Computer Aided Geometric Design 40 (2015) 1–25

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Optimal decomposition and recombination of isostatic geometric constraint systems for designing layered materials ☆

T. Baker *, M. Sitharam, M. Wang, J. Willoughby

University of Florida, CSE Building, Gainesville, FL 32611, USA

ARTICLE INFO

Article history:
Received 21 February 2015
Received in revised form 17 May 2015
Accepted 6 July 2015
Available online 30 July 2015

Keywords:
Rigidity
Geometric constraint solving
Configuration spaces
Self-similar structures
Layered materials

ABSTRACT

Optimal recursive decomposition (or *DR-planning*) is crucial for analyzing, designing, solving or finding realizations of geometric constraint systems. While the optimal DR-planning problem is NP-hard even for (general) 2D bar–joint constraint systems, we describe an $O(n^3)$ algorithm for a broad class of constraint systems that are isostatic or underconstrained. The algorithm achieves optimality by using the new notion of a canonical DR-plan that also meets various desirable, previously studied criteria. In addition, we leverage recent results on Cayley configuration spaces to show that the indecomposable systems – that are solved at the nodes of the optimal DR-plan by recombining solutions to child systems – can be minimally modified to become decomposable and have a small DR-plan, leading to efficient realization algorithms. We show formal connections to well-known problems such as completion of underconstrained systems. Well suited to these methods are classes of constraint systems that can be used to efficiently model, design and analyze quasi-uniform (aperiodic) and self-similar, layered material structures. We formally illustrate by modeling silica bilayers as body–hyperpin systems and cross-linking microfibrils as pinned line-incidence systems. A software implementation of our algorithms and videos demonstrating the software are publicly available online (visit http://cise.ufl.edu/~tbaker/drp/index.html).

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Geometric constraint systems have well-established, mature applications in mechanical engineering and robotics, and they continue to find emerging applications in diverse fields from machine learning to molecular modeling. Solving or realizing geometric constraint systems requires finding real solutions to a large multivariate polynomial system (of equalities and inequalities representing the constraints); this requires double exponential time in the number of variables, even if the type or orientation of the solution is specified. Thus, to realize a geometric constraint system, it is crucial to perform recursive decomposition into locally rigid subsystems (which have finitely many solutions), and then apply the reverse process of recombining the subsystem solutions. With the use of *decomposition–recombination (DR-) planning*, the complexity is dominated by the size of the largest subsystem that is solved, or recombined, from the solutions of its child subsystems, i.e. the maximum fan-in occurring in a DR-plan. In addition, navigating and analyzing the solution spaces, as well as designing constraint sys-

☆ This paper has been recommended for acceptance by Konrad Polthier.

* Corresponding author.
E-mail addresses: tbaker@cise.ufl.edu (T. Baker), sitharam@cise.ufl.edu (M. Sitharam), menghan@cise.ufl.edu (M. Wang), jw5@cise.ufl.edu (J. Willoughby).

http://dx.doi.org/10.1016/j.cagd.2015.07.001
0167-8396/© 2015 Elsevier B.V. All rights reserved.

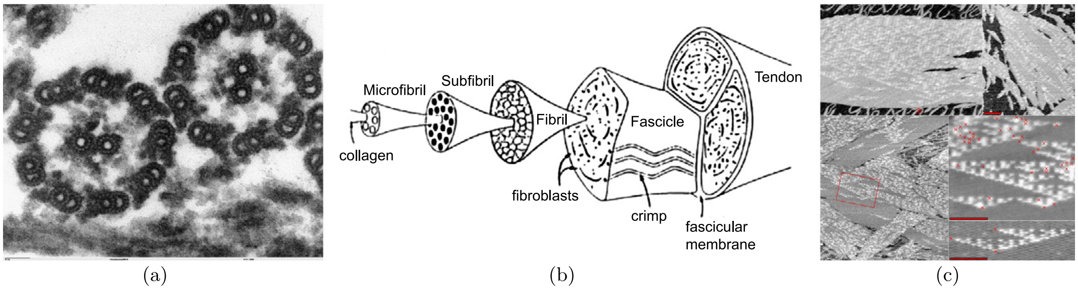


Fig. 1. (a) Cross section of the Chlamydomonas algae axoneme, a cilia composed of microtubules (Wikimedia Commons, 2007a). (b) Cross section of a tendon displaying the hierarchical structure (Hollister, 2007). (c) A DNA array exhibiting the Sierpinski triangle (Wikimedia Commons, 2007b).

tems with desired solution spaces, leads to the *optimal decomposition–recombination (DR-) planning* problem (Sitharam, 2005; Hoffman et al., 2001a, 2001b).

For a broad class of geometric constraint systems, local rigidity is characterized generically as a sparsity and tightness condition of the underlying constraint (hyper)graph (Laman, 1970; Streinu and Theran, 2009; Tay, 1976; White and Whiteley, 1987). This allows the generic DR-planning problem to be stated and treated as a combinatorial or (hyper)graph problem as we do in this paper.

Naïvely, the optimal DR-plan is used as follows. Each decomposed subsystem – a node of the DR-plan – is treated and solved as a polynomial system of constraints between its child subsystems. However, even in an optimal DR-plan, there can be arbitrarily many children at a node. In other words, even in the recursive decomposition given by an optimal DR-plan, the size of the maximal indecomposable subsystem could be arbitrarily large. It represents a bottleneck that dictates the complexity of solving or realizing the constraint system (Sitharam, 2006; Sitharam et al., 2010a, 2010b). We address this problem using the recently developed concept of *convex Cayley configuration spaces* (Sitharam and Gao, 2010; Sitharam et al., 2011a, 2011b; Sitharam and Wang, 2014; Wang and Sitharam, 2015). This allows for even greater reduction of the complexity by realizing large, indecomposable systems in a manner that avoids working with large systems of equations. Specifically, we give an efficient technique for *optimally modifying* large indecomposable subsystems in a manner that reduces their complexity while preserving desired solutions; the modification ensures a convex Cayley configuration space, and the space can be efficiently searched to find a realization that satisfies the additional constraints of the original system. This optimal modification problem is a generalization of the previously studied problem of optimal completion of underconstrained systems (Sitharam, 2005; Joan-Arinyo et al., 2003).

DR-plans are especially useful for constraint systems that exhibit some level of *self-similarity* and *quasi-uniformity*, in addition to isostaticity. These properties can be leveraged to further reduce the complexity of both optimal DR-plan construction and recombination. We consider 3 different types of constraint systems – which we collectively call *qusecs* – that are used to model, design, and analyze quasi-uniform or self-similar materials. In the remainder of this section, we motivate the materials application and give the contributions and organization of the paper.

1.1. Introducing qusecs

A large class of constraint systems that we call *qusecs*, a contraction of "quasi-uniform or self-similar constraint system", (a) can be treated combinatorially as described above and (b) occur as independent (isostatic or underconstrained) systems in materials applications. We discuss these next. Some natural and engineered materials can be analyzed by treating them as two dimensional (2D) layers. As illustrated by the examples below, the structure within each layer is often: self-similar[1] (Gaspar and Csermely, 2012), spanning multiple scales; generally aperiodic and quasi-uniform within any one scale; and composed of a few repeated motifs appearing in disordered arrangements. Note that a 2D layer is not necessarily planar (genus 0), it can consist of multiple, inter-constraining planar monolayers. Furthermore, a layer is often either *isostatic* or *underconstrained* (not *self-stressed/overconstrained*, see Section 2.1 for definitions). These properties, as well as quasi-uniformity, aperiodicity, self-similarity, and layered structure, are natural consequences of evolutionary pressures or design objectives such as stability, minimizing mass, optimally distributing external stresses, and participating in the assembly of diverse and multifunctional, larger structures.

The importance of an optimal DR-plan is particularly evident for a qusecs. The quasi-uniform or self-similar properties mean that the decomposition and solution for one subsystem can be used as the decomposition and solution for other subsystems, thus causing further reduction in the complexity of both DR-planning and recombination. This is shown in Figs. 3 and 10.

Some materials that are readily modeled as qusecs include:

[1] In this manuscript we only study finite 2D structures. *Self-similarity* refers to the result of finitely many levels of hierarchy or subdivision in an iterated scheme to generate self-similar structures.

1. Cross-sections of microtubule structures (Needleman et al., 2004) (Fig. 1a), e.g., in ciliary membranes and transitions (Garcia-Gonzalo and Reiter, 2012).
2. Cross-sections of organic tissue with hierarchical structure, e.g., compact bone and tendon (Fig. 1b).
3. Crosslinked cellulose or collagen microfibril monolayers, e.g., in cell-walls (Wikimedia Commons, 2010, 2007c), as well as crosslinked actin filaments in the cytoskeleton matrix. See Section 6.
4. More recent, engineered examples, including disordered graphene layers (Björkman et al., 2015; Eder et al., 2015) sometimes reinforced by microfibrils; and DNA assemblies including a recent Sierpinski gasket (Rothemund et al., 2015) (Fig. 1c), bringing other self-similar structures (Wikimedia Commons, 2012) within reach.
5. Silica bi-layers (Wilson et al., 2013), glass (Heyde, 2013), and materials that behave like assemblies of 2D particles under non-overlap constraints, i.e. like jammed disks on the plane (Donev et al., 2015). See Section 5.

1.2. Organization and contributions

In Section 2, we provide basic definitions in combinatorial rigidity theory, and formalize the new notion of qusecs (Sitharam et al., 2010a, 2010b; Sitharam, 2006). In addition, we define DR-plans and what it means for a DR-plan to be complete or optimal. We survey previous work on DR-planning algorithms, discussing other desirable criteria of DR-plans and their relation to the NP-hard optimality property of DR-plans.

In Section 3, we define a so-called *canonical DR-plan* and prove a strong Church–Rosser property: all canonical DR-plans for isostatic or underconstrained qusecs are optimal. In so doing, we navigate the NP-hardness barrier present in the general form of the DR-planning problem; the canonical DR-plan elucidates the essence of the NP-hardness of finding optimal DR-plans when a system is over-constrained. Furthermore, our optimal/canonical DR-plan satisfies desirable properties such as the previously studied *cluster minimality* (Hoffman et al., 2001a). Also in this section, an $O(n^3)$ time algorithm is provided to find an optimal DR-plan for independent bar–joint graphs. While this and the next section focus on bar–joint graphs, the theory is easily extended to other qusecs used to model the abovementioned types of materials, as shown in subsequent sections.

In Section 4, we give a method to deal with the algebraic complexity of recombining the realizations or solutions of child subsystems into a solution of the parent system (Sitharam et al., 2010a, 2010b; Sitharam, 2006). Specifically, we define the problem of minimally modifying the indecomposable recombination system so that it becomes decomposable via a small DR-plan and yet preserves the original solutions in an efficiently searchable manner. When the modifications are bounded, we obtain new, efficient algorithms for realizing both isostatic and underconstrained qusecs by leveraging recent results about Cayley parameters in Sitharam and Gao (2010), Sitharam et al. (2011a, 2011b) (see Sections 4.3 and 4.4). In Section 4.5, we show formal connection to well known problems such as optimal completion of underconstrained systems (Joan-Arinyo et al., 2003; Sitharam, 2005; Gao et al., 2006) and to find paths within the connected components.

In Sections 5 and 6, we briefly describe applications of the above techniques to modeling, analyzing, and designing specific properties in 2D material layers (Jackson and Jordán, 2008). We explicitly model these materials as qusecs. For Examples 4 and 5, we discuss boundary conditions for achieving various desired properties of body–hyperpin systems. For Example 3, we discuss canonical and optimal DR-plans for pinned line incidence systems (Sitharam et al., 2014).

Section 7 concludes the paper, and Section 7.1 lists open problems and conjectures. In particular, we conjecture that the methods of Section 3 extend in fact to a large class of (hyper)graphs, formally those with an underlying abstract rigidity matroid in which independence corresponds to some type of sparsity, and maximal independence (rigidity) is a tightness condition.

Throughout this paper, an asterisk after a formal statement indicates that its proof appears in Appendix A.

A software implementation of our algorithms and videos demonstrating the software are publicly available online.[2]

2. Preliminaries and background

We first give basic definitions and concepts in combinatorial rigidity, leading to a definition of a DR-plan, its properties, and how they relate. The section ends with a discussion of previous work on DR-plans.

2.1. Geometric constraint systems and combinatorial rigidity

In this paper, a *geometric constraint system* is a multivariate polynomial (usually bilinear or quadratic) system $G(x, \delta) = 0$, representing constraints with parameters δ between geometric primitives in $\mathbb{R}^2$ represented collectively as $x \in \mathbb{R}^n$. When the type of constraint (system) is fixed, the system is simply represented as (G, δ), where G is the underlying constraint (hyper)graph $G = (V, E)$ with the vertices V representing the geometric primitives in $\mathbb{R}^2$ and (hyper)edges E representing the constraints, each with an associated parameter δ. For example, a *bar–joint system or linkage* (G, δ), is a graph $G = (V, E)$ with fixed length bars as edges, i.e. $\delta : E \to \mathbb{R}$; this represents the distance constraint system $\|x_u - x_v\|_2 = \delta_{u,v}$ for $(u, v) \in E$, where $x_u \in \mathbb{R}^2$ represents the coordinates of $u \in V$.

[2] Visit http://cise.ufl.edu/~tbaker/drp/index.html.

In all types of geometric constraint systems we consider in this paper, a Cartesian *realization* or *solution* $G(p)$ of (G, δ) is an assignment of coordinates or Euclidean transformations (poses), $p : V \to \mathbb{R}^2$ or $\mathbb{R}^3$, to the vertices of G satisfying the constraints with parameters δ, modulo orientation preserving isometries (Euclidean rigid body motions).

Although the realization space itself depends on the constraint parameters δ, many relevant *generic* properties of the constraint system $G(x, \delta)$ are defined to be properties of the constraint (hyper)graph G and do not depend on δ (or they hold for all but a measure zero set of δ values). Many of these are properties of the Jacobian $\Delta_x G(x, \delta)$, often called the appropriate *rigidity matrix of* G (a matrix of indeterminates). For example, the *bar–joint rigidity matrix of the graph* $G = (V, E)$ is a matrix of indeterminates representing the Jacobian of the distance map $\|x_u - x_v\|_2$ for $(u, v) \in E$. The matrix has 2 columns per vertex in V and one row per edge in E, where the row corresponding to edge (u, v) contains the 2 coordinate indeterminates for $x_u - x_v$ (resp. $x_v - x_u$) in the 2 columns for u (resp. v), i.e. 4 non-zero entries per row.

One important property of a generic constraint system or (hyper)graph[3] is *rigidity*, i.e. the realizations or solutions of the corresponding constraint system being generically isolated and zero-dimensional. The result by Asimow and Roth (1978) shows a constraint (hyper)graph is rigid if and only if it is generically *infinitesimally rigid*, i.e. the number of independent rows of its appropriate rigidity matrix is at least the number of columns less the number of rigid body motions, which is 3 for 2D bar–joint systems.

Geometric constraint systems can also have inequalities in addition to equations, where the parameters in δ are small intervals rather than exact values. In this case, the definition of rigidity is approximate; the solutions are isolated, small, full-dimensional connected components.

Other generic constraint system or (hyper)graph properties are mentioned here. A constraint (hyper)graph G is *independent* if its appropriate rigidity matrix of indeterminates has independent rows (i.e. the determinant of some square submatrix is not identically zero). It is *isostatic* (*minimally rigid*, *well-constrained*) if it is both rigid and independent. It is *flexible* if it is not rigid, *underconstrained* if it is independent and not rigid, or *overconstrained* if it is not independent.

Defining the combinatorial independence of a subset of edges $E' \subseteq E$ to be the independence of corresponding rows in the rigidity matrix of indeterminates, we obtain the *rigidity matroid* of a constraint (hyper)graph $G = (V, E)$. There are various results on combinatorial characterization of independence, rigidity, and rigidity matroids for different types of (hyper)graphs. For bar–joint rigidity matroids, the famous Laman's theorem (Laman, 1970) states that the underlying graph is isostatic if and only if $|E| = 2|V| - 3$ and $|E'| \leq 2|V'| - 3$ for every induced subgraph with at least 2 vertices. The result by Lovasz and Yemini (1982) shows that all 6-vertex-connected graphs are rigid in the plane. For bar–body rigidity matroids, Tay (1976) proved that the underlying multigraph is isostatic if and only if it can be decomposed as 3 edge disjoint spanning trees. White and Whiteley (1987) gave the same characterization using a different technique to study the algebraic–geometric conditions of genericity, called pure condition. Lee et al. (2007) defined the (k, l)-*sparsity matroid*, where a hypergraph G is called (k, l)-*sparse* if $|E'| \leq k|V'| - l$ for any induced subgraph (V', E') with at least 2 vertices, and (k, l)-*tight* if it is (k, l)-sparse and $|E| = k|V| - l$. In general, given a d-uniform hypergraph, a (k, l)-sparsity condition is matroidal as long as $l \leq dk - 1$.

In this paper, a *qusecs* is any independent geometric constraint system of one of 3 types: bar–joint (defined formally in Section 2.1), *body–hyperpin* (defined formally in Section 5), and *pinned line-incidence* (defined formally in Section 6).

We note that the remainder of this section and Sections 3 and 4 we only consider bar–joint qusecs and graphs. Relevant formal analogies for the other 2 types of qusecs and (hyper)graphs are given along with their materials applications in Sections 5 and 6.

2.2. Decomposition–recombination (DR-)plans

Definition 1 *(DR-plan).* A *decomposition–recombination (DR-)plan* (Hoffman et al., 2001a) of graph G is defined as a forest that has the following properties:

1. Each node represents a rigid subgraph of G.[4]
2. A root node is a vertex-maximal rigid subgraph of G.
3. A node is the subgraph of G induced by the union of its children.
4. A leaf node is a single edge.

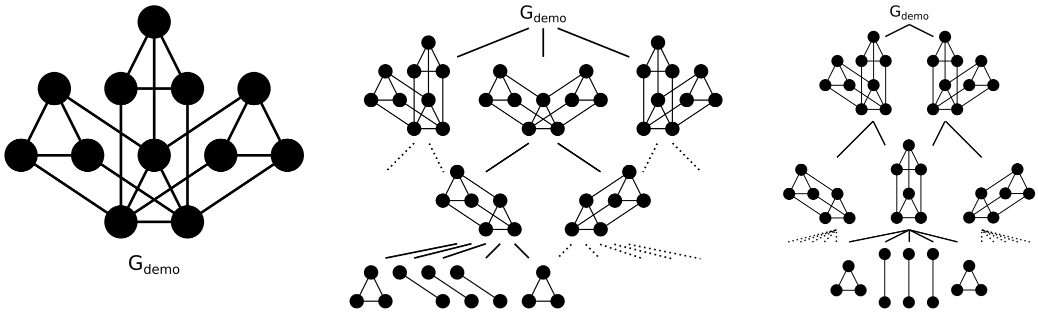
Note that this definition permits the same rigid subgraph to appear in multiple nodes of the DR-plan. Not permitting such duplication would, in general, require the DR-plan to be defined as a directed acyclic graph instead of a forest.

Definition 2 *(Complete DR-plan).* A DR-plan is *complete* if it satisfies an additional property: for an internal node C, its children are all of the rigid vertex-maximal proper subgraphs of C (which makes Property 3 of a DR-plan implicit).

Definition 3 *(Optimal DR-plan).* A DR-plan is *optimal* if it minimizes the maximum fan-in over all nodes in the forest.

[3] We refer to these as properties of the constraint system or as properties of the underlying (hyper)graph interchangeably.

[4] Nodes will be referred to interchangeably as "the node that represents or contains the (sub)graph C" and as simply "C".



(a) (b) (c)

Fig. 2. (a) A graph, G_{demo}, used to illustrate concepts throughout this and the next section. (b) The complete DR-plan of G_{demo}. Dashed lines indicate that the children repeat the same pattern as the others shown on this level. The children of triangles (3 edges) are omitted. (c) The canonical DR-plan of G_{demo}, which is optimal (see Section 3). The children of triangles are omitted.

Remark 4. More than one node (leaf) in a DR-plan forest may represent the same subgraph (vertex) of G. For a given graph, there could be exponentially many DR-plans – and even optimal DR-plans – in the size of the graph. A complete DR-plan is unique but may not be (and is usually not) optimal. DR-plans of self-similar graphs are self-similar.

See Figs. 2, 3, 6, and 10 for examples of DR-plans and how their properties relate to each other.

2.3. Previous work on DR-plans

We now briefly survey existing techniques for detecting rigidity and creating DR-plans of 2D constraint systems. The limitations of these techniques directly motivate the contributions of the next section.

2.3.1. Finding (vertex)-maximal, generically rigid subsystems

Fast, graph-based algorithms exist (pebble-game Jacobs and Hendrickson, 1997; Hoffmann et al., 1997; Jermann et al., 2006; Lee and Streinu, 2007), for locating all maximal, *generically rigid* subsystems (formally defined in Section 2.1). When the input itself is rigid, these algorithms do nothing, i.e. compute the identity function.

However, both for self-similar or just aperiodic 2D qusecs, it is imperative to recursively decompose rigid systems into their rigid subsystems, down to the level of geometric primitives, in order to understand or design properties at all scales, such as (formally defined in 2.1) *rigidity*, *flexes*, distribution of *external stresses*, boundary conditions for *isostaticity*, as well as behavior under constraint variations.

2.3.2. Optimal recursive decomposition (DR-planning)

Recursive decomposition of geometric constraint systems has been formalized (Hoffman et al., 2001a, 2001b) and well-studied (Lomonosov, 2004; Sitharam, 2005; Jermann et al., 2006) as the *Decomposition–Recombination (DR-)planning* problem (formally defined in Section 2.1). For the abovementioned classes of 2D qusecs, generic rigidity is a combinatorial property and hence each level of the decomposition should, in principle, be achievable by a graph-based algorithm without involving the geometric information in the constraint system. Since many such decompositions can exist for a given constraint system, criteria defining desirable or optimal DR-plans and DR-planning algorithms were given in Hoffman et al. (2001a). We conjecture (in Section 7.1) that one such decomposition, a version of Frontier (Hoffman et al., 2001a, 2001b; Lomonosov, 2004; Sitharam, 2005), which is a bottom-up, polynomial time method, also generates optimal DR-plans for independent systems.

However, for overconstrained 2D qusecs, even when restricted to bar–joint systems, the optimal DR-planning problem was shown to be NP-hard (Lomonosov, 2004; Sitharam, 2005). The NP-hardness of the optimal DR-planning problem for 2D bar–joint graphs is partly the consequence of possibly exponential number of DR-plans. On the other hand, although the complete DR-plan is unique it could have large average fan-in and exponentially many nodes making it far from optimal.

2.3.3. DR-plans for special classes and with other criteria

For a special class of 2D qusecs, namely *tree-decomposable* systems (Owen, 1991; Fudos and Hoffmann, 1997; Joan-Arinyo et al., 2004) common in computer aided mechanical design (which includes ruler-and-compass and Henneberg-I constructible systems), all DR-plans turn out to be optimal. This satisfies the Church–Rosser property, leading to highly efficient DR-planning algorithms. For general 2D qusecs, alternate criteria were suggested such as *cluster minimality* requiring parent systems to have a minimal set of at least 2 rigid proper subsystems as children (i.e. the union of no proper subset of size at least 2 child subsystems forms a rigid system); and *proper maximality*, requiring child subsystems to be maximal rigid proper subsystems of the parent system. See Section 2.1 for formal definitions.

While polynomial time algorithms were given to generate DR-plans meeting the cluster minimality criterion (Lomonosov, 2004), no such algorithm is known for the latter criterion.

3. Main result: canonical DR-plan, optimality, and algorithm

The goal of this section is to develop an $O(n^3)$ time complexity algorithm for finding an optimal DR-plan. To this end, we first introduce a canonical DR-plan to capture those aspects of an optimal DR-plan that mimic the uniqueness of a complete DR-plan, and we show that the nonunique aspects do not affect optimality for independent (underconstrained or isostatic) graphs. While useful for proving optimality, the canonical DR-plan is difficult to work with algorithmically. Therefore, we define the pseudosequential DR-plan which is derived from the canonical and is still optimal. However, the pseudosequential DR-plan lacks the essential uniqueness of the complete DR-plan. From the pseudosequential, we derive the sequential DR-plan, which can be found in the same time complexity and is essentially unique.

In this section and in Section 4, any reference to a graph G without further specification is assumed to be isostatic (i.e. well-constrained or (k,l)-tight). Furthermore, we only consider unions and intersections of graphs that are induced subgraphs of a single parent graph G. In this case unions and intersections are well defined. For example, the union (resp. intersection) of $F_1 = (V_1, E_1)$ and $F_2 = (V_2, E_2)$ is the subgraph of G induced by $V_1 \cup V_2$ (resp. $V_1 \cap V_2$).

3.1. Canonical DR-plan

Definition 5 *(Canonical DR-plan).* A *canonical DR-plan* is a DR-plan that satisfies the additional two properties:

1. Children are rigid vertex-maximal proper subgraphs of the parent.
2. If all pairs of rigid vertex-maximal proper subgraphs intersect trivially then all of them are children, otherwise exactly two that intersect non-trivially are children.

Definition 5 gives the canonical DR-plan a surprisingly strong Church–Rosser property, which is made explicit in Theorem 6, the main result of this section.

Theorem 6 *(Canonical is optimal). A canonical DR-plan exists for a graph G and any canonical DR-plan is optimal if G is independent.*

Proof. We show the existence of a canonical DR-plan by constructing it as follows:

Let P_G^{com} be the complete DR-plan of the rigid 2D bar–joint graph G. For all nodes C with children $C_1, \ldots, C_N$ retain children nodes according to the following rules:

(a) If $C_i \cap C_j$ is trivial then retain all $C_1, \ldots, C_N$ as children.
(b) If $C_i \cap C_j$ is rigid then select any two out of $C_1, \ldots, C_N$ as children.

This directly satisfies Properties (1) and (2) of a canonical DR-plan (see Definition 5), because all the nodes in P_G^{com} are rigid vertex-maximal proper subgraphs, which we shorten to *clusters*. To show that a canonical DR-plan is, in fact, a DR-plan: for Rule (a) above, since we start with a complete DR-plan, if we preserve all the children it is still a DR-plan; for Rule (b) above, we know that the union must be rigid as well and it cannot be anything other than C, otherwise we would have found a larger rigid proper subgraph of C, contradicting vertex-maximality.

Note that if we begin with an isostatic graph, “rigid” can be replaced with “isostatic” throughout the construction and preserve the above properties. The rigid proper subgraphs of an isostatic graph must be isostatic themselves.

Next we show that a canonical DR-plan is optimal.

First, note that any DR-plan of G, P_G, without Property (1) of a canonical DR-plan can always be modified (by introducing intermediate nodes) to satisfy Property (1) without increasing the max fan-in, since any child (a rigid proper subgraph) of node C in P_G is the subgraph of some cluster of C. Thus, without loss of generality, we can assume that an optimal DR-plan satisfies Property (1) of a canonical DR-plan.

The proof of optimality of a canonical DR-plan is by induction on its height. The base case trivially holds for canonical DR-plans of height 0, i.e. for single edges. The induction hypothesis is that canonical DR-plans of height t are optimal for the root node. For the induction step consider a canonical DR-plan P_C^{can} rooted at node C with height $t+1$. Notice that P_C^{can} contains a canonical DR-plan $P_{C_i}^{can}$ for the graphs C_i corresponding to each of C’s descendant nodes. Thus, from the induction hypothesis, we know that the $P_{C_i}^{can}$ is optimal for C_i.

To carry out the induction step, it is sufficient to demonstrate a set of nodes S (of height at most t) that must be present in any DR-plan P_C of graph C that satisfies Property (1), including a known optimal one; and furthermore, for any such DR-plan P_C, either (Claim 1) S must be the set of children of C; or (Claim 2) all the ancestors A of S that are descendants of C have the minimum possible fan-in of 2.

We show the two claims below. The first claim is that for a node C whose clusters have trivial pairwise intersections, any DR-plan of C that satisfies Property (1) must also satisfy Property (2) at C, i.e. the set of children S of C consists of all clusters of C. Because this is the only choice, it is the minimum fan-in at C for any DR-plan for C with Property (1), including a known optimal one. The second claim shows that in the case of nodes C whose rigid, vertex-maximal proper subgraphs have non-trivial pairwise intersections, every canonical DR-plan of C that uses any possible choice of two such subgraphs of C as children results in a minimum possible fan-in of 2 in the ancestor nodes A leading to the *same maximal antichain S of descendants D of C*. The antichain is maximal in the partial order of rigid subgraphs of C under containment. I.e. S satisfies the property that every proper vertex-maximal rigid subgraph of C is a superset of some D in S; this follows from properties of maximal antichains that no element of S is contained in the union of other elements of S; and the union of elements of S is C. Thus any DR-plan that satisfies Property (1) and hence contains two or more of the rigid vertex-maximal proper subgraphs of C as children must also contain every element of S. The two claims complete the proof of the induction step and thus the proof that every canonical DR-plan is optimal.

Proof of Claim 1. Let the set of clusters of node C be $C_1, \ldots, C_N$. If the pairwise intersection of clusters is trivial, all of the clusters must be children of C in an optimal DR-plan

We prove this claim by showing that the union of no subset of the children can be C, thereby requiring all of them to be included as children.

We prove by contradiction. Assume to the contrary that there is a strict subset S of the clusters such that U, the union of all elements in S, is isostatic. If $U \neq C$, then we found a larger proper subgraph contradicting vertex-maximality of the clusters in set S. So, it must be that $U = C$. However, since $C_i \cap C_j$ is trivial then for $C_k \notin S$ we know, by Lemma 8, Item 3, $U \cap C_k$ must be one or more vertices, i.e. disconnected trivial subgraphs. By definition of a DR-plan, $C_k = C \cap C_k$ and we know that $U = C$ so $C_k = U \cap C_k$. Thus, C_k is (i) a collection of disconnected vertices, and (ii) an isostatic subgraph of C, which is impossible. As C is isostatic, this means the union of no proper subset of $C_1, \ldots, C_N$ is isostatic, nor is it equal to C, proving Claim 1.

Furthermore, since a canonical DR-plan has nodes with proper rigid *vertex-maximal* subgraphs as children, if, as in this case, their pairwise intersection is trivial, it follows that any node has at most as many children as a DR-plan without this restriction, because the union of the children must contain all edges of the parent. Therefore, the canonical DR-plan is the optimal choice in this case of trivial intersections.

Proof of Claim 2. Let the set of clusters of node C be $C_1, \ldots, C_N$. If some pair of clusters has an isostatic (non-trivial) intersection, then choosing any two as children (minimum possible fan-in) will result in the same maximal antichain of descendants of node C.

To prove Claim 2, notice that if $C_i \cap C_j$ is isostatic, then, by Observation 7, $C_i \cup C_j$ is also isostatic. This means that, by Lemma 8, Point 2, the union of any two children of C is C itself. Thus, any two children can be chosen to make a canonical DR-plan and that is the minimum fan-in possible for a node of the DR-plan.

However, to guarantee that any two are the *optimal* choice, it must ensure minimum fan-in over all descendants leading up to a common maximal antichain S of subgraphs.

Let I denote the intersection of all the clusters; we call this the *core*. Let R_i be the graph induced by the edge set of C minus the edge set of C_i; we call these the *appendages*.[5] Note that C is the core plus all appendages, and cluster C_i is the core plus all appendages except R_i. Suppose C_i and C_j, where $i \neq j$, are taken to be the children of node C. The $N - 1$ clusters of C_i are the core plus all appendages except R_i and R_j, for each $j \neq i$. The pairwise intersection of any of these clusters of C_i will clearly be isostatic, so any two of them are viable children of node C_i. Beginning with node C, this pattern repeats for $N - 1$ levels. Every node in this subtree rooted at C has a fan-in of two (the minimum possible) up through this level. At level $N - 1$, we have a set of nodes where each node is the core plus some appendage (with every appendage appearing at least once). Thus, regardless of the sequence of choices of C_i and C_j, and of their descendants at each level, the DR-plan has the optimal fan-in of two for every node for $N - 1$ levels, and the collection of last level nodes contain the same maximal antichain of subgraphs (for all choices). □

This proof of the theorem relies on the following crucial observation and lemma. These will be used again in the application sections (Sections 5 and 6) of the paper, with modifications to work for other types of qusecs.

Observation* 7. If F_i and F_j are isostatic subgraphs of an independent graph then the following hold: (1) $F_i \cup F_j$ is not trivial; (2) $F_i \cup F_j$ is underconstrained if and only if $F_i \cap F_j$ is trivial; (3) $F_i \cup F_j$ is isostatic if and only if $F_i \cap F_j$ is isostatic; and (4) $F_i \cap F_j$ is not underconstrained.

The following key properties hold at the nodes of a canonical DR-plan.

Lemma* 8. *Let C be an isostatic node of a canonical DR-plan with distinct children $C_1, C_2, \ldots, C_N$. Assume $i \neq j$. Then*

1. *$C_i \cup C_j$ is isostatic if and only if $C_i \cup C_j = C$.*
2. *If $C_i \cup C_j$ is isostatic, then for all k, $C_i \cup C_k$ is isostatic. Alternatively, if $C_i \cup C_j = C$, then for all k, $C_i \cup C_k = C$.*
3. *If $C_i \cap C_j$ is trivial, then for all k, $C_i \cap C_k$ is trivial.*

[5] *Core* and *appendage* are used in Section 3.2 and are more formally defined in Definitions 12 and 20.

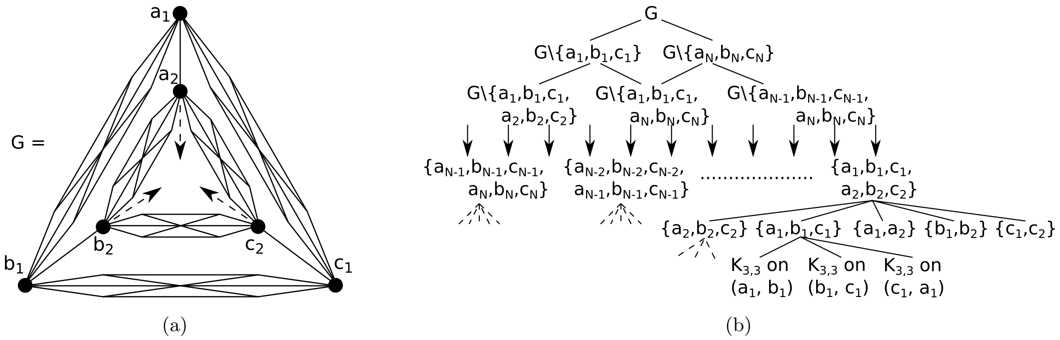


Fig. 3. (a) A sequence of doublets ($C_2 \times C_3$) intersecting on triangles, where the edges of the triangles are replaced by $K_{3,3}$'s. This pattern continues inwards for a total of N triangles, indicated by the dashed arrows. (b) A canonical DR-plan of G, drawn as a DAG. $G \setminus \{a_i, b_i, c_i\}$ is shorthand for G difference those nodes and all of the nodes in the corresponding $K_{3,3}$ subgraphs. Below the third level, the obvious pattern continues until only the individual doublets are present (fourth level) with the ellipses indicating the remaining doublets between those shown. Decomposition of one of these doublets is shown. The dashed lines indicated that this exact decomposition (of the similar nodes on the level) is repeated. Further decomposition of $K_{3,3}$ subgraphs into the separate 9 edges is omitted from the figure.

Remark 9. The first item in the above lemma generalizes to the union of any number of children, $C_1, \ldots, C_k$, resulting in the desirable property of *cluster minimality* (defined in Hoffman et al., 2001a and in Section 2.3) holding for canonical–optimal DR-plans.

Example 10 *(DR-plan for self-similar structure).* This example details the decomposition of the graph in Fig. 3, a canonical DR-plan of G. It begins with the whole (isostatic) graph as the root. The graph G has only two isostatic vertex-maximal subgraphs: G without the outermost triangle composed of $K_{3,3}$ graphs (triangle 1) and G without the inner triangle (triangle N). These intersect on G without triangle 1 and N which is clearly isostatic. As explained in the proof of Theorem 6, since there are only 2 possible children, their intersection must be a node 2 levels below the parent. As expected, it is on the third level, as a child of both of G's children.

Both of G's children are similar to G, but containing only $N - 1$ triangles. Therefore, the canonical DR-plans of these children follow the same pattern. This continues downward until the individual doublets are reached (there will be multiple occurrences of the same doublets at this level, but they can be represented as the same node in a DAG).

Further decomposition of one of these doublets is shown. The three edges between the triangles and the triangles themselves all intersect trivially pairwise. By Theorem 6, part 1, they must all be children in the DR-plan. Similarly, the triangles decompose into their three trivially intersecting $K_{3,3}$'s. Then the $K_{3,3}$ subgraphs decompose into their separate 9 edges.

The self-similar nature of this graph is evident in the canonical DR-plan. Many structures are repeated throughout the DR-plan, allowing for shared computation in both decomposition and recombination.

3.2. Algorithm

The algorithm for finding an optimal DR-plan relies on key structural properties of canonical DR-plans that are revealed by the proof of Theorem 6. We begin by redefining the canonical DR-plan in a recursive manner and also by recursively defining the new *pseudosequential DR-plan* to highlight the similarities. We show that this pseudosequential DR-plan is optimal (i.e. has the smallest maximum fan-in over all nodes) and has at most the same overall size (i.e. number of unique nodes) as a canonical DR-plan, and is in general smaller. By ensuring that no two nodes contain the same subgraph, the pseudosequential DR-plan is more malleable for algorithmic purposes.

Definition 11 *(Canonical DR-plan).* A *canonical DR-plan* of an isostatic G is recursively defined as follows:

1. Base case: When G is a single edge, the canonical DR-plan for G is G itself.
2. In case the pairwise intersections of the proper vertex-maximal rigid subgraphs C_i of G are all trivial, take the children of G to be the roots of the canonical DR-plans for C_i.
3. In case there are two proper vertex-maximal rigid subgraphs C_i and C_j of G with non-trivial intersection, take the children of G to be the roots of the canonical DR-plans for C_i and C_j.

Now we define the pseudosequential DR-plan in a manner analogous to Definition 11.

Definition 12 *(Pseudosequential DR-plan, appendage, partner).* A *pseudosequential DR-plan* of an isostatic G is recursively defined as follows:

1. Base case: When G is an edge, the pseudosequential DR-plan for G is G itself.
2. In case the pairwise intersections of the proper vertex-maximal rigid subgraphs C_i of G are all trivial, take the children of G to be the roots of the pseudosequential DR-plans for C_i.
3. In case there are two proper vertex-maximal rigid subgraphs C_i and C_j of G with non-trivial intersection, take the children of G to be the roots of pseudosequential DR-plans for $C_j \setminus C_i$ (called an *appendage*), and C_i (called its *partner*).

Remark 13 *(Size of pseudosequential DR-plan).* Any rigid subgraph appears in at most one node of a pseudosequential DR-plan. Also, if an edge e of G belongs in the rigid subgraph at any two nodes A and B of a pseudosequential DR-plan, then either $A \subset B$ or $B \subset A$. Since the leaves of pseudosequential DR-plan tree are the $O(|V(G)|)$ edges of the independent input graph, this implies the size of the DR-plan is $O(|V(G)|)$.

Observation 14 *(Pseudosequential is optimal).* Any pseudosequential DR-plan of an independent graph is optimal.

Proof. We need to show the statement: (*) the max fan-in (i.e. number of children) of any pseudosequential DR-plan for an independent graph G is no larger that of some canonical DR-plan for G. By Theorem 6, this would imply that the pseudosequential DR-plan is optimal.

Recall that both pseudosequential and canonical DR-plans are defined recursively. We will prove the statement (*) by induction on the height of a pseudosequential DR-plan for G. The base case (height of 0, i.e. single edges) trivially holds. Induction hypothesis: (*) holds for independent graphs G with pseudosequential DR-plans of height h.

Induction step: a pseudosequential DR-plan of height $h+1$ rooted at a node G consists of pseudosequential DR-plans (of height at most h) rooted at the children of G, which we call the set C. It is sufficient to show (a) the children C will exist somewhere in some canonical DR-plan, and (b) $|C|$ is less than or equal to the max fan-in of some canonical DR-plan. Given (a), then by the recursive definition of canonical DR-plans, these nodes are the roots of canonical DR-plans, thus the induction hypothesis applies to the nodes in C. Additionally given (b), the proof of the induction step of (*) is complete.

Case (1): When the isostatic vertex-maximal proper subgraphs of G have trivial intersections, the children of G in any canonical DR-plan are the same set C of children in any pseudosequential DR-plan. Thus both conditions (a) and (b) are immediately satisfied.

Case (2): When the isostatic vertex-maximal proper subgraphs of G have non-trivial intersections, the children C are the isostatic components of some appendage, A_i, and its partner. The partner will be a child of G in some canonical DR-plan. Furthermore, in any canonical DR-plan there will be some node $N-1$ levels below G (where N is the number of appendages) containing the core (the common intersection of all the isostatic vertex-maximal proper subgraphs of G) plus appendage A_i. In some canonical DR-plan, the children of this node will be the components of A_i and the core. This node has the same fan-in as G in the pseudosequential DR-plan, satisfying (b), and, along with the partner (a child of G), shows that some canonical DR-plan has all the nodes in C, thereby satisfying (a). □

Definition 15 *(Branch).* Given a pseudosequential DR-plan P_G for an isostatic graph G, and an edge $e \in G$, the $\textsc{Branch}(G, e, P_G)$ is the subtree of the DR-plan consisting of the path from the root containing G to the leaf containing e, together with the children of all the nodes on this path, which are the leaves of $\textsc{Branch}(G, e, P_G)$. See Fig. 4 for examples.

Observation 16 *(Pseudosequential DR-plan recursively from branches).* A pseudosequential DR-plan P_G for an isostatic graph G is obtained from $\textsc{Branch}(G, e, P_G)$ by recursively attaching to each of its leaves L a pseudosequential DR-plan P_L for L.

The next two lemmas are the crux of our $O(|V|^3)$ algorithm to find a pseudosequential DR-plan.

Lemma 17 *(Branch leaves from components). Let G be isostatic, e be an edge in G, and* $\textsc{Components}(G \setminus e)$ *be the set of maximal rigid components of $G \setminus e$. Then there is a pseudosequential DR-plan P_G for G such that* $\textsc{Components}(G \setminus e)$ *is exactly the set of leaves of* $\textsc{Branch}(G, e, P_G)$ *(minus e itself).*

Proof. Follows from the structure of a pseudosequential DR-plan and the definition of $\textsc{Branch}(G, e, P_G)$. □

Lemma 18 *(Branch from branch leaves). For an isostatic graph G containing an edge e, the* $\textsc{Branch}(G, e, P_G)$ *for a pseudosequential DR-plan P_G can be constructed from* $\textsc{Components}(G \setminus e)$*, i.e. from the set of leaves of* $\textsc{Branch}(G, e, P_G)$*, by carrying out – for each leaf L – one computation of* $\textsc{Components}(G \setminus f)$*, where f is any edge in L.*

Proof. First note that in order to obtain $\textsc{Branch}(G, e, P_G)$ from the set of subgraphs at its leaves, it is sufficient to find the subgraphs at the nodes along the path from G to e in P_G. Once these non-leaf nodes of the branch are known, the branch leaves can be organized by parent node (and level) thereby obtaining the branch.

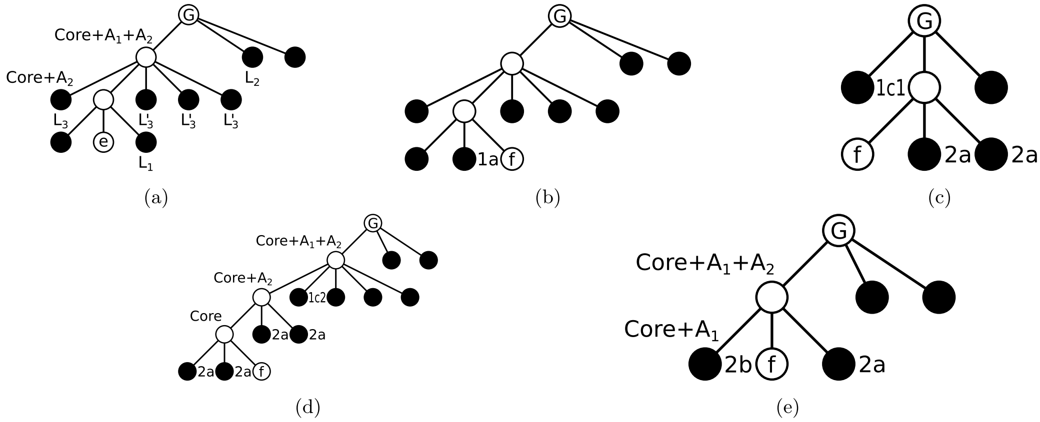


Fig. 4. This figure illustrates the notion of a branch, how it relates to the components of $G \setminus e$, and the variety of cases discussed in the proof of Lemma 18. Each subfigure is $\textsc{Branch}(G, e, P_G)$ for some edge e (resp. $\textsc{Branch}(G, f, P_G)$), the white nodes are the path from G to e (resp. f), and the black nodes are the set $\textsc{Components}(G \setminus e)$ (resp. $G \setminus f$). In subfigures (b), (c), and (d), nodes are labeled with the case they fall under in the proof of Lemma 18 if they are chosen to be D (given the choice of L discussed below), any unlabeled node falls under Case (1b). (a) has nodes labeled L_1 and L_2 (L'_1 will be discussed later), from which different edges f are taken in subsequent figures. Note that the node labeled 'Core+A_1+A_2' is a subgraph of G that has 2 isostatic vertex-maximal proper subgraphs intersecting non-trivially. The core is the intersection of these subgraphs and A_i is an appendage. (b) uses $f = L_1$. (c) uses an arbitrary $f \in L_2$. (d) uses an f in the core in L_3. (e) uses an f in the appendage A_1 in L_3. Note that this contains a node D that has Case (3). With this D, $D \cup L_3$ is 'Core+A_1+A_2' and $D \cap L_3$ is 'Core'. The leaves labeled L'_3 in (a) are the leaves that will allow us to find the node on the path from G to e that is their sibling (the parent of e).

For a leaf L and edge $f \in L$, we can classify the component $D \in \textsc{Components}(G \setminus f)$ as being one of the following cases:

Case (1): D and L are edge disjoint, and D contains (1a) no elements, (1b) exactly one element, or (1c) more than one element of $\textsc{Components}(G \setminus e)$. Case (2): D and L have non-empty intersection of edge sets, and D is (2a) a proper subgraph or (2b) not a subgraph of L. Note that this list is exhaustive. If D and L are edge disjoint, D cannot contain subgraphs of elements of $\textsc{Components}(G \setminus e)$. If D and L have non-empty intersection of edge sets, D cannot be exactly L.

Now observe that it is impossible for all $D \in \textsc{Components}(G \setminus f)$ to be Case (1b), as this would imply $f = e$.

Case (1a) implies that $D = e$, and is therefore a sibling of L and an element on the path from G to e. Furthermore, we can classify the rest of the leaves (i.e. $\textsc{Components}(G \setminus e)$) as children of the ancestors of D.

Similarly, Case (1c) implies that D is a sibling of L on the path from G to e. It also allows us to partition the leaves as either children of L (those components which are subgraphs of D) or as children of the ancestors of D. This case occurs when the siblings of L in (any pseudosequential DR-plan) P_G have either (1c1) trivial pairwise intersection or (1c2) non-trivial pairwise intersections, but the edge f is in their common intersection, i.e. the core.

Case (2a) implies that D is a leaf in $\textsc{Branch}(L, f, P_L)$ to be used in the next level of recursion of the algorithm.

Case (2b) implies that D is the partner to the appendage containing f. Node D will also contain some other leaves from $\textsc{Components}(G \setminus e)$, namely the siblings of L and their descendants. In this case, $D \cup L$ is a node along the path from G to e with the leaves contained in L being its descendants, L being its child, and all other components in $\textsc{Components}(G \setminus e)$ being children of the ancestors of $D \cup L$ in $\textsc{Branch}(G, e, P_G)$. In addition, $D \cap L$ is a leaf of $\textsc{Branch}(L, f, P_L)$ to be used in the next recursion level of the algorithm.

One subtle obstacle to overcome in Case (2b) is that the newly found node along the path from G to e is the parent of L as opposed to a sibling of L as in Case (1a) and (1c). On the surface, this is problematic because the sibling D' of L along the path from G to e may never be found. However a closer inspection of Case (2b) reveals that L is the partner of an appendage A containing e, and A, being underconstrained by Observation 7 and Lemma 8, must have maximal rigid components with nontrivial intersections, which means that there must be another sibling of D' that is a leaf L' of $\textsc{Branch}(G, e, P_G)$. Hence for some edge f' in L', $\textsc{Components}(G \setminus f')$ will find D' within Case (1a) or (1c).

Since each node along the path from G to e is found by the above procedure, the proof is complete. See Fig. 4 for examples of the cases. □

Theorem 19 *(Complexity of the algorithm). Computing a pseudosequential DR-plan for a graph G has time complexity* $O(|V(G)|^3)$.

Proof. The previous two lemmas and the $O(|V(G)|^2)$ complexity of the pebble game algorithm for computing $\textsc{Components}(G \setminus e)$ show that the procedure for computing $\textsc{Branch}(G, e, P_G)$ given G and e takes time $O(M|V(G)|^2)$, using M $\textsc{Components}()$ computations, where M is the number of leaves L of $\textsc{Branch}(G, e, P_G)$. Furthermore, in the process, the leaves of all the branches $\textsc{Branch}(L, f, P_L)$ have already been computed. Recursive computation of a pseudosequential DR-plan P_G as in Observation 16 now proceeds by computing branches $\textsc{Branch}(L, f, P_L)$ for each leaf L of $\textsc{Branch}(G, e, P_G)$. Since each node of P_G appears as the leaf L of a branch exactly once during the above recursive procedure, overall one

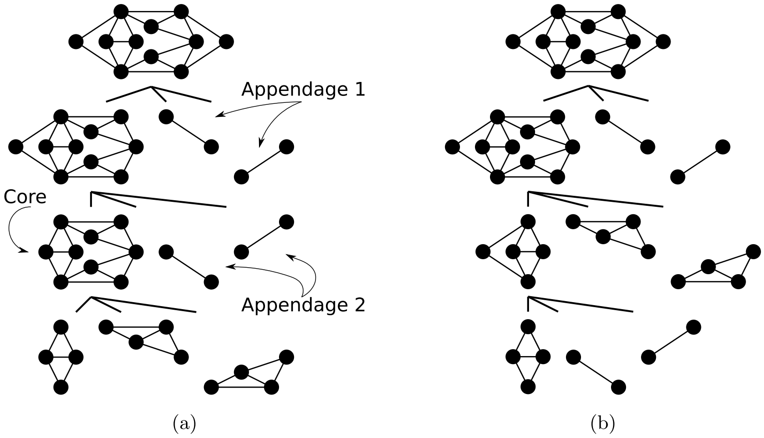


Fig. 5. Two pseudosequential DR-plans of the same isostatic input graph, further decomposition of the diamonds is omitted. (a) also satisfies Property (4) of Definition 20 and is therefore a sequential DR-plan as well. The core and the appendages are marked in the figure. (b) uses an alternative decomposition of the partner of appendage 1; since the isostatic vertex-maximal proper subgraphs of this node have non-trivial intersections as well, there is a choice in which appendage to decompose first. This choice pushes appendage 2 of the input graph further down the plan and the core is not present.

$O(|V(L)|^2)$ computation of Components() is carried out for each of the $O(|V(G)|)$ nodes L of P_G, resulting in an overall complexity of $O(|V(G)|^3)$. □

3.3. The 'essentially unique' sequential DR-plan

The class of sequential DR-plans is defined as a subset of pseudosequential and satisfies all properties discussed in the previous section. Furthermore, we show that a sequential DR-plan can always be obtained from a pseudosequential DR-plan. A sequential DR-plan is desirable because it is 'essentially' unique, and illustrates that the canonical DR-plan in fact retains the essential uniqueness of a complete DR-plan.

Definition 20 *(Sequential DR-plan, core).* A *sequential DR-plan* is a pseudosequential DR-plan that additionally satisfies the following requirement:

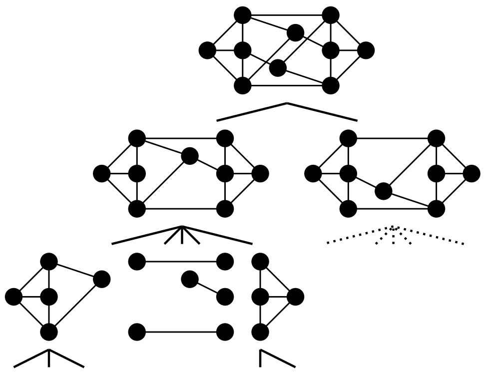
4. Let C be a node and C_s the set of its siblings in a sequential DR-plan. If there is a descendant D of C, with siblings D_s possessing the property that $C \cup C_s \setminus D_s$ is rigid, then for the contiguous sequence of nodes D' on the path from C to D, we require that $C \cup C_s \setminus D'_s$ be rigid, where D'_s is the set of siblings of D'. Here C, D and D' are the partners of the appendages C_s, D_s and D'_s.

Take an independent graph with a sequential DR-plan (i.e. Property (4) holds). Now, if the parent of $C \cup C_s$ falls under Property (2) of Definition 12, while $C \cup C_s$ falls under Property (3) of Definition 12, then the lowest descendant D as above is called the *core* of $C \cup C_s$.

For sequential DR-plans of independent graphs, Property (3) of Definition 12 appears asymmetric with respect to i and j; but in fact, i and j can be switched, using the appendage $C_i \setminus C_j$ instead. Let $C_1, C_2, \ldots, C_N$ be a complete list of proper vertex-maximal rigid subgraphs of C. Their pairwise intersections must all be nontrivial, and their common intersection, called a *core*, is isostatic by Lemma 8. Denote by A_i the ith appendage, where $A_i = C_j \setminus C_i$, for any $j \neq i$. Note that $C_i = C \setminus A_i$; and $C_i = \bigcap_i C_i \bigcup_{j\neq i} A_j$. Choosing a particular ordering of the C_i, i.e. choosing the maximal rigid components of a particular appendage, say A_1, and its partner C_1 to be the children of C simply pushes down the nodes corresponding to the appendages $A_2, A_3, \ldots, A_N$ to a lower level of the sequential DR-plan and the corresponding partners are created as $C_1 \cap C_2, C_1 \cap C_2 \cap C_3, \ldots$; the last appendage, A_N, will always have the core, $\bigcap_i C_i$, as its partner. Thus, we come to the following conclusion.

Remark 21 *(Essential uniqueness of sequential DR-plans).* Modulo the ordering of appendages and the corresponding partners, the sequential DR-plan is unique.

For an example of a pseudosequential DR-plan that is not sequential (i.e. Property (4) does not hold), see Fig. 5. Note that the core is not present in the pseudosequential DR-plan (Fig. 5b) and the order of the appendages cannot be easily changed. Nevertheless, we will now show that a sequential DR-plan can be found from any pseudosequential DR-plan in linear time (in the size of the input graph), making it a worthwhile tool in practice.



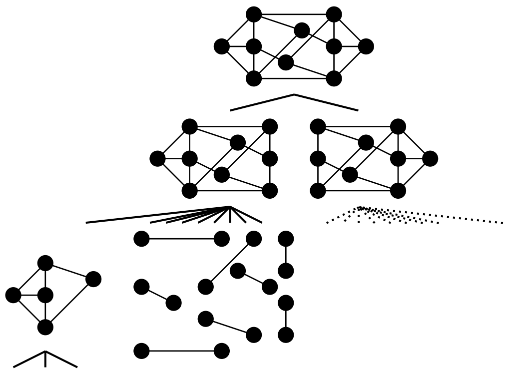


Fig. 6. Both figures are canonical and cluster-minimal DR-plans of the same singly overconstrained rigid graph. Further decomposition of the bottom level is omitted (indicated by tree edges) and dashed lines indicate a decomposition similar to the other nodes on the same level. (a) is an optimal DR-plan, with a fan-in of 5. (b) has a fan-in of 9 and is non-optimal, shown by the preceding counter-example.

Lemma 22 *(Pseudosequential to sequential DR-plan). Any pseudosequential DR-plan for an independent graph can be converted to a sequential DR-plan (satisfying Property (4)) in time $O(|V(G)|)$.*

Proof. For independent graphs, note that Properties (2) and (3) of Definition 12 automatically imply that the following holds for a pseudosequential DR-plan: for a node G that falls under Property (2), there is no child C of $G = C \cup C_s$ for which there is a descendant D with siblings D_s possessing the property that $C \cup C_s \setminus D_s$ is rigid; for a node G that falls under Property (3), such a descendant D exists for a unique child C of the node $G = C \cup C_s$, as well as for unique children of all D' on the path from C to D. Furthermore, for all such D', it automatically holds that $D' \cup D'_s \setminus D_s$ is rigid. We call this *Property (4′)* (since it is a parallel property of Property (4) of Definition 20).

Given a pseudosequential DR-plan of an independent graph where nodes with Property (3) of Definition 12 are labeled, we show how to enforce Property (4) of Definition 20, for a node $C \cup C_s$ with Property (3), i.e. when there is a descendant D with siblings D_s possessing the property that $C \setminus D_s \cup C_s$ is rigid.

Let D be any such descendant where Property (4) does not hold and let D^* (with siblings D^*_s) be the highest node along the path from C to D where $C \cup C_s \setminus D^*_s$ is not rigid. Since we know that $C \cup C_s \setminus D_s$ is in fact rigid by the hypothesis of Property (4), and since Property (4′) holds for D^*, i.e. $D^* \cup D^*_s \setminus D_s$, we “switch D^*_s with D_s” by setting up the partners and appendages along the path from D^* to D as follows.

At the level of each partner node D' on the path starting from D to a child of D^* the appendage becomes parent$(D')_s$ and the new partner becomes $D' \cup D'_s \setminus D_s$ (which we know to be rigid by Property (4′)). In particular, at the level of the partner node D the appendage becomes parent$(D)_s$ and the new partner remains D. However, at the next higher level the old partner node parent$(D) = D \cup D_s$ becomes the new partner $D \cup$ parent$(D)_s$ of the new appendage is parent(parent$(D))_s$.

At the level of the node D^*, the appendage becomes D_s and the partner becomes $D^* \cup D^*_s \setminus D_s$ (which we know to be rigid by Property (4)).

Once the above switch has been completed, moving D_s up the path from D to C, it is unnecessary to inspect any of the siblings of any of the nodes along this path. This is because C is the unique child of $C \cup C_s$ for which such a descendant D even exists, and the same uniqueness holds for every node that is on the path from C to D.

Note that in the same iteration, the algorithm can simultaneously perform the above process on all such descendants of C for which Property (4) does not hold, since they must all be on the same descending path from C. Once Property (4) has been enforced for all such descendants of C, the algorithm has found the core for $C \cup C_s$, namely the last node in a contiguous path of nodes D' starting down from C, for which $C \cup C_s \setminus D'_s$ is rigid. The order of appendages of all of these nodes are interchangeable (with appropriate partners) as indicated in Remark 21.

The algorithm then proceeds to the next node $C \cup C_s$ with Property (3), for which there is a descendant D with siblings D_s possessing the property that $C \setminus D_s \cup C_s$ is rigid and for which Property (4) does not hold.

This process continues until Property (4) always holds, resulting in a sequential DR-plan. It is clear that the algorithm visits any given node of the sequential DR-plan at most $O(1)$ times resulting in a linear time complexity. □

3.4. Overconstrained graphs and NP-hardness of optimal DR-planning

For overconstrained (not independent) graphs, a canonical DR-plan is still well-defined. However, it may be far from optimal. The proofs of Theorem 6, Observation 7, and Lemma 8 all fail for overconstrained graphs. It is important to note that, regardless whether the graph is overconstrained, if every node in a canonical DR-plan R has clusters whose pairwise intersection is trivial, then the DR-plan is the unique one satisfying Property (2), and since we know that there is an optimal

DR-plan that satisfies Property (2), R is in fact optimal. The problem arises when some node in a DR-plan has clusters whose pairwise intersection is non-trivial. In this case, an arbitrary choice of a pair of clusters as children of an overconstrained node in a canonical DR-plan may not result in an optimal DR-plan. This is in contrast to independent graphs, which, as shown in Theorem 6, exhibit the strong Church–Rosser property that any choice yields an optimal DR-plan. A good source of examples of overconstrained graphs with canonical DR-plans that are not optimal are graphs whose cluster-minimal DR-plans that are not optimal. The example shown in Fig. 6 is a canonical, cluster-minimal DR-plan that is not optimal; an optimal DR-plan is also shown in the figure. The root cause of the NP-hardness is encapsulated in this figure: because the different choices of vertex-maximal subgraphs for overconstrained input do not incur the same fan-in, finding the optimal DR-plan becomes a search problem with a combinatorial explosion of options.

As mentioned earlier, the Modified Frontier algorithm version given in Lomonosov (2004) runs in polynomial time and finds a cluster-minimal DR-plan for any graph. Similarly, the algorithm given above finds a canonical DR-plan also for any input graph. However neither of these DR-plans may be optimal for overconstrained graphs as shown in Fig. 6.

While the canonical DR-plan is optimal only if the input graph is independent, when there are only k overconstraints for some fixed k, we can still find the optimal DR-plan using a straightforward modification of the above algorithm. However, the time complexity is exponential in k.

This exponential growth of time complexity for overconstrained graphs is in fact captured in the proof of NP-hardness of optimal DR-planning in Sitharam (2005), Lomonosov (2004).

4. Recombination and problem relationships

In this section, we consider the *optimal recombination* problem of combining specific solutions of subsystems in a DR-plan into a solution of their parent system I (without loss of generality, at the top level of the DR-plan). In the case of isostatic qusecs, the parent system I is isostatic (the root of the DR-plan), and we seek *solution(s) (among a finite large number of solutions) with a specific orientation or chirality*. In the case of underconstrained qusecs the subsystems are the multiple roots of the DR-plan, the parent system I is underconstrained, and we typically seek an efficient algorithmic description of *connected component(s) of solutions with a specific orientation or chirality*.

The main barrier in recombination when given an optimal DR-plan (of smallest possible size or max fan-in) for a system S, is that the number of children of the root (resp. number of roots of the DR-plan) – and correspondingly the size and complexity of the (indecomposable) algebraic system I to be solved – could be arbitrarily large as a function of the size of S. This difficulty can persist even after optimal parametrization of the indecomposable system I as in Sitharam et al. (2010a) to minimize its algebraic complexity.

4.1. Previous work

We now briefly survey existing techniques for handling the complexity of recombination of DR-plans for qusecs. The limitations of these techniques directly motivate the contributions in this section.

4.1.1. Optimal recombination and solution space navigation

For the entire DR-plan, finding all desired solutions is barely tractable even if recombination of solved subsystems is easy for each indecomposable parent system in the DR-plan. This is because even for the simplest, highly decomposable systems with small DR-plans, the problem of finding even a single solution to the input system at the root of the DR-plan is NP-hard (Saxe, 1979) and there is a combinatorial explosion of solutions (Borcea and Streinu, 2004). Typically, however, the desired solution has a given orientation type, in which case, the crux of the difficulty is concentrated in the algebraic complexity of (re)combining child system solutions to give a solution to the parent system at any given level of the DR-plan. For fairly general 3D constraint systems, there are algorithms with formal guarantees that uncover underlying matroids to combinatorially obtain an optimal parameterization to minimize the algebraic complexity of the indecomposable parent (sub)systems that occur in the DR-plan (Sitharam et al., 2010a, 2010b; Sitharam, 2006), provided the DR-plan meets some of the abovementioned criteria.

However, the generality of these algorithms trades-off against efficiency, and, despite the optimization, the best algorithms can still take exponential time in the number of child subsystems (which can be arbitrarily large even for optimal DR-plans) in order to guarantee all solutions of a given orientation type, even for a single (sub)system in a DR-plan. They are prohibitively slow in practice. We note that, utilizing the DR-plan and optimal recombination as a principled basis, high performance heuristics and software exists (Sitharam et al., 2006) to tame combinatorial explosion via user intervention.

4.1.2. Configuration spaces of underconstrained systems

For underconstrained 2D bar–joint and body–hyperpin qusecs obtained from various subclasses of tree-decomposable systems, algorithms have been developed to complete them into isostatic systems (Joan-Arinyo et al., 2003; Sitharam, 2005; Gao et al., 2006; Sitharam and Gao, 2010) and to find paths within the connected components (Sitharam et al., 2011a; Hidalgo and Joan-Arinyo, 2011) of standard Cartesian configuration spaces. Most of the algorithms with formal guarantees leverage Cayley configuration space theory (Sitharam and Gao, 2010; Sitharam et al., 2011a, 2011b) to characterize subclasses of graphs and additional constraints that control combinatorial explosion, and provide faithful bijective representation of connected components and paths. These algorithms have decreasing efficiency as the subclass of systems gets bigger, with highest efficiency for underlying partial 2-tree graphs (alternately called tree-width 2, series-parallel, and K_4 minor avoiding), moderate efficiency for 1 degree-of-freedom (dof) graphs with low Cayley complexity (which include common linkages such as the Strandbeest, Limacon, and Cardioid), and decreased efficiency for general 1-dof tree-decomposable graphs. While software suites exist (Key Curriculum, 1995; Porta et al., 2014; Siemens, 1999; Todd, 2007), no such formal algorithms and guarantees are known for non-tree-decomposable systems.

4.2. Optimal modification for recombination

In the following, we formulate the problem of *optimal modification* of an indecomposable algebraic system I at some node of a (possibly optimal) DR-plan into a decomposable system with a small DR-plan (low algebraic complexity). Leveraging recent results on *Cayley configuration spaces*, our approach to the optimal modification problem achieves the following:

(a) *Small DR-plan.* We obtain a parameterized family of systems I_{λ_F} – one for each value λ_F for the parameters F, all of which have small DR-plans. Thus, given a value v for λ_F, the system I_v can potentially be solved or realized easily once the orientation type of the solution is known (when the DR-plan size is small enough).

(b) *Solution preservation.* Moreover, the union of solution spaces of the systems in the family I_{λ_F} is guaranteed to contain all of I's solutions.

(c) *Efficient search.* Finally, the so-called *Cayley* or distance parameter space λ_F is convex or otherwise easy to traverse in order to search for I's solution (or connected component) of the desired orientation type. For the case when the modification (number of Cayley parameters) is bounded, this approach provides an efficient algorithm for recombination. We first define the decision version of the problem of optimal modification for decomposition. The standard optimization versions are straightforward.

Optimal Modification for Decomposition (OMD) problem. Given a generically independent graph $G = (V, E)$ with no non-trivial proper isostatic subgraph (indecomposable) and 2 constants k and s, does there exist a set of at most k edges E_1 and a set of non-edges F such that the modified graph $H = (V, E \setminus E_1 \cup F)$ has a DR-plan of size at most s? The OMD_k problem is OMD where k is a fixed bound (not part of the input). We say that such a tuple (G, s) is a member of the set OMD_k. We loosely refer to graphs G as OMD with appropriately small k and s or OMD_k with appropriately small s.

It is immediately clear that indecomposable graphs G that belong in OMD_k for small k and s lend themselves to modification into decomposable graphs satisfying Criteria (a) and (b) above. However, it is not clear how Criterion (c) is met by OMD graphs. Before we consider this question, we discuss previous work on recombination of DR-plans.

4.3. Using convex Cayley configuration spaces

Next we provide the necessary background to describe a specific approach for achieving the requirements (a)–(c) mentioned above, by restricting the class of reduced graphs $G' = G \setminus E_1$ and their isostatic completions H in the above definition of the OMD problem, and using a key theorem of Convex Cayley configuration spaces (Sitharam and Gao, 2010). This theorem characterizes the class of graphs H and non-edges F (Cayley parameters), such that the set of vectors λ_F of attainable lengths of the non-edges F is always convex for any given lengths δ for the edges of H (i.e. over all the realizations of the bar–joint constraint system or linkage (H, δ) in 2 dimensions). This set is called the (2-dimensional) *Cayley configuration space* of the linkage (H, δ) on the Cayley parameters F, denoted $\Phi_F(H, \delta)$ and can be viewed as a "projection" of the Cartesian realization space of (H, δ) on the Cayley parameters F. Such graphs H are said to have *convexifiable Cayley configuration spaces for some parameters* F (short: H is *convexifiable*).

To state the theorem, we first have to define the notion of *2-sums* and *2-trees*. Let H_1 and H_2 be two graphs on disjoint sets of vertices V_1 and V_2, with edge sets E_1 and E_2 containing edges (u, v) and (w, x) respectively. A *2-sum* of H_1 and H_2 is a graph H obtained by taking the union of H_1 and H_2 and identifying $u = w$ and $v = w$. In this case, H_1 and H_2 are called *2-sum components* of H. A *minimal 2-sum component* of H is one that cannot be further split into 2-sum components. A *2-tree* is recursively obtained by taking a 2-sum of 2-trees, with the base case of a 2-tree being a triangle. A *partial 2-tree* is a 2-tree minus some edges. Partial 2-trees have an alternate characterization as the graphs that avoid K_4 minors, and are also called series-parallel graphs.

Theorem 23. *(See Sitharam and Gao, 2010.) H has a convexifiable Cayley configuration space with parameters F if and only if for each $f \in F$ all the minimal 2-sum components of $H \cup F$ that contain both endpoints of f are partial 2-trees. The Cayley configuration space $\Phi_F(H, \delta)$ of a bar–joint system or linkage (H, δ) is a convex polytope. When $H \cup F$ is a 2-tree, the bounding hyperplanes of this polytope are triangle inequalities relating the lengths of edges of the triangles in $H \cup F$.*

The idea of our approach to achieve the criteria (a)–(c) begins with the following simple but useful theorem.

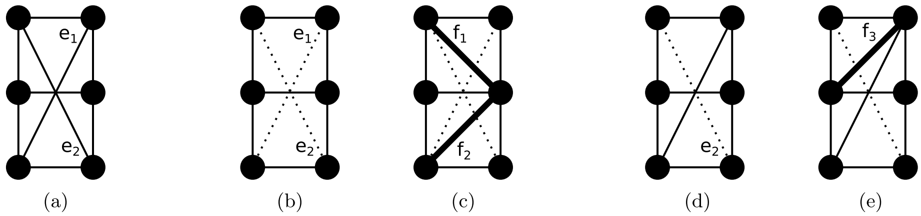


Fig. 7. (a) The $K_{3,3}$ with two labeled edges, e_1 and e_2. (b) The $K_{3,3}$ with e_1 and e_2 removed (dashed lines) and rearranged to illustrate that it is now a partial 2-tree. (c) The $K_{3,3}$ with $\{e_1, e_2\}$ removed and $\{f_1, f_2\}$ (bold lines) added to make a 2-tree, showing that the $K_{3,3}$ is at least OMD_2. (d) The $K_{3,3}$ with only e_2 removed (dashed line). (e) The $K_{3,3}$ with e_2 removed and f_3 (bold line) added to make a low Cayley complexity graph, showing that the $K_{3,3}$ is OMD_1.

Theorem 24. *Given an indecomposable graph G, let G' be a spanning partial 2-tree subgraph in G with k fewer edges than G. Then $(G, 2)$ belongs in the set OMD_k.*

Proof. The proof follows from the fact that 2-trees are well decomposable and have simple DR-plans of size 2. We know that G can be reduced by removing k edges to create a partial 2-tree G' which can then be completed to an (isostatic) 2-tree by adding some set of non-edges F. Thus the modified graph $H = G' \cup F$ has a DR-plan of size 2, proving the theorem. $\square$

We refer to such graphs G in short as *k-approximately convexifiable*, where the reduced graphs G' and isostatic completions H are convexifiable. As observed earlier, since graphs such as G are in OMD_k, Criteria (a) and (b) are automatically met for small enough k. Criterion (c) is addressed as described in the following efficient search procedure which clarifies the dependence of the complexity on the number and ranges of Cayley parameters F.

Theorem* 25 *(Efficient search). For an indecomposable, k-approximately convexifiable graph $G = (V, E)$, let $G' = (V, E' = E \setminus D)$ be a spanning partial 2-tree subgraph where $|D| \leq k$. Let F be a set of non-edges of G such that $H = (V, E' \cup F)$ is a 2-tree. Each solution p (or connected component of a solution space) of (G, δ) of an orientation type σ_p can be found in time $O(\log(W))$ where W is the number of cells of desired accuracy (discrete volume) of the convex polytope $\Phi_F(G', \delta'_E)$. The (discrete) volume W is exponential in $|F|$ and polynomial in the (discrete range) of the parameters in F.*

Note that a major advantage of the convex Cayley method is that it is completely unaffected when δ are intervals of values rather than exact values (Sitharam and Gao, 2010).

Example 26 *(Using Cayley configuration space).* A graph $G = K_{3,3}$ cannot be decomposed into any nontrivial isostatic graphs, i.e. its DR-plan has a root and 9 children corresponding to the 9 edges. Solving or recombining the system (G, δ) corresponding to the root of this DR-plan involves solving a quadratic system with 8 equations and variables. Instead of simultaneously solving this system, we could instead use the fact that $G = K_{3,3}$ is in OMD_2: remove the edges e_1, e_2 in Fig. 7 to give a partial 2-tree G'. Now add the non-edges f_1, f_2 to give a 2-tree H with a DR-plan of size 2. The Cayley configuration space $\Phi_f(G', \delta_{E\setminus e})$ is a single interval of attainable length values λ_F for the edge f. When δ is generic, i.e. does not admit collinearities or coincidences in the realizations of (G, δ), the realization space of $(H, \langle \delta_{E\setminus e}, \lambda_f \rangle)$ has 16 solutions $q^p_{\lambda_f}$ (modulo orientation preserving isometries), with distinct orientation types σ_p (two orientation choices for each of the 4 triangles) that can be obtained by solving a sequence of 4 single quadratics in 1 variable (DR-plan of size 2). By subdivided binary search in the interval $\lambda_f \in \Phi_f(G', \delta_{E\setminus e})$, the desired solution p of (G, δ) is found when the length of the nonedge e in the realization $q^p_{\lambda_f}$ is δ_e.

In fact, we can show that $G = K_{3,3}$ is in OMD_1 by removing a single edge to reduce (as shown in Fig. 7) to a tree-decomposable graph of low Cayley complexity (which includes the class of partial 2-trees). In Section 4.4, we discuss this issue of why the largest class of reduced graphs is desirable.

4.4. Optimized modification by enlarging the class of reduced graphs

It is possible in principle to decrease k for an OMD_k graph (i.e. the number of edges to be removed to ensure an isostatic completion that is decomposable with a small DR-plan) by considering reduced graphs G' (and modified graphs H) that come from a larger class than partial 2-trees but nevertheless have convex Cayley configuration spaces at least when the realization space is restricted to a sufficiently comprehensive orientation type. In particular, the so-called *tree-decomposable graphs of low Cayley complexity* (Sitharam et al., 2011a, 2011b) include the partial 2-trees and many others that are not partial 2-trees. See an illustration in Fig. 7. These too result in DR-plans of size 2 or 3, putting G in the class OMD_k and thus meeting Criteria (a) and (b). The Criterion (c) is met – for example when $k = 1$ – because 1-dof Cayley configuration spaces of linkages based on such graphs G' are known to be single intervals when a comprehensive orientation type σ_p of the sought solution p is given. In addition, the bounds of these intervals are of low algebraic complexity. More precisely, the bounds can themselves be computed using a DR-plan of size 2 or 3, i.e. the computation of these bounds is tree-decomposable. Alternatively, the bounds are in a simple quadratic or radically solvable extension field over the rationals, or they can be computed by solving a triangularized system of quadratics.

4.5. Problem relationships

In this section we provide a unified view of the various problems studied in the previous 2 sections, along with formal reductions between them. We discuss their relationship to other known problems and results as well as open questions.

4.5.1. Special classes of small DR-plans

As seen in the previous section, 2-trees and tree-decomposable graphs have not only small, but also special DR-plans that permit easy solving – essentially by solving a single quadratic at a time.

The *restricted optimal DR-planning problem* requires DR-plans of one of these types, which reduces to recognizing if the input graph is a 2-tree or a tree-decomposable graph for which simple near-linear time algorithms are available (Valdes et al., 1979; Fudos and Hoffmann, 1997) and the DR-plan is a by-product output of the recognition algorithm.

In the recombination setting, the corresponding *restricted* OMD_k *problem* requires the reduced graph G' and its isostatic completion H to be 2-trees as in Section 4.3 or to be a low Cayley complexity tree-decomposable graph as in Section 4.4. Clearly these problems have deterministic polynomial time algorithms in n, but the algorithms run in time exponential in k.

We discuss the complexity of the restricted OMD problem (when k is part of the input) in the open-problem Section 7.1.

4.5.2. Optimal modification, completion and recombination: previous work and formal connections

The OMD problem is closely related to a well-studied problem of completion of an underconstrained system to an isostatic one with a small DR-plan.

Observation 27. The (decision version of) the **optimal completion problem (OC)** from Sitharam (2005), Joan-Arinyo et al. (2003), Zhang and Gao (2006) is OMD_0.

In fact, a *restricted OC* problem was studied by Joan-Arinyo et al. (2003) requiring the completion to be tree-decomposable.

We now connect the OMD problem to the informal *optimal recombination (OR)* problem mentioned as motivation at the beginning of Section 4.

In order to connect the OR problem to OMD, when the input graph is the isostatic graph at the DR-plan root, we do not consider the case where the two child *solved subgraphs* (corresponding to already solved subsystems) have a nontrivial intersection (in this case the recombination is trivial). We only consider the case where no two child solved subgraphs (resp. two root subgraphs when the input graph is underconstrained) share more than 1 vertex. We replace such solved subgraphs by isostatic graphs as follows. If a solved subgraph shares at most one vertex with the remainder of the graph, simply replace it by an edge one of whose endpoints is the shared vertex. Otherwise, replace it by a 2-tree graph of the shared vertices. Finally, we add the additional restriction to the OM problem that when any edge in a solved subgraph is chosen among the k edges to be removed, in fact the entire solved subgraph must be removed and all of its edges must be counted in k.

This reduction is used also for adapting algorithms for optimal DR-planning, recombination, completion, OMD, and other problems from bar–joint systems to so-called *body–hyperpin*, defined in Section 5, by showing that the problems for the latter are reduced to the corresponding problems on bar–joint systems.

5. Application: finding completions of underconstrained glassy structures from underconstrained to isostatic

We can use qusecs DR-plans to design materials such as disordered graphene and silica bi-layers (Wilson et al., 2013; Heyde, 2013). We investigate a more specific problem in a somewhat more general setting: the problem of finding boundary conditions (additional constraints) to add to an underconstrained monolayer to make it isostatic. This can be done in a number of ways: (1) pin together 2 underconstrained monolayers in such a way that the resulting bi-layer becomes isostatic (see Fig. 8); (2) pin the boundary of (or in general, add constraints to) a layer (possibly a genus 0 monolayer) so that it becomes isostatic; or (3) design a broader class of structures to ensure they are isostatic, self-similar (via some subdivision rule) and in addition isostatic at each level of the subdivision (see Fig. 9).

In all cases, we are specifically interested in how to add additional constraints such that the resulting isostatic structure has a small DR-plan; this way a realization can be found, allowing efficient stress, flex and other property design related to the rigidity matrix. To answer these questions, we first introduce the qusecs that are used to model these materials. In this section, we discuss Item (2) in detail.

5.1. Body–hyperpin qusecs

Definition 28 *(Body–hyperpin qusecs).* A *body–hyperpin qusecs* is a constraint system where the objects are rigid bodies and the constraints are incidences of object subsets at a common point, i.e. a pinning of bodies.

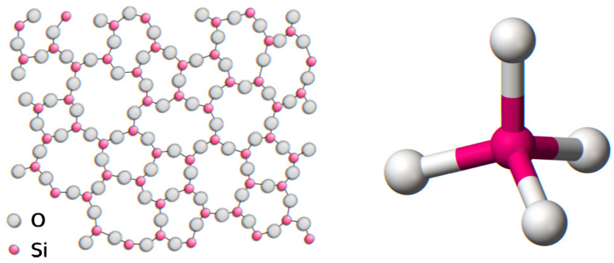


Fig. 8. Example of a single monolayer of a silica (silicon and oxygen) glassy structure. This can be viewed as a triangle–multipin qusecs where the silicon atoms are the triangles and the oxygen atoms are the pins. Not shown here are other monolayers stacked on this one. Each silicon atom has the structure seen to the right, and so binds to another oxygen atom in an adjacent monolayer. Pictures taken from Wikimedia Commons (2006, 2008).

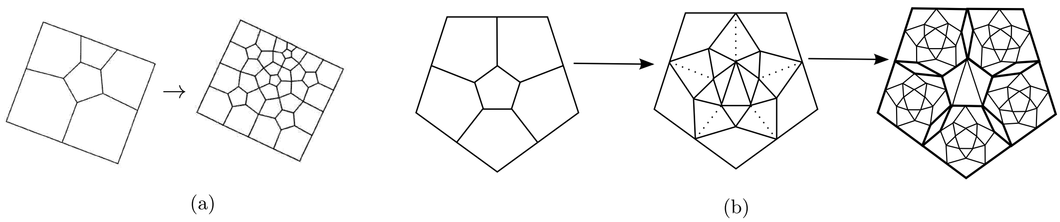


Fig. 9. Examples of self-similarity via repeated subdivision. In (a) (Canon et al., 2001), there is a simple subdivision scheme that does not guarantee isostaticity. In (b), a more complicated scheme is used, ensuring that the resulting graph is isostatic but not tree-decomposable.

Remark* 29. A body–hyperpin qusecs is a special case of bar–joint qusecs of the previous sections of the paper. As such, the DR-planning for isostatic systems discussed in Section 3 is unchanged and the results of Section 4 still go through with minor modifications.

For the remainder of this section, we deal only with the DR-plan of such qusecs. Hence, we refer only to the combinatorics or underlying hypergraph of the qusecs. We now introduce 2 sub-classes of body–hyperpin graphs for modeling Examples 4 and 5 in Section 1, for which the optimal completion problem is significantly easier.

Definition 30 *(Body–pin graph).* A *body–pin graph* is a body–hyperpin graph with the following conditions: (1) each pin is shared by at most two bodies; and (2) no two bodies share more than one pin.

Such a body–pin graph, G_{BP}, can also be seen as a *body–bar graph*, G_{BB}, where the bodies of G_{BB} are the original bodies of G_{BP} and each pin between bodies in G_{BP} are replaced with 2 bars in G_{BB} between the same bodies. Such body–bar graphs with 1 and 2-dof can be characterized by being $(3,4)$ and $(3,5)$-tight respectively (Lee and Streinu, 2007; Streinu and Theran, 2009) (defined in Section 2.1). See Fig. 10b.

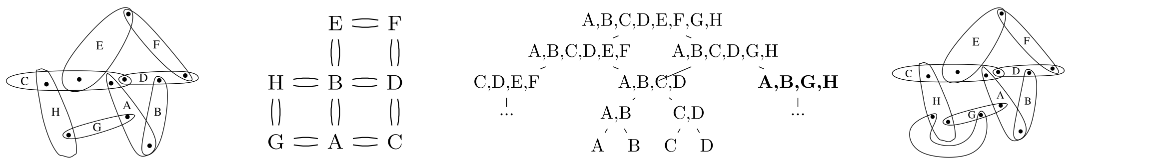
Definition 31 *(Triangle–hyperpin graph).* A *triangle–hyperpin graph* is a body–hyperpin graph where each body is a triangle, i.e. it shares pins with at most 3 other bodies. This is also represented as a hyper-graph where each pin is a vertex and each triangle represents a tri-hyperedge. For such hypergraphs, *1- and 2-dof* can be characterized by $(2,4)$- and $(2,5)$-tightness respectively (Lee and Streinu, 2007; Streinu and Theran, 2009).

Body–pin graphs are of particular interest to us in the context of Example 4 in Section 1. Triangle–multipin graphs can be used to represent the silica bi-layers and glassy structures described in Example 5 of Section 1, where each triangle is the junction of "disks" in the plane (see Fig. 8). Typically, these systems are not isostatic, so to relate the work of this paper to the systems, we define a slightly different kind of DR-plan using the notion of (k,l)-sparsity and tightness.

Definition 32 *((k,l)-tight DR-plan).* A *(k,l)-tight DR-plan* is one in which each child node is either a vertex maximal proper (k,l)-tight subgraph of the parent node or it is trivial. In our case, the trivial nodes are the bodies.

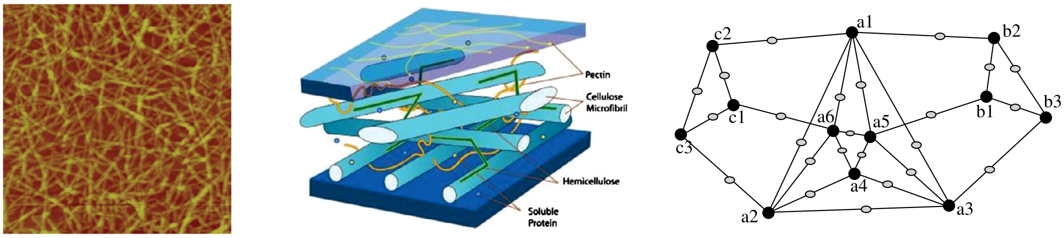
Provided such (k,l)-sparse graphs are matroidal (conditions given in Lee and Streinu, 2007), the notion of a canonical DR-plan extends directly to the case when the hypergraph is (k,l)-sparse (i.e. independent) using the straightforward notion of trivial and non-trivial intersections and (k,l)-tightness conditions as in Section 3. In particular, we define canonical DR-plans with similar properties for the 1 and 2-dof body–pin and triangle–hyperpin systems defined above.

Observation* 33. For the 1-dof body–pin graphs described above that are $(3,4)$-sparse, a $(3,4)$-tight canonical DR-plan exists where every node of a $(3,4)$-sparse graph satisfies one of the following: (1) its children are 2 proper vertex-maximal



(a) (b) (c) (d)

Fig. 10. (a) A 1-dof body–pin graph. (b) The corresponding body–bar graph, explained in Definition 30. (c) The 1-dof DR-plan for the graph. In this case, to obtain an isostatic system, we would need to add a body and 2 pins to one of the nodes in the second level. (d) The result of adding one such body to the bold-faced node.



(a) (b) (c)

Fig. 11. (a) Microfibrils of carboxymethylated nanocellulose adsorbed on a silica surface (Wikimedia Commons, 2010). (b) Cross-linking of cellulose microfibrils (Wikimedia Commons, 2007c). (c) An isostatic pinned line-incidence graph, representing fibrils and their attachments.

1-dof graphs that intersect on another 1-dof graph; or (2) its children are all of the proper maximal 1-dof subgraphs, pairwise sharing at most one body.

As in Section 3, a strong Church–Rosser property holds, making all canonical DR-plans optimal:

Observation* 34. When the input is independent, all $(3,4)$-tight canonical DR-plans are optimal. We can find such a DR-plan in the same time complexity as the $(2,3)$-tight case for bar–joint graphs discussed in Section 3.

The abovementioned algorithm exists because such $(3,4)$-tight graphs are matroidal and have a pebble game (Lee and Streinu, 2007).

The above discussion leads to the main theorem:

Theorem* 35. *Given a 1-dof body–pin or triangle–multipin graph and corresponding 1-dof DR-plan, there is a quadratic algorithm for the 1-dof optimal completion problem of Section 4.*

Observation* 36. For the 2-dof case, provided an analogous statement to Observation 34 holds, then Theorem 35 holds for the 2-dof systems.

Remark 37. While the proof for Theorem 35 gives us a DR-plan for the isostatic completion with minimum fan-in (a reasonable measure of algebraic complexity), a more nuanced measure that treats solutions of 1-dof and 2-dof systems as 1 or 2 parameter families would no longer be optimized by the algorithm given in that proof. In particular, the complexity of the standard algorithm in the k-dof case would be exponential in k (even if the case were matroidal and an optimal DR-plan is known).

6. Application: finding optimal DR-plans and realizations for cross-linking microfibrils

The canonical DR-plan of Section 3 can additionally be applied to analyze and solve the structure of cross-linking collagen microfibrils in animals, cellulose microfibrils in plant cell walls, and actin filaments in the cytoskeleton by modeling these structures as a third type of qusecs, *pinned line-incidence systems*.

Collagen is an important protein material in biological tissues with highly elastic mechanical properties (Buehler, 2008). *Cellulose* is the most important constituent of the cell wall of plants (see Fig. 11a) (Fall et al., 2013; Smith, 1971). Both of these substances consist of a large number *microfibrils*, each of which is cross-linked at 2 places with usually 3 other fibrils, where the *cross-linking* is like an incidence constraint that the crosslinked fibrils can slide against each other while remaining incident (see Fig. 11b).

6.1. Modeling the fibrils as a pinned line-incidence system

The cross-linking microfibrils can be modeled as a pinned line-incidence constraint system in $\mathbb{R}^2$, where incidence constraints are used instead of distance constraints.

Definition 38 *(Pinned line-incidence system).* A *pinned line-incidence system* (G, δ) is a graph $G = (V, E)$ together with parameters δ specifying $|E|$ *pins* with fixed positions in $\mathbb{R}^2$, such that each edge is constrained to lie on a line passing through the corresponding pin, i.e. $\delta : E \to \mathbb{R}^2$.

A pinned line-incidence graph G is rigid if $|E| = 2|V|$ and $|E'| \leq 2|V'|$ for every induced subgraph (V', E') (Sitharam et al., 2014). Note that no trivial motion exists since the pins have fixed positions on the plane. Euclidean transformations are not factored out. In particular, both a single vertex and a single edge are underconstrained graphs.

In the case of microfibril cross-linking, each fibril is attached to some fixed larger organelle/membrane at one site. Consequently, each fibril can be modeled as an edge of the graph, with the attachment being the corresponding pin. The two cross-linkings in which the fibril participates are modeled as the two vertices in V defining the edge.

Fig. 11c shows an example of a pinned line-incidence graph, where the gray ovals denote pins representing attachments of fibrils, and the vertices $a_1, a_2, \ldots, c_3$ represent cross-linkings. The graph is isostatic, with 12 vertices and 24 edges/pins.

6.2. Optimal DR-plan for pinned line-incidence systems

In this section, we will adapt the results in Section 3 to give the canonical DR-plan for pinned line-incidence graphs. First, we note that *an isostatic pinned line-incidence graph can be disconnected*, being the disjoint union of two or more isostatic subgraphs. This is because the pins have fixed positions on the plane. We define a *trivial* graph to be a single vertex and make the following modification to the definition of the canonical DR-plan:

Definition 39 *(DR-plans of pinned line-incidence systems).* The *DR-plan* of a pinned line-incidence graph G is one in which (1) each child node of a non-leaf node C is either a *connected* rigid vertex-induced subgraph of C, or an edge not contained in any proper rigid subgraph of C, and (2) a leaf node is a single edge.

The *canonical DR-plan* of G is one in which the child rigid subgraphs are connected, isostatic vertex-maximal subgraphs of the parent.

Theorem 6 holds for pinned line-incidence graphs with this modified definition. The proof is similar to the original proof (in Section 3) using the same set of lemmas and the following modified version of Observation 7, which can be proved using a simple counting based argument.

Observation 40. Let F_i and F_j be subgraphs of the same isostatic graph F, where each of them can be either a single edge or a connected isostatic subgraph. There are only two possible cases: (1) at least one of F_i, F_j is an edge, if and only if $F_i \cup F_j$ is underconstrained, if and only if $F_i \cap F_j$ is trivial; and (2) both F_i and F_j are isostatic, if and only if $F_i \cup F_j$ is isostatic, if and only if $F_i \cap F_j$ is isostatic.

Given Observation 40, Lemma 8, Points 1 and 3, straightforwardly extend to pinned line-incidence graphs. The proof of Point 2 for pinned line-incidence graphs is given in Appendix A.4.1. Thus it is straightforward to adapt the proof of Theorem 6 to pinned line-incidence graphs. Consequently, we can efficiently find the optimal DR-plan for pinned line-incidence graphs using basically the same algorithm as for bar–joint graphs.

Note that the recombination problem for pinned line-incidence systems is trivial. Since the pins are given fixed positions in the plane, the solutions of an isostatic subsystem will automatically be consistent with the solutions of the other subsystems.

7. Open problems and conclusion

7.1. Open problems

The results of this paper lead to a number of open problems. The first set of problems are from Section 3:

Open Problem 1. Is there a more efficient algorithm than $O(|V|^3)$ to find the canonical DR-plan of isostatic 2D bar–joint graphs?

Conjecture 41. *The Modified Frontier Algorithm (MFA) (Lomonosov, 2004) finds a canonical, and hence optimal, DR-plan.*

The difficulty of proving Conjecture 41 arises from the fact that MFA, although running in time $O(n^3)$, is a bottom-up algorithm, involving complex data structures. However, a proof of optimality, even if it exists, would not be possible without the new notion of a canonical DR-plan at hand. The intuition for this conjecture comes from the similarity of the DR-plan generated by MFA to that of the pseudosequential decomposition described in the proof of Theorem 19. Since it is known (Lomonosov, 2004) that the DR-plan generated by MFA is cluster-minimal, an alternate conjecture is the following.

Conjecture 42. *For independent graphs, cluster-minimal DR-plans are optimal. In fact, for independent graphs, cluster-minimality and canonical are equivalent properties of a DR-plan.*

Open Problem 2. Although generic rigidity is a property of graphs, and moreover, in the case of qusecs, generic rigidity has a combinatorial sparsity and tightness-based characterization, the original definition of independence in the rigidity matroid requires an algebraic notion of independence of vectors of indeterminates over $\mathbb{R}$. Thus the definition of the DR-plan requires algebra over the reals. In fact, the recursive decomposition problem is not tied to geometric constraint graphs or an algebraic–geometric or mechanical notion of rigidity, and can be defined for any graph using the notion of an abstract rigidity matroid (Graver et al., 1993). This is a type of matroid with two additional matroid axioms; abstract rigidity matroids can be defined in a purely graph-theoretic manner, with no need for algebra in their definition. However, such abstract rigidity need not have a sparsity characterization. On the other hand, there are sparsity matroids that do not correspond to any notion of abstract rigidity. However, when an abstract rigidity matroid is also a sparsity matroid, then the techniques of this paper directly apply and we can obtain purely combinatorially defined recursive decompositions of graphs.

A few natural open questions concern the following common theme that runs through the optimal recombination and later sections of the paper:

Open Problem 3. For fixed k, we have polynomial time optimal DR-planning (Section 3), recombination (modification) in the presence of k overconstraints, optimal modification for decomposition $\mathrm{OMD}_k(G)$ when at most k constraints are removed (Section 4), and also optimal completion using at most $k \leq 2$ constraints in the body–pin and triangle–multipin cases for a somewhat different optimization of the DR-plan (Section 4.5). However, in the running time of all of these algorithms, k appears in the exponent. Can k be removed from the exponent?

One problem in the above theme is from Section 5.

Open Problem 4. What is the complexity of the optimal completion problem when the given graph has more than 2-dofs? Our proof for the 1- and 2-dof cases relied heavily on the matroidal properties of their corresponding (k, l)-tightness. For higher number of dofs, the (k, l) characterization is no longer matroidal (Lee and Streinu, 2007). As a result, the major obstacle is that there is no easy way of obtaining an optimal or canonical k-dof DR-plan in general. Even assuming such a DR-plan is available, if higher dofs had the same characteristics, Observation 37 raises questions about the correct measure of DR-plan size that captures algebraic complexity for recombining graphs with many dofs (this is not an issue in the isostatic case). Unless some restrictions can be found and taken advantage of, the k-dof optimal completion problem would have complexity exponential in k.

Another problem from the above theme is from Section 4

Open Problem 5. What is the complexity of the restricted OMD (optimal modification for decomposition) problem? This has the potential to be difficult. For example, when the isostatic completion is required to be a 2-tree the restricted OMD problem is reducible to the maximum spanning series-parallel subgraph problem shown by Cai and Maffray (1993) to be NP-complete even if the input graph is planar of maximum degree at most 6. However, since the OMD problem has other input restrictions such as not having any proper isostatic subgraphs, it is not clear if the reverse reduction exists and hence it is unclear whether the OMD problem is NP-complete.

The same holds for the restricted OMD problem where the isostatic completion is required to be a tree-decomposable graph of low Cayley complexity (i.e. have special, small DR-plans). One potential obstacle to an indecomposable graph G's membership in the restricted OMD_k for small k is if G is tri-connected and has large girth. In fact, 6-connected (hence rigid) graphs with arbitrarily large girth have been constructed in Servatius (2000).

The next is the reverse direction of Observation 27 in Section 4.5.

Open Problem 6. Is the OMD (optimal modification for decomposition) problem reducible to the OC (optimal completion) problem?

7.2. Conclusion

We have clarified the main source of complexity for the optimal DR-plan and recombination problems. For the former problem, when there are no overconstraints (as is the case for 2D qusecs whose realizations are many common types of layered materials), we defined a canonical DR-plan and showed that any canonical DR-plan is guaranteed to be optimal, a strong Church–Rosser property. This gives an efficient ($O(n^3)$) algorithm to find an optimal DR-plan that satisfies other desirable characteristics.

We have also described a novel method of efficiently realizing a 2D qusecs from the optimal DR-plan by modifying the otherwise indecomposable systems at nodes of a DR-plan. These results rely on a recent theory of convex Cayley configuration spaces. Relationships and reductions between these and previously studied problems were formally clarified.

We then modeled specific layered materials using extensions of the above theoretical results including the motivating Examples 1–5 in the introduction.

Finally, we detailed a number of open problems that were motivated by the work in this paper.

Appendix A. Proofs

A.1. Proofs from Section 3

A.1.1. Proof of Observation 7

Proof. For (1), simply note that if $F_i \cup F_j$ were trivial, then, by definition, F_i and F_j must be trivial.

For the next parts, we use the quantity $d(G) = 2|V| - |E|$, which we call *density*. For (2), observe that underconstrained subgraphs of isostatic graphs must have density less than 3. For (3), observe that, given an isostatic graph, a subgraph with density 3 must also be isostatic. Then, use the fact that, by definition, $d(F_i) = 3$ and $d(F_j) = 3$. Then it is straightforward application of the inclusion–exclusion $d(F_i \cap F_j) = d(F_i) + d(F_j) - d(F_i \cup F_j)$.

For (4), because subgraphs of an isostatic graph can only be trivial, underconstrained, or isostatic, all cases have already been exhausted. $\square$

A.1.2. Proof of Lemma 8, Point 1

Proof. Assume $C_i \cup C_j \neq C$. This would contradict the proper vertex-maximality of C_i, C_j. In the reverse direction, we know C is either a non-leaf node (isostatic by definition of a DR-plan) or G itself (isostatic by definition of the problem). Thus, $C_i \cup C_j = C$ is isostatic. $\square$

A.1.3. Proof of Lemma 8, Point 2

Proof. Take two children of node C, called C_i and C_j. Let R_i be the graph induced in node C by the edge set of C minus the edge set of child C_i ($C \setminus C_i$ for convenience); let $R_j = C \setminus C_j$. Let $D_{i,j} = C_i \cap C_j = C \setminus (R_i \cup R_j)$. Let $R'_i \subset R_i$, $R'_j \subset R_j$, and $D'_{i,j} \subset D_{i,j}$, and take these proper subgraphs to be non-empty.

If there are two children ($N = 2$) then the proof is simple, it follows from the definition of a DR-plan: the union of the children is the parent which is isostatic. Assume that $N > 2$ and take a third child, called C_k. We want to determine what C_k can be, in terms of i and j. C_k can possibly be composed of an element from $\{\emptyset, D'_{i,j}, D_{i,j}\}$, an element from $\{\emptyset, R'_i, R_i\}$, and an element from $\{\emptyset, R'_j, R_j\}$, for a total of 27 cases. We will exhaustively show that it must be $R_i \cup R_j \cup D'_{i,j}$. First, we make the following observation:

Observation 43. If $C_i \cup C_j$ is isostatic, then there can be no edges in C between the vertices of R_i and R_j.

Proof. Lemma 8, Point 1, shows that $C_i \cup C_j$ must equal the parent graph C. $\square$

- 3 cases: C_k cannot be $C = R_i \cup R_j \cup D_{i,j}$, $C_i = R_j \cup D_{i,j}$, or $C_j = R_i \cup D_{i,j}$. This is by definition.
- 13 cases: C_k cannot be a proper subgraph of C_i and C_j or else C_k would not be vertex-maximal. These are the graphs $R'_i \cup D_{i,j}$, $R'_j \cup D_{i,j}$, $D_{i,j}$, $R_i \cup D'_{i,j}$, $R_j \cup D'_{i,j}$, $R'_i \cup D'_{i,j}$, $R'_j \cup D'_{i,j}$, $D'_{i,j}$, R_i, R_j, R'_i, R'_j, and $\emptyset$.
- 2 cases: C_k cannot contain C_i or C_j as proper subgraphs, or else they are not vertex-maximal. These are the graphs $R'_i \cup R_j \cup D_{i,j}$ and $R_i \cup R'_j \cup D_{i,j}$ respectively.
- 4 cases: C_k cannot be $R_i \cup R_j$, $R'_i \cup R_j$, $R_i \cup R'_j$, or $R'_i \cup R'_j$ because these are all disconnected (Observation 43) and cannot be isostatic.
- 1 case: $C_k = R'_i \cup R'_j \cup D_{i,j}$ is not possible. Since $C_i \cup C_k = R'_i \cup R_j \cup D_{i,j} \neq C$ we have from Lemma 8, Point 1, that $C_i \cup C_k$ cannot be isostatic. We also know it cannot be trivial because it contains isostatic subgraphs. This means it must be

underconstrained. From Observation 7, we know that $C_i \cap C_k = R'_j \cup D_{i,j}$ must then be trivial. This is impossible because $D_{i,j}$ is isostatic, thereby contradicting the assumption that C_k is isostatic.

- 1 case: $C_k = R'_i \cup R'_j \cup D'_{i,j}$ is not possible. Since $C_i \cup C_k \neq C$ (and $C_j \cup C_k \neq C$), we know by the same logic as the previous case that the $C_i \cap C_k$ must be trivial (a single node). However, $C_i \cap C_k = R'_j \cup D'_{i,j}$. This causes a contradiction, the intersection cannot be trivial because R'_j and $D'_{i,j}$ are not empty sets and are disjoint.
- 2 cases: $C_k = R'_i \cup R_j \cup D'_{i,j}$ and $C_k = R_i \cup R'_j \cup D'_{i,j}$ are not possible. The proof mirrors the previous case, except here you must choose C_i and C_j respectively.
- 1 case: $C_k = R_i \cup R_j \cup D'_{i,j}$ is all that remains.

Since $D_{i,j} \subset C_i, C_j$ it means that $C_k \cup C_i = C_k \cup C_j = C$, thus proving Point 2. The alternative phrasing of this point is straightforward from Point 1. □

A.1.4. Proof of Lemma 8, Point 3

Proof. Assume there is some k such that $C_i \cap C_k$ is not trivial. By Observation 7, $C_i \cap C_k$ must be isostatic. Then, by Lemma 8, Point 2, the intersection between any two children must be isostatic. This means that $C_i \cap C_j$ is isostatic. Therefore, such a k cannot exist and all intersections are trivial. □

A.2. Proofs from Section 4

A.2.1. Proof of Theorem 25

Proof. The Cartesian realization space of $(H, \langle \delta_{E'}, \lambda_F \rangle)$ is computed easily with a DR-plan of size 2, and is the union of 2^t solutions (modulo orientation preserving isometries) each with a distinct orientation type, where t is the number of triangles in the 2-tree H; here δ_S is the restriction of the length vector δ to the edges in S. A desired solution p (or connected component of a solution space) of (G, δ) of an orientation type σ_p can be found by a subdivided binary search of the Cartesian realization space of $(H, \langle \delta_{E'}, \lambda_F \rangle)$ of orientation type σ_p, as λ_F ranges over the discretized convex polytope $\Phi_F(G', \delta'_E)$ with bounding hyperplanes described in Theorem 23. A solution p is found when the lengths for nonedges in D match δ_D. □

A.3. Proofs from Section 5

A.3.1. Proof of Remark 29

Proof. We can replace each body that has only one pin by a single vertex. A body with 2 pins can be replaced by an edge. In general, a body with n pins can be replaced by a 2-tree on n vertices. When finding a DR-plan, we treat each body as trivial, so they become the leaves of the DR-plan. The optimal recombination problem and approach of Section 4 are unchanged. The optimal completion via the optimal modification problem in Section 4 now has an additional restriction that all edges in the 2-tree representation of the bodies must be removed together, not individually. □

A.3.2. Proof of Observation 33

Proof. The existence of this canonical DR-plan follows from the same arguments as in the proof of Theorem 6. The only difference is the definition of a trivial intersection. In this case, when two subgraphs share more than 1 body, they become rigid (in fact over constrained). Sharing a pin is not considered an intersection. Such a structure is viewed as two subgraphs each sharing 1 body with a third 1-dof subgraph which essentially just consists of those two bodies pinned together. □

A.3.3. Proof of Theorem 35

Proof. Suppose we are given a body–pin graph and its corresponding body–bar graph G and have obtained the 1-dof DR-plan T. Each node of T is then a vertex-maximal proper 1-dof subgraph of G.

To make the graph isostatic, we need only add one body and pin it to 2 other bodies. Doing so will cause G to become $(3, 3)$-tight.

We adopt the following algorithm. Choose the 2 bodies to pin to by choosing a node b in T and looking at its children. From Observation 34, we know that the children can only share a single pin or a subgraph. Pin the new body to bodies in two separate children. Doing so will ensure that all children of b will have 1-dof and all ancestors of b (including b) will now be isostatic.

Such a pinning covers all possible ways of adding a new body. Assume a new body b is added to the input graph and pin it to b_i and b_j to make it isostatic. Then, there is a lowest 1-dof node v in T such that b_i and b_j appear in v. Thus, pinning v in the manner described yields an equivalent isostatic DR-plan to pinning b to b_i and b_j.

For each node b, assign a size of the T_b denoted $|T_b|$. $|T_b| = \max_{v \in T_b} \textsc{FanIn}(v)$. We are looking for b that minimizes $|T_b|$. Denote the sub-tree of T rooted at v by T^v and the number of leaves in a tree T by $\mathrm{NL}(T)$. Note that $\textsc{FanIn}(b) = \mathrm{NL}(T^b)$ because no descendant of b is isostatic. Similarly, for any ancestor w of b, $\textsc{FanIn}(w) = \mathrm{NL}(T^w) - \mathrm{NL}(T^{b'}) + 1$, where b' is the child leading to b. All other nodes are not isostatic and hence do not appear in the isostatic DR-plan.

The node to be pinned is always the deepest nontrivial node of some path in T. Suppose a node b is pinned that has a nontrivial child v. Then, $\textsc{FanIn}_b(b) = \mathrm{NL}(T^b) = \mathrm{NL}(T^v) + n$, where n is essentially the number of leaves between b and v. If we had instead chosen to pin v, then $\textsc{FanIn}_v(b) = \mathrm{NL}(T^b) - \mathrm{NL}(T^{b'}) + 1 \le \textsc{FanIn}_b(v)$. And for each ancestor w of b, $\textsc{FanIn}(w)$ is unchanged, meaning $|T_v| \le |T_b|$. Thus we only have to check the deepest non-trivial nodes.

Running the above algorithm brute force gives running time quadratic in the number of bodies of the given body–pin system.

For the multi-triangle pin graphs, we can do the same thing, except we need to add a single triangle to one of the nodes to cause it to become isostatic. □

A.3.4. Proof of Observation 36

Proof. The only difference from the 1-dof case is that now we need to remove 2-dofs from our graph. Start with a 2-dof DR-plan T. Like in the previous proof, we need to add a body and 2 pins to 2 nodes to obtain an isostatic DR-plan.

Suppose we pin 2 distinct nodes v_i and v_j. Then, there must exist a common ancestor a of v_i and v_j. Then, in T_{v_i,v_j}, $\textsc{FanIn}_{v_i,v_j}(a) = \mathrm{NL}(T^a)$. However, if we chose to pin one of v_i and v_j twice, then $\textsc{FanIn}_v(a) = \mathrm{NL}(T^a) - \mathrm{NL}(T^{a'}) + 1$. Thus $\textsc{FanIn}_v(a)' \le \textsc{FanIn}_{v_i,v_j}(a)$. All ancestors of a are unchanged. So $|T_v| \le |T_{v_i,v_j}|$.

Thus the only choice is to pin a single node twice. Hence, we can run the same algorithm as the 1-dof case and simply pin twice instead of once. □

A.3.5. Proof of Observation 37

Proof. An isostatic graph has 3 parameters that define its position and orientation. These are the Euclidean motions. A 1-dof graph has 4 parameters: the 3 Euclidean motions and a dof parameter. A 2-dof has 5 parameters. The number of parameters roughly correlates with the algebraic complexity of obtaining a realization.

Thus, starting with a T as described in the proof for Remark 35, when a node b is pinned, the same structure is preserved as before. Suppose v is an isostatic node after pinning b. Then, the children of v (except one if $v \ne b$) have 1-dof. The realization complexity for v is simply that of realizing each of its children. In general, the number of parameters for v will be $\mathrm{NP}(v) = 4\mathrm{NC}_1(v) + 3$, if $v \ne b$ and $\mathrm{NP}(b) = 4\mathrm{NC}_1(b)$, where $\mathrm{NC}_k(v)$ is the number of k-dof children of v.

Minimizing the algebraic complexity requires minimizing the maximum $\mathrm{NP}(v)$ for any node v. In this case, it is not possible to always choose to pin a node closest to a leaf in the tree, because it could have high fan-in. So we try brute force by pinning all nodes to pick the one with the lowest algebraic complexity. This algorithm is still quadratic for the 1-dof case.

For the 2-dof situation, there are more cases to consider. If we pin the same node twice as above, we have $\mathrm{NP}(v) = 5\mathrm{NC}_2(v) + 3$ for any ancestor $v \ne b$ and $\mathrm{NP}(b) = 5\mathrm{NC}_2(b)$. If we pin a node v and one of its ancestors v', then any nodes between v' and v will be 1-dof, any nodes above v' will be isostatic, and nodes below v will be 2-dof. Note that solving or realizing v' will also realize v. Next, we need to consider nodes above and including v' in our complexity: $\mathrm{NP}(v') = 5\mathrm{NC}_2(v') + 4$ and $\mathrm{NP}(a) = 3 + 5\mathrm{NC}_2(a)$ for a an ancestor of v'.

The only remaining case is pinning two nodes that are incomparable, i.e. do not have a descendant/ancestor relationship. The only change from the previous case is that for the lowest common ancestor of the nodes v', $\mathrm{NP}(v') = 2 * 4 + 5\mathrm{NC}_2(v')$. For any ancestor a of v', we still have $\mathrm{NP}(a) = 3 + 5\mathrm{NC}_2(a)$.

Like the 1-dof case, we again cannot simply choose the nodes deepest in the tree to pin. However, neither can we assume pinning one node twice will give us the best algebraic complexity. Hence, we will need to check each pair of nodes to pin. This makes our brute–force algorithm $O(b^3)$, where b is the number of bodies. □

A.4. Proofs from Section 6

A.4.1. Proof of Lemma 8, Point 2 — for 2-dimensional pinned line incidence graphs

Proof. We use the same notation as in the original proof of Lemma 8, Point 2, given above. Without loss of generality, all graphs are the induced graphs on C.

First notice that since $C_i \cup C_j$ is isostatic, by Observation 40, both C_i and C_j are connected isostatic vertex-maximal proper subgraphs of C. Since $C_j \cup C_j = C$, there are no edges in C that is not contained in an isostatic subgraph, so C does not have any single-edge child node, and C_k is a connected isostatic vertex-maximal proper subgraph of C.

We analyze all the possible cases for C_k.

- 1 case: $C_k = R'_i \cup R'_j \cup D_{i,j}$ is not possible. Since $C_k \cup C_i = R'_i \cup R_j \cup D_{i,j} \neq C$ we have from Lemma 8, Point 1, that $C_k \cup C_i$ cannot be isostatic. By Lemma 40, it must be underconstrained, so one of C_i and C_k must be an edge, contradicting the assumption that both C_i and C_k are isostatic.
- 1 case: $C_k = R'_i \cup R'_j \cup D'_{i,j}$ is not possible. The proof is similar to the previous case.
- 2 cases: $C_k = R'_i \cup R_j \cup D'_{i,j}$ and $C_k = R_i \cup R'_j \cup D'_{i,j}$ are not possible. The proof is similar to the previous case.

All remaining cases are similar to the original proof for 2D bar–joint graphs. □

References

Asimow, L., Roth, B., 1978. The rigidity of graphs. Trans. Am. Math. Soc. 245, 279–289. http://dx.doi.org/10.1090/S0002-9947-1978-0511410-9.

Björkman, T., Kurasch, S., Lehtinen, O., Kotakoski, J., Yazyev, O.V., Srivastava, A., Skakalova, V., Smet, J.H., Kaiser, U., Krasheninnikov, A.V., 2015. Defects in bilayer silica and graphene: common trends in diverse hexagonal two-dimensional systems. Sci. Rep. 3 (3482). http://dx.doi.org/10.1038/srep03482.

Borcea, C., Streinu, I., 2004. The number of embeddings of minimally rigid graphs. Discrete Comput. Geom. 31 (2), 287–303. http://dx.doi.org/10.1007/s00454-003-2902-0.

Buehler, M.J., 2008. Nanomechanics of collagen fibrils under varying cross-link densities: atomistic and continuum studies. J. Mech. Behav. Biomed. Mater. 1 (1), 59–67.

Cai, L., Maffray, F., 1993. On the {SPANNING} *k*-tree problem. Discrete Appl. Math. 44 (1–3), 139–156. http://dx.doi.org/10.1016/0166-218X(93)90228-G. http://www.sciencedirect.com/science/article/pii/0166218X9390228G.

Canon, J.W., Floyd, W.J., Parry, W.R., 2001. Finite subdivision rules. Conform. Geom. Dyn., 153–196.

Donev, A., Torquato, S., Stillinger, F.H., Connelly, R., 2015. Jamming in hard sphere and disk packings. J. Appl. Phys. 95 (3).

Eder, F.R., Kotakoski, J., Kaiser, U., Meyer, J.C., 2015. A journey from order to disorder—atom by atom transformation from graphene to a 2D carbon glass. Sci. Rep. 4 (4060). http://dx.doi.org/10.1038/srep04060.

Fall, A.B., Lindström, S.B., Sprakel, J., Wågberg, L., 2013. A physical cross-linking process of cellulose nanofibril gels with shear-controlled fibril orientation. Soft Matter 9 (6), 1852–1863.

Fudos, I., Hoffmann, C.M., 1997. A graph-constructive approach to solving systems of geometric constraints. ACM Trans. Graph. 16 (2), 179–216. http://dx.doi.org/10.1145/248210.248223.

Gao, X.-S., Lin, Q., Zhang, G.-F., 2006. A c-tree decomposition algorithm for 2d and 3d geometric constraint solving. Comput. Aided Des. 38 (1), 1–13. http://dx.doi.org/10.1016/j.cad.2005.03.002.

Garcia-Gonzalo, F.R., Reiter, J.F., 2012. Scoring a backstage pass: mechanisms of ciliogenesis and ciliary access. J. Cell Biol. 197 (6), 697–709. http://dx.doi.org/10.1083/jcb.201111146. http://jcb.rupress.org/content/197/6/697.full.pdf+html. http://jcb.rupress.org/content/197/6/697.abstract.

Gaspar, M.E., Csermely, P., 2012. Rigidity and flexibility of biological networks. arXiv:1204.6389.

Graver, J.E., Servatius, B., Servatius, H., 1993. Combinatorial Rigidity. Graduate Studies in Mathematics. American Mathematical Society, Providence (RI). http://opac.inria.fr/record=b1131810.

Heyde, M., 2013. Structure and motion of a 2D glass. Science 342 (6155), 201–202. http://dx.doi.org/10.1126/science.1245217. http://www.sciencemag.org/content/342/6155/201.short. http://www.sciencemag.org/content/342/6155/201.full.pdf.

Hidalgo, M.R., Joan-Arinyo, R., 2011. The reachability problem in constructive geometric constraint solving based dynamic geometry. J. Autom. Reason., 1–24.

Hoffman, C.M., Lomonosov, A., Sitharam, M., 2001a. Decomposition plans for geometric constraint systems, Part I: Performance measures for cad. J. Symb. Comput. 31 (4), 367–408. http://dx.doi.org/10.1006/jsco.2000.0402.

Hoffman, C.M., Lomonosov, A., Sitharam, M., 2001b. Decomposition plans for geometric constraint problems, Part II: New algorithms. J. Symb. Comput. 31 (4), 409–427. http://dx.doi.org/10.1006/jsco.2000.0403.

Hoffmann, C.M., Lomonosov, A., Sitharam, M., 1997. Finding solvable subsets of constraint graphs. In: Smolka, G. (Ed.), Principles and Practice of Constraint Programming. CP97. In: Lect. Notes Comput. Sci., vol. 1330. Springer, Berlin, Heidelberg, pp. 463–477.

Hollister, S.J., 2007. Bme 332: introduction to biosolid mechanics. http://www.umich.edu/~bme332/ch10ligten/ligten2.jpg.

Jackson, B., Jordán, T., 2008. Pin-collinear body-and-pin frameworks and the molecular conjecture. Discrete Comput. Geom. 40 (2), 258–278. http://dx.doi.org/10.1007/s00454-008-9100-z.

Jacobs, D.J., Hendrickson, B., 1997. An algorithm for two-dimensional rigidity percolation: the pebble game. J. Comput. Phys. 137 (2), 346–365. http://dx.doi.org/10.1006/jcph.1997.5809. http://www.sciencedirect.com/science/article/pii/S0021999197958095.

Jermann, C., Trombettoni, G., Neveu, B., Mathis, P., 2006. Decomposition of geometric constraint systems: a survey. Int. J. Comput. Geom. Appl. 16 (5–6), 379–414.

Joan-Arinyo, R., Soto-Riera, A., Vila-Marta, S., Vilaplana-Pastó, J., 2003. Transforming an under-constrained geometric constraint problem into a well-constrained one. In: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications. SM'03. ACM, New York, NY, USA, pp. 33–44.

Joan-Arinyo, R., Soto-Riera, A., Vila-Marta, S., Vilaplana-Pastó, J., 2004. Revisiting decomposition analysis of geometric constraint graphs. In: Solid Modeling and Applications. Comput. Aided Des. 36 (2), 123–140. http://dx.doi.org/10.1016/S0010-4485(03)00057-5. http://www.sciencedirect.com/science/article/pii/S0010448503000575.

Key Curriculum, 1995. The geometer's sketchpad. http://www.keycurriculum.com/.

Laman, G., 1970. On graphs and rigidity of plane skeletal structures. J. Eng. Math. 4, 331–340. http://dx.doi.org/10.1007/BF01534980.

Lee, A., Streinu, I., 2007. Pebble game algorithms and sparse graphs. arXiv:math/0702129.

Lee, A., Streinu, I., Theran, L., 2007. Graded sparse graphs and matroids. J. Univers. Comput. Sci. 13 (11), 1671–1679.

Lomonosov, A., 2004. Graph and combinatorial analysis for geometric constraint graphs. Ph.D. thesis. University of Florida.

Lovasz, L., Yemini, Y., 1982. On generic rigidity in the plane. SIAM J. Algebr. Discrete Methods 3 (1), 91–98.

Needleman, D.J., Ojeda-Lopez, M.A., Raviv, U., Miller, H.P., Wilson, L., Safinya, C.R., 2004. Higher-order assembly of microtubules by counterions: from hexagonal bundles to living necklaces. Proc. Natl. Acad. Sci. USA 101 (46), 16099–16103. http://dx.doi.org/10.1073/pnas.0406076101. http://www.pnas.org/content/101/46/16099.abstract. http://www.pnas.org/content/101/46/16099.full.pdf+html.

Owen, J.C., 1991. Algebraic solution for geometry from dimensional constraints. In: Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications. SMA'91. ACM, New York, NY, USA, pp. 397–407.

Porta, J.M., Ros, L., Bohigas, O., Manubens, M., Rosales, C., Jaillet, L., 2014. An open-source toolbox for motion analysis of closed-chain mechanisms. In: Computational Kinematics. Springer, pp. 147–154. cUIK.

Rothemund, P.W.K., Papadakis, N., Winfree, E., 2015. Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biol. 2 (12). http://dx.doi.org/10.1371/journal.pbio.0020424.
Saxe, J., 1979. Embeddability of weighted graphs in k-space is strongly np-hard. In: Proceedings of 17th Allerton Conference in Communications, Control and Computing, pp. 480–489.
Servatius, B., 2000. On the rigidity of Ramanujan graphs. Ann. Univ. Sci. Bp. Eötvös, Sect. Math. 43, 165–170.
Siemens, 1999. D-cubed. http://www.plm.automation.siemens.com/en_us/products/open/d-cubed/.
Sitharam, M., 2005. Combinatorial approaches to geometric constraint solving: problems, progress, and directions. In: DIMACS: Ser. Discret. Math. Theor. Comput. Sci., vol. 67, p. 117.
Sitharam, M., 2006. Well-formed systems of point incidences for resolving collections of rigid bodies. Int. J. Comput. Geom. Appl. 16 (5–6), 591–615.
Sitharam, M., Gao, H., 2010. Characterizing graphs with convex and connected configuration spaces. Discrete Comput. Geom. 43 (3), 594–625. http://dx.doi.org/10.1007/s00454-009-9160-8.
Sitharam, M., Wang, M., 2014. How the beast really moves: Cayley analysis of mechanism realization spaces using caymos. In: SIAM Conference on Geometric and Physical Modeling. 2013, Comput. Aided Des. 46, 205–210.
Sitharam, M., Arbree, A., Zhou, Y., Kohareswaran, N., 2006. Solution space navigation for geometric constraint systems. ACM Trans. Graph. 25 (2), 194–213. http://dx.doi.org/10.1145/1138450.1138452.
Sitharam, M., Peters, J., Zhou, Y., 2010a. Optimized parametrization of systems of incidences between rigid bodies. J. Symb. Comput. 45 (4), 481–498.
Sitharam, M., Zhou, Y., Peters, J., 2010b. Reconciling conflicting combinatorial preprocessors for geometric constraint systems. Int. J. Comput. Geom. Appl. 20 (6), 631–651.
Sitharam, M., Wang, M., Gao, H., 2011a. Cayley configuration spaces of 2d mechanisms, Part I. Extreme points, continuous motion paths and minimal representations. Under review, arXiv:1112.6008.
Sitharam, M., Wang, M., Gao, H., 2011b. Cayley configuration spaces of 2d mechanisms, Part II. Combinatorial characterization. Under review, arXiv:1112.6009.
Sitharam, M., Tarifi, M., Wang, M., 2014. An incidence geometry approach to dictionary learning. In: Proceedings of CCCG 2014. Springer.
Smith, G., 1971. Plant cell wall structure and cell wall growth. Tuatara 19, 43–50.
Streinu, I., Theran, L., 2009. Sparse hypergraphs and pebble game algorithms. Eur. J. Comb. 30 (8), 1944–1964. http://dx.doi.org/10.1016/j.ejc.2008.12.018.
Tay, T.-S., 1976. Rigidity problems in bar and joint frameworks. Ph.D. thesis. University of Waterloo.
Todd, P., 2007. Geometry expressions: a constraint based interactive symbolic geometry system. In: Automated Deduction in Geometry. Springer, pp. 189–202.
Valdes, J., Tarjan, R.E., Lawler, E.L., 1979. The recognition of series parallel digraphs. In: Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing. ACM, pp. 1–12.
Wang, M., Sitharam, M., 2015. Algorithm 951: Cayley analysis of mechanism configuration spaces using CayMos: software functionalities and architecture. ACM Trans. Math. Softw. 41 (4).
White, N., Whiteley, W., 1987. The algebraic geometry of motions of bar-and-body frameworks. SIAM J. Algebr. Discrete Methods 8 (1), 1–32.
Wikimedia Commons, 2006. Tetrahedral 3d balls. https://commons.wikimedia.org/wiki/File:Tetrahedral-3D-balls.png.
Wikimedia Commons, 2007a. Chlamydomonas TEM 17. https://commons.wikimedia.org/wiki/File:Chlamydomonas_TEM_17.jpg.
Wikimedia Commons, 2007b. Rothemund DNA Sierpinski gasket. https://commons.wikimedia.org/wiki/File:Rothemund-DNA-SierpinskiGasket.jpg.
Wikimedia Commons, 2007c. Plant cell wall diagram. https://commons.wikimedia.org/wiki/File:Plant_cell_wall_diagram.svg.
Wikimedia Commons, 2008. Silica. https://commons.wikimedia.org/wiki/File:Silica.svg.
Wikimedia Commons, 2010. AFM height image of carboxymethylated nanocellulose adsorbed on a silica surface. https://commons.wikimedia.org/wiki/File:AFM_Innventia_nanocellulose.JPG.
Wikimedia Commons, 2012. Finite subdivision of a radial link. https://commons.wikimedia.org/wiki/File:Finite_subdivision_of_a_radial_link.png.
Wilson, M., Kumar, A., Sherrington, D., Thorpe, M.F., 2013. Modeling vitreous silica bilayers. Phys. Rev. B 87 (21), 214108. http://dx.doi.org/10.1103/PhysRevB.87.214108. arXiv:1303.5898.
Zhang, G.-F., Gao, X.-S., 2006. Well-constrained completion and decomposition for under-constrained geometric constraint problems. Int. J. Comput. Geom. Appl. 16 (5–6), 461–478.