Combinatorial Classification of 2D Underconstrained Systems (Abstract)

Heping Gao*, Meera Sitharam *†

September 6, 2005

Keywords: Underconstrained systems, Constraint graphs, Mechanisms, Degrees of Freedom, Decomposition of geometric constraint systems, Combinatorial Rigidity, Variational geometric constraint solving.

1 Introduction and Significance

Approaches for characterizing, classifying, decomposing, solving and navigating the solution set of generically wellconstrained geometric constraint systems have been studied extensively, both in 2D [16, 8, 9, 10, 15, 13] and in 3D [18, 4]. Significant progress has also been made in understanding generically overconstrained systems [14, 7]. However, while the study of underconstrained systems is acknowledged to be important and crucial for both the classical CAD, Robotics and newer molecular modeling applications of geometric constraint solving, this study is still at a nascent stage partly due to the following reason. In the process of solving and navigating the solution set of well and overconstrained systems, the combinatorial and algebraic ingredients are naturally demarcated: there are clearly defined questions that concern only their constraint graphs. These questions are, for example, related to characterization, classification, and recursive decomposition of constraint graphs and their subgraphs. Answering these questions is necessary for efficient solving and solution set navigation. Moreover, one obtains combinatorial measures of complexity for solving a generic or worst case system that corresponds to a particular constraint graph. However, the combinatorial and algebraic ingredients in the process of solving underconstrained systems are not, to date, clearly demarcated.

Organization. In Section 2 we recall the problem **(*)** of "solution existence determination and solution set navigation of geometric constraint systems." We dwell at some length on commonly accepted formalizations of these problems for generically well-constrained systems in order to motivate their extension to generically underconstrained systems, but point out the difficulties in formulating such an extension. In Section 3 we first lay out a program of study by isolating a set of well-defined combinatorial questions that are natural and meaningful and, in our opinion, necessary, to make progress on **(*)** for underconstrained systems. We motivate these with simple examples. We then state preliminary results that initiate this program of study.

2 Statement of the Problem

We restrict ourselves here to 2D *distance constraint systems* $(G, \mathbf{d})$ which consist of a *distance constraint graph* G whose vertices represent point objects and the edges represent pairwise distance constraints, and where the distance values associated with the edges give the tuple $\mathbf{d}$. We refer the reader to [4] for definitions of *generically well, over and underconstrained systems* - these properties depend only on the constraint graphs of these systems and we will hence refer to *well, over and underconstrained graphs*.

The *solution existence* problem for a constraint system is to determine (constructively) if there exists (and find) a 2D realization or embedding of the points on the Euclidean plane so that the distance constraints are satisfied.

In the case of well-constrained systems, the solution set is finite, and we generally assume the problem of *navigating the solution set* to mean: giving *all* such solutions or realizations. For well-constrained graphs

*University of Florida; Work supported in part by NSF Grants EIA 02-18435, CCF 04-04116.

†*corresponding author: sitharam@cise.ufl.edu*

(generically well-constrained systems) the two problems are generally believed to have the same worst case time complexity. Intuitively, for every well-constrained graph G, there is some corresponding system or distance tuple $\mathbf{d}_G^*$ which has the largest number m_G of realizations; and given any algorithm A that solves or finds one realization, there is some distance tuple $\mathbf{d}$ for which A would have to check at least m_G candidate realizations before it finds one.

Furthermore, in the case of generically well-constrained systems, most algorithms - for both the existence and the navigation problems - deal with two distinct subproblems. The first subproblem **(a)** is to obtain a so-called *Decomposition-Recombination* plan [1], which is a purely combinatorial object that involves only the constraint graph G. A DR-plan D_G for a well-constrained graph G can be viewed as an efficient combinatorial *description* or roadmap of a set S_G of *candidate* realizations that is meaningful for all generically well-constrained systems $(G, \mathbf{d})$ that correspond to G.

The second subproblem **(b)** involves the distance tuple $\mathbf{d}$: the problem is to effectively *search the candidate set* S_G and obtain one realization $r_{G,\mathbf{d}}$ (in the case of the existence problem) or obtain the entire set $R_{G,\mathbf{d}}$ of realizations (in the case of the navigation problem). Both these involve finding all real solutions to the subsystems corresponding to the nodes in the DR-plan. Intuitively, the candidate solution set S_G is a combinatorial description that captures the set of all possible realizations.

For example, in subproblem **(a)**, if the DR-plan D_G decomposes G into k child subgraphs $C_1, \ldots, C_k$, which are combined by a set C of *active* edges to form G, then the candidate solution set S_G described by this DR-plan is recursively defined as $S_C \times (S_{C_1} \times \ldots \times S_{C_k})$. In subproblem **(b)**, recursively searching the S_{C_i} corresponds to finding all real solutions $R_{C_i,\mathbf{d}}$ to the subsystems $C_i, \mathbf{d}$. Then, each element of $R_{C_1,\mathbf{d}} \times \ldots R_{C_k,\mathbf{d}}$ yields a different subsystem corresponding to the edges C: finding the real solutions to m of these systems corresponds to m searches through S_C.

It is important to notice the tradeoff relationship between the complexities of the two subproblems **(a)** and **(b)**. For example, one correct, but trivial DR-plan would be a single node representing G itself, i.e., one solution to the first subproblem would be to not decompose G at all. While this reduces the complexity of the subproblem **(a)**, this does not yield a particularly efficient description of the candidate solution set as it does not offer any assistance for the subproblem **(b)**. On the other hand, an *optimal* DR-plan, is an efficient description as it minimizes the size of the largest subsystem for which the subproblem **(b)** finds real solutions. This is a combinatorial measure of *size* of the DR-plan that captures the complexity of subproblem **(b)**. Finer such combinatorial measures of algebraic complexity can be found in [6]. In this context, the complexity of subproblem **(b)** is defined for the worst case tuple $\mathbf{d}$ and is hence treated as a combinatorial property dependent only on the constraint graph G. This complexity could be associated with a specific type of algorithm for finding the realization, given the DR-plan D_G: the definition of the DR-plan's *size* would vary accordingly. However, for most common notions of size, finding an optimally sized DR-plan is NP-hard [17] even for the 2D distance constraint systems being considered here. In other words, the efficiency of description of the candidate solution set is directly related to the complexity of the subproblem **(b)** and inversely related to the complexity of subproblem **(a)**.

The above discussion shows that we can extract the combinatorial content of **(b)** and incorporate it into **(a)** to get unified combinatorial questions useful for answering problem **(*)** for well-constrained graphs G. Many questions falling into these categories have been studied in the literature.

(w1*) Characterize interesting subclasses of well-constrained graphs G that have a a DR-plan of size at most $b_1(|G|)$ and relate these classes to each other. Give an efficient algorithm to recognize the graphs in such a class. For example, triangle decomposable graphs [5], Henneberg 1 graphs [16], and quadratically solvable graphs [10, 11, 12] are such classes. In general, choosing a particular algorithm or method A for finding DR-plans, characterize the class C_A of well-constrained graphs G for which A finds a DR-plan of size at most $b_1(|G|)$. Further, given 2 such algorithms A and A' how do the corresponding classes C_A and $C_{A'}$ relate to each other?

(w2*) For general graphs G, or graphs G in a class defined in **(w1*)**, give an algorithm that finds such a DR-plan in time at most $b_2(|G|)$.

There are clear difficulties in extending such formalizations meaningfully for generically underconstrained systems. As a start, while the existence problem is still meaningful as stated above, the navigation problem is not, since the solution set is not finite. Moreover, the 2 subproblems of the existence problem are not meaningful: unlike in the case of well-constrained graphs, the subproblem **(a)** of finding a good DR-plan is not particularly useful for subproblem **(b)**. This is because any DR-plan of an input underconstrained graph stops with a complete set of maximal well-constrained subgraphs [2, 17]: even an optimal DR-plan leaves the "underconstrained part" of the graph untouched. It does not provide an efficient description of a candidate solution set from which a realization is significantly easier to obtain.

3 Contribution

We first formulate extensions of the combinatorial questions in Section 2 that are meaningful for underconstrained graphs. We motivate these with simple examples. Next we briefly state preliminary new results that initiate this program of study.

3.1 Formulating meaningful combinatorial questions about underconstrained graphs

As pointed out in Section 2, while a good DR-plan is an efficient combinatorial description of the solution set of generically well-constrained systems, this is not directly true for generically underconstrained systems.

A natural way to leverage the concept of a DR-plan is to *parametrize* the (infinite) solution set of a *k-degree-of-freedom (k-dof)* underconstrained system (also called *mechanism*) $(G, \mathbf{d})$. The parameters are k independent distances that are not explicitly specified, i.e., an *appropriately chosen set U of edges* not present in G such that $G \cup U$ is well-constrained. Specifically, the DR-plan of $G \cup U$ would serve as an efficient description of the (finite) candidate solution set for a generically well-constrained system $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$, for any tuple of distance values $\mathbf{d_U}$ for the edge set U. The *valid* set of distance values $\mathbf{d_U}$ - for which a realization exists for $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ - is the *projection* of the real solution variety of the system $(G, \mathbf{d})$ onto the distance variables given by U.

The choice of this set U of edges, i.e., the choice of parametrization or projection, gives rise to a first set of combinatorial questions about underconstrained graphs G that address one part of **(*)**.

(u1*) Given an efficient algorithm to choose a set of edges U such that $G \cup U$ is a well-constrained graph, i.e., a so-called *completion* of G [18]. Some care needs to be taken to ensure that U does not cause overconstrained subgraphs. An simple, efficient method for obtaining such completions for arbitrary underconstrained graphs G - from a so-called *complete* DR-plan for G [17] - is given in [18].

(u2*) We can now directly extend questions **(w1*,w2*)** by combining them with **(u1*)** above: characterize and relate classes of underconstrained graphs for which a set of edges U can be chosen such that the resulting well-constrained graph $G \cup U$ has a DR-plan of small size. We can add on the requirement that the DR-plan be found quickly by some specified algorithm. Here "small" and "quickly" are appropriately specified. In particular, combining **(w1*)** and **(u1*)**, we can ask to characterize and relate classes of underconstrained graphs G for which there is a set of edges U for which $G \cup U$ has a small DR-plan, is triangle decomposable, Henneberg 1, quadratically solvable etc.

For one such class, [3] gives an algorithm for finding such edges. However, we observe that, without further qualification, the picture is not rosy for general 2D distance constraint graphs. We can find underconstrained graphs of arbitrary size that are 1-degree-of-freedom mechanisms (have 1 extra dof more than well-constrained graphs) such that for any (singleton) completion edge e the well-constrained graph $G \cup \{e\}$ is effectively not decomposable, i.e., any DR-plan has size $\Omega(|G|)$. In fact, Figure 1 shows even a *planar* graph with this property: two n-cycles are connected as shown, with 1 additional vertex connected to alternate vertices in the inner cycle and a second additional vertex connected to alternate vertices in the outer cycle. The largest 1-dof proper subgraphs are the quadrilaterals: no larger 1-dof proper subgraph exists.

The above combinatorial problems address only a portion of the problem **(*)** for underconstrained graphs. A *complete* combinatorial description C_G of the underconstrained solution set for G would consist not only of the following:
(i) the set of completion edges U; and

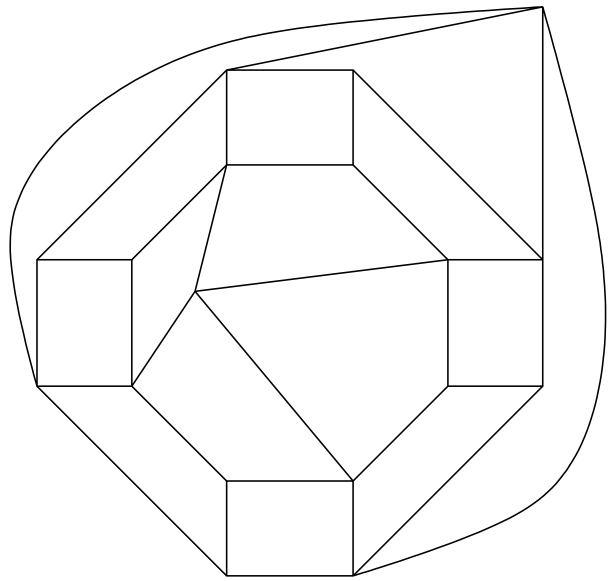

Figure 1: Arbitrarily large graph G such that for any additional edge e, $G \cup \{e\}$ only has DR-plans of size atleast $\Omega(|G|)$

(ii) a DR-plan $D_{G,U}$ for $G \cup U$ which serves as a description of the (finite) candidate solution set $S_{G,U}$ for a generically well-constrained system $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$, for any tuple of distance values $\mathbf{d_U}$ for the edge set U. In addition, C_G it should also contain
(iii) a description $V_{G,U}$ of a candidate *finite set of distance values* $T_{G,U}$ from which it can be efficiently determined whether there is a *valid* value of d_U for which a realization exists for $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ and from which such a value of d_U can be efficiently found. In other words, $T_{G,U}$ captures *at least one* valid value of of d_U for which $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ has a realization (provided $(G, \mathbf{d})$ has a realization).

Note. The above definiton of candidate value set differs crucially from the candidate solution set described by a DR-plan for well-constrained systems such as in (ii) following way: the latter captures all realizations, but the former is guaranteed to capture only one. Specifically, the set of all valid values for the $\mathbf{d_U}$ (i.e., the projection mentioned above) is infinite for generic underconstrained systems $(G, \mathbf{d})$ but $T_{G,U}$ is finite. Another point of difference is that the DR-plan is a canonical or uniform type of description, although special algorithms for finding DR-plans may result in special types of DR-plans. At this early stage in the investigation of underconstrained systems, it seems prudent to allow somewhat more leeway in the type of the description $V_{G,U}$, although it is likely that a canonical type of description will evolve that encompasses all of these types.

Recall that the size and other measures of the DR-plan were required to identify and/or rule out trivial DR-plans and to capture the complexity with which a (all) realization(s) could be obtained from the DR-plan (recall the the complexity of subproblem **(b)** in Section 2). Now we use similar measures which we, as in Section 2, collectively refer to as *size*: these capture the efficiency of the description $V_{G,U}$. As in Section 2, these measures, in turn, capture the complexity of *searching* $T_{G,U}$, in other words, determining - from the description $V_{G,U}$ - whether there is a value of $\mathbf{d_U}$ for which a realization exists for $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ and finding such a value of $\mathbf{d_U}$. As in Section 2, this complexity is understood to be worst case, over all $\mathbf{d}$, and could be associated with a specific type of algorithm for processing the description $V_{G,U}$: the definition of its *size* or efficiency varies accordingly.

In general, when possible, we favor *separable* descriptions $V_{G,U}$, i.e., those from which existence of a realization and a valid value of d_U can be determined *without* having to determine a (partial) realization i.e., without solving for position values for any of the points using the DR-plan for $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ from (ii). Such descriptions permit items (ii) and (iii) above to be dealt with separately.

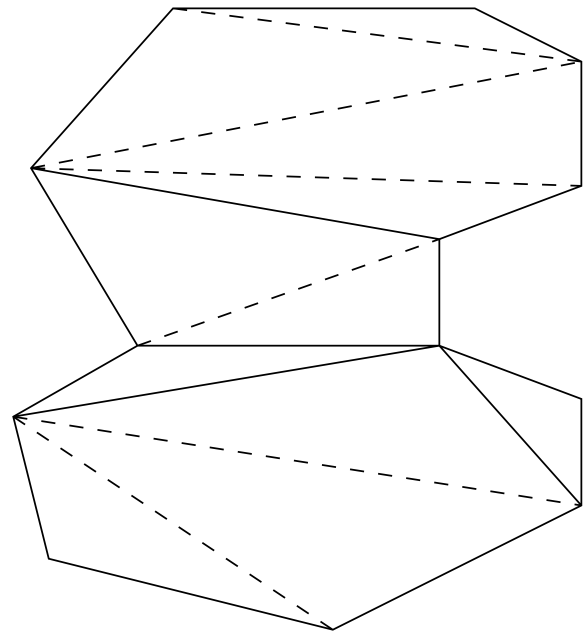

Figure 2: A well-triangulable graph: dotted are completion edges in U

3.1.1 Example

We give a simple example that illustrates the concepts discussed above concerning Item (iii) of the description C_G. Consider an underconstrained graph G which is an n-cycle with bridge edges or diagonals that do not cross. See Figure 2.

The graph can be made well-constrained by triangulating it. We call such graphs *well-triangulable*. This additional set U of edges ensure a simple version of Henneberg 1 DR-plan $D_{G,U}$ for the graph $G \cup U$, taking care of Parts (i) and (ii) of the description C_G. In this case, this DR-plan also directly yields a *linear polytope* description $V_{G,U}$ consisting of a set of triangle inequality expressions relating variables representing $\mathbf{d}$ and $\mathbf{d_U}$. For specific values of $\mathbf{d}$, $V_{G,U}$ defines a polytope in $\mathbf{R}^{|U|}$ containing exactly the valid values of $\mathbf{d_U}$. Technically, the description $V_{G,U}$ gives a a generic such polytope whose extreme points could be taken as the finite candidate set $T_{G,U}$ of valid values for $\mathbf{d_U}$: they are guaranteed to contain at least one valid value for which $(G\mathbf{d})$ has a realization. (In fact, in this case, the polytope describes the generically infinite set of valid values - the projection of the real solution variety of $(G\mathbf{d})$ onto the distance variables defined by U). This description $V_{G,U}$ is separable since given specific values of $\mathbf{d}$, we can determine whether or not this polytope is empty (i.e., determine whether a valid value of $\mathbf{d_U}$ exists and find it) without actually finding a (partial) realization of $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$. ♣

Thus, Item (iii) of the description C_G gives rise to further combinatorial questions about underconstrained graphs, necessary for answering **(*)**. Just as question **(u2*)** concerned Item (ii) of the description C_G, namely the difficulty of obtaining an efficient DR-plan $D_{G,U}$, the following question concerns Item (iii): the difficulty of obtaining an efficient, (possibly separable) description $V_{G,U}$.

(u3*) Characterize and relate classes of underconstrained graphs G for which there is a set of edges U, for which $G \cup U$ has an efficient or small size description $V_{G,U}$ (of some specified type). We can add on the requirement that $V_{G,U}$ be found quickly by some specified algorithm. Here, "small size" and "quickly" are appropriately specified.

We can further combine **(u2*)** and **(u3*)** to ask the following.

(u4*) Characterize and relate classes of underconstrained graphs G for which there is a set of edges U, for which $G \cup U$ has a small size description $V_{G,U}$ (of some specified type), *and* a small size DR-plan D_G. We can add on the requirements that $V_{G,U}$ and D_G be found quickly by some specified algorithms. Here, "small size" and "quickly" are appropriately specified.

3.2 Preliminary results

We now state (informally and without proof) some preliminary results we have obtained related to Questions **(u1*)-(u4*)** given above.

- We give a characterization of a large subclass of graphs for which there is a set of edges U, for which $G \cup U$ has an efficient description $V_{G,U}$ satisfying a natural generalization of the linear polytope property given in the example above. We further give an efficient algorithm for finding such a description, including the set U. We additionally generalize and quantify the notion of separable, by which the new type of description turns out to be *nearly separable*: we give an efficient method, given specific values $\mathbf{d}$ of determining existence of (and finding) a valid value $\mathbf{d_U}$ for which a realization of $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ exists without determining most of the realization.

- We characterize and relate two subclasses of 1-dof underconstrained graphs obtained by removing 1 edge from triangle decomposable, well-constrained graphs. These classes are defined by the existence of 2 types of efficient descriptions $V_{G,U}$, which we call respectively *backward propagation* description and *extremal distance* description. These two could be viewed as further generalizations of the linear polytope description, however, the extremal distance description is not known to be separable. Since these are 1-dof graphs, the set U is a singleton set consisting of a single edge and the set of valid values of $\mathbf{d_U}$ is generically a set of intervals in $\mathbf{R}$. The candidate value sets $T_{D,U}$ corresponding to these 2 descriptions are guaranteed to contain all the end points of these intervals. Interestingly these descriptions and their efficiency and size are based on not just one but several DR-plans for $G \cup U$. As in the previous result, we give efficient algorithms for finding such descriptions (including the set U); and for determining existence of (and finding) valid values $\mathbf{d_U}$ for which a realization of $(G \cup U, < \mathbf{d}, \mathbf{d_U} >)$ exists. (Since the extremal distance description is not known to be separable, realizations are found in the process).

References

[1] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Decomposition of geometric constraints systems, part i: performance measures. *Journal of Symbolic Computation*, 31(4), 2001.

[2] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Decomposition of geometric constraints systems, part ii: new algorithms. *Journal of Symbolic Computation*, 31(4), 2001.

[3] Robert Joan-Arinyo and Antoni Soto-Riera and S. Vila-Marta and Josep Vilaplana-Pasto. Transforming an under-constrained geometric constraint problem into a well-constrained one. In *Symposium on Solid Modeling and Applications 2003*, pages 33–44, 2003.

[4] Jack E. Graver and Brigitte Servatius and Herman Servatius. *Combinatorial Rigidity*. Graduate Studies in Math., AMS, 1993.

[5] I. Fudos and C. M. Hoffmann. Correctness proof of a geometric constraint solver. *Intl. J. of Computational Geometry and Applications*, 6:405–420, 1996.

[6] Jörg Peters and JianHua Fan and Meera Sitharam and Yong Zhou. Elimination in generically rigid 3d geometric constraint systems. In *Proceedings of Algebraic Geometry and Geometric Modeling,Nice, 27-29 September 2004*, pages 1–16. Springer Verlag, 2005.

[7] C Hoffman and M Sitharam and B Yuan. Making constraint solvers more useable: the overconstraint problem. *CAD*, 36(4):377–399, 2004.

[8] B. Bruderlin and R. Roller ed.s. *Geometric constraint solving and applications*. Springer-Verlag, 1998.

[9] X. S. Gao and S. C. Chou. Solving geometric constraint systems. I. a global propagation approach. *CAD*, 30:47–54, 1998.

[10] X. S. Gao and S. C. Chou. Solving geometric constraint systems. II. a symbolic approach and decision of rc-constructibility. *CAD*, 30:115–122, 1998.

[11] John C. Owen and Steve C. Power. The nonsolvability by radicals of generic 3-connected planar graphs. In *Automated Deduction in Geometry*, pages 124–131, 2002.

[12] John C. Owen and Steve C. Power. Are all 3-connected generic constriant configurations of points on a plane non-radical. In *Automated Deduction in Geometry*, 2004.

[13] I. Fudos. *Geometric Constraint Solving*. PhD thesis, Purdue University, Dept of Computer Science, 1995.

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

[15] G. Kramer. *Solving Geometric Constraint Systems*. MIT Press, 1992.

[16] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Engrg. Math.*, 4:331–340, 1970.

[17] Andrew Lomonosov. Graph and Combinatorial Analysis for Geometric Constraint Graphs. Technical report, Ph.D thesis, Univ. of Florida, Gainesville, Dept. of Computer a nd Information Science, Gainesville, FL, 32611-6120, USA, 2004.

[18] M Sitharam. Graph based geometric constraint solving: problems, progress and directions. In D. Dutta and R. Janardhan and M. Smid, editor, *AMS-DIMACS volume on Computer Aided Design*, 2005.