On the atomic decomposition length of graphs and tensegrities

Jose David Fernández Rodríguez
Universidad de Málaga
Málaga, Málaga (Spain)
josedavid@geb.uma.es

David Orden Martín
Universidad de Alcalá
Alcalá de Henares, Madrid (Spain)
david.orden@uah.es

Abstract

In this study, a complexity measure for graphs and tensegrities is proposed, based on the concept of atomic decomposition.We state several results on the relationship between atomic decompositions and spaces of self-stresses for generically rigid graphs, and study the computational complexity of finding atomic decompositions of minimal length.

1 Introduction

The concept of tensegrity is nowadays widely used in many disciplines, from biomechanics to structural engineering. Loosely speaking, a tensegrity is a structure whose elements are bound together in a stressed state, while the whole is in equilibrium. Over the years, each discipline has tailored the meaning of the word tensegrity to its own needs. Consequently, there are many definitions of tensegrity, each one subtly different from the others. We will stick to one of the simplest, broadest and more tractable definitions.

Up to date, it is usual to discuss about the complexity of tensegrity structures without trying to measure it, among other reasons because finding an agreeable complexity measure is problematic. Furthermore, existing complexity measures are usually too general to be useful in a specific context. In this work we propose a method to measure the complexity of tensegrity structures, taking into account their tensegrity nature.

Before starting our discussion, let us present here briefly the concepts and results (mostly) formulated in [11] that we need to present our work:

- A finite *point configuration* $P := \{p_1, ..., p_n\}$ in $\mathbb{R}^d$ is in *general position* if no $d+1$ points lie on the same hyperplane. More restrictively, if the points are algebraically independent, the position is *generic*.
- A *framework* $G(P)$ in $\mathbb{R}^d$ is an embedding of the abstract graph $G = (V, E)$ on a finite point configuration P in $\mathbb{R}^d$ in general position, with straight edges.
- A *self-stress* w on a framework is an assignment of scalars w_{ij} (called tensions) to its edges, such that for each vertex i, the scaled sum of incident vectors $p_i - p_j$ is zero:

$$\forall i, \quad \sum_{ij \in E} w_{ij}(p_i - p_j) = 0$$

  Observe that self-stresses form a vector space.
- If a framework $G(P)$ has n vertices and e edges, its associated *rigidity matrix* $R(P)$ has e rows and nd columns, such that:
  - There is a row per edge ij of the framework, with $i < j$ and in lexicographic order.
  - Each block of d columns is associated to a vertex p_i, and it contains zeros except for each row corresponding to an incident edge ij, where it contains the d coordinates $p_i - p_j$.

J. Cáceres, C. Grima, A. Márquez, M.L. Puertas (eds.); Avances en Matemática Discreta en Andalucía (vol.2), Carmona (Sevilla), Noviembre 2011, pp. 1–9

Observe that if w is a self-stress on $G(P)$, then $w \cdot R(P) = 0$.

- A framework with a self-stress non-null on every edge is called a *tensegrity*, denoted as $G(P,w)$.
- An *atom* A in dimension d is a complete graph K_{d+2} embedded in $\mathbb{R}^d$. If the embedding is in general position, its space of self-stresses has dimension 1. Otherwise, it only admits a null stress, that is to say, $w_{ij} = 0,\ \forall i, j$.
- An *atomic decomposition* of a tensegrity $G(P,w)$ is a finite set of atoms (each atom corresponding to a set of $d+2$ points in P) such that the sum of their self-stresses is w. Note that the atoms in the decomposition may have edges ij not present in G, which cancel out to $w_{ij} = 0$ when the atom stresses are added up. In general, decompositions are not unique. The *length* of the decomposition is the cardinality of the set of atoms.

One of the main results in [11] is the development of an algorithm to generate atomic decompositions for tensegrity structures. The algorithm can be applied to a tensegrity $G(P,w)$ (Theorem 3.2 in [11]), or to an abstract graph $G = (V,E)$ (Algorithm 3.4 in [11]). In this work, we will refer to these variants as the *geometric* and *combinatorial algorithms*, respectively. Below, the combinatorial version of the algorithm is reproduced:

Algorithm 1.1. (Adapted from Algorithm 3.4 in [11]) *Atomic combinatorial decomposition*
INPUT: abstract graph $G = (V,E)$ and dimension d.
OUTPUT: (L, M, F), where L is a list of "atoms" (subsets of $(d+2)$ elements of V), M is a list containing the number of edges added for each atom in L, and F is a list of intermediate graphs.

1. *Initialize $L = \varnothing$, $M = \varnothing$, $F = [G]$.*
2. *While E is not empty, choose a vertex $a \in V$ and:*
   - 2.1 *If a has degree $d+1$, let $a_0, \ldots, a_d$ be its neighbors. Remove the edges aa_i from E. Let E' be the set of all the edges a_ia_j between the neighbours that were not in E.*
   - 2.2 *If a has degree at least $d+2$, choose $d+1$ neighbors $a_0, \ldots, a_d$ of a. Remove the edge aa_0 from E. Let E' be the set of all the edges a_ia_j between the neighbours that were not in E.*
   - 2.3 *If a has degree $\leq d$, remove its incident edges from E.*

   In cases 2.1 and 2.2, also add the edges from E' to E, insert the atom $\{a, a_0, \ldots, a_d\}$ to the list L, and $|E'|$ to the list M. In any case, also update the graph with the new set of edges and removing unconnected vertices, and add it to the list F.
3. *Return (L, M, F).*

It is important to note that both the geometric and the combinatorial algorithms are non-deterministic: different decompositions can be obtained by making different sets of choices at several points in the algorithms. In the combinatorial algorithm, the resulting decomposition of a graph G represents a set of constraints between the positions of vertices and/or self-stresses of edges of tensegrities with underlying graph G. More details in [11].

We will use the acronym SAL to refer to the smallest atomic length, i.e., the smallest size of the list L over all the possible outputs of Algorithm 1.1. We are interested in studying the SAL because it can be seen as a tool to understand the complexity of a tensegrity structure: it represents the minimal way to interlock a set of one-dimensional atoms to generate the structure. However, as we will see, the SAL itself is not a completely satisfying tool for measuring complexity and we will develop a related concept.

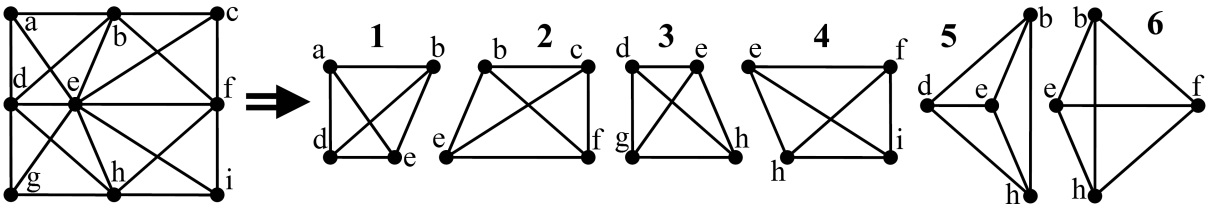

Figure 1: An abstract graph with one example of a combinatorial atomic decomposition.

Remark 1.2. *While the combinatorial algorithm formally operates in the domain of abstract graphs, it implicitly assumes that the graph is embedded in some unspecified tensegrity. In this context, it is significant to note that some valid tensegrities $G(P,w)$ might have a SAL shorter than the combinatorial SAL for G. Consider the decomposition shown in Figure 1: if we set a general point configuration P and define a self-stress w which is the sum of atoms $1,\ldots,4$, we can obtain a tensegrity $G(P,w)$ whose underlying graph G is the same as the one depicted in the figure. Therefore, while the geometric SAL of this specifically constructed $G(P,w)$ is 4, the combinatorial SAL for G is 6, because the combinatorial algorithm tacitly assumes that G is implicitly embedded in a tensegrity $G(P,w)$ as generic as possible, both in terms of the position P and of the self-stress w. However, as we will see in Section 2.2, the combinatorial problem can be reduced to the geometric one.*

From rigidity theory [9], we know that the set W of all possible self-stresses of a framework $G(P)$ is, in fact, the left kernel of the matrix $R(P)$. Similarly, the space of infinitesimal motions is the right kernel of $R(P)$, whose dimension (or number of degrees of freedom) is 0 if $G(P)$ is rigid. The following proposition relates the dimensions of both spaces:

Proposition 1.3. (Theorem 2.4.1 in [9]) *Let $G(P)$ be a framework in general position P in dimension d with $G=(V,E)$, $\left|W_{G(P)}\right|$ the dimension of the self-stress space and* $\mathrm{df}(G(P))$ *the number of degrees of freedom of $G(P)$. Then:*

$$\mathrm{df}(G(P)) = \begin{cases} \left|W_{G(P)}\right| - \binom{d+1}{2} + d\cdot|V| - |E|, & \text{if } |V| \geq d \\ \binom{|V|}{2} - |E|, & \text{if } |V| \leq d+1 \end{cases} \tag{1}$$

2 The structure of the space of self-stresses

For all practical purposes, we will be interested in graphs which are generically rigid, that is to say, rigid in any generic position. Non-generically rigid graphs can be tensegrities only in very degenerated positions. We will use $|W_G|$ to mean the dimension of $W_{G(P)}$ if P is a generic position, since it is constant for every generic position.

For generically rigid graphs, the space of self-stresses of the intermediate graphs (list F) in the decomposition algorithm changes according to the number of intermediate edges inserted along the decomposition (list M), and these numbers are directly related to the dimension of the space of self-stresses. To show this, we start with the following proposition:

Proposition 2.1. *If a graph $G=(V,E)$ is generically rigid in $\mathbb{R}^d$, then every intermediate graph in any decomposition in $\mathbb{R}^d$ will be generically rigid.*

Proof. Suppose that, for a given decomposition, the list of intermediate graphs is $F = [\ldots, G_i, G_{i+1} \ldots,]$. We can prove the proposition by showing that if an intermediate graph G_i is generically rigid, then G_{i+1} also is. We do it by cases:

- If the transition is done by step 2.1, let E' be the set of the added edges from G_i to G_{i+1}. If G_i is generically rigid, consider the rigidity of the graph G'_{i+1}, induced from G_{i+1} by removing the edges in E'. If G'_{i+1} is not generically rigid, then the edges incident to a in G_i must remove all degrees of freedom from G'_{i+1}, which allows the movement of some of its neighbours relative to others. In G_{i+1}, all distances between these neighbours are fixed by the edges added in E', so G_{i+1} is also generically rigid.

- If the transition is done by step 2.2, let aa_1 be the only edge removed from G_i to G_{i+1}. In G_{i+1}, the subgraph induced by the vertices $a, a_1, \ldots, a_d$ is a clique K_{d+2} minus an edge. A clique K_{d+2} is a generic rigidity circuit in dimension d (Theorem 3.11.9.a in [9]). If an edge is removed from a rigidity circuit, the resulting subgraph is still generically rigid (in chapter 3 in [9]). Therefore, the relative positions of a and a_1 are fixed in G_{i+1}, and hence this is also generically rigid.

- If the transition is done by step 2.3, at most d edges have been removed from G_i to G_{i+1}. Reasoning by the number of vertices in G_i:

  - if $|V_i| > d$, the vertex a must have exactly d incident edges for G_i to be rigid, and any self-stress must be always zero in these edges. Therefore, the equilibrium at other vertices in G_i is independent from these edges, and $|W_G|$ remains the same in G_i and G_{i+1}. By applying the first case of Equation 1, we get that $\mathrm{df}(G_{i+1}) = 0$, so G_{i+1} is also generically rigid.
  - if $|V_i| \leq d$, G_i must be a complete graph in order to be generically rigid by the second case of Equation 1, so G_{i+1} will also be a complete graph, hence also generically rigid. □

Therefore, generic rigidity is a property conserved through all intermediate graphs in the decomposition algorithm. To take advantage of this, we define the *Laman bound*:

Definition 2.2. *The* Laman bound *of a generically rigid graph $G = (V, E)$ in dimension d is defined as:*

$$B = \binom{d+1}{2} - d \cdot |V| + |E|$$

The Laman bound is modified by the decomposition algorithm in a very specific way:

Proposition 2.3. *Let G_i, G_{i+1} be two successive intermediate graphs in a combinatorial atomic decomposition in dimension d, with every vertex in G_i having degree at least d. Let B_i and B_{i+1} be their Laman bounds, respectively, and let e_i be the number of edges added from G_i to G_{i+1}. Then, $B_{i+1} = B_i + e_i - 1$.*

Proof. By the definition of Laman bound, it is easy to see that the equality holds in any case (steps 2.1, 2.2 and 2.3). □

If a graph is generically rigid and has at least d vertices, then $B = |W_G|$ by Proposition 1.3. Hence, by Proposition 2.1, each atom considered in the decomposition algorithm changes the dimension of the space of self-stresses of the intermediate graph, according to the number of edges added to the graph. Atoms adding no edges represent an independent dimension in W_G, atoms adding one edge must be tuned to cancel out the stress in that edge, so they do not affect the dimension of W_G, and atoms adding two or more edges represent an interlock between several other atoms. These considerations are summarized in the following result:

Proposition 2.4. *Let G be a generically rigid graph and B its Laman bound, and consider the lists L (of atoms) and* $M = \left[e_1, \ldots, e_{|L|}\right]$ *(of amounts of added edges) produced by an atomic decomposition of G. Then, the number of atoms in the decomposition is the Laman bound plus the total amount of (possibly repeated) edges added during the decomposition:*

$$|L| = B + \sum_{i=1}^{|L|} e_i \tag{2}$$

Proof. As G is generically rigid, Proposition 2.3 can be applied to every intermediate graph generated by the algorithm, so an atom introducing e edges changes B by $e-1$. Equality 2 is implied by combining Proposition 2.3 with the fact that the Laman bound must change from $B_0 = B$ to $B_{|L|} = 0$. □

The previous proposition provides a characterization of the SAL: it corresponds to the decompositions introducing the fewest edges in the intermediate steps. This holds even if the graph is not generically rigid.

Definition 2.5. *A decomposition is defined as* atomistic *it no edges are added in any intermediate step. By extension, a graph is* atomistic *if it admits an atomistic decomposition.*

It is clear that, in an atomistic decomposition, the self-stresses of the atoms form a basis for W_G. It is also evident that if a graph $G = (V,E)$ is decomposed and E' is the set of all edges added in the intermediate steps, then the graph $G' = (V, E \cup E')$ is atomistic. Also, chordal graphs that are generically rigid are also atomistic, and so are cliques K_n, whose minimal decomposition length is $\binom{n-d}{2}$.

2.1 Tensegrity complexity revisited

At first glance, it seems natural to define the complexity of a tensegrity as the smallest $|L|$ over all the possible outputs of Algorithm 1.1. Then, cliques would be maximally complex under this definition and, in general, very dense graphs would tend to have relatively large SALs. However, we can use the relationship between atomic decompositions and the structure of the generic self-stress space W_G of a graph G, in order to get a more descriptive definition. Since each edge added during the decomposition represents an interlock between the self-stresses of several atoms, the more edges, the more complex must be the interlock between the self-stresses of the atoms in the decomposition, in order to form the desired tensegrity. Thus, the number of edges added during the decomposition seems to be a better measure of the complexity of a tensegrity, bearing in mind that it should be used to compare graphs with the same or nearly similar number of vertices.

2.2 An algebraic characterization

Given a tensegrity $G(P,w)$, the geometric decomposition algorithm (check the combinatorial version in Algorithm 1.1 or the geometric version in [11]) admits an algebraic reformulation.

Remark 2.6. *Note that we are moving from combinatorics to geometry now. Most of the (combinatorial) graphs we are interested in can be realized as a (geometric) tensegrity although, as described in Remark 1.2, some valid tensegrities $G(P,w)$ might have a SAL shorter than the combinatorial SAL for G. If a generically rigid graph $G = (V,E)$ can be a tensegrity, the way to find P and w to construct a tensegrity $G(P,w)$ depends on the value of $|W_G|$:*

- *If $|W_G| > 0$, and no edge in E must have a null self-stress for a generic position P, any generic P will be suited to construct a tensegrity $G(P,w)$. Given a generic P (uniformly random configurations are almost certainly generic), a suitable self-stress can be generated almost certainly by*

taking a basis $\{w_1,\dots,w_k\}$ for $W_{G(P)}$, and finding a linear combination $w=\sum a_i w_i$ where coefficients a_i are drawn from a uniform distribution $U(0,1)$.

- *If $|W_G|=0$ and a tensegrity $G(P,w)$, can be defined, then the configuration P must be non-generic. In some cases, it can be found relatively easily (for example, for graphs with edge-inserting decompositions, as described in [11]).*

Definition 2.7. *Let P be a point configuration in dimension d. We define Q_m as the collection of all sets with exactly m vertices from P. We also define the* atomic self-stresses matrix*, named S, as follows:*

- *There is a row for each possible set $\{p_i,p_j\}\in Q_2$. The rows are ordered by the indices i, j, with $i<j$ and in lexicographic order.*
- *There is a column for each possible set $\{p_1,\dots,p_{d+2}\}\in Q_{d+2}$. As these $d+2$ points represent an atom A, let w_A be an unitary, non-null self-stress of A. Then, the column associated to A contains, for each row corresponding to a pair of points p_i,p_j of A, the self-stress assigned by w_A to the edge p_ip_j. In every other row, the value is 0. As with the rows, the columns are ordered in lexicographic order by the indices of the points.*

For a given tensegrity $G(P,w)$, the self-stress w can be represented as a column vector $\overline{w}$ of length $\binom{n}{d+2}$, where there is a position for each possible pair p_i,p_j of vertices of the framework, with $i<j$ and in lexicographic order, and the value in $\overline{w}$ for the position corresponding to the pair p_i,p_j is w_{ij} if there is an edge between them, and 0 otherwise.

It is easy to see that if w is a valid self-stress of $G(P)$, then every solution x to the underdetermined system of linear equations $S\cdot x=\overline{w}$ represents a linear combination of atoms which can be used to construct a tensegrity $G(P,w)$. Furthermore, let $\|x\|_0$ be number of non-zero elements in x. The problem of finding the SAL can be recast as finding a sparsest solution (with minimal $\|x\|_0$) to $S\cdot x=\overline{w}$. Since by Remark 2.6 the combinatorial algorithm can be restated in these terms for most graphs, this leads to some interesting properties for S:

Corollary 2.8. *If a configuration P in n vertices is in general position, the rank of the corresponding atomic self-stresses matrix S is $\binom{n-d}{2}$.*

Proof. K_n is an atomistic graph, so any SAL minimal decomposition corresponds to a basis for W_{K_n}. As the columns of the matrix S are the self-stresses for all possible atoms in K_n, its rank must be exactly the size of this basis, i.e., $\binom{n-d}{2}$. □

It is interesting to consider how Proposition 2.4 translates into this algebraic setting. First, we need some definitions:

Definition 2.9. *Let $G(P)$ be a framework, and Q_2,Q_{d+2} as in Definition 2.7. Given a set of pairs $D\subseteq Q_2$, we define its collection of* associated atoms *$A_D\subseteq Q_{d+2}$ as the collection of all sets of $d+2$ points including some member of D, i.e., $A_D=\{\{\dots,p_i,\dots,p_j,\dots\}\in Q_{d+2}\mid\{p_i,p_j\}\in D\}$.*

Definition 2.10. *Let $G(P,w)$ be a tensegrity, with $G=(V,E)$ its underlying graph, S its atomic self-stresses matrix and $\overline{w}$ the column vector associated to w, as in Definition 2.7. Let $E'\subseteq Q_2$ be a set of edges containing E. Let $G'=(V,E')$ be a graph formed by adding the edges in $E'-E$ to G. Let $A_{E'}\subseteq Q_{d+2}$ be the set of atoms associated to E'. The rows (resp. columns) of S correspond one-to-one to sets in Q_2 (resp. Q_{d+2}). We define the* core *of S (resp. $\overline{w}$) with respect to E', denoted $S_{E'}$ (resp. $\overline{w}_{E'}$), as a the submatrix of S (resp. $\overline{w}$) induced by E', whose rows correspond to E' and whose columns correspond to $A_{E'}$.*

Now, it is worth to note that any solution $x_{E'}$ to $S_{E'} \cdot x_{E'} = \overline{w}_{E'}$ induces a solution x (generated by padding $x_{E'}$ with zeros for the rows in S but not in $S_{E'}$) to $S \cdot x = \overline{w}$ (in fact, $\|x_{E'}\|_0 = \|x\|_0$). The following insight relates the structure of a combinatorial decomposition to the solution to the linear system of equations $S \cdot x = \overline{w}$:

Proposition 2.11. *Let $G(P,w)$ be a tensegrity with graph $G = (V,E)$, such that Proposition 2.4 holds (as in Remark 2.6). Then, the SAL corresponds to a minimal set of edges E_a such that, for $E' = E \cup E_a$, it holds that* $\operatorname{rank}(S_{E'}) = \operatorname{rank}([S_{E'} \,|\, \overline{w}_{E'}])$*, where $S_{E'}$ and $\overline{w}_{E'}$ are the cores of S and $\overline{w}$.*

Proof. Let L be any geometric decomposition of $G(P,w)$, let E_a be the set of edges added during decomposition L, and let $E' = E \cup E_a$. L can be recast as a solution $x_{E'}$ to $S_{E'} \cdot x_{E'} = \overline{w}_{E'}$. Then, Proposition 2.4 means that $\|x_{E'}\|_0$ is the sum of the dimension of the space of self-stresses of $G(P)$ and the number of rows in $S_{E'}$ corresponding to edges in E_a. Therefore, a SAL will correspond to a minimal set of edges E_a such that the system $S_{E'} \cdot x_{E'} = \overline{w}_{E'}$ is solvable. The solvability condition can be restated in terms of the ranks of the matrix and the augmented matrix, $\operatorname{rank}(S_{E'}) = \operatorname{rank}([S_{E'} \,|\, \overline{w}_{E'}])$. □

As a result, atomistic graphs can be characterized in algebraic terms: if a suitable tensegrity $G(P,w)$ is defined on a graph $G = (V,E)$ (as in Remark 2.6), then it is atomistic if and only if $\operatorname{rank}(S_E) = \operatorname{rank}([S_E \,|\, \overline{w}_E])$.

3 Computational complexity

The problem of computing the SAL seems to be NP-complete, but the question remains open. As the combinatorial atomic decomposition is defined through an algorithm with some non-deterministic steps, a choice (selected atom) taken in a step affects in highly convoluted ways to the choices available in all subsequent steps. Because of this, it is very difficult to reason directly about the computational complexity of finding the SAL. While several NP-complete problems seem to be very related to it, no obvious ways to reduce them to it have been devised:

The NP-complete **fixed clique covering problem** (FCC) [12, 5] is (for any given n) the problem of finding the minimal amount of copies of K_n needed to cover all the edges of a graph G, where these copies are not required to be induced subgraphs of G. FCC can be deemed as a non-trivial lower bound on the SAL in dimension d, but, unfortunately, the FCC tends to grossly underestimate the SAL.

The NP-complete **minimum fill-in** (MFI) [18] of a graph G is the minimal amount of edges whose addition makes the graph chordal. If a graph can be instantiated as a tensegrity in dimension d, its MFI plus its Laman bound can be regarded as an upper bound on its SAL, as the resulting chordal supergraph will necessarily have at least as many edges as a minimal atomistic supergraph of G (in fact the chordal supergraph will induce an atomic decomposition of G, although not necessarily minimal). As chordal graphs are also atomistic, the MFI induces a SAL in many cases. These facts hint that, even if the problem is not NP-complete, it must be relatively hard to solve.

By Definition 2.7, finding the SAL is equivalent to finding a sparsest solution to $S \cdot x = \overline{w}$. While this algebraic formulation of the problem may seem more tractable than the combinatorial one, the highly structured nature of the matrix S implies severe restrictions on the ways to reduce any known NP-complete problem to this one. Interestingly, the superproblem of finding a **sparsest solution to a general linear system of equations** $A \cdot x = b$ (over the real numbers) has been thoroughly studied [1, 2, 6, 7, 8, 10, 13, 14, 15, 17, 3], as it is relevant in many engineering applications, and is NP-complete [16]. However, many theoretical results have been developed, defining conditions on the matrix A for the sparsest solution to be calculable in polynomial time [4], provided that A is full-rank or at least has a large spark. Unfortunately, the matrix S has always low rank (Corollary 2.8), and the spark

is always extremely low (at most seven for dimension $d = 2$, since six vertices induce seven linearly dependent atoms). As a result, these methods yield very sub-optimal solutions to the SAL in most cases.

4 Acknowledgements

The present work was developed during a stay of Jose David Fernández Rodríguez at the Universidad de Alcalá, partially supported by the Departamento de Matemáticas de la Universidad de Alcalá. Jose David Fernández Rodríguez is partially supported by the Ministerio de Educación del Gobierno de España through a FPU grant (AP2007-03704). David Orden Martín is partially supported by grants MTM2008-04699-C03-02/MTM and MTM2011-22792. The second author wants to gratefully acknowledge fruitful discussions with Carlos Marijuan.

References

[1] A. Aspremont and L. El Ghaoui, Testing the nullspace property using semidefinite programming, *Mathematical Programming* (2010), 1–22.

[2] A.M. Bruckstein, D.L. Donoho and M. Elad, From sparse solutions of systems of equations to sparse modelling of signals and images, *SIAM Review* **51** (2009), 34–81.

[3] E. Candès, M. Wakin and S. Boyd, Enhancing sparsity by reweighted L1 minimization, *Journal of Fourier Analysis and Applications* **14** (2008), 877–905.

[4] S.S. Chen, D.L. Donoho and M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM Review* **43** (2001), 129–159.

[5] D.G. Corneil and J. Fonlupt, The complexity of generalized clique covering, *Discrete Applied Mathematics* **22** (1989), 109–118.

[6] D. Donoho, For most large underdetermined systems of linear equations the minimal L1-norm solution is also the sparsest solution, *Communications on pure and applied mathematics* **59** (2006), 797–829.

[7] S. Foucart and M.J. Lai, Sparsest solutions of underdetermined linear systems via L_q-minimization for $0 < q <= 1$, *Applied and Computational Harmonic Analysis* **26** (2009), 395–407.

[8] J.J. Fuchs, Sparsity and uniqueness for some specific under-determined linear systems, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (2005), v/729 - v/732 Vol. 5.

[9] J.E. Graver, B. Servatius and H. Servatius, *Combinatorial Rigidity*, Graduate Studies in Mathematics, Vol. 2, American Mathematical Society, 1993.

[10] R. Gribonval and M. Nielsen, Sparse representations in unions of bases, *IEEE Transactions on Information Theory* **49** (2003), 3320–3325.

[11] M. Guzman and D. Orden, From graphs to tensegrity structures: geometric and symbolic approaches, *Publicacions Matematiques* **50** (2006), 279–299.

[12] L.T. Kou, L.J. Stockmeyer and C.K. Wong, Covering edges by cliques with regard to keyword conflicts and intersection graphs, *Communications of the ACM* **21** (1978), 135–139.

[13] M. Lai, On sparse solutions of underdetermined linear systems, *Journal of Concrete and Applicable Mathematics* **8** (2010), 296–327.

[14] Y. Li and S.I. Amari, Two conditions for equivalence of 0-norm solution and 1-norm solution in sparse representation, *IEEE Transactions On Neural Networks* **21** (2010), 1189–1196.

[15] D. Malioutov, M. Cetin and A. Willsky, Optimal sparse representations in general overcomplete bases, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (2004), ii - 793-6 vol.2.

[16] B.K. Natarajan, Sparse approximate solutions to linear systems, *SIAM Journal on Computing* **24** (1995), 227–234.

[17] Y. Tsaig and D.L. Donoho, Breakdown of equivalence between the minimal L_1-norm solution and the sparsest solution, *Signal Processing* **86** (2006), 533–548.

[18] M. Yannakakis, Computing the minimum fill-in is NP-complete, *SIAM Journal on Algebraic and Discrete Methods* **2** (1981), 77–79.