Directed Graphs, Decompositions, and Spatial Linkages

Offer Shai *
Faculty of Engineering, Tel-Aviv University. shai@eng.tau.ac.Il
Adnan Sljoka†
Department of Mathematics and Statistics
York University, 4700 Keele Street
Toronto, ON M3J1P3, Canada and
Walter Whiteley‡
Department of Mathematics and Statistics
York University, 4700 Keele Street
Toronto, ON M3J1P3, Canada

August 4, 2012

Abstract

The decomposition of a linkage into minimal components is a central tool of analysis and synthesis of linkages. In this paper we prove that every pinned d-isostatic (minimally rigid) graph (grounded linkage) has a unique decomposition into minimal strongly connected components (in the sense of directed graphs), or equivalently into minimal pinned isostatic graphs, which we call d-Assur graphs. We also study key properties of motions induced by removing an edge in a d-Assur graph - defining a stronger sub-class of strongly d-Assur graphs by the property that all inner vertices go into motion, for each removed edge. The strongly 3-Assur graphs are the central building blocks for kinematic linkages in 3-space and the 3-Assur graphs are components in the analysis of built linkages. The d-Assur graphs share a number of key combinatorial and geometric properties with the 2-Assur graphs, including an associated lower block-triangular decomposition of the pinned rigidity matrix which provides modular information for extending the motion induced by inserting one driver in a bottom Assur linkage to the joints of the entire linkage. We also highlight some problems in combinatorial rigidity in higher dimensions ($d \geq 3$) which cause the distinction between d-Assur and strongly d-Assur which did not occur in the plane.

*Supported by a grant from ISF (Israel)
†Supported in part under a grant from NSERC (Canada).
‡Supported by a grant from NSERC (Canada).

Key Words: isostatic framework, decomposition, Assur graphs, directed graph, algorithms

1 Introduction

The decomposition of a system of constraints into small basic components is an important tool of design and analysis. In particular, the decomposition of a mechanical engineering linkage into minimal mechanical components is a central tool of analysis and synthesis of mechanisms [2, 13, 20, 21]. Figure 1 illustrates the analysis: the initial plane linkage (Figure 1(a)), is transformed into a flexible pinned framework (Figure 1(b)); designating one of the links to be a driver, then adding an extra bar, or pinning the end of the driver to the ground, this becomes an isostatic pinned framework (Figure 1(c,d)) (see §4).

The focus of this paper is the study of specific decompositions of such associated pinned isostatic frameworks in dimensions 2 and 3. However, the theorems and definitions will be generalized whenever possible to all dimensions $d \geq 2$. We emphasize that this decomposition goes further into the framework than the decomposition of a flexible framework such as (b) into rigid components.

For plane linkages, this decomposition has been an important tool in the mechanical engineering literature over some decades [2, 13]. Our central goal with this paper is to extend the previous decomposition for plane pinned isostatic frameworks [20, 21] to pinned isostatic frameworks in d-space [22]. In developing this extension, we present some key additional properties of the Assur decomposition in the plane (and in d-space) in terms of lower block-triangular decompositions of the associated pinned rigidity matrix (§3). In the analysis, we also draw some new connections to the theory of strongly connected decompositions of directed graphs and their associated algorithms.

Section 2 highlights a unique decomposition of constraint graphs, using only properties of the graphs of the constraints captured in directions on the edges. A key first step is generating a directed graph for the constraints, which in §3 will be directed towards the 'ground' of the linkage. This directed graph is then decomposed into strongly connected components (components in directed cycles in the directed graph), using a standard combinatorial result which can be implemented using various algorithms, such as Tarjan [25]. Overall, the strongly connected decomposition is presented as an acylic graph with condensed nodes for the strongly connected components (Figure 2). In this decomposition, the strongly connected components can be recognized visually as separated in the original directed graph by directed cut-sets.

As a useful invariant property of this decomposition, we note that two equivalent orientations of a given (multi)-graph (orientations with an assigned out-degree for each vertex) will produce the same strongly connected components, in the strongly connected decomposition. This is relevant to techniques in §3 in this paper and to the algorithms such as the pebble game, which include some choices of orientations for edges, and note that these choices do not alter the

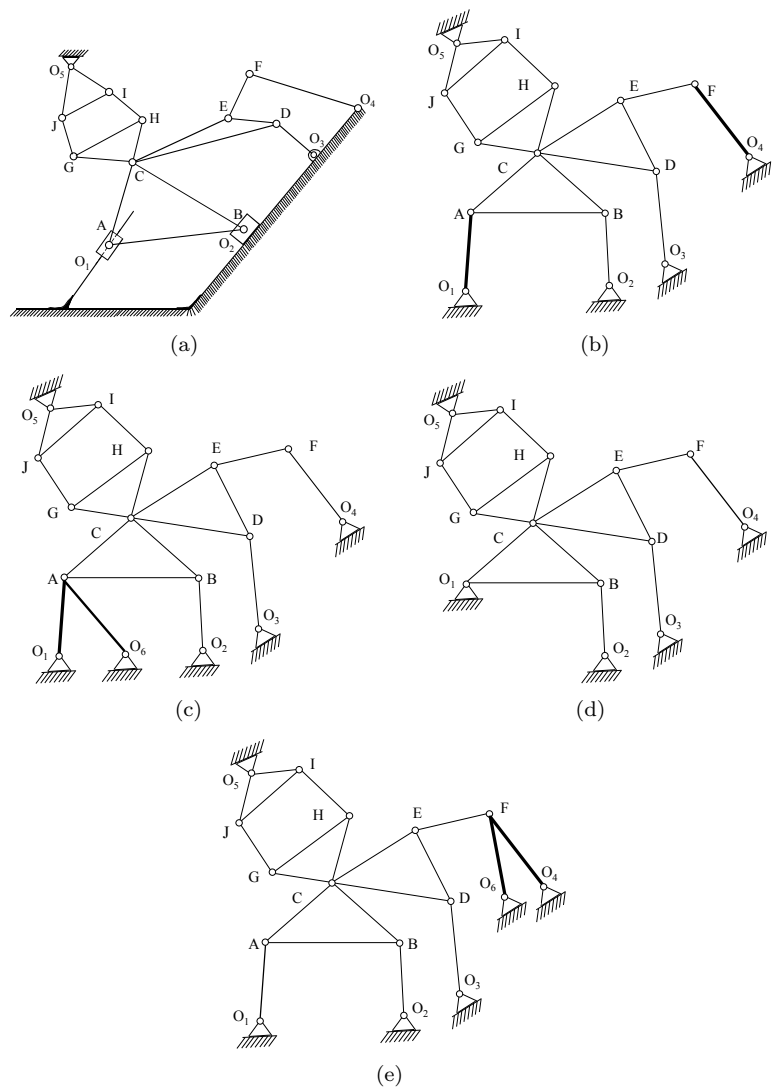


Figure 1: A plane linkage (a) is transformed into a flexible pinned framework (b). Assuming the chosen driver is the link from A to the ground, with one added bar (c) or by pinning the end of the driver to the ground (d) this becomes an isostatic pinned framework. If we specify the link from F to the ground as a designated driver, with one added bar, this results in an alternate pinned isostatic framework (e).

decomposition.

In Section 3, we show that, for an appropriate orientation (a d-directed orientation) this strongly connected graph decomposition gives a decomposition

of a pinned isostatic graph which coincides several other key decompositions of a pinned isostatic graph: (i) with a block triangular decomposition of the pinned rigidity matrix (§3.2) and (ii) an associated decomposition into minimal pinned isostatic graphs (§3.4). For pinned linkages in the plane, this shared decomposition coincides with the 2-Assur decomposition in [20] .

Here we extend these decompositions to all higher dimensions $d \geq 2$. The minimal pinned isostatic graphs in this decomposition are called d-Assur, and coincide with the plane 2-Assur components of the previous work. This connection to the pinned rigidity matrix also provides a way to generate a directed graph from the original undirected graph, so that we can apply the strongly directed decomposition of §2, if directions were not already supplied by other analysis. This explicit connection to lower triangular matrices is new in the plane, as well.

In Section 4, we consider extensions to dimension d of a further key property of 2-Assur graphs: removal of any single edge, at a generic configuration, gives a non-trivial motion at all inner (unpinned) vertices. This property represents the desirable property that a driver replacing this edge causes all parts of the mechanism to be in motion. While this property follows for minimal pinned isostatic graphs in the plane, it is a stronger property in higher dimensions. We use this added property to define the strongly d-Assur graphs in all dimensions, as a restricted subclass of d-Assur graphs. This distinction between d-Assur graphs and strongly d-Assur graphs adds another view on the complexity of obtaining a combinatorial characterization of generic rigidity 3-space or higher. We will offer examples which illustrate this distinction.

This capacity to break the overall analysis of a pinned framework down into smaller pieces, and recombine them efficiently, is the central contribution of these decompositions to the analysis and synthesis of mechanical linkages [13, 15, 19]. We conjecture that other special geometric properties, such as those studied in [21], are also inherited by these strongly d-Assur graphs in higher dimensions.

In §5, we mention some further directions and extension of these techniques and decompositions to the alternate body-bar frameworks and related mechanisms. In keeping with the general work on rigidity in higher dimensions, these alternative structures have good theories with d-Assur again coinciding with strongly d-Assur, and good fast algorithms for these structures. Since many built 3-D linkages have this modified structure, and have important applications to fields such as mechanical engineering and robotics, Assur body-bar frameworks and decomposition algorithms are currently being explored.

We note that the strongly connected decomposition of a directed graph is included in the code of current Computer Algebra Systems such as Maple, Mathematica and SAGE. However, we alert the reader that the lower block-triangular decomposition for a matrix implemented in these CAS systems is weaker than the decomposition we present here for constraint graphs. We comment further on this in §3.2.

This general focus on decompositions of constraints into irreducible com-

ponents resembles other work done in electronics. For example, Kron's diakoptics [11], describes a method that decomposes any electrical network into sub-networks which can be solved independently and then joined back obtaining the solution of the whole network. The overall goal of simplifying analysis (and synthesis) through decomposition into pieces which can be analyzed and then recombined through known boundary connections is part of a wide array of methods in systems engineering [4].

In particular, the directed graph decomposition presented in this paper connects to general work on decomposition of CAD systems [16, 17, 31]. In the latter systems the problem of decomposing a cluster configuration into a sequence of clusters [9, 10] corresponds to the decomposition of isostatic graphs into d-Assur graphs. Some of that work also uses the strongly connected decomposition highlighted here as well as other related methods.

This paper initiates a program to develop and apply the unique decomposition of pinned linkages in d-space into minimal components - the d-Assur graphs. We present an algorithm which decomposes a linkage, as well as some fundamental properties associated with the decomposition. We also highlight further work in progress, as well as key unsolved problems in the rigidity theory for frameworks in d-space which this analysis brings back into focus in terms of pinned structures.

Aknowledgment The authors thank the reviewer for a number of valuable comments which were of great assistance to us.

2 Decomposition of Pinned Directed Graphs

In combinatorial rigidity theory, the rigidity of a given framework is a property of an underlying undirected graph. However, directed graphs come up in several important applications and algorithms in rigidity theory. For example, in the control theory of formations of autonomous agents, the constraints on distances between agents are represented as a directed graph [5, 8]. The output of the fast pebble game algorithm [12, 22, 23] as part of the algorithmic verification of critical counts and, implicitly, for decomposing rigid and flexible regions in the framework graph is also represented by a directed graph. Directed graphs also appear in the practice of mechanical engineering when synthesizing and analyzing linkages. We will use the directed graphs and their decompositions given in this section to obtain the d-Assur decomposition of a pinned isostatic framework in d-space in §3.

In this section we state some basic definitions and background from the theory of directed graphs and present the decomposition of a directed pinned graph into strongly connected components and develop a few simple extensions to confirm the invariance of the decomposition under some natural variations of the directed graph.

2.1 Strongly connected component decomposition

A *graph* $G = (V, E)$ has a vertex set $V = \{1, 2, ..., n\}$ and an edge set E, where E is a collection of unordered pairs of *vertices* called the *edges* of the graph. We define a *direction assignment* $\overrightarrow{G}$ to graph $G = (V, E)$ as a pair of maps init:$E \to V$ and ter:$E \to V$ assigning to every edge e an *initial vertex* init(e) and a *terminal vertex* ter(e). The edge e is said to be *directed out* of init(e) and *into* ter(e). We refer to $\overrightarrow{G}$ as a *directed graph* associated with G. A directed graph may have multiple directed edges between the same two vertices, v and w. We will call such graphs directed *multi-graphs*.

Remark. For simplicity most definitions and proofs are presented in terms of the vocabulary of simple graphs. However, every proof in this paper will also apply to multi-graphs. The functions init:$E \to V$ and ter:$E \to V$ work well in this multi-graph setting, where an edge is no longer uniquely represented by an ordered pair. □

A *cycle* of a graph G is a subset of the edge set of G which forms a path such that the start vertex and end vertex are the same. A *directed cycle* is an oriented cycle such that all directed edges are oriented in the same direction along the cycle. A directed graph $\overrightarrow{G}$ is *acyclic* if it does not contain any directed cycle. The *degree* or *valence* of a vertex v is the number of edges that have v as an endvertex. The *out-degree* of a given vertex in a directed graph is the number of edges directed out of that vertex. A vertex which has out-degree 0 is called a *sink*. Sink vertices will be *pinned vertices* and directed graphs that have some pinned vertices will be called *pinned graphs*.

A directed graph is called *strongly connected* if and only if for any two vertices i and j in $\overrightarrow{G}$, there is a directed path from i to j and from j to i. The *strongly connected components* of a graph are its maximal strongly connected subgraphs. It is maximal in the sense that a subgraph cannot be enlarged to another strongly connected subgraph by including additional vertices and its associated edges (see Fig 2 (b), (c)). One can determine the strongly connected components of a directed graph using the $O(|\mathrm{E}|)$ Tarjan's algorithm [25] which is implemented in several computer algebra packages, such as Maple, Mathematica and SAGE.

To illustrate the decomposition process of a pinned directed graph, we first condense (ground) all pinned vertices into a single ground vertex (sink) (Figure 2(a, b)). (More detailed definitions of pinned graphs and grounding will be presented in the next section.) After we have identified the strongly connected components, as a simplification, we will ignore the orientation of edges within the strongly connected components, and just keep the orientation of edges between the components (Figure 2 (c)). In the final step, we apply condensation to each strongly connected component by contracting it to a single vertex, obtaining an acyclic graph and a partial order (Figure 2(d)). We have schematized the partial order, so that there are no multiple edges appearing between any components.

So far, we were not concerned how we obtained the directed graphs. In the next section, we will find directed graphs for our linkages, and use those for our decomposition. There are also inductive constructions [24, 30], which are currently being further investigated, and there are additional fast algorithmic techniques for generating the desired directed graphs for linkages (e.g. the pebble game) [22].

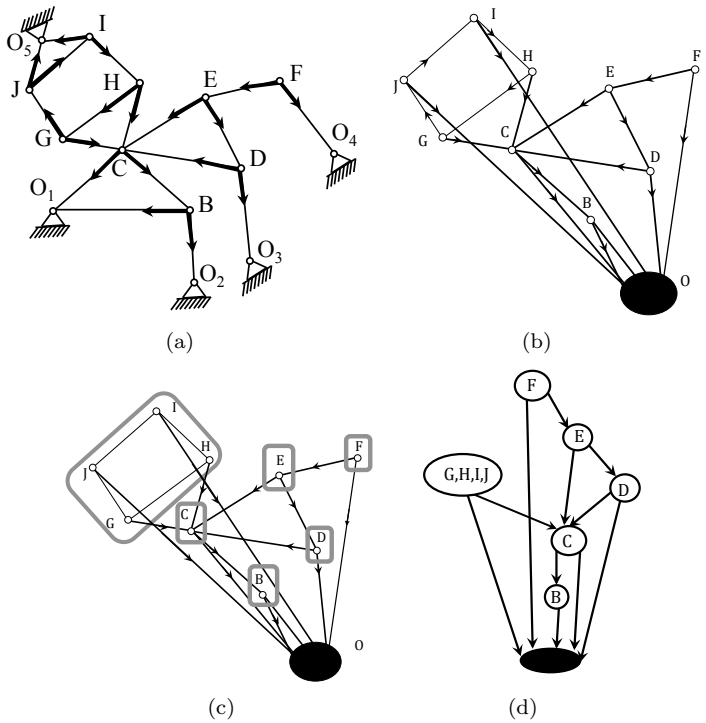


Figure 2: Decomposition of a directed pinned graph: The directed pinned graph (a) has the pinned vertices condensed to the ground (sink) (b), with the corresponding strongly connected decomposition (c) and the partial order (d).

2.2 Equivalent orientations of a graph

We now show that choices in orientations of edges which conserve a fixed outdegree of each of the vertices do not alter the decompositions. This is a useful observation for the proofs in the next section and will assist the verification of algorithms for generating the directed graphs and related decompositions [22]. For example, all plays of the pebble game algorithm with the same number of final pebbles at the corresponding vertices will give the same strongly connected decompositions [12, 23]. We thank Jack Snoeyink for conversations which clarified these arguments.

Definition 2.1 *Given a multi-graph G and two direction assignments $\overrightarrow{G^1}$ and $\overrightarrow{G^2}$. We say that $\overrightarrow{G^1}$ and $\overrightarrow{G^2}$ are* equivalent orientations *on G if the corresponding vertices have the same out-degree.*

Such alternate orientations appear when comparing assigned direction assignments by hand (common engineering practice) vs those attained by algorithms. We confirm we obtain same decomposition regardless how we obtained the equivalent direction assignment.

Lemma 2.1 *Given two equivalent orientations $\overrightarrow{G^1}$ and $\overrightarrow{G^2}$ on G, then the two orientations differ by reversals on a set of directed cycles.*

Proof. Pick an edge $e = (u, v)$ in $\overrightarrow{G^1}$ that is oppositely directed in $\overrightarrow{G^2}$. So, in $\overrightarrow{G^1}$ edge e is incoming at vertex v. Assume there are k outgoing edges at v in $\overrightarrow{G^1}$ Because v has same out-degree k in $\overrightarrow{G^2}$, and this edge is reversed to be outgoing in $\overrightarrow{G^2}$ there must exist an out-going edge from v in $\overrightarrow{G^1}$, say $f = (v, w)$, that is oppositely directed in $\overrightarrow{G^2}$ (Figure 3(a,b)). We walk out of v along f in $\overrightarrow{G^1}$ (Figure 3(c)).

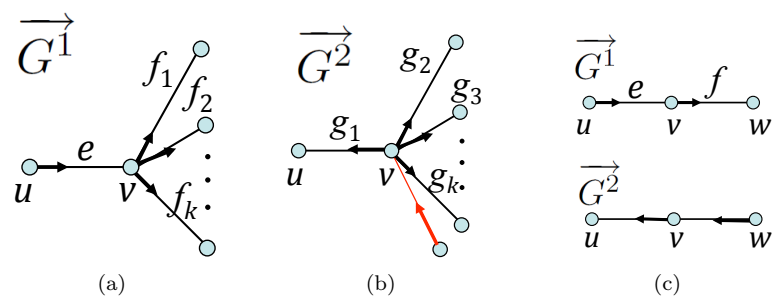


Figure 3: Since vertex v has same out-degree in G^1 and in G^2, there exists an edge in G^1 (a) that is oppositely oriented in G^2 (b). We show such a selection f in (c,d).

As we enter a new vertex, we again apply this same argument. This identifies a directed path in $\overrightarrow{G^1}$ that is oppositely directed in $\overrightarrow{G^2}$. As there is only a finite collection of edges that have opposite direction, we walk along this directed path until we come back to some vertex on this path, identifying a directed cycle in $\overrightarrow{G^1}$ that has an opposite orientation in $\overrightarrow{G^2}$.

We reverse the orientation of such a cycle from the orientation in $\overrightarrow{G^1}$ towards the orientation in $\overrightarrow{G^2}$ This decreases the number of edges in $\overrightarrow{G^1}$ that are oppositely directed from $\overrightarrow{G^2}$ (see Figure 4). We continue reversing identified cycles, until all edges are directed following $\overrightarrow{G^2}$. □

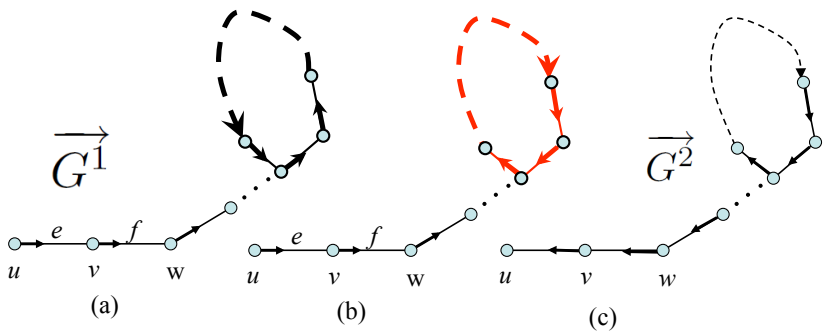


Figure 4: Locating a directed path in G^1 that is oppositely oriented in G^2 (a), reversing a first cycle in G^1 (b), which now has the same orientation as in G^2 (c).

We end with a valuable corollary to this lemma:

Corollary 2.2 *Given two equivalent orientations $\overrightarrow{G^1}$ and $\overrightarrow{G^2}$, the strongly connected components of the decompositions are the same.*

Proof. From Lemma 2.1, $\overrightarrow{G^1}$ and $\overrightarrow{G^2}$ differ by reversal of set of directed cycles. As cycle reversals do not change the strongly connected components the decompositions are the same. □

3 Decomposition of Pinned d-isostatic Graphs

Paper [20] described the 2-Assur decomposition of a pinned generically isostatic graph in the plane. Here we will nuance this decomposition with the connections to the directed graph decomposition of §2, and directly extend the central decomposition to all dimensions. This extended decomposition carries significant properties of the 2-Assur decomposition, with one key exception which we examine in §4. Because of applications in mechanical engineering, we are primarily interested in the Assur decompositions in 2-space and 3-space. For this reason all the figures will be either 2 or 3-dimensional.

Recall that the 2-Assur graphs shared a suite of equivalent properties [20, 21]. We will begin with the extended definitions in d-space, collectively defining the d-Assur graphs (which coincide with Assur graphs for dimension 2). All the results in this section apply to d-Assur graphs. Accordingly, when we just speak of Assur graphs in this section, it should be understood that we are referring to d-Assur graphs. For mathematical completeness, whenever possible we state all the theorems and definitions in d-dimensional space.

In §4 we will explore one further property of 2-Assur graphs that does not always extend to higher dimensions - the response of inner vertices to removing

one edge.

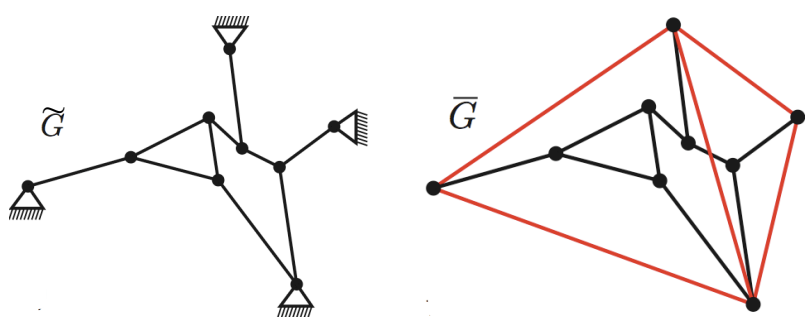
3.1 Pinned d-isostatic graphs and pinned rigidity matrix

In the remainder of the paper, we will focus on pinned frameworks and the analysis of their associated graphs and directed graphs. The rational for this focus on pinned frameworks is as follows:

> Given a framework associated with a linkage, we are interested in its internal motions, not the trivial ones. Following the mechanical engineers, we pin the framework by prescribing, for example in 2-space, the coordinates of the endpoints of some edges, or equivalently, by fixing the position of the vertices of some rigid subgraph (see Figure 5).

We call these vertices with fixed positions *pinned*, the other vertices are *inner*. (Inner vertices are sometimes called *free* or *unpinned* in the literature on linkages.) Edges between pinned vertices are irrelevant to the analysis of a pinned framework. More formally, we denote a *pinned framework* as $(\widetilde{G}, p) = ((I, P; E), p)$, where $\widetilde{G} = (I, P; E)$ is a pinned graph, I is the set of inner vertices, P is the set of pinned vertices, E is the set of edges, where each edge has at least one endpoint in I, together with an assignment p of points in d-space to the vertices of $\widetilde{G}$ (i.e. p is a fixed configuration (embedding) of V into $\mathbb{R}^d$).

In the previous papers [20, 21], some of the rigidity analysis was routed through grounding the pinned graphs through an isostatic framework for the ground (Figure 5). Here we will give an alternative version that directly analyzes the pinned rigidity matrix, which only has columns for the inner vertices, corresponding to the possible variations in their positions, if the linkage is flexible with the pin fixed.



(a) (b)

Figure 5: The framework (a) is pinned 2-isostatic because framework (b) is 2-isostatic.

For a pinned framework $(\widetilde{G}, p) = ((I, P; E), p)$ we define the $|E| \times d|I|$ d-space *pinned rigidity matrix*, which unlike the regular rigidity matrix, only has

columns for the inner vertices:

$$\mathbf{R}(\widetilde{G},p) \;=\; \begin{array}{c} \\ \{i,j\} \\ \\ \{i,k\} \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccccccc} & & & i & & & & j & & & \end{array} \\ \begin{pmatrix} & \ddots & & & & \vdots & & & & \iddots & \\ 0 & \dots & 0 & (p_i-p_j) & 0 & \dots & 0 & (p_j-p_i) & 0 & \dots & 0 \\ & \vdots & & & & \vdots & & & & \vdots & \\ 0 & \dots & 0 & (p_i-p_k) & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & \iddots & & & & \vdots & & & & \ddots & \end{pmatrix} \end{array},$$

where $i,j \in I$ and $k \in P$. Note that this matrix has d columns for each inner vertex.

The solutions U to the equation: $\mathbf{R}(\widetilde{G},p) \times U^{tr} = 0$ are called *infinitesimal motions* of the pinned framework. A framework $(\widetilde{G},p)$ is *pinned d-rigid* if the only infinitesimal motion is the zero motion or equivalently, if pinned rigidity matrix $\mathbf{R}(\widetilde{G},p)$ has full rank $d|I|$. A framework is *pinned d-independent* if the rows of $\mathbf{R}(\widetilde{G},p)$ are independent. A framework $(\widetilde{G},p)$ is *pinned d-isostatic* if it is both *pinned d-rigid* and *pinned d-independent*. In particular, if the framework is pinned d-isostatic then $|E| = d|I|$ and the pinned rigidity matrix is a square matrix.

If we vary the configuration p over all of $\mathbb{R}^{d|I|+d|P|}$, then the pinned rigidity matrix achieves some maximal rank, and this maximal rank occurs for an open dense subset of $\mathbb{R}^{d|I|+d|P|}$ - the *generic rank* of the *d-space pinned rigidity matrix* for the graph. In particular, for a pinned isostatic graph, the configurations that drop the rank are captured by a non-zero polynomial in variables for the vertices. We conclude that if one configuration p achieves the full rank $d|I|$, then almost all configurations (all points in this open dense subset) achieve this rank, and we call all configurations in the open dense subset *generic* or *regular*.

We say a pinned framework $(\widetilde{G},p)$ is *generic* if the configuration p of the joints p is generic. (A special subset of the generic configurations is those whose coordinates satisfy no algebraic equations - also a dense set, but not an open set.)

Theorem 3.1 *Given a pinned graph $\widetilde{G} = (I,P;E)$, the following are equivalent:*

1. *There exists a pinned d-isostatic realization p of $\widetilde{G}$ in d-space;*
2. *For all placements $p|_P$ of the pins P in generic position in d-space, and all generic positions of vertices in I the resulting pinned framework is pinned d-isostatic;*
3. *$\widetilde{G} = (I,P;E)$ is generically pinned d-rigid and $|E| = d|V|$;*
4. *$\widetilde{G} = (I,P;E)$ is generically pinned d-independent and $|E| = d|V|$.*

Proof. These statements are a simple translation of standard results for isostatic frameworks to the pinned d-isostatic setting [30]. □

Results in the plane, and other recent work suggests the *conjecture* that (2) can be refined to require only that the pins are in general position within a d-hyperplane (a line in $\mathbb{R}^2$, a plane in $\mathbb{R}^3$).

We call any graph $\widetilde{G} = (I, P; E)$ satisfying the equivalent conditions of Theorem 3.1 *pinned d-isostatic*. A *d-Assur graph* is a *minimal* pinned d-isostatic graph. By minimal we mean there is no proper subgraph which is also a pinned d-isostatic graph. In §4, we will define a strongly d-Assur graph as a d-Assur graph with the added property that removal of any edge puts all inner vertices in motion. We will look at the difference between the two types of graphs in more detail in Section 4, but note no distinction between 2-Assur graphs and strongly 2-Assur graphs (see also Section 4), so they will always be called 2-Assur.

3.2 Directed graph decomposition of the pinned rigidity matrix

We now use the d-space pinned rigidity matrix to generate a special type of directed graph for any pinned graph in d-dimension with $|E| = d|V|$. We then connect the corresponding directed graph decomposition from §2 and to a block decomposition of the pinned rigidity matrix, which will be central to the rest of the paper.

Matching the shape of the pinned rigidity matrix, we develop *d-directed orientations of the pinned graph*: one in which each inner vertex has out-degree d and each pinned vertex has out-degree 0 - a sink in the directed graph.

Proposition 3.2 *Every pinned d-isostatic graph has a d-directed orientation, with all inner vertices of out-degree d and all pinned vertices of out-degree* 0.

Proof. Take the determinant of the $d|V|\times d|V|$ pinned rigidity matrix. We know the determinant of the square matrix is non-zero if and only if the framework is infinitesimally rigid. For this determinant to be non-zero, there must be a non-zero term in the Laplace expansion of the determinant of this matrix, in $d \times d$ blocks following the d columns for each inner vertex. Take any such non-zero term. This will associate d rows (edges) with each inner vertex i, and we direct these d edges out from vertex i (Figure 6). This gives the desired d-directed orientation. □

We can now apply the strongly connected graph decomposition from §2 to decompose the pinned d-isostatic graph, with the pinned vertices identified in the directed graph decomposition. Each strongly connected component is extended to include the outgoing edges from the component. (In §3.3, we show that these extended components are minimal pinned d-isostatic graphs.) We make the connections through a block-triangular decomposition of the pinned

d-rigidity matrix (Figure 6). In this decomposition, a permutation of the vertices and the edges of the pinned graph generates a lower triangular matrix.

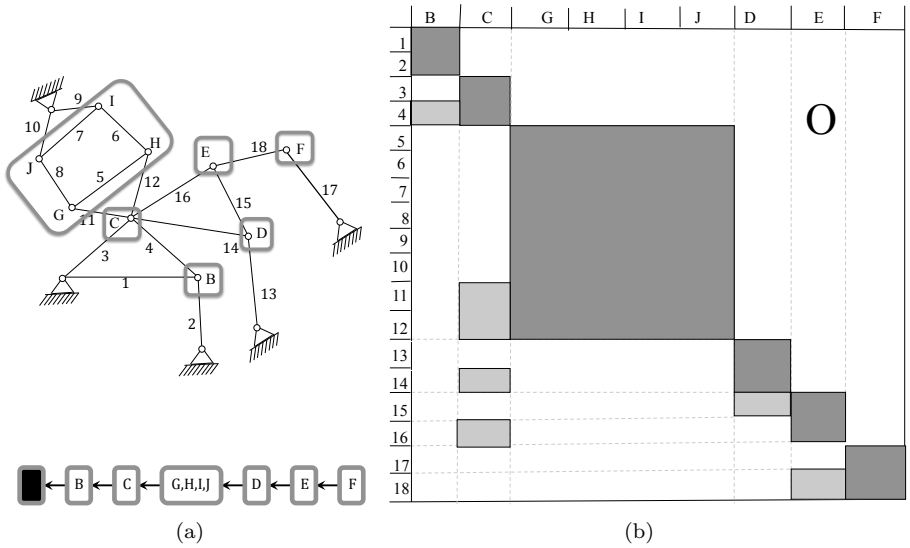


Figure 6: The pinned 2-isostatic framework in (a), with the directed graph decomposition of Figure 2 as a partial order, extended to a linear order generates a block triangular matrix in (b).

Theorem 3.3 *For a pinned d-isostatic graph with a pinned d-directed orientation (all inner vertices of out-degree d and all pinned vertices of out-degree 0), the strongly connected decomposition (with the pinned vertices identified) coincides with the block-triangular decomposition of the pinned rigidity matrix with a maximal number of diagonal blocks, for some linear order of the blocks extending the partial order of the directed graph decomposition.*

Proof. Given a pinned d-isostatic graph with a pinned d-directed orientation, we apply the techniques of §2 to the pinned graph with the pinned vertices identified to obtain a strongly connected decomposition. If this has more than one component, then focus some bottom strongly connected component A, extended with its edges to the ground. With a permutation of the rows and a permutation of the columns, we place these vertices and edges at the top left of the matrix. The rest of the rows and columns for a second block B (that may not be strongly connected). (For notation in the matrices below, we assume with $i, j \in A$ and $\ell, m \in B$ and k is a pinned vertex, where B is the component.) This gives the form:

	$\dots$	i	$\dots$	j	$\dots$	$\dots$	ℓ	$\dots$	m	$\dots$
	$\ddots$		$\vdots$		$\iddots$	$\ddots$		$\vdots$		$\iddots$
$\{i,j\}$	$\dots 0$	$p_i - p_j$	$0 \dots 0$	$p_j - p_i$	$\dots$	$0 \dots$	0	$\dots$	0	$\dots 0$
$\{i,k\}$	$\dots 0$	$p_i - p_k$	$0 \dots$	0	$\dots$	$0 \dots$	0	$\dots$	0	$\dots 0$
	$\iddots$		$\vdots$		$\ddots$	$\iddots$		$\vdots$		$\ddots$
	$\ddots$		$\vdots$		$\iddots$	$\ddots$		$\vdots$		$\iddots$
$\{i,\ell\}$	$\dots 0$	$p_i - p_\ell$	$0 \dots$	0	$\dots$	$\dots 0$	$p_\ell - p_i$	$0 \dots 0$	0	$0 \dots$
$\{\ell,m\}$	$\dots 0$	0	$\dots$	0	$\dots$	$\dots 0$	$p_\ell - p_m$	$0 \dots 0$	$p_m - p_\ell$	$0 \dots$
	$\iddots$		$\vdots$		$\ddots$	$\iddots$		$\vdots$		$\ddots$

or simply:

$$\mathbf{R}(G) = \begin{array}{|c|c|} \hline \mathbf{R}(A) & 0 \\ \hline X & \mathbf{R}(B) \\ \hline \end{array}$$

We have an initial block decomposition. Repeating this for each of the blocks up the acyclic strongly connected decomposition, we find a diagonal matrix block for each component of the decomposition, and the matrix has the desired block-triangular form (with $B_2, ...B_3$ as the remaining blocks):

$$\mathbf{R}(G) = \begin{array}{|c|c|c|c|c|} \hline \mathbf{R}(A) & 0 & 0 & \dots & 0 \\ \hline X_{12} & \mathbf{R}(B_2) & 0 & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \ddots & \vdots \\ \hline X_{1n} & X_{2n} & X_{3n} & \dots & \mathbf{R}(B_n) \\ \hline \end{array}$$

Note that the acyclic strongly connected decomposition graph only gives a partial order. The matrix block-triangular decomposition imposes a linear order which extends this partial order. The linear order will not be unique, as it is possible that some blocks may be incomparable in the partial order - which will show up here as some off diagonal blocks with $X_{ij} = 0$. (See Figure 6 where component $(GHIJ)$ could be moved anywhere to the right in the linear order.)

Conversely, if we have such a block-triangular decomposition, then the corresponding Laplace decomposition of the determinant will produce a pinned d-directed orientation in which all edges are either oriented within the block, or directed towards vertices in blocks to the left. In short, we have a decomposition with an acyclic decomposition graph extracted from the linear order of the blocks.

If there is a further decomposition from the d-directed graph, then this will show up as a further block-triangular decomposition with more components, and vice versa. $\square$

Recall that any two pinned d-directed orientations are equivalent, so they give the same decomposition in the algorithm of §2. Therefore in the block-triangular decomposition of the pinned rigidity matrix, the linear order extends this partial order. In particular, each other non-zero term in the Laplace block expansion gives an equivalent pinned d-directed orientation, and an equivalent block-triangular decomposition of the pinned rigidity matrix. Moreover, it is easy to check that any pinned d-directed orientation of a graph can be used to create a non-zero term in the Laplace decomposition of the determinant. There is a bijection between these d-directed orientations and the non-zero terms of the Laplace expansion.

Remark. We note that CAS programs such as Maple, Mathematica and SAGE have 'strongly connected decompositions of matrices' into lower triangular form [14]. However, these are not the same - only analogous. The 'directed graph' used in their algorithm comes from treating the rows and columns of the square matrix as indexed by the same vertex set, and using any non-zero entry $a_{i,j}$ as a directed edge (i, j). The strongly connected decomposition then corresponds to a simultaneous permutation of the rows and columns, so that we go from A to the similar matrix $[D] = P[A]P^{-1}$. In our process, we have identified a different graph with fewer vertices ($|V|$ instead of $d|V|$) and different edges. Our shift allows distinct permutations of the rows and columns, generating a more general $[D'] = P[A]Q$. With more possible permutations, we get all the decompositions that flow from the standard CAS code - and more. In particular, a simple permutation of the rows of the pinned rigidity matrix will change a matrix which Maple declares does not decompose, into one that Maple notices does decompose! $\square$

3.3 d-Assur graphs and minimal components

We now focus on the minimal components in these decompositions of a pinned d-isostatic graph $\widetilde{G} = (I, P; E)$. We will verify that these are minimal pinned d-isostatic components of $\widetilde{G}$, i.e. the *d-Assur graphs*. We also verify some key properties of these graphs.

Theorem 3.4 (d-Assur Graphs) *For a pinned d-isostatic graph $\widetilde{G}$ the following are equivalent:*

1. *the graph contains no proper pinned d-isostatic subgraphs;*
2. *the graph is indecomposible for some (any) d-directed orientation;*
3. *the pinned rigidity matrix has no proper block triangular decomposition.*

Proof. Theorem 3.3 shows the equivalence of (2) and (3). It remains to show that (1) is equivalent to these.

Assume the graph is not minimal and there is a proper pinned isostatic subgraph $\widehat{G}$. We will show the pinned rigidity matrix decomposes. If we permute the inner vertices and all the edges of $\widehat{G}$ to the upper left corner of the pinned rigidity matrix, the rest of these rows from $\widehat{G}$ are 0 and the remaining columns and rows form a second block. This gives a block triangular decomposition of the pinned rigidity matrix. The contrapositive says that if the pinned rigidity matrix does not have a proper block triangular decomposition, then the pinned isostatic graph is minimal.

Conversely, if the pinned rigidity matrix for $\widetilde{G}$ has a proper block triangular decomposition, then it is clear the upper left block represents the inner vertices and the edges (including the ground edges) of a proper pinned isostatic subgraph $\widehat{G}$. The contrapositive confirms that if the graph $\widehat{G}$ is minimal then the pinned rigidity matrix does not decompose. □

Any graph $\widetilde{G}$ which satisfies one of these three equivalent properties will be called d-Assur. We can use the concept of a minimal pinned isostatic graph (d-Assur graph) to build up a third route to the decomposition of a larger pinned d-isostatic graph into d-Assur graphs. We follow the same track used for 2-Assur graphs in [20].

We just recall the outline of the process, since the decomposition will match what we already have from Theorem 3.3. (See Figure 7.) The ground is the bottom layer. Find a minimal d-isostatic pinned subgraph. This will be above the ground component. Combine that into the ground, and seek another minimal d-isostatic pinned subgraph. This will be a component above all components to which its ground edges attach. Repeat until all vertices are in some component. This is the *d-Assur decomposition* of the pinned d-isostatic graph $\widetilde{G}$.

3.4 Summary Assur Decomposition

We now pull these connections into a summary theorem about the three equivalent ways to decompose a pinned d-isostatic graph.

Theorem 3.5 (d-Assur Decomposition) *Given a pinned d-isostatic graph $\widetilde{G}$, there is a d-directed orientation of $\widetilde{G}$. With any such d-directed orientation, the following decompositions are equivalent:*

1. *the d-Assur decomposition of $\widetilde{G}$;*
2. *the strongly connected decomposition into extended components associated with the d-directed orientation (with all pins identified);*
3. *the block-triangular decomposition of the pinned rigidity matrix into a maximal number of components for some linear order extending the partial order of (i) or equivalently (ii).*

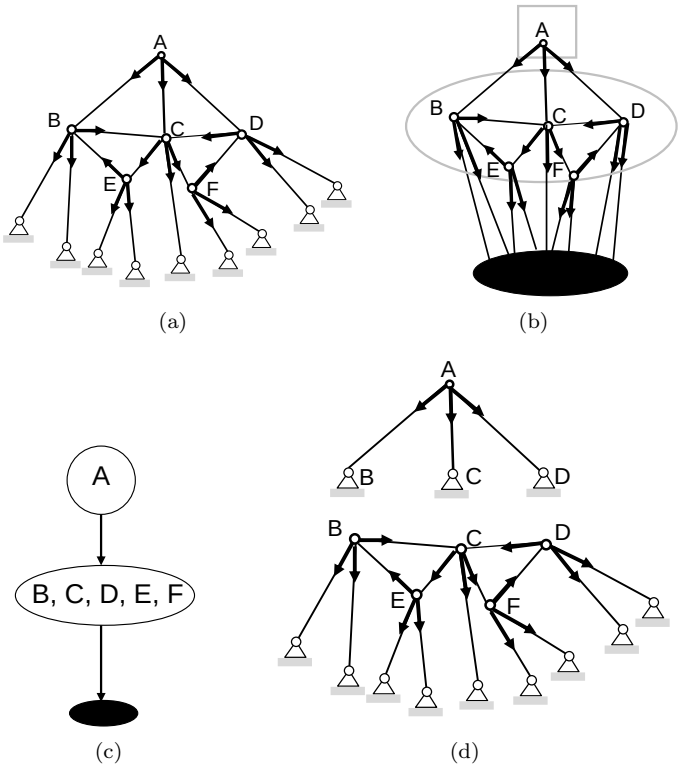
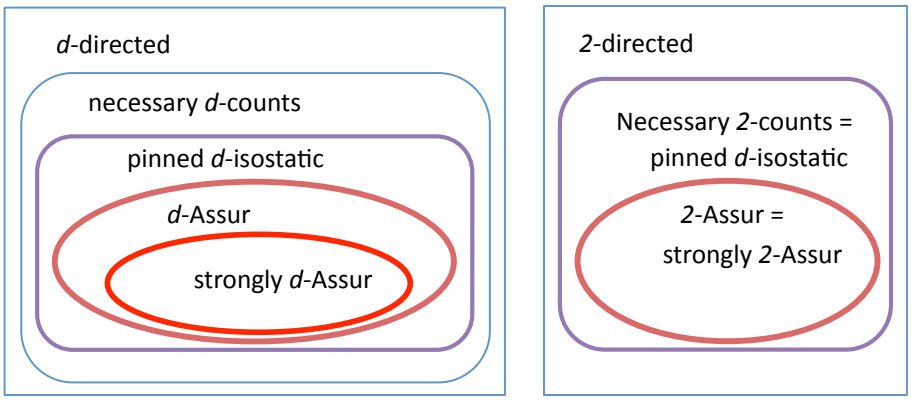


Figure 7: A pinned 3-isostatic framework (a) has a unique decomposition into 3-Assur graphs through condensing the pins to ground (rigid region) (b) and creating a partial order (c) which also corresponds to a scheme of Assur graphs (d).

Proof. Theorem 3.3 give the equivalence of (2) and (3). The equivalence of (1) and (2) follows from Theorem 3.4. The construction process for the d-Assur decomposition above guarantees that for two components A, B A is above B in the d-Assur partial order if and only if A is above B in the d-directed partial order. $\square$

3.5 Pinned d-counts and insufficiency of d-directions

While having a d-directed orientation is a necessary condition for being d-isostatic, it is not sufficient. These orientations are also connected to some basic necessary counting conditions - conditions that have been proven sufficient in the plane, but fail to be sufficient in 3-space. Since there have been efforts to transform this necessary condition into a sufficient condition, we give a few key observations and examples to clarify the connections.



(a) (b)

Figure 8: (a) shows the general nesting of conditions for dimension d, including strongly d-Assur graphs, while (b) shows the simplified nesting in the plane.

Theorem 3.6 (Necessary Pinned d-Counts) *Given a pinned d-isostatic graph $\widetilde{G} = (I, P; E)$, the following properties hold*

(1) $|E| = d|I|$.

(2) for all subgraphs $\widetilde{G}' = (I', P'; E')$

(i) $|E'| \leq d|I'|$ *if* $|P'| \geq d$,

(ii) $|E'| \leq d|I'| - \binom{d+1-k}{2}$ *if* $|P'| = k$, $1 < k < d$,

(iii) $|E'| \leq d|I'| - \binom{d}{2}$ *if* $|P'| = 1$ *and*

(iv) $|E'| \leq d|I'| - \binom{d+1}{2} = d|I'| - (d + \binom{d}{2})$ *if* $P' = \emptyset$.

Proof. The count (1) is just the condition for the pinned d-rigidity matrix to be square. For the subgraphs, the count 2(i) is again the subgraph count on the submatrix of the columns of I', whose failure would guarantee a row dependence.

In 2(ii) $\binom{d+1-k}{2}$ is the count of remaining rotational degrees of freedom with k points pinned. For instance, with $d = 3$ and $k = 2$ ($|P'| = 1$), there must be a $\binom{2}{2} = 1$ remaining degree of freedom, corresponding to a rotation of the entire framework about a line passing through these two points. Note that pinning one point eliminates all d translational degrees of freedom. As we continue to pin additional vertices we are eliminating more rotational degrees of freedom. With k points fixed, $k - 1$ dimensional space is fixed. The rotational axis in any dimension d is of dimension $d - 2$ (i.e. point in plane, line in $3D$, plane in $4D$, etc). The possible rotations occur in the orthogonal complement to the rotational axis, which is a 2-dimensional space. With k points fixed from the remaining $d - (k - 1)$ space we choose all such possible 2-dimensional

orthogonal complements which gives us the remaining $\binom{d-(k-1)}{2} = \binom{d+1-k}{2}$ space of rotations (DOF). For 2 (iii), there is a space of $\binom{d}{2}$ rotations fixing the single vertex in dimension d, and for 2(iv) there is a $\binom{d+1}{2}$-space of trivial motions (combination of d translations and $\binom{d}{2}$ rotations), so the bounds on the count of edges is the maximum number of independent rows. □

Note that the formula for the subgraph count in 2(ii) of Theorem 3.6 is also the correct form for all $k \geq 0$.

In dimension 2, the necessary pinned 2-counts become (*Pinned Plane Framework Conditions* for $\widetilde{G} = (I, P; E)$) [20]:

1. $|E| = 2|I|$ and
2. for all subgraphs $\widetilde{G}' = (I', P'; E')$ the following conditions hold:
   (i) $|E'| \leq 2|I'|$ if $|P'| \geq 2$,
   (ii) $|E'| \leq 2|I'| - 1$ if $|P'| = 1$, and
   (iii) $|E'| \leq 2|I'| - 3$ if $P' = \emptyset$.

For completeness, the Necessary pinned 3-counts are:

(1) $|E| = 3|I|$.

(2) for all subgraphs $\widetilde{G}' = (I', P'; E')$
   (i) $|E'| \leq 3|I'|$ if $|P'| \geq 3$,
   (ii) $|E'| \leq 3|I'| - 1$ if $|P'| = 2$, and
   (iii) $|E'| \leq 3|I'| - 3$ if $|P'| = 1$.
   (iv) $|E'| \leq 3|I'| - 6$ if $P' = \emptyset$.

A useful observation is that all pinned graphs in dimension d that satisfy the necessary pinned d-counts are d-directed:

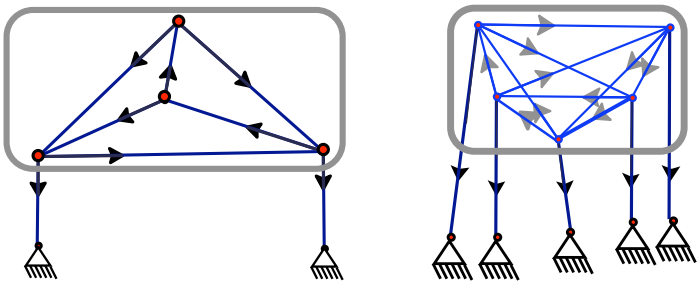
Theorem 3.7 *If a pinned graph $\widetilde{G} = (I, P; E)$ satisfies the Necessary pinned d-counts for some d, then it has a d-directed orientation with all inner vertices of out-degree d and inner vertices of out-degree* 0.

One proof is to use the pebble game algorithm operations, essentially playing a $d|V|$ pebble game on both inner edges and ground edges, giving us the desired d-directed orientation of the graph [22].

Having stated the necessary pinned d-counts, we can now illustrate that d-directed orientation is not a sufficient condition for being d-isostatic. In Figure 9 (a) we have an example of a graph which is 2-directed but not 2-isostatic. It is not 2-isostatic as it fails the necessary pinned 2-counts. Such a graph will have a non-zero term in the Laplace expansion, and a block decomposition. However

the determinant is zero, overall, with cancellation among non-zero terms. The example in Figure 9 (b) shows a similar example in 3-space, it has a proper 3-directed orientation, although it is not pinned 3-isostatic. These two examples confirm that having a 2 and 3-directed orientation and being indecomposible does not capture the very basic subgraph counting conditions of Theorem 3.6.

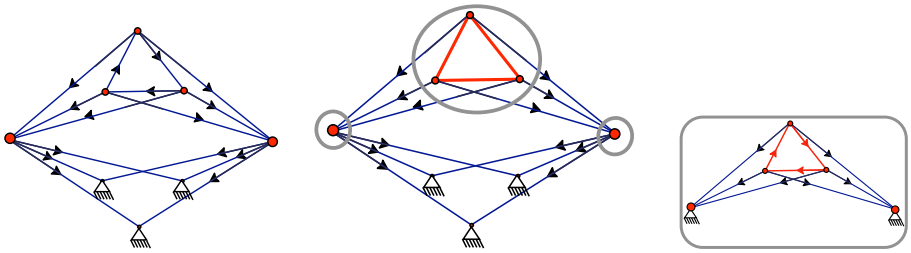
In Figure 10 we have another example which has a 3-directed orientation (a), although it is not pinned 3-isostatic (see remark below for further discussion on this special example). This graph is decomposible (b), but what is interesting is that one of the components (c) now visibly fails the necessary pinned 3-count though is still 3-directed.



(b)

(b)

Figure 9: A pinned graph which is 2-directed, indecomposible but not 2-isostatic (the subgraph in the grey box is overcounted - it does not satisfy the necessary pinned 2-counts). (a). A pinned graph which is 3-directed, indecomposible but not 3-isostatic (the subgraph in grey box does not satisfy the necessary 3-counts) (b).



(a) (b) (c)

Figure 10: A pinned graph graph which is 3-directed (a), has a strongly connected decomposition (b) but the top extended component (c) is not 3-isostatic.

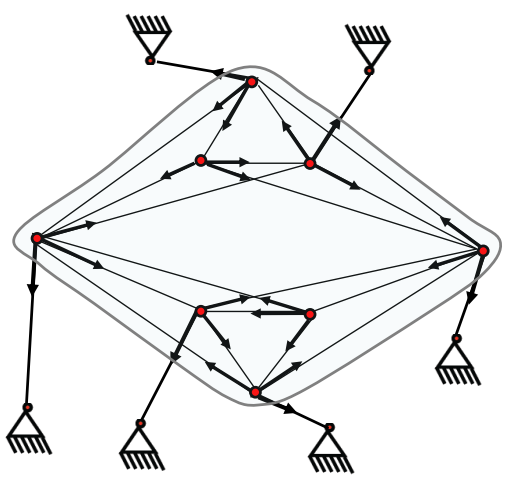
Remark. By translating the general results for generically isostatic graphs in dimension 1 and 2, the pinned counts in dimension 1 and 2 are also sufficient for a graph to be pinned 1 and 2-isostatic, respectively. In dimension 2, the necessity

and sufficiency of the counts is captured in the Pinned Laman Theorem [20], which states that $\widetilde{G}$ satisfies *Pinned Plane Framework Conditions* if and only if it there exists a pinned 2-isostatic realization of $\widetilde{G}$ in the plane.

In dimensions $d > 2$ however, these counting conditions are not sufficient for d-isostatic graphs. There are classic examples which show that these counts are not sufficient in 3-space. Figure 10 is one such classic example (analogue to a well known 'double banana' example in 3D unpinned bar and joint frameworks, also known in mechanical engineering community as floating frameworks). □

A d-directed graph has an immediate directed graph decomposition. Whether the graph is pinned d-isostatic will depend on whether all of the components are pinned d-isostatic.

While such a directed graph decomposition detects some failures, via failed necessary counts on subgraphs, detecting from a graph if a d-pinned framework is d-isostatic for $d > 2$ is generally difficult. Alternatively, we would have to resort to analysis of the pinned rigidity matrices. Figure 11 shows a 3-directed indecomposable graph which satisfies all the subgraph necessary pinned 3-counts of Theorem 3.6 but is still not pinned 3-isostatic. Even combined with the d-Assur decompositions, we do not have necessary and sufficient counting conditions for a graph to be d-Assur when $d > 2$. There are simple algorithms to detect the failure of the type in Figure 9, but no known polynomial algorithms for the failures of type Figure 11 [12, 22].



(a)

Figure 11: Pinned graph which is 3-directed, indecomposible, satisfies the necessary pinned 3-count, but it is not 3-isostatic

In 2D it is easier to check whether any pinned graph is 2-Assur (Figure 9(b)). We need to test both (i) the complete set of Pinned Plane Framework Conditions and (ii) the indecomposibility. While the decomposition algorithm is linear in

$|E|$, the process of extracting the 2-directed orientation from the pinned rigidity matrix is exponential. In [22] we have presented a polynomial time ($O(|I|^3)$) algorithm to test both (i) and (ii). This algorithm is based on the strongly connected decompositions of the graph and the pinned version of the pebble game algorithm.

However, if we already know that the graph is pinned d-isostatic, regardless of the dimension d, we can use the pinned pebble game algorithm to directly check whether the graph is d-Assur or if it can be further decomposed to individual d-Assur components [22].

4 Tracing Motions in Linkages through Assur Decompositions

In the previous papers for plane Assur graphs [20, 21], some additional properties were explored. In the context of linkages, a central property is how an associated framework responds when one of the links (edges) is replaced by a 'driver' so that this distance between the end points is controlled by a piston, or we control an angle at a hinge, as in a robot arm.

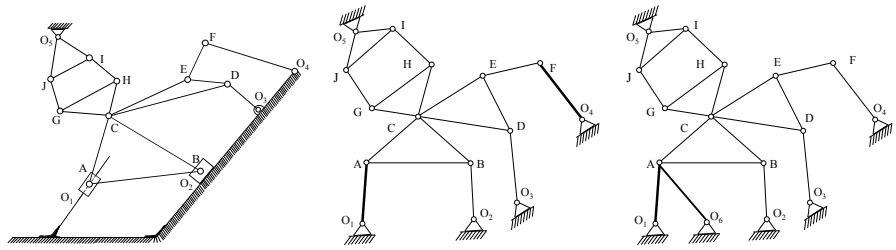


Figure 12: A plane linkage (a) is translated to a framework scheme with one degree of freedom (b) which is then made pinned 2-isostatic by adding a bar to block a chosen driver (c) .

We first recall how some types of motions of linkages were translated into pinned frameworks. Then we consider the infinitesimal motions of inner vertices which arise when one edge is removed. This is followed by some additional variations in how 'drivers' are inserted into pinned isostatic frameworks. We will word the theorems whenever possible for d-Assur graphs, though several key results only hold for 2-Assur graphs.

In §4.2 we will distinguish the special class of *strongly d-Assur graphs* by the feature that all inner vertices go into motion for every choice of inserted driver. This is the direct extension of a stronger key property of 2-Assur graphs [20]. This is not a matter of a deeper decomposition - but a difference in the characteristics of the underlying components we are analyzing (or synthesizing).

4.1 From linkages to structural schemes

In the introduction, we presented a figure of a linkage, complete with some slider joints, side by side with the *structural scheme* - corresponding flexible pinned bar and joint framework. The results in §3 applied to pinned isostatic frameworks, and those in §2 applied whenever we generated a directed graph.

It is appropriate in an applied mathematics to say a bit more about how some unusual features in the linkage were translated in the graph and about framework constraints. Most links in the plane linkage were represented as bars with pin joints at their ends. The translation for these is clear: links go to fixed length bars and pin-joints go to vertices. There were two more exceptional cases that we need to address - highlighted in Figure 13.

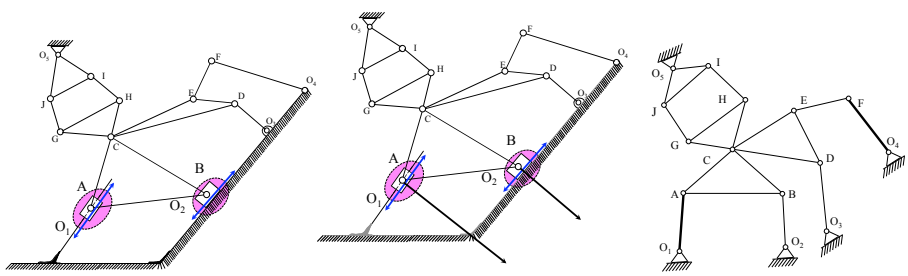


Figure 13: Slider joints in a linkage (a), function like joints to infinite bars perpendicular to the motion (b) and are then translated to schematic bars and joints in the structural scheme (c).

In this transfer, the slider joints are first seen as equivalent to a link to a pinned vertex very far away (at 'infinity'), which leaves only a translation (perpendicular to the bar) (b) and then schematically represented as a pin at a finite location (c). Geometrically, the inclusion of slide constraints as 'projective points at infinity' is part of the mechanical folklore and the corresponding projective theory of rigidity [3]. In this literature, it is recognized that the infinitesimal rigidity of a framework, the rank of the rigidity matrices, and the dimension of the space of non-trivial velocities is projectively invariant [3, 30].

It is also well known in the engineering literature ([15]) that both sliders and pinned joints are of the same type - lower kinematic pairs. Therefore, from the view point of rigidity (counts of the degrees of freedom) the sliders can be treated as pinned joints as is done in the structural scheme.

With a finite framework, it is possible to use such a projective transformation to bring all joints to finite locations. The rest of the schematic, and the analysis in §2,§3 is combinatorial, not specifically geometric. In the pinned rigidity matrices above, and our work below, we will continue to use the 'simpler' Euclidean representation with finite points. However, there is a full matrix representation and associated analysis that explicitly includes joints at infinity.

4.2 Motions generated by removing an edge: d-Assur vs strongly d-Assur graphs

The following result generates a 1 DOF linkage by removing an edge from an d-isostatic framework. Recall that an edge is part of a unique d-Assur graph in the extended decomposition, in which each strongly connected component is extended to include its outward directed edges. Recall that a *strongly d-Assur graph* is a d-Assur graph (minimal pinned d-isostatic graph) with the added property that removal of any edge induces a motion of all the inner vertices. In the plane all 2-Assur graphs are strongly 2-Assur graphs, as we will show.

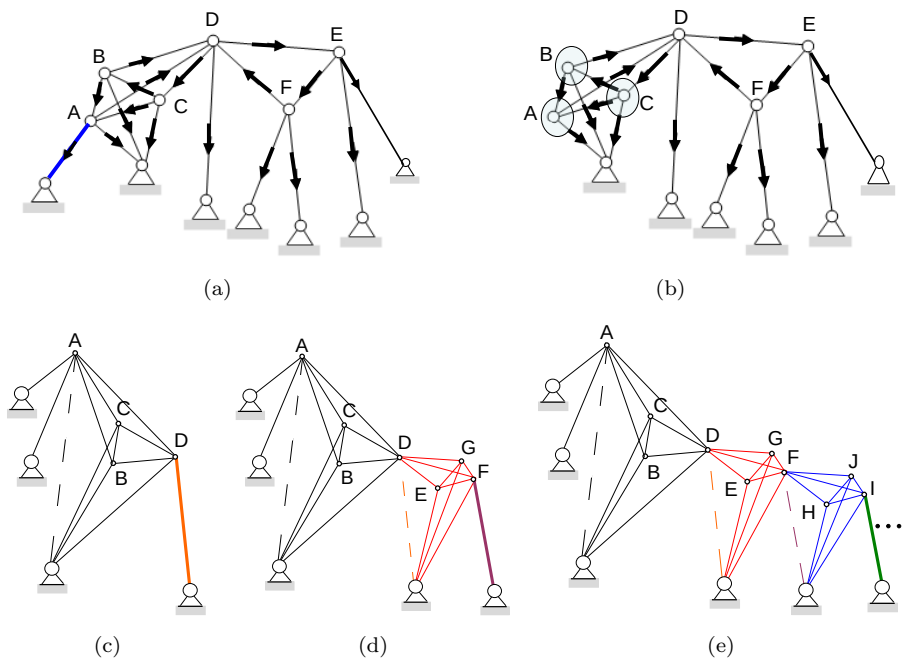


Figure 14: Examples of 3-Assur graphs which are not strongly 3-Assur . Removal of a blue edge (driver) (a) results in a motion of only the circled vertices in (b). In the examples in (c, d, e), introduce a new copy of a banana graph, illustrate how removal of certain edges induces smaller sets of vertices to move. Removal of an orange edge in (c) causes all vertices to move except A, which is held rigidity to the ground by an implicit (imaginary) edge indicated with a dashed line. This process is continued through (d,e).

Proposition 4.1 *If $\widetilde{G}$ is a pinned d-isostatic graph, and at a generic configuration p, removal of any edge from $\widetilde{G}$ causes an infinitesimal motion which is non-zero at all its inner vertices, then $\widetilde{G}$ is strongly d-Assur.*

Proof. We need to show that $\widetilde{G}$ is a 'minimal' pinned d-isostatic graph. By assumption $\widetilde{G}$ is a pinned d-isostatic graph. Assume $\widetilde{G}$ is not minimal pinned d-isostatic. Then by Theorem 3.4 there is a block-triangular matrix decomposition with more than one block. Removing an edge from any block which is lower right in the matrix will leave the graph associated with the upper left block (equivalently at the bottom of the directed graph decomposition) as pinned d-isostatic. This guarantees that the solution $R(\widetilde{G}, p) \times U = 0$ is zero on all vertices of this upper block (i.e. these vertices have no motion), a contradiction. Therefore $\widetilde{G}$ must be a minimal pinned d-isostatic graph. Since $\widetilde{G}$ is minimal and by assumption removal of any edge from $\widetilde{G}$ causes all inner vertices to go into motion, $\widetilde{G}$ is strongly d-Assur . □

Note that in Proposition 4.1, we could have assumed that $\widetilde{G}$ is pinned d-rigid, and the same conclusion would still follow. The assumption of independence was not necessary, as removal of an edge which causes a motion in the graph, indicates that the edge is independent.

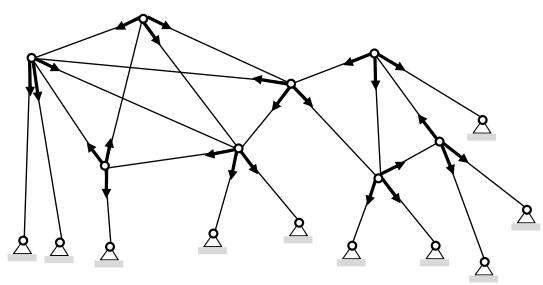


Figure 15: An example of a strongly 3-Assur graph. Removal of any ground edge or inner edge induces a motion of all inner vertices.

We have the following stronger statement about 2-Assur graphs (Figure 8(b):

Proposition 4.2 *If we remove any edge from a 2-Assur graph $\widetilde{G}$ then this leaves a graph which, at a generic configuration p has an infinitesimal motion which is non-zero at all inner vertices. Therefore $\widetilde{G}$ is strongly 2-Assur.*

Proof. Assume the graph $\widetilde{G}$ is 2-Assur. This means that $|E| = 2|I|$. Removing any one edge will leave the pinned rigidity matrix with $|E| = 2|I| - 1$, so there must be a nontrivial solution U to the matrix equation $R(\widetilde{G}, p) \times U = 0$. If $U_i = 0$ on some inner vertex i, then that vertex is still rigidly connected to the ground, and therefore must be in a pinned subgraph $\widetilde{G}'$ with $|E'| = 2|I'|$. This subgraph $\widetilde{G}'$ would itself be a pinned 2-isostatic graph which could not include at least one of the vertices of the removed edge. Such a subgraph contradicts the minimality of the original 2-Assur graph $\widetilde{G}$. □

The analog of Proposition 4.2 in d space ($d > 2$) fails, and there are explicit counter-examples. The example in Figure 15 is a strongly 3-Assur graph since

removal of any edge causes all inner vertices to be mobile, while the examples in Figure 14 are only 3-Assur. We should point out that all the examples that were presented in §3 were of strongly 2 or 3-Assur graphs.

The examples in Figure 14 (c, d, e) are particularly interesting. Starting with a 3-Assur graph in (c) we keep constructing new 3-Assur graphs as shown in (d) and (e). In these set of examples, removal of some edges (drivers) induces a full motion of inner vertices while removal of other edges induces motions in a smaller collection of inner vertices. One could create a further partial order of edges (drivers) in the 3-Assur graph - those edges whose removal puts all the inner vertices in motion which we call *regular* drivers, and further classification of edges called *weak* drivers. The partial order among the weak drivers would depend on the partial order of subsets of inner vertices which are sent into motion by a removal of this weak driver.

In 2D this distinction among the drivers would only occur in the graphs that are not 2-Assur, where removal of any edge in one component will only cause the motion in that 2-Assur component and all the other 2-Assur components below it in the acyclic 2-Assur decomposition.

The fact that not all 3-Assur graphs are strongly 3-Assur is connected to combinatorial obstacles to a good combinatorial (counting) characterization for 3-dimensional bar and joint frameworks, also noted in the lack of necessary and sufficient counting conditions in higher dimensional frameworks. In this context, detecting the difference between d-Assur graphs and strongly d-Assur graphs can be equally challenging. We pose this difficult open problem in combinatorially rigidity: *Given a d-Assur graph ($d > 2$) $\widetilde{G}$, find a combinatorial method (using only the information from the graph) that determines if $\widetilde{G}$ is strongly d-Assur .*

Remark. Despite these difficulties in higher dimensional space, having a comprehensive understanding of 2-dimensional Assur graphs can be very useful to the analysis of 3-dimensional linkages. A frequent practice by mechanical engineers and in robotics community is to decompose the 3-dimensional structure to several 2-dimensional components, which significantly simplifies the analysis.

In many cases, 3-dimensional linkages are built up by carefully connected copies of planar structures. Often identical 2-dimensional structures are reused forming the larger 3-dimensional structure, where the motions of individual 2-dimensional structures are restricted to the plane by the geometry of the other constraints. □

By general results from algebraic geometry at a generic configuration p for the vertices (the length constraints are algebraic conditions), the infinitesimal motion U extends to a non-trivial finite path $p(t)$ within the configuration space $\mathbb{R}^{d|I|}$ which preserves the constrained lengths [1]. In fact, the mechanism is typically designed to create a specific finite path in the configuration space, or perhaps for a given inner vertex to trace a specific path in $\mathbb{R}^d$. Thus, we can summarize the two propositions from this section in terms of finite motions in the following corollaries:

Corollary 4.3 *If $\widetilde{G}$ is a pinned d-isostatic graph, and at a generic configuration p, then removal of any edge from $\widetilde{G}$ causes a finite motion which is non-zero at all its inner vertices if any only if $\widetilde{G}$ is strongly d-Assur.*

One of the central motivations for decomposing a linkage is to break down the analysis of the paths being generated from one set of large polynomial constraints for the entire linkage into the analysis of smaller polynomials for each of the pieces, plus a set of linking equations for composing the results for the components into a single larger analysis. This is precisely what the d-Assur decomposition lets us do. Here, we present a linearized version of this process, for the decomposition into d-Assur graphs.

Assume we have a d-isostatic linkage (pinned framework) $(\widetilde{G_k}, p)$, and we assign a vector r of *drive velocities* to the pinned vertices. We need the $e \times 1$ *drive matrix*:

$$\mathbf{D}(\widetilde{G_k}, p, r) = \begin{array}{l} \{i,j\} \ |i,j \in I \\ \{i,k\} \ |k \in P \end{array} \begin{pmatrix} \vdots \\ 0 \\ (p_i - p_k) \cdot r_k \\ \vdots \end{pmatrix}$$

Proposition 4.4 *For an assignment of drive velocities r to the pinned vertices of a d-Assur graph $\widetilde{G_k}$, and a generic configuration p for the vertices, the drive equation $\mathbf{R}(\widetilde{G_k}, p) \times U = \mathbf{D}(\widetilde{G_k}, p, r)$ has a unique solution.*

Proof. The essential property is that $\mathbf{R}(\widetilde{G_k}, p)$ is invertible, so the system of non-homogeneous equations has the unique solution:

$$U = [\mathbf{R}(\widetilde{G_k}, p)]^{-1} \mathbf{D}(\widetilde{G_k}, p, r). \qquad \square$$

With this observation in hand, we see that if we decompose the original 3-isostatic linkage, and replace one edge of a bottom 3-Assur linkage (A_1, p) with a driver, then we can compute the velocities of all the inner vertices of (A_1, p). These velocities can then be used as drive velocities for other components in the linkage. Iteration up the decomposition gives solutions for all inner vertices of the entire linkage. All computations are reduced to computations within Assur components. This is one primary value for the Assur decompositions of linkages [2, 13, 15].

Notice that this capacity to propagate drive velocities through the decomposition does not require that all the components are strongly d-Assur. However, it is possible that if some are not, the induced velocity will be zero at a number of inner vertices of components above the driver.

Remark. If we go to a non-generic configuration p for a d-Assur graph A, then the rank of the pinned rigidity matrix can drop, creating a *singular configuration.*

In these singular configurations, two things happen: (i) there are drive velocities for which there are no solutions; and (ii) for some drive velocities (including the 0 drive velocity), there are multiple solutions. Both of these events are a serious problem for a linkage in mechanical engineering.

The configurations p which make this happen can be *dead end positions*, a geometric subclass of the singular positions - depending on the driver, or drive velocities. There is an extensive geometric literature for dead end positions [18, 21]. Again, this geometric analysis is simplified by working with the geometry of the components of the Assur decomposition. We note that the singular positions of an Assur component depend on the geometry of the configuration p of both the inner and the pinned vertices. This geometry is the subject for ongoing investigations, often for specific linkages or classes of linkages. □

4.3 Re-pinning of inner vertices and release of pinned vertices

An alternative operation which mechanical engineers use in 2D linkages is to 'replace a driver' in a structural scheme and create a 2-pinned isostatic graph is to shift an inner vertex into a pinned vertex (Figure 16(c) to (a)). This can be represented by a two step process passing through Figure 16(b). We add a bar to freeze out the motion from the inner end of the driver. Then we convert this small dyad (which includes the old driver) into a new pinned vertex Figure 16(a).

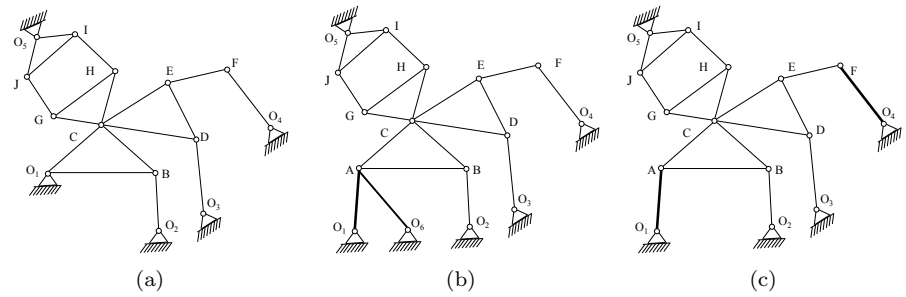


Figure 16: Shifting from a pinned vertex O_3 (a), through a dyad at A (b), to a flexible structural scheme by removing an edge (c).

The reverse operation, which also applies in higher dimensional space, is to 'release' a pinned vertex into a new inner vertex, moving from Figure 16(a) to (b), using d edges in d-space ($d = 2, 3$) and creating a further strongly d-Assur graph (dyad in the plane, triplet in 3-space). If we started with a pinned d-isostatic framework, this released framework is also a pinned d-isostatic framework, with one added strongly d-Assur component (b). Then we remove an edge from this minimal d-Assur graph (dyad) (c), generating a local motion

at this new inner vertex. This motion propagates on into rest of the inner vertices, using the matrix as above.

More generally, we can take a d-isostatic graph, and add a single velocity to any one pin (or set of pins). This then proceeds as above with the drive matrix connecting these pins to the other parts of the framework, and solving for the velocities at all the inner vertices. In following this propagation, it is still valuable to be able to do the work on one d-Assur component at a time, which was the essential feature of the previous subsection.

4.4 Vertex removal

An additional engineering technique in testing for decompositions is to remove an inner vertex. Removing any pinned vertex will generate a finite space of non-trivial motions - the dimension of the space being the valence of the pinned vertex. We now focus on removing an inner vertex. Any edge will always contain at least one inner vertex, leading to a vertex analog of Theorem 4.3.

Theorem 4.5 *If $\widetilde{G}$ is a pinned d-isostatic graph, and at a generic configuration p, removal of any inner vertex from $\widetilde{G}$ causes a finite motion which is non zero at all inner vertices, then $\widetilde{G}$ is d-Assur.*

Proof. Case 1: $\widetilde{G}$ has one inner vertex. $\widetilde{G}$ is trivially strongly d-Assur.

Case 2: There is more than one inner vertex. We want to show that $\widetilde{G}$ is minimal pinned d-isostatic. If some vertex has valence d, removing it will not cause the other vertices to move, which contradicts the assumption. Therefore the vertex we remove from $\widetilde{G}$ has valence at least $d+1$.

Since $\widetilde{G}$ is a pinned d-isostatic graph, there is a block-triangular matrix decomposition. Assume $\widetilde{G}$ is not minimal pinned d-isostatic. Removing any vertex of degree $\geq d+1$ from any block which is lower right, will leave the upper left block (near the bottom of the directed graph decomposition) as pinned d-isostatic. This guarantees that the solution $R(\widetilde{G}, p) \times U = 0$ is zero on all vertices of this upper block (i.e. these vertices have no motion), a contradiction. Therefore $\widetilde{G}$ is minimal and d-Assur. $\square$.

In Theorem 4.5 it may be surprising that we cannot conclude that the graph will be strongly d-Assur ($d > 2$). Consider the example in Figure 14 (a), removing any inner vertex from this graph will cause all other inner vertices to be mobile, yet this is a 3-Assur graph, not strongly 3-Assur.

Theorem 4.6 *If we remove any inner vertex from a strongly d-Assur graph $\widetilde{G}$, then at a generic configuration p, $\widetilde{G}$ has a finite motion which is non zero at all inner vertices.*

Proof. Case 1: $\widetilde{G}$ has one inner vertex, removing it leaves no inner vertices.

Case 2: There is more than one inner vertex. Since $\widetilde{G}$ is strongly d-Assur , every inner vertex has to be at least valence $d+1$, as any inner vertex of degree

d (and its outgoing edges) in a pinned d-isostatic graph is a strongly d-Assur component (a d-dyad)). Choose any inner vertex v and remove its edges. As $\widetilde{G}$ is strongly d-Assur, removal of these edges, in fact any single edge, will cause a motion at all inner vertices. Now remove the vertex v, all other inner vertices are still in motion. □

This result is expected, as removal of an inner vertex should cause at least as much motion as a removal of a single edge incident to that vertex. In particular, if we remove a vertex of degree higher than $d+1$ there will be more flexibility (more DOF) caused then removal of any edge at that vertex.

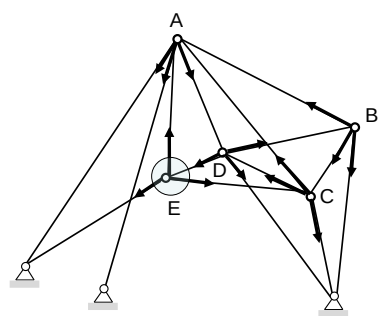


Figure 17: A 3-Assur graph, removal of vertex E causes motion in vertices B, C and D, but not in A.

Remark. Most of these concepts and results in §3,4 were illustrated using examples in 3-space. All these observations are true in all dimensions. To give higher dimensional examples, one can easily start with a 3-dimensional example and construct similar examples in any higher dimension, using the technique of coning [28], which transfers the rigidity, orientation (out-degree), counts, etc. from a framework in dimension d to dimension $d+1$. □

5 Further Work

The results in this paper are contributions to the the synthesis and analysis of linkages in d-space, and provide new extensions of 2-dimensional Assur graphs to higher dimensional space, and new connections and better understanding of the difficulties with higher dimensional rigidity. There are several key further directions that are part of the ongoing research, some of which are addressed in papers such as [22] and others which hope to follow up in future papers. We highlight some of current research and envision other developments.

5.1 Applications to more general kinematic structures

Section 3 developed decompositions for minimal pinned isostatic frameworks (d-Assur), which traced back to the strongly connected decompositions of Section 2. Looking back, the techniques as developed only depended on two properties:

1. we had an underlying constraint multi-graph;
2. we have a square constraint matrix for the multi-graph with rows indexed by the edges and columns indexed (in groups) by the vertices.

With these two properties, we can use the constraint matrix to generate directions on the multi-graphs, with the out-degree corresponding to the number of columns for a given vertex (as in §3). With this in place, the Laplace block decomposition of the determinant of the constraint matrix facilitates the certificate that we can generate an orientation of the constraint multi-graph. This orientation will yield a strongly connected decomposition of the original multi-graph.

The key results in §2 and §3 also did not depend on each type of vertex having the same degree in the constraint graph (equivalently, having the same number of columns in the pinned constraint matrix). They only depended on the out-degree being constant when we switched to another orientation of the graph or equivalently on there being an associated number of columns or variables for each type of vertex in the graph. The results also extend to multi-graphs as the constraint input, as long as there is a row in the constraint matrix for each edge of the multi-graph. Everything generalizes directly to these broader settings.

We can apply the entire suite of decomposition techniques from general pinned bar and joint frameworks to analyze isostatic body-bar frameworks in 3-space[26]. For these special frameworks, we not only have the required constraint matrix on the multi-graph, but we have complete necessary and sufficient counting properties to fully test the components of the decomposition for whether they are generically 3-isostatic [26]. In fact, there is a complete theory, with good algorithms, and these are associated with some types of linkages, such as the well-known Stewart platform. In addition, for d-body-Assur graphs, all minimal components are strongly d-Assur, in the sense that removal of any one edge causes all inner bodies to go into relative motion.

We could also adapt these decomposition techniques to mixed frameworks in the plane, in which we have bodies, bars, joints, pins between bodies, prismatic or pistons drivers (Figure 18). For these more general kinematic chains, the classical counting rules are associated with the name of Grübler [7, 20]. These counting rules provide necessary conditions on a linkage to be generically isostatic. While they are not usually elaborated into necessary and sufficient conditions in the engineering literature, they can be reworked using the generic theory of plane rigidity to give such necessary and sufficient conditions. All of the results for the plane decomposition extend to mixed frameworks, providing a complete theory of 2-Assur decompositions, and the associated block-triangular decomposition of the pinned rigidity matrix for the constraint multi-graph.

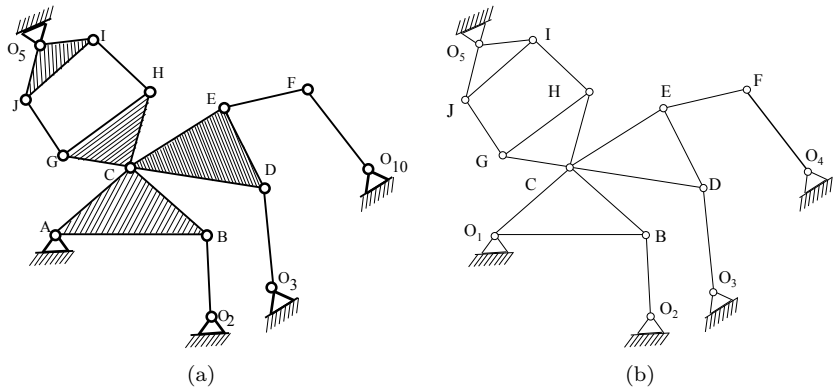


Figure 18: The unified structural scheme (b) of the linkage with bodies (a).

We can further adapt these decomposition techniques to mixed frameworks in 3-space, with bodies, bars, joints, revolute joints between bodies (hinges), and prismatic joints [7, 15]. For these general kinematic multi-loop mechanisms, the more general counting rules are associated with the names of Chebychev, Grübler and Kutzbach [7]. In such a broad extension, the prismatic joints again become 'revolute joints at infinity' in the general projective theory of 3-space rigidity [3].

5.2 Algorithms for generating directions and Assur decompositions

The algorithms for assigning directions to the graph presented in this paper are, in principle, exponential: searching over terms in the Laplace expansion of the determinant of the pinned rigidity matrix. However, fast pebble game algorithm can be used to generate the 2-directed orientations, as well as for the necessary and sufficient counting conditions and to decompose the graph into 2-Assur components. Some of this work using the pebble game algorithm and techniques from rigidity theory was for the first time presented to a mechanical engineering community [22], which provides fast analysis and synthesis of linkages.

As we have seen, for $d \geq 3$-space, there is no known necessary and sufficient counting (polynomial time) algorithm for infinitesimal rigidity or independence. The examples in §3.5 illustrated some of the problems. However, if we start with a pinned d-isostatic graph (or at least without redundance - no dependant rows in the pinned rigidity matrix), we have shown elsewhere how to adapt the pebble game to give the d-directions and d-Assur decomposition [22]. We are currently investigating more on these pebble games techniques.

For general body-bar frameworks in d-space, as well as their specialization as body-hinge and molecular frameworks, mentioned in §5.1, there are full, efficient pebble games to generate the directed graphs and to complete the decomposi-

tion. Body and bar frameworks are routinely used in study of general linkages and in robotics. Given this motivation of important applications to mechanical engineering community, we are currently adapting some of the techniques and algorithms to body bar Assur graphs.

5.3 Inductive constructions

An alternative process for analyzing pinned 3-isostatic graphs is through inductions. Variations on this theme are widely used in various forms [24, 30]. In addition, there are selected forms of these inductions which produce minimal 3-isostatic graphs - the 3-Assur graphs. For the synthesis of linkages, various inductive steps are widely used, even if not all possible linkages can be generated [15, 19].

It is an ongoing research project to survey the known and conjectured inductive constructions for 3-space and some new connections to inductions which generate 3-directed graphs and which can be driven by the 3-directions of a proposed final graph.

In the end, we will still only be left with a conjecture for sufficient steps to construct all 3-Assur graphs. These conjectures are an extension of the conjectures of Tay and Whiteley [24].

5.4 Concluding remarks

As this paper confirms, there are a wide variety of important, and mathematically interesting unsolved questions for investigation. In the last 30 years, there has been a broad development in the general theory of rigid and flexible structures. In particular, during the last decade, there has been a renewed interaction of these developments in the mathematical theory of rigidity, and the results on rigidity of structures, with the parallel analysis of linkages, which has its own rich history. Our work on Assur graphs, offers some additional tools for decomposition of pinned bar and joint structures, probing into the difficult and not well understood rigidity of bar and joint structures in higher dimensional space.

There is a wide field of fruitful directions for further investigation. We invite the reader to join in these investigations.

References

[1] L. Asimov and B. Roth, The Rigidity Of Graphs, AMS 245 (1978), 279–289.

[2] L. V. Assur. Issledovanie ploskih sterznevyh mehanizmov s nizsimi parami s tocki zreniya ih struktury i klassikacii. Izdat. Akad. Nauk SSSR, Edited by I. I. Artobolevski. 1952.

[3] H. Crapo and W. Whiteley, Statics of frameworks and motions of panel structures: a projective geometric introduction, Structural Topology 6 (1982), 43-82.

[4] K. Bowden, On general physical systems theories, Intern. J. Systems Theories 18(1990), 61-79. 2000.

[5] T. Eren, W. Whiteley, B. D.O. Anderson, A. S. Morse, P. Belhumeur, Information Structures to Secure Control of Rigid Formations with Leader-Follower Structure, In Proc. of the American Control Conference, pp. 2966-2971, Portland, Oregon, June 2005.

[6] S. Foufou, D. Michelucci, and J-P. Jurzak: Numerical Decomposition of Geometric Constraints, 2005 ACM 1-59593-015-9/05/0006 2, (2005) ACM.

[7] G. Gogu, Chebychev-Grubler-Kutzbach's criterion for mobility calculation of multi-loop mechanisms revisited via theory of linear transformations, European Journal of Mechanics / A Solids (May 2005), 24 (3), pg. 427-441

[8] J. Hendrickx, B.D.O.Anderson, J-C. Delvenne, and V. Blondel, International Journal of Robust and Nonlinear Control, Volume 17, Issue 10-11, 2006.

[9] C. Hoffman, A. Lomonosov, M. Sitharam Decomposition plans for geometric constraint systems, parts I : Performance measures for CAD, J. Symbolic Computation 31, 2001 367408.

[10] C. Hoffman, A. Lomonosov, M. Sitharam Decomposition plans for geometric constraint problems, part II : New algorithms, J. Symbolic Computation 31, 2001, 409427.

[11] G. Kron, Diakoptics: The Piecewise Solution of Large Scale Systems, MacDonald, 1963.

[12] A. Lee and I. Streinu. Pebble Game Algorihms and (k, l)-sparse graphs, 2005 European Conference on Combinatorics, Graph Theory and Applications (EuroComb 05), DMTCS Proceedings: 181186, 2005

[13] S. Mitsi, K.-D. Bouzakis, G. Mansour, and I. Popescu, Position analysis in polynomial form of planar mechanisms with Assur groups of class 3 including revolute and prismatic joints. Mech. Mach. Theory, 38(12):1325 1344, 2003.

[14] S. Lo, M. Monagan, A. Wittkopf, Strongly Connected Graph Components and Computing Characteristic Polynomials of Integer Matrices in Maple , preprint 2006 from http://www.cecm.sfu.ca/CAG/papers/CharPoly.pdf.

[15] R. L. Norton, Design of Machinery: An Introduction To The Synthesis and Analysis of Mechanisms and Machines. McGraw Hill, New York, 2004.

[16] J . Owen, Algebraic solution for geometry from dimensional constraints. In Proc. of the Symp. on Solid Modeling Foundations and CAD/CAM Applications, (1991) 397407.

[17] J . Owen, Constraints on simple geometry in two and three dimensions. Int. J. of Computational Geometry and Applications 6, (1996), 421434.

[18] R. Penne, Relative Centers of Motion, Implicit Bars and Dead-Center Positions for Planar Mechanisms, European Journal of Combinatorics 31 (2010).

[19] O. Shai, Topological Synthesis of All 2D Mechanisms through Assur Graphs, ASME Design Engineering Technical Conferences, August 15-18, Montreal, Quebec, Canada, (2010).

[20] O. Shai, B. Servatius, and W. Whiteley, Combinatorial char- acterization of the assur graphs from engineering. European Journal of Combinatorics 31, (2010), 1091-1104.

[21] O. Shai, B. Servatius, and W. Whiteley, Geometric Properties of Assur Graphs. European Journal of Combinatorics 31 (2010), 1105-1120.

[22] A. Sljoka, O. Shai. and W. Whiteley, Checking Mobility and Decomposition of Linkages via Pebble Game Algorithm, ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Aug 28-31, 2011, Washington, USA, IDETC/CIE 2011.

[23] A. Sljoka, Counting for Rigidity, Flexibility and extensions vis the Pebble Game Algorithm, Masters Thesis, York University, Septemeber 2006.

[24] T-S. Tay and Walter Whiteley, Generating isostatic frameworks. Structural Topology, (11) (1985), 2169.

[25] R. Tarjan, Depth-first search and linear graph algorithms, SIAM Journal on Computing. 1972, Vol. 1, No. 2, P. 146-160.

[26] N. White and W. Whiteley, The algebraic geometry of motions of bar-and-body frameworks. SIAM J. Algebraic Discrete Methods, 8(1):1 32, 1987.

[27] W. Whiteley, A matroid on hypergraphs, with applications in scene analysis and geometry, Disc. and Comp. Geometry 4 (1988), 75-95

[28] W. Whiteley, Cones, infinity and one-story buildings, Structural Topology 8 (1983), pp 53 - 70.

[29] W. Whiteley, Union of Matriods and the rigidity of frameworks; SIAM J. Disc. Meth. 1 (1988), 237-255.

[30] W. Whiteley, Some Matroids from Discrete Applied Geometry, Contemporary Mathematics, AMS **197** (1996), 171–311.

[31] G.F. Zhang and X.S. Gao, Spatial Geometric Constraint Solving Based on k-connected Graph Decomposition, SAC06 April 23-27, 2006, Dijon, France, ACM.