone.
2. Since for cubic and quartic equations, we may write their roots as formulas of radicals and give explicit formulas to determine the number of real roots [1], the computation process is almost as simple as the case of using ruler and compass.

2.3 Geometric Construction with Conics

In this section, we will develop a rule-based approach for GCS with conics. We use a modification of the global propagation method presented in [9] to solve constraint problems with lines, circles and conics.

Algorithm 2.2 *The algorithm takes a constraint problem as input and outputs a construction sequence for it. The algorithm is a search process with backtracking and has two phases.*

Let PS and CT be the point and geometric constraint sets of the problem. Let $CS = \phi$ be the construction sequence to be found.

1. **Find locus of the point to be drawn.** The points constructed in CS are considered already drawn. The basic idea of propagation is to find a way to construct an un-drawn point with the points already drawn. To draw a point P, we will try to find two loci containing P. Then P is the intersection of the two loci.

   Take a point P from PS. For each constraint $T_i \in CT, i = 1, \cdots, s$ involving P, determine an Lc_i of P satisfying T_i, assuming all points constructed in CS are known.

2. **Find construction.** We will find a construction introducing point P, and remove P from PS.

   - If $s = 0$, point P is an arbitrarily chosen (free) point. We add a new construction POINT(P) to CS.
   - There exist $i \neq j$ such that $T_i \neq T_j$ and Lc_i and Lc_j are not parallel lines, concentric circles, nor conics having no intersection points. We add a new construction INTERSECTION(P, Lc_i, Lc_j) to CS and remove T_i and T_j from CT.
   - Point P is a semi-free point that can move freely on a line, a circle or a conics. We add a new construction ON(P, Lc_i) to CS and remove T_i from CT.

The algorithm finds a construction sequence successfully, if all objects in PS are introduced in this way. The crucial step of the algorithm is how to determine the locus Lc from a constraint T in the first phase. In [9], methods for determine lines and circles are presented. We now give methods to determine conics.

Let us first note that, conics may be introduced by using the following constraints.

- **SUM(A,B,C,s):** the sum of $|AB|$ and $|AC|$ is a fixed value s, where A,B,C are points.

- **DIF(A,B,C,d):** the difference of $|AB|$ and $|AC|$ is a fixed value d, where A,B,C are points.

- **EQU(A,P,L):** the distance from A to P is equal to the distance from A to line L, where A,P are points and L is a line.

Let us assume that B and C are fixed points. The locus of the points A satisfying constraint **SUM(**A,B,C,d**)** is an ellipse whose focal points are B and C. The locus of the points A satisfying the constraint **DIF(**A,B,C,d**)** is a hyperbola whose focal points are B and C. We use HYPERBOLA(B, C, d) to denote this hyperbola. Let RHYPERBOLA(B, C, d) be the branch close to point C. The locus of the points A that satisfy **EQU(**A,P,L**)** is a parabola with focal point P and directrix L.

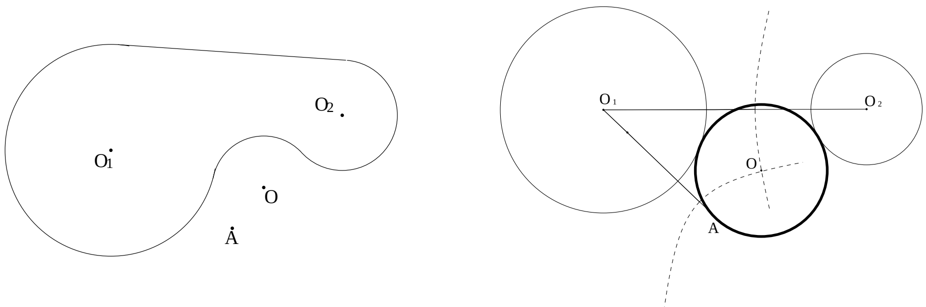

Figure 1: Solving an *An Apollonian drawing problem* with conics

Before presenting a more complicated method of determining conics, we give an illustrative example. Let us consider the machine part in the left side of Figure 1. Circles O_1 and O_2 are fixed with radii r_1 and r_2 respectively, and we need to blend these two circles with an circle O. If we know the radius of the circle O, then it is easy to draw. In this problem, we require that circle O passes through a fixed point A. Since circle O passes through A and is tangent to circle O_1 (externally), we have $|OO_1| - |OA| = r_1$, or equivalently DIF(O, O_1, A, r_1) is valid. Since O_1 and A are fixed, point O is on a hyperbola RHYPERBOLA(O_1, A, r_1). Similarly, point O is on another hyperbola RHYPERBOLA(O_2, A, r_2), and O is the intersection of these two hyperbolas. This problem has four solutions.

The above example is a special case of the *Apollonian drawing problem*, which is to determine a circle satisfying conditions $(C_i, C_j, C_k), (1 \le i, j, k \le 3)$, where,

C_1 : the circle passes through a known point,

C_2 : the circle is tangent to a known line,

C_3 : the circle is tangent to a known circle.

There are altogether ten drawing problems. Although these drawing problems can be solved with ruler and compass[9], the solutions thus given involves many auxiliary points. In [19], the Gröbner basis is used to solve these problems. To solve these problems with conics, we need only compute one intersection point.

The following solution to the Apollonian problem could be used as general construction rules to generate conics. The key step is to determine the center of the circle. Note that the circle satisfies three constraints. If we remove one constraint, the remaining two constraints would determine a locus for the center of the circle. In this way, we obtain three loci and the center of the circle will be determined by computing the intersection point of any two loci. Thus we need to find the locus of the center of circles satisfying one of the six sets of conditions: (C_1, C_1), (C_1, C_2), (C_1, C_3), (C_2, C_2), (C_2, C_3), (C_3, C_3). In the figures below, the bold circle and bold line represent the known circle and known line; the dotted circle and dotted line represent the unknown circle and auxiliary line respectively; the solid curves are the loci of the center of the circle.

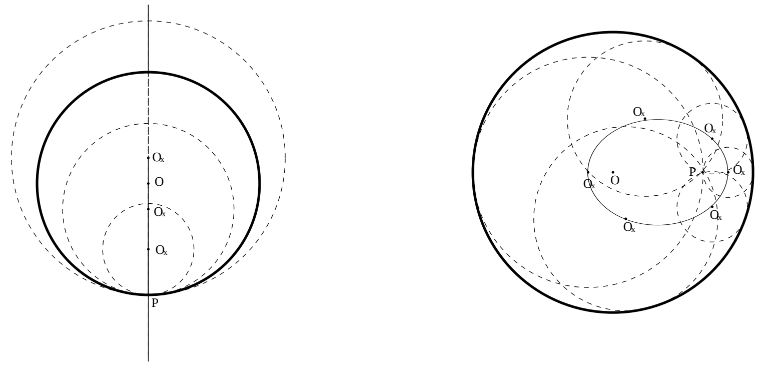

Figure 2: Case (C_1, C_3): point inside of or on circle

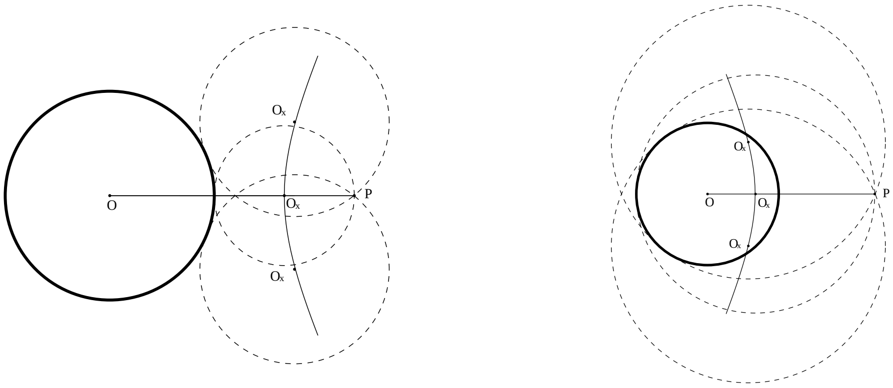

Figure 3: Case (C_1, C_3): point outside of circle

$(\mathbf{C_1}, \mathbf{C_1})$: The circle passes through two known points. Clearly, the locus of the center of the circle is the perpendicular bisector of the segment connecting the two points.

$(\mathbf{C_1}, \mathbf{C_2})$: The circle passes through a known point and is tangent to a known line. In this case, the distance from the center of the circle to the known point equals the distance from the center of the circle to the known line. The locus of the center of the circle is a parabola whose focal point is the known point and directrix the known line.

$(\mathbf{C_2}, \mathbf{C_2})$: The circle is tangent to two known lines simultaneously. If two known lines are parallel, the locus of the center of the circle is a parallel line equidistant to the two known lines. Otherwise, it is the two bisectors of the angle formed by two known lines.

$(\mathbf{C_1}, \mathbf{C_3})$: The circle passes through a known point and is tangent to a known circle. Different positions between the point and circle will result in different loci. When the known point is inside the known circle (Figure 2), the absolute value of the sum of the distance from the center of the circle to the known point and from the center of the circle to the center of the known circle is a fixed value. Hence the locus is an ellipse whose focal points are the known point and the center of the known circle. When the known point is on the known circle (Figure 2), the locus is the line passing through the known point and the center of the known circle. When the known point is outside of the known circle (Figure 3), the absolute value of the differences between the distances from the center of the circle to the known point and from the center of the circle to the center of the known circle is a fixed value. Hence the locus of the center of the circle is a hyperbola whose focal point is the known point and the center of the known circle.

In what below, we only give the conclusion without analyzing.

$(\mathbf{C_2}, \mathbf{C_3})$: The circle is tangent to a known line and a circle. When the known circle and the known line are intersecting, the locus is a parabola whose focal point is the center of the known circle and the directrix is a line having fixed distance to the known line. When the known line is tangent to the known circle, the locus is the line perpendicular to the known line and passing through the center of the known circle or the parabola whose focal point is the center of the known circle and the directrix is a line having fixed distance to the known line. When the known circle and the known line are separate, the locus is the parabola whose focal point is the center of the known circle and the directrix is a line having fixed distance to the known line.

$(\mathbf{C_3}, \mathbf{C_3})$: The circle is tangent to two known circles. When the two known circles and the circle are tangent at the same point, the locus is a line passing through the centers of two known circles. When the two known circles are internally tangent, the locus is an ellipse whose focal points are the centers of the two known circles. When the two known circles are intersecting and the circle is an inscribed circle of one known circle and a circumscribed circle of another known circle, the locus is an ellipse whose focal points are the centers of the two known circles. Except the above cases, the locus is a hyperbola whose focal points are the centers of the two known circles.

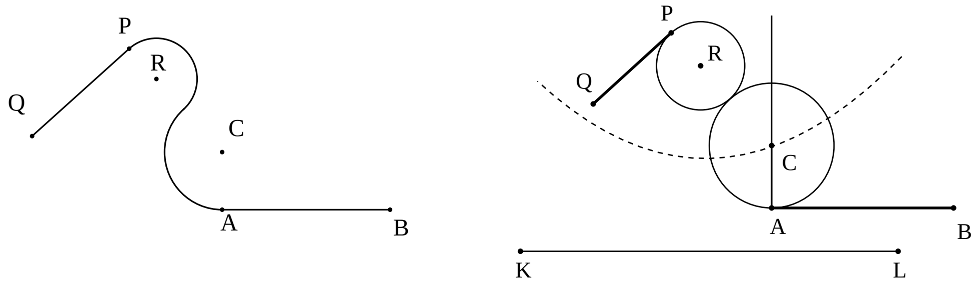

Figure 4: Blending two segments with two circles

In [7], conics are used to blend two line segments. As an example, we will show how to do this with two circles. The problem is to find two tangent circles with centers R and C, which pass through points A and P and tangent to AB and and PQ respectively (Figure 4). Since two circles have six free parameters and the problem has five constraints, we may fix the radius of one circle, say the circle with center R. Now the problem becomes to determine a circle passing through point A, tangent to line AB and circle R. This is an Apollonian problem (C_1, C_2, C_3). Since points A, B, P, Q are fixed, we need to draw points R and C according to Algorithm 2.2. Since we assume that the radius PR is known, we may draw point R easily. We are going to draw point C. From constraint $CA \perp AB$, C is on a line l passing through point A and perpendicular to AB. Since circle C is tangent to line AB and circle R (externally), $|RC| - |CA| = |PR|$ which is a fixed number. Let KL be a line with distance $|PR|$ from line AB. Then the distance from C to R is the same as the distance from C to line KL. Then point C is on a parabola c with R as focal point and with KL as the directrix. It is clear that point C is the intersection of l and c.

3 Construction with Linkages

3.1 Scope of Construction with Linkages

In this paper, by a *linkage*, we mean a mechanism with one degree of freedom and consisting of links with fixed lengths and rotation joins only [17]. One linkage is shown in Figure 5. In the linkage, points A and B are fixed and when point D rotates on circle $\mathrm{CIR}(B, |DB|)$, point E generates a locus which is represented by the dotted curve.

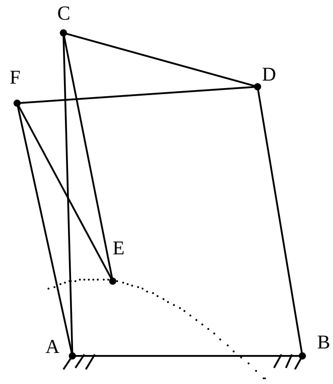

Figure 5: A linkage with eight bars

Since we can draw lines and conics with linkages [8], drawing with ruler, compass and conics is a special case of drawing with linkages. So it is natural to ask what is the drawing scope of linkages as drawing tools. We will prove that linkages as drawing tools are complete in certain sense. First, we will show that the locus of a linkage is an algebraic curve. Introducing coordinates for the joint-points in the linkage, each distance constraint may be transformed into an algebraic equation. Let PS be the equations thus obtained. Since the linkage is of one degree of freedom, the solution of PS is an algebraic set of dimension one. Let (x, y) be the coordinates of the locus point. With an elimination theory, such as Wu-Ritt's method [32, 10], we may obtain an algebraic equation $f(x, y) = 0$ which determines an algebraic curve as the locus of the linkage. Details of the method may be found in [6].

Now suppose that a constraint problem may be drawn with linkages sequentially. Then we may follow the argument in Section 2.1 to obtain a set of equations (2.6) in triangular form, where each g_i is an algebraic equation of arbitrary degree. The nice thing is that the converse of the above process is also true. We will show that if a diagram may be described with a set of equations of form (2.6) then it can be constructed with linkages sequentially. This is valid because of a famous result of Kempe [22] which states that we may design a linkage to draw any given planar algebraic curve. In [13], we improved and implemented Kempe's result and showed that the complexity of the Kempe linkage is $O(n^4)$ where n is the degree of the curve. We have proved the following result.

Theorem 3.1 *A diagram can be drawn with linkages sequentially if and only if it can be described by a set of triangular algebraic equations like (2.6).*

3.2 Geometric Construction with Linkages

We still use the global propagation method introduced in Section 2.2 for GCS with linkages. As we have mentioned that the crucial step of the method is to find a locus for the point to be drawn. This means that we need to find a linkage which may be used to draw a locus containing the point. This is the task of the following algorithm.

Algorithm 3.2 *Let QS be the set of points in the constraint problem which have been drawn, and P_0 the point selected to be drawn in the next step. The algorithm is going to find a set of points L containing P_0 such that L together with the distance constraints consist of a linkage. This linkage may be used to draw a locus for point P_0.*

S1 If there is a point $Q \in QS$ such that $|P_0Q|$ is known, then P_0 is on a circle with center Q. The algorithm terminates. Otherwise, let $S_0 = \{P_0\}$ and go to S2.

S2 Let S_1 be the set of points P such that $\exists Q \in S_0$ and $|PQ|$ is known. If S_1 is an empty set, the algorithm terminates without finding a linkage.

S3 Let d be the number of distance constraints between pairs of points in $S_1 \cup S_0$, at least one of the point is un-drawn. Let n be the number of un-drawn points in $S_1 \cup S_0$.

S4 If $d = 2n-1$, then the points in $S_1 \cup S_0$ consist of a linkage. The algorithm terminates.

S5 If $d > 2n-1$, then point P_0 is in an over-constrained sub-diagram. The algorithm terminates without finding a linkage. Otherwise, i.e., $d < 2n-1$, let $S_0 = S_1 \cup S_0$ and go to S2.

The above algorithm is linear in terms of the number of geometric objects and constraints in the problem. Note that we do not show how to check whether the linkage is well-constraint, which may be done with the method in [26]. We will show how to compute the locus of the linkage in the next subsection.

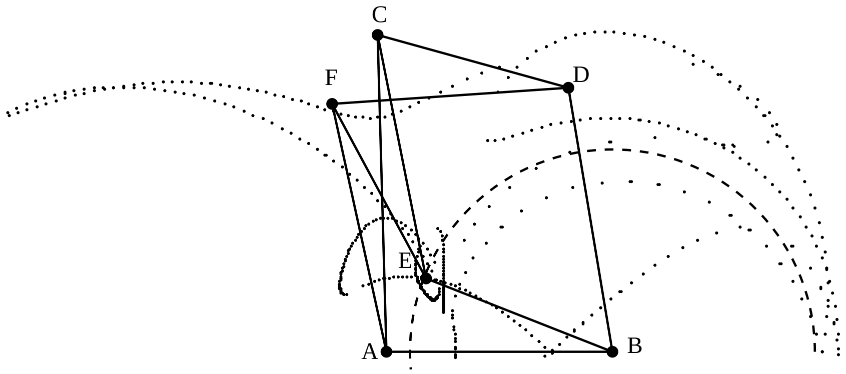

Figure 6: A constraint problem with 16 solutions

Figure 6 is a constraint problem consists of nine segments whose lengths are known. With the global propagation method, we may first draw points A, B, and then draw point E. Since $|BE|$ is known, E is on a circle $l_1 = \text{CIR}(B, |BE|)$, where $\text{CIR}(B, |BE|)$ is the circle with center B and radius $|BE|$. Remove this constraint from the constraint set. We will show how to find a linkage l_2 to draw point E with Algorithm 3.2. In Step

S2, since $S_0 = \{E\}$ and $|EC|$, $|EF|$ are known, $S_1 = \{C, F\}$. Since $n = 3, d = 2$ and $d < 2n - 1$, we need to go back to S2. Since $|CA|$, $|CD|$, $|FA|$, and $|FD|$ are known, $S_1 = \{A, D\}$. We still have $d < 2n-1$. After another execution of S2, we have $S_1 = \{B\}$, and $l_2 = \{E, C, F, D, A, B\}$ consists of a linkage for point E. Point E is the intersection of circle l_1 and the locus for l_2. Note that linkage l_2 is the one in Figure 5.

The above example is one of the smallest constraint problem containing point-to-point distance constraints only that cannot be solved with ruler and compass. We will show that all constraint problems of this kind can be solved by linkages constructively.

Theorem 3.3 *All structurally well or under constrained problems with point-to-point non-zero distance constraints only can be solved with linkages constructively.*

Proof. We need only consider well-constrained problems since under-constrained problems may become well-constrained problems by adding appropriate number of point-to-point distance constraints, using the method presented in [26]. By structurally well-constraint, we mean that each sub-diagram of the linkage with k joins must have $2k - 3$ distance constraints. We assume that the problem contains n points. Then it must have $2n - 3$ constraints. Let us assume that $|AB|$ is known. We first draw A, B. Let $\mathbf{S} = \{A, B\}$. If C is a point such that $|AC|$ and $|BC|$ are known, add C to $\mathbf{S}$. We may construct C as the intersection of two circles. Repeat the above process until we cannot go further. Let $\mathbf{T}$ be the set of the remaining points, and $k = |\mathbf{S}|, t = |\mathbf{T}|$. Then $k + t = n$.

There must be points $P \in \mathbf{T}$ and $Q \in \mathbf{S}$ such that $|PQ|$ is known. We will construct P which is already on a circle since $|PQ|$ is known. The number of constraints used in $\mathbf{S}$ is $2k-3$. Since $|PQ|$ is also used, we have $(2n-3)-(2k-3)-1 = 2(n-k)-1$ constraints left. For the point set $\mathbf{T}$ to form a linkage, we need $2t-1 = 2(n-k)-1$ constraints. Then by Algorithm 2.2, $\mathbf{T}$ forms a linkage, and P is the intersection of a circle and the locus of this linkage. From the next subsection, we may see that to compute the intersection, we will generate the coordinates for all points in the linkage. So we may add all points in the linkage to S and repeat the above process for the remaining points similarly. ∎

3.3 Computation of the Construction Sequences of Linkages by Locus Intersection

Given a construction sequence $(C_1, C_2, \cdots, C_n)$, by introducing coordinates properly, we will have an equation system (2.5). Now we will show how to solve this equation system. Basically, we need to solve two algebraic equations.

$$\begin{cases} f(u_1, \cdots, u_m, x, y) = 0 \\ g(u_1, \cdots, u_m, x, y) = 0. \end{cases} \tag{3.1}$$

Please notice that in certain cases, f and g are the equations of the loci for some linkages, which are not explicitly given. In some special cases, the equations f and g in (3.1) are explicitly given. For instance, since lines, circles, and conics can be drawn with linkages, we may use them as special cases of linkages. Another particular linkage must mention is the four-bar linkage whose locus is an algebraic curve of degree six [17].

If the equations in (3.1) are explicitly given, we may use numerical methods such as the Newton-Ralphon iteration method to solve it. Since there are only two equations, we may convert equation (3.1) into the following form by resultant computation.

$$\begin{cases} t_1(u_1, \cdots, u_m, x) = 0 \\ t_2(u_1, \cdots, u_m, x, y) = 0 \end{cases}$$

which can be solved by the roots isolation method presented in [16]. In practice, we may use a hybrid approach: using the resultant and roots-isolation method to find the initial values and use the Newton-Ralphon method to find high precision solutions.

If the linkage is complex, then to find the equation for its locus is a difficult task. In this case, we propose to use the *locus intersection method* to find the solutions of (3.1). The locus intersection method has three main steps.

Step 1. Generate a construction sequence for the linkage. Let L_1 and L_2 be two linkages whose loci are defined by equations in (3.1). To draw the loci for the linkages, we need first to find a construction sequence for them.

Algorithm 3.4 *For a linkage found with Algorithm 3.2 in a constraint problem, let L be the set of points in the linkage, P the locus point. Let S be the set of points in L which are already drawn in the constraint problem. Points in S are considered to have fixed positions in the linkage. The algorithm will find a construction sequence for the linkage to generate a locus of P.*

Before giving the algorithm, we analysis what kinds of constructions are needed. For each point Q, let $DEG(Q)$ be the number of constraints involving Q. If $DEG(Q) = 2$, then there exist two distance constraints $|QU| = d_1$ and $|QV| = d_2$ and Q can be constructed as the intersection of two circles $\mathrm{CIR}(U, d_1)$ and $\mathrm{CIR}(V, d_2)$. If $DEG(Q) = 1$, then there exists one distance constraint $|QW| = d_1$, and Q can be constructed as a semi-free point on circle $\mathrm{CIR}(W, d_1)$. From the above analysis, if $DEG(Q) \leq 2$ then we have a construction for point Q.

The algorithm is to repeatedly remove points satisfying $DEG(Q) \leq 2$. If all points are removed in this way, we find a construction sequence for the linkage. The following algorithm is a linear realization of the above idea.

1. For each point in $L - S$, we assign a FLAG to it and the initial value of the FLAG is zero.

2. For each point Q satisfying $DEG(Q) \leq 2$, we obtain a construction for Q, move Q to an ordered list l, and set the FLAG part of Q to one. Also remove Q and the constraints involving Q from the problem. If l is empty, the algorithm terminates without finding a result.

3. For each point R in l do the following: for each point $U \in L - S$ such that distance(R, U) is known, DEG$(U) \leq 2$ and FLAG$(U) = 0$, we may obtain a construction for U, add U to the end of l, set FLAG(U) to be one, and remove U and the constraints involving U from the problem.

4. If all points in $L - S$ are removed in this way, then every un-drawn point in the linkage has been constructed and we find a construction sequence for the linkage. Otherwise, the algorithm fails.

By using a graph representation [19, 29] for the linkage, the above algorithm is linear in terms of the number of points in the linkage. This method works for simple linkages only. For complex linkages, i.e. linkages with loops, more complex methods are needed to generate the construction sequence [11].

For the linkage in Figure 5, suppose that we have drawn A, B and will draw point E with the linkage $\{E, C, F, D, A, B\}$. Then $S = \{A, B\}$. There exists one point satisfying DEG$(E) = 2$, which leads to the following construction for point E.

$$\mathrm{INTERSECTION}(E, \mathrm{CIR}(C, |CE|), \mathrm{CIR}(F, |FE|)) \tag{3.2}$$

After removing E and constraints involving E, we have DEG$(F) = 2$ and DEG$(C) = 2$, which lead to the following constructions for C and F.

$$\begin{array}{l} \mathrm{INTERSECTION}(C, \mathrm{CIR}(A, |AC|), \mathrm{CIR}(D, |DC|)) \\ \mathrm{INTERSECTION}(F, \mathrm{CIR}(A, |AF|), \mathrm{CIR}(D, |DF|)) \end{array} \tag{3.3}$$

Next, we have DEG$(D) = 1$, which leads to the following construction for D.

$$\mathrm{ON}(D, \mathrm{CIR}(B, |BD|)). \tag{3.4}$$

Since A and B are fixed points, (3.4), (3.3), and (3.2) consist of a construction sequence for the linkage in Figure 5.

Step 2. Generate Locus. Locus generation is a basic function of *dynamic geometry* [12]. It works as follows.

1. Find a driving point. Since a linkage has one degree of freedom, there must exists a construction like $\mathrm{ON}(D, \mathrm{CIR}(A, d))$ in the construction sequence. D may move freely on a circle and will be treated as a driving point.

2. While the driving point moves on a circle, for each position of the driving point, we may compute the coordinates of all points in the linkage according to the construction sequence. In particular, the coordinates of the *locus point*.

3. Repeat the preceding step, we have a set of coordinates of the locus point. The locus is the set of points generated in this way.

For the linkage in Figure 6, from (3.4) point D is a driving point. For each position of D, we may compute the positions of F, C, and E from (3.3) and (3.2) respectively. The dotted curve is the locus of point E generated in this way. Note that each construction in (3.2) and (3.3) will lead to two solutions. So the locus may have eight branches. The locus in Figure 5 is only one of the branches. The thin dotted curves in Figure 6 are the eight branches. Note that we only draw the curve above line AB.

Step 3. Find Intersection. After the two loci for L_1 and L_2 were generated, we search them to find two points $P_1 \in L_1$ and $P_2 \in L_2$ such that $|P_1P_2|$ has minimal value. We may use the middle point of P_1P_2 as an approximate solution.

Figure 6 shows the solution to the constraint problem in this way, which is generated by a software named Geometry Expert [12]. The thin dotted curves are the loci of the linkage $ABCDFE$ with locus point E. The thick dotted curve is the circle $\mathrm{CIR}(B, |BE|)$. From this diagram, it is easily seen that the curve and the circle have eight intersection points. Hence, the problem has 8 solutions. Since the loci are symmetric with AB, we have 16 solutions in total.

Notice that all solutions may be generated in this way. In practice, this method is quite efficient, because to generate the locus we need only to solve linear and quadratic equations which have closed form solutions [12]. We use the following strategy to enhance the efficiency: at the beginning, we may use relatively large steps for the driving points so that less points for the locus will be generated. After finding the approximate solutions, we will use small steps for the driving points near the approximate solution to find high precision solutions. A drawback of this method is that we do not know the number of solutions before the computation.

4 Solution to a Smallest Tri-connected Constraint Graph

Hoffmann and Owen's triangle decomposition methods are among the fundamental methods of GCS. The basic idea is to divide the constraint graph into three parts, construct the each part separately, and then assemble them together. The difference is that Hoffmann uses a bottom-up approach and Owen uses a top-down approach. Constraint graph that can be solved by these methods are non-tri-connected graphs [19, 29]. A simplest constraint graph that cannot be solved with these methods is the one in Figure 7.

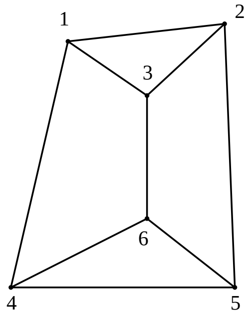

Figure 7: A smallest tri-connected graph

The vertices of the graph could be a point or a line. The edges represent geometric constraints:

Pair of vertices	Geometric constraint represented by the edge
Point/Point	Distance between two points
Line/Line	Angle formed by the two lines
Point/Line	Coincidence or distance from point to line

Thus Figure 7 represents 13 types of constraint problems. We introduce a notation to represent the graph:

$$(V_1V_2V_3, V_4V_5V_6)$$

where V_i could be P or L. If $V_i = P$, then the i-th position in Figure 7 is a point. If $V_i = L$, then the i-th position in Figure 7 is a line.

In order to solve these problems, we need three types of *four-bar linkages*.

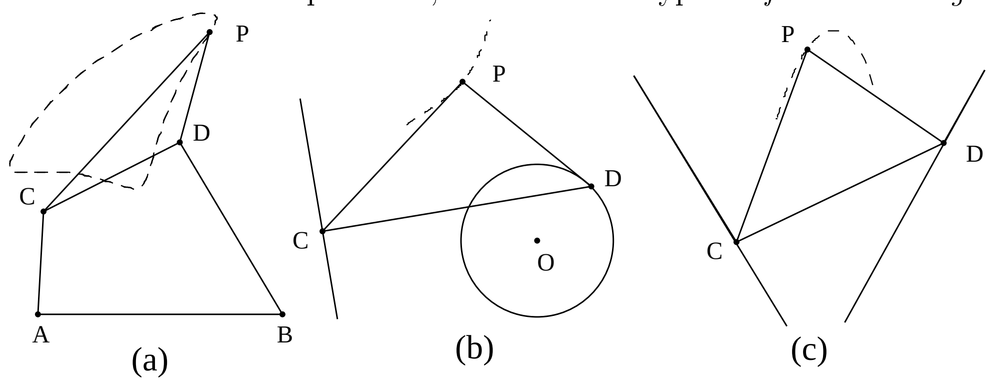

Figure 8: Three types of four-bar linkages

The linkage in Figure 8(a) is the *four-bar linkage*, and is denoted by $(ABCD, P)$. With points A, B fixed and C rotating on a circle, point P will generate the locus of the linkage.

An *lc-four-bar* linkage consists of a fixed line l, a fixed circle c, and a triangle PCD with fixed shape such that $C \in l$ and $D \in c$ (Figure 8(b)). The locus is generated by point P. This linkage is denoted by $(ucCD, P)$ and is a general form of the slider-crane linkage [17].

An *ll-four-bar* linkage consists of two fixed lines u, v and a triangle PCD with fixed shape such that $C \in u$ and $D \in v$ (Figure 8(c)). The locus is generated by point P. This linkage is denoted by $(uvCD, P)$.

In the above linkages, we need to move a point on a line, which can be done with the Peaucelier linkage [17].

Case 8 and 8' are solved in [3] with the Gröbner basis method. In [27], they are solved with numerical computation. We will show that all of the thirteen cases can be solved constructively if linkages are allowed as drawing tools.

Theorem 4.1 *All the 13 constraint problems represented by the graph in Figure 7 can be solved with the three types of four-bar linkages sequentially.*

The following table gives the information on how to solve the 13 problems.

NO	Problem	P/L	Type	R/C	Loci
1	(PPP,PPP)		Well	No	circle/four-bar
2	(PPP,PPL)		Well	No	line/four-bar
3	(PPP,PLL)	Coincidence	Well	Yes	circle/circle
3'	(PPP,PLL)	Distance	Well	No	line/ll-four-bar
4	(PPP,LLL)		over		
5	(PPL,PPL)		Well	Yes	circle/circle
6	(PPL,LPP)	Coincidence	Under	Yes	line/circle
6'	(PPL,LPP)	Distance	Well	No	line/lc-four-bar
7	(PPL,PLL)		Well	Yes	line/circle
8	(PPL,LLP)	Coincidence	Under	Yes	line/circle
8'	(PPL,LLP)	Distance	Well	No	line/ll-four-bar
9	(PPL,LLL)		over		
10	(PLL,PLL)		over		
11	(PLL,LLP)		well	Yes	line/circle
12	(PLL,LLL)		over		
13	(LLL,LLL)		over		

Table 1. The thirteen tri-connected constraint graphs

In the above table, **P/L** means the type of point-line constraint. **Type** means whether the problem is well-, over- or under-constrained. **R/C** means whether the problem can be drawn with ruler and compass. **Loci** means the two most complicated loci or linkages the intersection of which is needed in the construction.

Of the thirteen cases, five use linkages: cases 1, 2, 3', 6' and 8'.

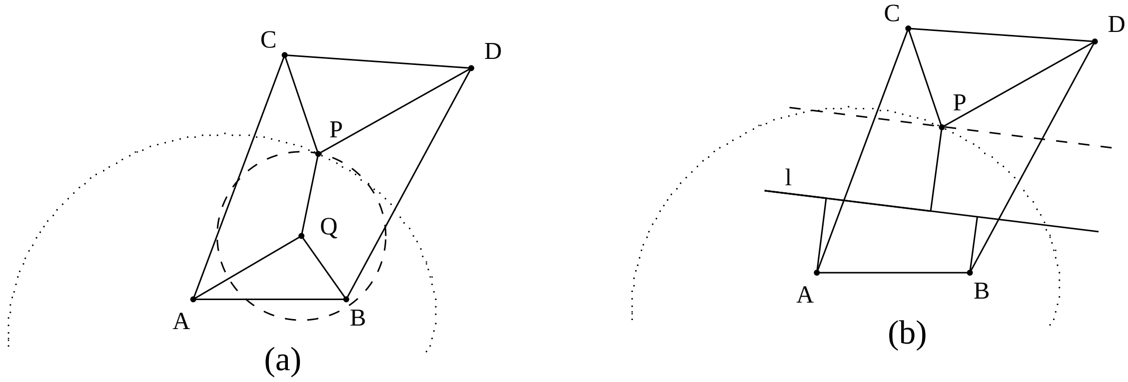

Figure 9: Cases (PPP,PPP) and (PPP,PPL)

Case 1 is shown in Figure 9(a). We may first draw triangle ABQ. Next, we will determine point P. Since $|QP|$ is known, P is on a circle. Remove this constraint from the constraint problem, we may find that point P is on the locus of a four-bar linkage $(ABCD, P)$. Then P is the intersection of the circle and the locus of the four-bar linkage $(ABCD, P)$ (Figure 9(a)). Case 2 is similar to case 1. We may first draw A, B, l, and P is the intersection of a line and the locus of the four-bar linkage $(ABCD, P)$.

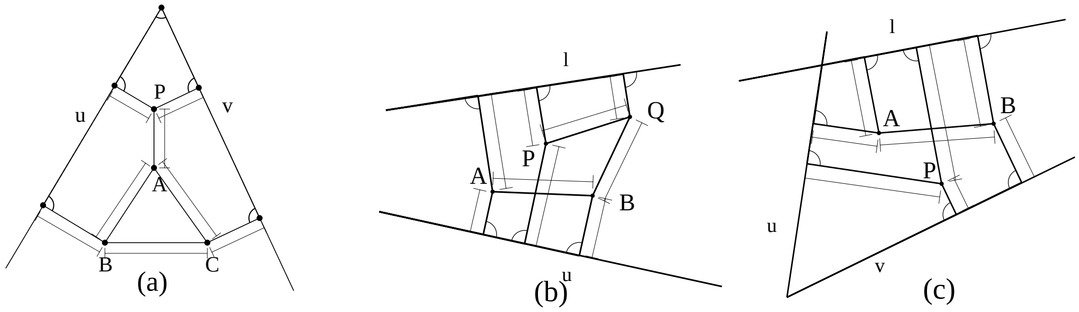

Figure 10: Cases ()

Figure 10(a) is the geometric diagram for case 3'. We may first draw uvP. Since distance(B, u) and distance(C, v) are known, B and C move on two lines and we have an ll-linkage $(uvBC, A)$. Now A is the intersection of circle CIR$(P, |PA|)$ and the locus of the ll-four-bar linkage $(uvBC, A)$.

Cases 6' and 8' need special explanation. Figure 10(c) is the geometric diagram for case 8'. We first draw the diagram Puv. Next, we will draw line l. Since distance(l, P) is known, l is tangent to a circle. Then we may generate one locus of l. Note that ABl is a rigid body and points A and B move on two lines. Then we may use an ll-four-bar linkage to simulate the movement of the rigid body ABl, and to generate a second locus of l. The position of l can be determined as the intersection of the two *loci of lines*. Case 6' can be treated similarly.

Cases 5, 7 and 11 can be drawn with ruler and compasses. They can be solved with

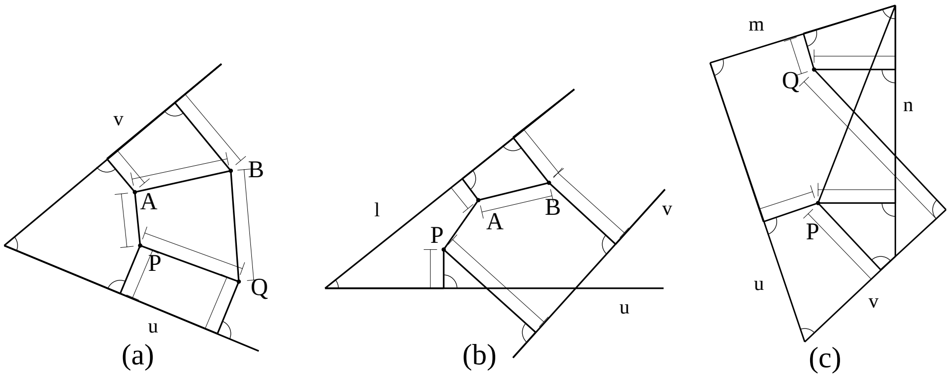

Figure 11: Problems can be solved with ruler and compass

the global propagation method presented in [9].

- To solve case 5 (Figure 11(a)), we first draw PQu. Since ABv and PQu are rigid bodies, we know the angle formed by lines AB and PQ. Now the problem is transformed into the following problem: "draw a quadrilateral if we know the lengths of its four sides and the angle formed by a pair of opposite sides," which has been solved in [9, 31].

- To solve case 6 (Figure 11(b)), we first draw uvP. Next, we will draw point A. Since $|PA|$ is known, A is on a circle c. Similar to case 5, we know the angle between lines AB and u. Since distance(B, v) is known, B is on a line l_1. Since $|AB|$, the direction of line AB are known and B moves on line l_1, by transformation $B \to A$, A must move on another line l_2. A is the intersection of c and l_2. For details about this kind of transformations, see [9].

- To solve case 11 (Figure 11(c)), we first draw uvP. Next, we will draw line n. Since distance$|nP|$ is known, n is tangent to a circle. Since $\angle(n, m)$ and $\angle(m, u)$ are known, we know the direction of n. Now n is a line with known direction and tangent to a known circle, and thus can be determined.

Cases 4, 9, 10, 12, 13 are over-constrained, because there are conflicting constraints. However, if these constraints in them are compatible, all of the constrained systems are under-constrained system and can be drawn with ruler and compass easily.

As an example, let us consider the *Pappus' Drawing Problem* [10, 27]: put a rectangle into a rectangle $ABCD$ as shown in Figure 12. This is actually a variant of case $8'$. Points E, F and line CE are the PPL part and lines DB, AB and point C are the LLP part. It

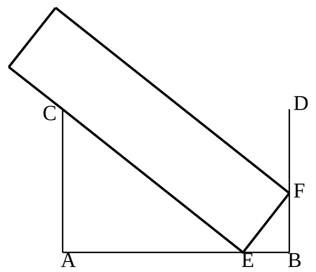

Figure 12: *Pappus' Problem*

is shown that this problem cannot be drawn with ruler and compass in [10]. As we have discussed above, we may solve this problem with an ll-four-bar linkage.

The above example suggests that we may use the solution patterns presented in this section as general construction rules to a rule-based method such as the one in [9].

5 Conclusion

Geometric construction approaches are usually among the first choices by GCS systems because these approaches may provide complete and efficient solutions to geometric constraint problems. In this paper, we extend the scope of construction approaches by introducing two new kinds of drawing tools: conics and linkages. Using these tools, construction approaches become a complete method for all kinds of constraint problems theoretically. Practically, many interesting constraint problems have been solved in this way.

We present our results in 2D case. It is interesting to extend this method to 3D case, because there are more difficult constraint problems similar to the smallest tri-connected problem reported in Section 4, which are still not satisfactorily solved [21]. In [21], one 3D problem is solved using a method similar to the one used in this paper.

Acknowledgment. The first author wants to thank Profs. Shang-Ching Chou and Jian-Xin Ge for valuable discussions.

References

[1] D. S. Arnon and M. Mishra, On Mechanical Quantifier Elimination for Elementary Algebra and Geometry, *J. of Symbolic Computation*, **5** 237–60, 1988.

[2] B. Bruderlin, Using Geometric Rewriting Rules for Solving Geometric Problems Symbolically, *Theoretical Computer Science*, **116**, 291-303, 1993.

[3] W. Bouma, I. Fudos, C. Hoffmann, J. Cai and R. Paige, Geometric Constraint Solver, *Computer Aided Design*, **27**, 487-501, 1995.

[4] S.A. Buchanan and A. de Pennington, Constraint Definition System: A Computer Algebra Based Approach to Solving Geometric Problems, *Computer Aided Design*, **25**(12), 740-750, 1993.

[5] S. C. Chou, *Mechanical Geometry Theorem Proving*, Kluwer, Netherlands, 1987.

[6] S.C. Chou and X.S. Gao, Mechanical Formula Derivation in Elementary Geometries, *Proc. ISSAC-90*, ACM Press, New York, pp. 265–270, 1990.

[7] I. Fudos and C. M. Hoffmann, Constraint-based Parametric Conics for CAD, *Computer Aided Design*, **28**(2), 91-100, 1996.

[8] X. S. Gao, Automated Geometry Diagram Construction and Engineering Geometry, in *Automated Deduction in Geometry*, pp. 226-252, Springer, 1999.

[9] X. S. Gao and S. C. Chou, Solving Geometric Constraint Systems I. A Global Propagation Approach, *Computer Aided Design*, **30**(1), 47-54, 1998.

[10] X. S. Gao and S. C. Chou, Solving Geometric Constraint Systems II. A Symbolic Approach and Decision of Rc-constructibility, *Computer-Aided Design*, **30**(2), 115-122, 1998.

[11] X. S. Gao and W. Q. Yang, Simulation of Complex Linkages, Preprint, Institute of Systems Science, Academia Sinica, 2000.

[12] X. S. Gao, J. Z. Zhang and S. C. Chou, *Geometry Expert*, Nine Chapter Pub., Taiwan, 1998 (in Chinese).

[13] X. S. Gao and C. C. Zhu, Automated Generation of Kempe Linkage and Its Complexity, *J. of Computer Science and Technology*, **14**, 460-467, 1999.

[14] R. Joan-Arinyo and A. Soto, A Correct Rule-Based Geometric Constraint Solver, *Comput. and Graphics*, **21**(5), 599-609, 1997.

[15] R. Joan-Arinyo and A. Soto, Combining Constructive and Equational Geometric Constraint-Solving Techniques, *ACM Trac. on Graphics*, **18**(1), 35-55, 1999.

[16] J. R. Johnson, Algorithms for Polynomial Real Roots Isolation, in *Quantifier Elimination and Cylindrical Algebraic Decomposition*, B. F. Caviness and J.R. Johnson (eds), Springer-Verlag, New Work, pp.269-299, 1998.

[17] R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*, McGraw-Hill Book, 1964.

[18] A. Heydon and G. Nelson, The Juno-2 Constraint-Based Drawing Editor, SRC Research Report 131a, 1994.

[19] C. Hoffmann, Geometric Constraint Solving in R^2 and R^3, in *Computing in Euclidean Geometry*, D.Z.Du and F.Huang(eds), Word Scientific, 1995, pp. 266-298.

[20] C. M. Hoffmann, A. Lomonosov and M. Sitharam, Finding Solvable Subsets of Constraint Graphs, in *LNCS*, NO.1330, pp.163-197, Springer, 1997.

[21] C. M. Hoffmann, Spatial Constraints on Lines, invited speach on ADG'2000, Zurith, Switzerland, September, 2000.

[22] A. B. Kempe, On a General Method of Describing Plane Curves of the nth Degree by Linkwork, *Proc. of L.M.S.*, 213–216, 1876; voir aussi Messenger of Math., T. VI., 143–144.

[23] G. A. Kramer, *Solving Geometric Constraints Systems: A Case Study in Kinematics*, MIT Press, 1992.

[24] K. Kondo, Algebraic Method for Manipulation of Dimensional Relationships in Geometric Models, *Computer Aided Design*, **24**(3), 141-147, 1992.

[25] H. Lamure and D. Michelucci, Solving Geometric Constraints By Homotopy, *IEEE Trans on Visualization and Computer Graphics*, **2**(1):28-34, 1996.

[26] R. S. Latheam and A. E. Middleditch, Connectivity Analysis: a Tool for Processing Geometric Constraints, *Computer Aided Design*, **28**(11), 917-928, 1994.

[27] J. Y. Lee and K. Kim, Geometric Reasoning for Knowledge-Based Design Using Graph Representation, *Computer-Aided Design*, **28**(10), 831-841, 1998.

[28] V. C. Lin, D. C. Gossard and R. A. Light, Variational Geometry in Computer-Aided Design, *Computer Graphics*, **15**(3), 171-177, 1981.

[29] J. Owen, Algebraic Solution for Geometry from Dimensional Constraints, in *ACM Symp., Found of Solid Modeling*, ACM Press, Austin TX, 1991, pp. 397-407.

[30] G. Sunde, Specification of Shape by Dimension and Other Geometric Constraints, *Geometric Modeling for CAD Application*, North-Holland, pp.199-213, 1998.

[31] A. Verroust, F. Schonek and D. Roller, Rule-Oriented Method for Parameterized Computer-Aided Design, *Computer Aided Design*, **24**(10), 531-540, 1992.

[32] W. T. Wu, *Basic Principles of Mechanical Theorem Proving in Geometries*, Science Press, 1984; English Version, Springer-Verlag, 1994.