MM Research Preprints, 48–58
No. 18, Dec. 1999. Beijing

Geometric Constraint Solving with Linkages and Geometric Method for Polynomial Equations-Solving[1)]

Xiao-Shan Gao and Chang-Cai Zhu
Institute of Systems Science, Academia Sinica, Beijing 100080

Abstract. In this paper, we introduce linkages as new tools and show that this tool is complete in certain sense, i.e., any diagram that can be described constructively can be drawn with linkages. As an application, we show that the simplest constrained graph which is beyond the scope of Owen and Hoffmann's triangle decomposition methods can be transformed to pure geometric constructive form. To solve the equations raised from linkage constructions, we propose a geometric method which is based on dynamic locus generation.

1. Introduction

Automated geometry diagram construction or geometric constraint solving (GCS) is the the central topic in much of the current work of developing parametric CAD systems. It also has applications in linkage design, computer vision and computer aided instruction [**?**]. There are four main approaches to GCS: the graph analysis approach [**?**, **?**, **?**], the rule-based approach [**?**, **?**, **?**, **?**], the numerical computation approach [**?**, **?**], and the symbolic computation approach [**?**, **?**, **?**, **?**]. In practice, most people use a combination of these approaches to get the best result.

The graph analysis and the rule-based approaches are also called geometric 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, this is equivalent to construct the diagram sequentially with ruler and compass. This can also be understood as drawing the diagram with geometric tools. But with ruler and compass, we can only draw a small portion of the diagrams. It is well known that using ruler and compass alone, we can describe diagrams whose equation systems are a sequence of triangularized equations of degree less than or equal to two. In [**?**], a new tool, conics, is added to enlarge the solving scope to diagrams that can be described by a sequence of triangularized equations of degree less than or equal to four. In this paper, we will introduce linkages as new tools and show that this tool is complete in certain sense, i.e., any general constructive diagram can be drawn with linkages sequentially. We also give an algorithm to find linkages in a constrained diagram.

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

[1)] This work is supported by "973" project and by the CNSF under an outstanding youth grant(NO. 69725002).

on the loci. This geometric method is based on *dynamic generation* of geometric locus which is widely used in *dynamic geometry* softwares [**?**, **?**, **?**].

Most of the results presented in this paper can be extended to 3D case.

As an application of the method introduced in this paper, we show that the simplest constrained graph which is beyond the scope of Owen and Hoffmann's triangle decomposition methods can be transformed to pure geometric constructive form if linkages are allowed as construction tools.

The rest of this paper is organized as follows. In Section 2, we will show the drawing scope of using linkages as construction tools. In Section 3, we will present the geometric method for solving equations. In Section 4, we will show how to solve the simplest constrained graph.

2. Construction with Linkages

Most of the geometric approaches of GCS is to transform a constrained problem into constructive form with ruler and compass. We generalize this concept as follows. A geometric diagram can be drawn constructively or in *constructive form* if the geometric objects in it can be listed in an order

$$(O_1, O_2, \cdots, O_m)$$

such that each O_i can be determined by $(O_1, \cdots, O_{i-1})$. Since all geometric objects can be treated as functions of points, we may assume without loss of generality that the geometric objects are points. The algebraic equations for a diagram in constructive form is naturally divided into blocks, because the points in the diagrams are introduced one by one and hence the coordinates are introduced at most two by two. In other words, the algebraic equations are as follows.

$$\begin{cases} f_{1,1}(u_1, \cdots, u_m, x_1, x_2) = 0 \\ f_{1,2}(u_2, \cdots, u_m, x_1, x_2) = 0 \\ f_{2,1}(u_1, \cdots, u_m, x_1, x_3, x_4) = 0 \\ f_{2,2}(u_2, \cdots, u_m, x_2, x_3, x_4) = 0 \\ \qquad \vdots \\ f_{l,2}(u_1, \cdots, u_m, x_1, \cdots, x_p) = 0. \end{cases} \tag{2.1}$$

Since the variables are introduced one by one or two by two, we may triangularize (2.1) easily, say, using resultant. Let the triangularized equations be

$$\begin{cases} t_1(u_1, \cdots, u_m, x_1) = 0 \\ t_2(u_2, \cdots, u_m, x_1, x_2) = 0 \\ \qquad \vdots \\ t_p(u_1, \cdots, u_m, x_1, \cdots, x_p) = 0. \end{cases} \tag{2.2}$$

Comparing to constructing diagrams with ruler and compass, the above definition is not constructive. But with ruler and compass, we can only describe a small portion of diagrams. It is well known that using ruler and compass alone, we can describe diagrams whose equation systems are of the form (2.2) and degree$(t_i) \leq 2$. In [**?**], a new tool, conics, is added to enlarge the scope to solve equation systems of form (2.1) and degree$(t_i) \leq 4$. A natural question is: can we add more tools such that the diagram can be drawn with these tools cover all diagrams in constructive form. The answer is positive.

By a *linkage*, we mean a mechanism with one degree of freedom and consisting of links with fixed lengths and rotation joins. One example is the following *four-bar* linkage $ABCD-P$. Since a linkage has one degree of freedom it can be used to draw a locus. The locus of the four-bar linkage in the figure is generated as follows: with points A, B fixed and C rotating on a circle, point P will generate the locus.

Fig. 1. The four-bar linkage and its locus

A diagram can be drawn with linkages constructively if the points in the diagram can be listed in an order

$$(P_1, P_2, \cdots, P_m)$$

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

1. POINT(P): takes a free point P in the plane.
2. ON(P, L): takes a semi-free point P on the locus L of a linkage.
3. INTER(P, L_1, L_2): takes the intersection P of L_1 and L_2 which are the loci of two linkages.

Theorem 2.1 *A diagram is in constructive form iff it can be drawn with linkages.*

Proof. This is valid because of a famous result of Kempe [**?**] which states that we may design a linkage to draw any given algebraic curve $f(x, y) = 0$. In [**?**], 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 f. ∎

Since linkages could be very complicated, it seems that rule-based approaches are more appropriate to transform a constraint system into constructive form. For a rule-based system, like the global propagation method described in [**?**], we may add the following algorithm to find a linkage.

Algorithm 2.2 *Suppose that we need to construct point P_0. We will find a linkage containing P_0. A point is said to be known if it has been already been constructed.*

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

S2 Let S_1 be the set of points such that $\forall P \in S_1, \exists Q \in S_0, s.t. |PQ|$ 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 $S_1 \cup S_0$, n the number of unknown points in $S_1 \cup S_0$, and $m = |S_1 \cup S_0|$.

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

S5 If $d > 2m - 3$, then there is an over-constrained sub-diagram. The algorithm terminates without finding a linkage. Otherwise, let $S_0 = S_1 \cup S_0$ and go to S2.

Example 2.3 *In Fig. 2, the lengths of the nine segments are known. Try to draw the diagram.*

We may first draw triangle ABC. Next, we will determine point P. Since $|CP|$ is known, P is on a circle. Using the above algorithm we can find that point P is on a four-bar linkage $ABUV - P$. Then P is the intersection of a circle and the locus of the four-bar linkage $ABUV - P$.

Fig. 2. Point P is the intersection of two loci

3. Evaluation of Construction Sequence of Linkages

Given a construction sequence $(C_1, C_2, \cdots, C_n)$, by introducing coordinates properly, we may obtain an equation system (2.1). Now we will show how to solve this equation system. Basically, we need to solve two algebraic equations.

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

Please notice that in certain cases, the equation f and g are the equations of the loci for some linkages, which are not explicitly given.

3.1. Algebraic and Numerical Methods

For the concise of description, we did not mention the three more simpler tools: lines, circles, and conics which can be generated by linkages due to Kempe's theorem. In practice, we will still use these geometric tools. One convenience of using them is that we have their equations explicitly. There are other linkages the equation of whose locus can be written explicitly. Of these linkages, a particular one must mention is the four-bar whose locus is an algebraic curve $f(x, y) = 0$ of degree six [**?**].

If the equations in (3.1) are explicitly given, we may use numerical methods such as Newton-Ralphon iteration method or symbolic methods to solve it. Since there are only two equations, we may use Collins' method to compute the resultant r of f and g and then the equation (3.1) is converted into the following form

$$\begin{cases} r(x) = 0 \\ f(x, y) = 0 \end{cases}$$

which can be solved by the roots isolation method presented in [**?**].

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.

3.2. Geometric Method

If the linkage is complex, then it is difficult to find the equation of its locus. In this case, we may use the *locus intersection method* to find the solutions of (3.1). The locus intersection method has two main steps.

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

1. Find a driving point which will move freely on a circle. In Fig. 1, a driving point could be C.
2. Starting from this driving point, find a sequence of constructions with line and circle to construct the whole linkage.
3. Take a value for the driving point, we may compute the coordinates of the points in the linkage. In particular, the coordinates of the *locus point*.
4. Repeat the preceding step, we have a set of coordinates of the locus point. We may use lines or Bezier curves to connect two neighboring points to form the locus.

Find Intersection. After the two loci 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. Notice that there might be more than one solutions.

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.

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.

3.3. Point/Point Distance Constraint Problems

We first use an example to illustrate the method presented in the preceding section.

Example 3.1 *As in Fig. 3, the lengths of the nine segments are known. Try to draw the diagram.*

Fig. 3. A second constrained problem with six points

We may first draw points E, B. Now point D is the intersection of a circle and the locus of a linkage $EBFCA$. Let F be the driving point. The construction sequence for the linkage is as follows

ON(F, CIR($E, |EF|$))
INTER(C, CIR($B, |BC|$), CIR($F, |FC|$))
INTER(A, CIR($E, |EA|$), CIR($C, |CA|$))
INTER(D, CIR($A, |AD|$), CIR($F, |FD|$))

where CIR($B, |BC|$) represents the circle with center B and radius $|BC|$. With the above construction sequence, we may generate the locus for point D when point F moves on CIR($E, |EF|$). Fig. 3 is actually generated in this way by Geometry Expert [**?**] in this way.

Both Examples **??** and **??** contain point-to-point distance only. It is not difficult to check that they are the two smallest possible constraint problems of this kind that can not be solved by ruler and compass construction. We will show that all constraint problems of this kind can be solved by linkages constructively.

Theorem 3.2 *All well or under constrained problems containing point-to-point 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. 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 C be a point such that AC is known. We will construct point C. Since AC is known, it is already on a circle. If BC is also known, we may construct C as the intersection of two circles. Repeat the above process until we cannot go further. Let $\mathbf{S}$ be the set of points constructed in this way, $\mathbf{T}$ 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 not used in $\mathbf{S}$ is $2(n-k)$. Since $|PQ|$ is also used, we have $2(n-k)-1$ constraints left. For the point set $\mathbf{T}$ to form a linkage, we need $2t-1 = 2(n-k)-1$ constraints. Then by Algorithm **??**, $\mathbf{T}$ form a linkage, and P is the intersection of a circle and the locus of this linkage. The remaining points can be treated similarly.

4. Solution of a Smallest Triconnected Constrained Graphs

Hoffmann and Owen's triangle decomposition method is one of the most popular method of GCS. The basic idea is to decompose the constrained graph into three parts, construct the three parts separately, and then assemble the three parts together. Constrained graph that can be solved by these methods are non-triconnnetted graphs [**?**, **?**]. As it is pointed out in [**?**], the simplest constrained graph that cannot be solved with these methods is the following graph. The vertices of the graph could be a point or a line. The edges represent

Fig. 4. Smallest triconnected graph

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

Since each vertex of the constrained graph in Fig. 4 could be a point or a line, we may 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 Fig. 4 is a point. If $V_i = L$, then the i-th position in Fig. 4 is a line. With this notation, Fig. 4 represents 13 types of constrained graphs.

Theorem 4.1 *All the 13 problems can be solved with linkages constructively.*

The following tables give the information on how to solve the 13 problems.

NO	Problem	P/L	Type	R/C	Locus One	Locus Two
1	(PPP,PPP)	Any	Well	No	circle	four-bar
2	(PPP,PPL)	Any	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)	Any	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)	Any	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)	Any	well	Yes	line	circle
12	(PLL,LLL)		over			
13	(LLL,LLL)		over			

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.

Locus One (Two) means the most complicated loci or linkages needed in the construction. The constructed point or (line) will be the intersection of these two loci.

Of the thirteen cases, five use linkages. Case 1 is Example **??**. Case 2 is similar to case 1. Cases 3', 6' and 8' need two new types of linkages.

Fig. 5. Two new four-bar linkages

An *ll-four-bar* linkage consists of two fixed lines u, v and a triangle PAB with fixed shape such that $A \in u$ and $B \in v$. The locus is generated by point P (Fig. 5(a)). This linkage is denoted by $(uvAB, P)$.

An *lc-four-bar* linkage consists of a fixed line l, a fixed circle c and a triangle PAB with fixed shape such that $A \in l$ and $B \in c$. The locus is generated by point P (Fig. 5(b)). This linkage is denoted by $(ucAB, P)$.

By the definition of linkages, a point cannot move on a line. This problem can be solved with the famous Peaucelier linkage which may generate a straight line (Fig. 5(c))).

Fig. 6. Three constraint problems need four-bar linkages

Fig. 6(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 $\mathrm{CIR}(P, |PA|)$ and the locus of the ll-four-bar linkage $(uvBC, A)$.

Cases 6' and 8' need special explanation. Fig. 6(c) is the geometry 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 the 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 the 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 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.

Cases 5,7 and 11 can be drawn with ruler and compasses. They can be solved with the Global Propagation method in [**?**].

- To solve case 5 (Fig. 7(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 [**?**].

- To solve case 6 (Fig. 7(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 $B \rightarrow A$, 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 [**?**].

Fig. 7. Three problems can be solved with ruler and compass

- To solve case 11 (Fig. 7(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.

References

[1] B. Brudelin,*Constructing Three-Dimensional Geometric Objects Defined by Constraints*, in Workshop on Interactive 3D Graphics, ACM, 1986, pp. 111-129.

[2] GABRI Geometry II, Texas Instruments, Dallas, Texas, 1994.

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

[4] X. S. Gao and S. C. Chou, *Solving Geometric Constraint Systems I. A Global Propagation Approach*, Computer-Aided Design, Vol.30, No.1,1 998, pp.47-54.

[5] X. S. Gao and S. C. Chou, *Solving Geometric Constraint Systems II. A Symbolic Approach and Decision of Rc-constructibility*, Computer-Aided Design, Vol.30, No.2, 1998, pp. 115-122.

[6] X. S. Gao and K. Jiang, Geometric Constraint Solving with Conics, this issue of MM-Preprints.

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

[8] 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.

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

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

[11] 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.

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

[13] N. Jakiw, *Geometer's Sketchpad*, User guide and Reference Manual, Key Curriculum Press, 1994.

[14] J. R. Johnson, Algorithms doe 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.

[15] 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.

[16] J. King and D. Schattschneider, *Geometry Turned On*, The Mathematical Association of America, 1997.

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

[18] K. Kondo, *Algebraic Method for Manipulation of Dimensional Relationships in Geometric Models*, Computer Aided Design, Vol. 24, No. 3, 1992, pp. 141-147.

[19] J. Y. Lee and K. Kim, *Geometric Reasoning for Knowledge-Based Design Using Graph Representation*, Computer-Aided Design, Vol.28, No.10, 1998, pp.831-841.

[20] V. C. Lin, D. C. Gossard and R. A. Light, *Variational Geometry in Computer-Aided Design*, Computer Graphics, Vol. 15, No. 3, 1981, pp. 171-177.

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

[22] A. Verroust, F. Schonek, and D. Roller, *Rule-Oriented Method for Parameterized Computer-Aided Design* , Computer-Aided Design, Vol.24, No.10, October,1992, pp.531-540.

[23] D. M. Wang, Geother: a geometry theorem prover. In: Proc. CADE-13, New Brunswick, 1996.

[24] W.T. Wu, Mathematics mechanization, Science Press, 2000.