Decomposition of arrow type positive semidefinite matrices with application to topology optimization

Michal Kočvara

Noname manuscript No.
(will be inserted by the editor)

Decomposition of arrow type positive semidefinite matrices with application to topology optimization

Michal Kočvara

Received: date / Accepted: date

Abstract Decomposition of large matrix inequalities for matrices with chordal sparsity graph has been recently used by Kojima et al. [8] to reduce problem size of large scale semidefinite optimization (SDO) problems and thus increase efficiency of standard SDO software. A by-product of such a decomposition is the introduction of new dense small-size matrix variables. We will show that for arrow type matrices satisfying suitable assumptions, the additional matrix variables have rank one and can thus be replaced by vector variables of the same dimensions. This leads to significant improvement in efficiency of standard SDO software. We will apply this idea to the problem of topology optimization formulated as a large scale linear semidefinite optimization problem. Numerical examples will demonstrate tremendous speed-up in the solution of the decomposed problems, as compared to the original large scale problem. In our numerical example the decomposed problems exhibit linear growth in complexity, compared to the more than cubic growth in the original problem formulation. We will also give a connection of our approach to the standard theory of domain decomposition and show that the additional vector variables are outcomes of the corresponding discrete Steklov-Poincaré operators.

Keywords Semidefinite optimization · Positive semidefinite matrices · Chordal graphs · Domain decomposition · Topology optimization

Mathematics Subject Classification (2010) 90C22 · 74P05 · 65N55 · 05C69

1 Introduction

General purpose algorithms and software for semidefinite optimization (SDO) are dominated by interior point and barrier type methods. Any such software exhibits two bottlenecks regarding computational complexity, and thus CPU time, and memory requirements.

This work has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement 813211 (POEMA).

Michal Kočvara
School of Mathematics, University of Birmingham, Birmingham B15 2TT, UK and Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 18208 Praha 8, Czech Republic, E-mail: m.kocvara@bham.ac.uk

The first one is the evaluation of the system matrix (Schur complement matrix or Hessian of augmented Lagrangian) in every step of the underlying Newton method. The second one is then the solution of a linear system with this matrix. For problems with large matrix inequalities, it is often the first bottleneck that dominates the CPU time and that prevents the user from solving large scale problems.

To circumvent this obstacle, the technique of decomposition of a large matrix inequality into several smaller ones proved to be efficient, at least for certain classes of problems. Decomposition of positive semidefinite matrices with a certain sparsity pattern was first investigated in Agler et al. [1] and, independently, by Griewank and Toint [4]. An extensive study has been recently published by Vandenberghe and Andersen [13]. We will call this technique *chordal decomposition*. It was first used in semidefinite optimization by Kojima and his co-workers; see [3, 10] and, more recently, [8]. The group also developed a preprocessing software for semidefinite optimization named SparseCoLO [2] that performs the decomposition of matrix constraints automatically.

The goal of this paper is twofold. Firstly, we introduce a new decomposition of arrow type positive semidefinite matrices called *arrow decomposition*. Unlike the chordal decomposition that generates additional dense matrix variables, arrow decomposition only requires additional vector variables of the same size, leading to significant reduction of number of variables in the decomposed problem. The second goal is to apply both decomposition techniques to the topology optimization problem. This problem arises from finite element discretization of a partial differential equation. We will show that techniques known from domain decomposition can be used to define the matrix decomposition. In particular, we will be able to control the number and size of the decomposed matrix inequalities. Even when using the chordal decomposition, this will allow us to gain tremendous speed-up when compared to approaches based on automatic chordal completion as in [2]. We will also give a connection of the arrow decomposition with the theory of domain decomposition and show that the additional vector variables are outcomes of the corresponding discrete Steklov-Poincaré operators.

To solve all semidefinite optimization problems, we will use the state of the art solver MOSEK [9]. Numerical examples will demonstrate tremendous speed-up in the solution of the decomposed problems, as compared to the original large scale problem. Moreover, in our numerical examples the arrow decomposition exhibits linear growth in complexity, compared to the higher than cubic growth when solving the original problem formulation.

Notation Let $\mathbb{S}^n$ be the space of $n \times n$ symmetric matrices, $A \in \mathbb{S}^n$, and $I \subset \{1, \dots, n\}$ with $s = |I|$. We denote

- by $(A)_{i,j}$ the (i,j)-th element of A;
- by $(A)_I$ the restriction of A to $\mathbb{S}^s$, i.e., the $s \times s$ submatrix of A with row and column indices from I ;
- by $O_{m,n}$ the $m \times n$ zero matrix; when the dimensions are clear from the context, we simply use O.

A matrix is called *dense* if all its elements are non-zeros. Otherwise, the matrix is called *sparse*. A matrix-valued function $A(x)$ is called dense if there exists $\bar{x}$ such that $A(\bar{x})$ is dense.

Let $A \in \mathbb{S}^n$. The undirected graph $G(N, E)$ with $N = \{1, \dots, n\}$ is called *sparsity graph of* A (or just *graph of* A) when $(i,j) \in E$ if and only if $(A)_{i,j} \neq 0$.

For an index set $I \subset \{1, \ldots, n\}$ we define

$$\mathbb{S}^n(I) := \{Y \in \mathbb{S}^n \mid (Y)_{i,j} = 0 \text{ if } (i,j) \notin I \times I\}$$
$$\mathbb{S}^n_+(I) := \{Y \in \mathbb{S}^n(I) \mid Y \succeq 0\}\,.$$

Furthermore, let $G(N,E)$ be an undirected graph with $N = \{1, \ldots, n\}$ and edge set $E \subseteq N \times N$. We define

$$\mathbb{S}^n(G) := \{Y \in \mathbb{S}^n \mid (Y)_{ij} = 0 \text{ if } (i,j) \notin E \cup \{(i,i)\}\}$$

and analogously $\mathbb{S}^n_+(G)$.

Let $G_s(N_s, E_s)$ be an induced subgraph of $G(N,E)$. Notice the difference between $\mathbb{S}^n(G_s)$ and $\mathbb{S}^n(N_s)$. If $A \in \mathbb{S}^n(N_s)$ then its restriction $(A)_{N_s}$ is a dense matrix. This is not true for $A \in \mathbb{S}^n(G_s)$, the sparsity pattern of which is given by the set of edges E_s. In particular, $\mathbb{S}^n(G_s) = \mathbb{S}^n(N_s)$ if and only if G_s is a maximal clique.

Finally, for functions from $\mathbb{R}^d \to \mathbb{R}$ we will use bold italics (such as $\boldsymbol{u}$ or $\boldsymbol{u}(\xi)$), while for vectors resulting from finite element discretization of these functions, we will use the same symbol but in italics (e.g. $u \in \mathbb{R}^n$).

2 Decomposition of positive semidefinite matrices

2.1 Matrices with chordal sparsity graphs

We first recall the well-studied case of matrices with chordal sparsity graph. The following theorem was proved independently by Grone, et al. [5], Griewank and Toint [4] and by Agler et al. [1]. A new, shorter proof can be found in [7].

Theorem 1 *Let $G(N,E)$ be an undirected graph with maximal cliques $C_1, \ldots, C_p$. The following two statements are equivalent:*

(i) $G(N,E)$ is chordal.
(ii) For any $A \in \mathbb{S}^n(G)$, $A \succeq 0$, there are matrices $Y_k \in \mathbb{S}^n_+(C_k)$, $k = 1, \ldots, p$, such that $A = Y_1 + Y_2 + \ldots + Y_p$.

Notice that this decomposition is not unique. However, Kakimura [7] has shown that there exist matrices Y_k^* minimizing $\sum_{k=1}^{p} \operatorname{rank} Y_k$ subject to $\sum_{k=1}^{p} Y_k = A$ and $Y_k \in \mathbb{S}^n_+(C_k)$ $(k = 1, \ldots, p)$ and that $\sum_{k=1}^{p} \operatorname{rank} Y_k^* = \operatorname{rank} A$.

2.2 Matrices embedded in those with a chordal sparsity graph

Let $A \in \mathbb{S}^n$, $n \geq 3$, with a sparsity graph $G = (N,E)$. Let the set of nodes $N = \{1, 2, \ldots, n\}$ be partitioned into $p \geq 2$ overlapping sets

$$N = I_1 \cup I_2 \cup \ldots \cup I_p\,.$$

Let $I_{k,\ell}$ denote the intersection of the kth and ℓth set, i.e.,

$$I_{k,\ell} := I_k \cap I_\ell\,, \quad (k,\ell) \in \Theta_p$$

with

$$\Theta_p := \{(i,j) \mid i = 1, \ldots, p-1;\ j = 2, \ldots, p;\ i < j\}\,.$$

Assumption 1. Let $1 \le k \le p$. There exists at least one index ℓ with $1 \le \ell \le p$, $\ell \ne k$, such that $I_k \cap I_\ell \ne \emptyset$.

Assumption 2. $I_k \cup I_\ell \ne I_k$ for all $1 \le k, \ell \le p$, $k \ne \ell$, i.e., no I_ℓ is a subset of any I_k.

Assumption 3. The intersections are "sparse" in the sense that for each $k \in \{1, \dots, p\}$ there are at most p_k indices ℓ_i such that $I_k \cap I_{\ell_i} \ne \emptyset$, $i = 1, \dots, p_k$, where $1 \le p_k \ll p$.

In a typical situation only $I_{k,k+1}$, $k = 1, \dots, p-1$, are not empty (corresponding to a block diagonal matrix with overlapping blocks) or I_k has a non-empty intersection with up to eight other sets (see Section 4).

Denote the induced subgraphs of $G(N, E)$ corresponding to I_k by $G_k(I_k, E_k)$, $k = 1, \dots, p$. *These subgraphs are not necessarily cliques.*

Lemma 1 *Let A be defined as above and Assumptions 1–3 hold. Then there exist matrices $Q_k \in \mathbb{S}^n(G_k)$ such that*

$$A = \sum_{k=1}^{p} Q_k \,.$$

The proof of the above lemma is obvious, as the principal submatrices associated with the subgraphs $G_k(I_k, E_k)$ cover all the nonzeros in A. However, although matrices Q_k are obviously non-unique, in our application in Section 3 they will be specified a priori and will be, in fact, used for the construction of the matrix A.

For all $k = 1, \dots, p$, let $\widehat{G}_k(I_k, \widehat{E}_k)$ denote a completion of $G_k(I_k, E_k)$, i.e., a clique in $G(N, E)$. According to Assumption 2, $\widehat{G}_k(I_k, \widehat{E}_k)$ are even maximal cliques. Clearly, $Q_k \in \mathbb{S}^n(\widehat{G}_k)$.

Assumption 4. The union $\widehat{G}(N, \widehat{E}) := \bigcup_{k=1}^{p} \widehat{G}_k(I_k, \widehat{E}_k)$ is a chordal graph.

The graph $\widehat{G}(N, \widehat{E}) \supset G(N, E)$ is called a *chordal extension* of $G(N, E)$; see, e.g., [13, Section 8.3].

Notice that the rather restrictive Assumption 4 is satisfied when A is a block diagonal matrix with overlapping blocks. It may *not* be satisfied in the application in Section 4; we will see, however, that it will not be needed in this application.

Theorem 2 *Let A be defined as above and Assumptions 1–4 hold. The following two statements are equivalent:*

(i) $A \succeq 0$.

(ii) There exist matrices $S_{k,\ell} \in \mathbb{S}^n(I_{k,\ell})$, $(k, \ell) \in \Theta_p$, such that

$$A = \sum_{k=1}^{p} \widetilde{Q}_k \text{ with } \widetilde{Q}_k = Q_k - \sum_{\ell:\ell<k} S_{\ell,k} + \sum_{\ell:\ell>k} S_{k,\ell}$$

and

$$\widetilde{Q}_k \succeq 0 \quad k = 1, \dots, p.$$

If $I_{k,\ell} = \emptyset$ or is not defined then $S_{k,\ell}$ is a zero matrix.

Proof Using the chordal extension $\widehat{G}(N,\widehat{E})$ of $G(N,E)$, we embed the matrix A into a set of matrices with chordal sparsity graphs with maximal cliques $\widehat{G}_k(I_k,\widehat{E}_k)$, $k=1,\ldots,p$. Then we can apply Theorem 1. Hence there exist matrices $Y_k \in \mathbb{S}^n_+(I_k)$, $k=1,\ldots,p$, such that $A = Y_1 + \ldots + Y_p$. Now, Y_k must be equal to Q_k for the "internal" indices of I_k, i.e., for all $(i,j) \in \big(I_k \setminus \big(\bigcup_{\ell:\ell>k}(I_{k,\ell}) \cup \bigcup_{\ell:\ell<k}(I_{\ell,k})\big)\big)^2$. Therefore the unknown elements of Y_k reduce to the overlaps $I_{k,\ell}$.

Having Q_k and Y_k, $k=1,\ldots,p$, we will now define the matrices the $S_{k,\ell}$ as follows. Firstly, for $k=1$ we select any solution $\{S_{1,\ell}\}_{I_{1,\ell}\neq\emptyset}$ of the equation

$$Y_1 = Q_1 + \sum_{\substack{\ell:\ell>1\\ I_{1,\ell}\neq\emptyset}} S_{1,\ell}\,.$$

Notice that many elements of matrices $S_{1,\ell}$ $(I_{1,\ell\neq\emptyset})$ are uniquely defined by this equation. Only elements with indices from nonempty intersections $I_{1,\ell}\cap I_{1,k}$ are not unique, as they appear in more than one matrix $S_{\bullet,\bullet}$ in the above equation.

Now, for $1<k<p$, we solve the equation

$$Y_k = Q_k - \sum_{\substack{\ell:\ell<k\\ I_{\ell,k}\neq\emptyset}} S_{\ell,k} + \sum_{\substack{\ell:\ell>k\\ I_{k,\ell}\neq\emptyset}} S_{k,\ell}\,.$$

All matrices $S_{\ell,k}$, $\ell<k$, were defined in steps $1,\ldots,k-1$, hence we are in the same situation as above and select any solution $\{S_{k,\ell}\}_{\ell>k,I_{k,\ell}\neq\emptyset}$ of the above equation. Any selection of the non-unique elements of $S_{\bullet,\bullet}$ will be consistent with the last equation

$$Y_p = Q_p - \sum_{\substack{\ell:\ell<p\\ I_{\ell,p}\neq\emptyset}} S_{\ell,p}$$

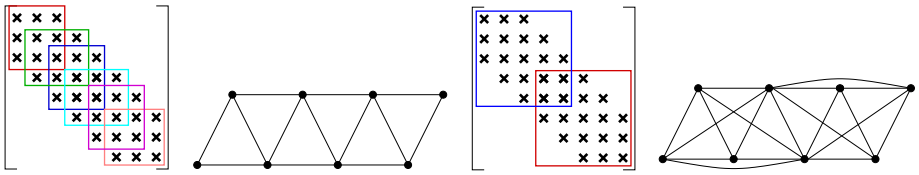
because we know that $A=\sum_{k=1}^p Y_k = \sum_{k=1}^p Q_k$. Therefore $A=\sum_{k=1}^p \widetilde{Q}_k$ and the assertion follows. □

Theorem 2 allows us to define the decomposition at our will, within the limits of the assumptions; even for matrices with chordal sparsity graph the decomposition does not have to be driven by the maximal cliques. This is illustrated in the following example.

Example 1 Consider a 7×7 five-diagonal matrix as schematically depicted in Fig. 1a; here crosses represent nonzero real numbers. The sparsity graph of this matrix is shown in Fig. 1b. This graph is chordal with six maximal cliques corresponding to the six triangles. Following Theorem 1, the chordal decomposition will use six principal submatrices shown in Fig. 1a. However, using Theorem 2 we can choose to decompose the matrix to only two principal submatrices, as shown in Fig. 1c. In this case, $I_1=\{1,2,3,4,5\}$, $I_2=\{4,5,6,7,8\}$ and $I_{1,2}=\{4,5\}$. The corresponding chordal extension with two maximal cliques is shown in Fig. 1d. ◊

2.3 Arrow type matrices

Let us now consider a particular type of sparse matrices, the arrow type matrices. Let again $A\in\mathbb{S}^n$, $n\geq 3$, and let I_k, $I_{k,\ell}$ and $G_k(I_k,E_k)$, $k=1,\ldots,p$, be defined as in the previous section.



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

Fig. 1: Decomposition of a five-diagonal matrix by maximal cliques, according to Theorem 1 (a), the sparsity graph of this matrix (b), decomposition to two blocks according to Theorem 2 (c), and the corresponding chordal extension of the graph with two maximal cliques (d).

Assume again that A is a sum of matrices associated with G_k:

$$A = \sum_{k=1}^{p} A_k, \quad A_k \in \mathbb{S}^n(G_k)\,.$$

Further, let $B \in \mathbb{R}^{n\times m}$, $B = \sum_{k=1}^{p} B_k$ with B_k, $k = 1,\dots,p$, being rectangular matrices such that

$$(B_k)_{i,j} = 0 \quad \text{for } i \notin I_k$$

and assume that

$$m < \min_{\substack{k,\ell=1,\dots,p\\ k<\ell}} |I_{k,\ell}|\,. \tag{1}$$

We also define

$$\widehat{I}_k = I_k \cup \{n+1,\dots,n+m\}\,, \quad k = 1,\dots,p$$

and

$$\widehat{I}_{k,\ell} = I_{k,\ell} \cup \{n+1,\dots,n+m\}\,, \quad (k,l) \in \Theta_p\,. \tag{2}$$

Finally, let $C \in \mathbb{S}^m$ be positive definite. We define the following *arrow type matrix*:

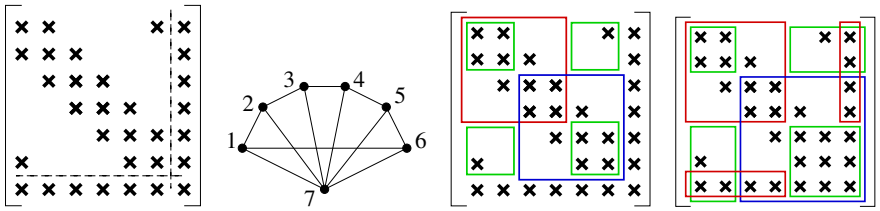
$$M = \sum_{k=1}^{p} M_k + \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} \quad \text{where } M_k = \begin{bmatrix} A_k & B_k \\ B_k^\top & 0 \end{bmatrix},\ k = 1\dots,p\,. \tag{3}$$

According to the definition of A_k and B_k, we have that

$$(M_k)_{i,j} = 0 \text{ for } (i,j) \notin \widehat{I}_k \times \widehat{I}_k, \quad k = 1,\dots,p\,. \tag{4}$$

The simplest example of an arrow type matrix is a block diagonal matrix with overlapping blocks and with additional rows and columns corresponding to matrices B and C. The next example presents another typical situation.

Example 2 Consider a 7×7 matrix as shown in Fig. 2a. The sparsity graph of this matrix is shown in Fig. 2b. This graph is not chordal: the cycle 1–2–3–4–5–6–1 does not have a chord. Let $I_1 = \{1,2,3,4\}$, $I_2 = \{3,4,5,6\}$, $I_3 = \{1,2,5,6\}$ and thus $\widehat{I}_k = I_k \cup \{7\}$. Notice that, due to (1), I_3 must contain at least two nodes from I_1. This decomposition satisfies Assumptions 1–4. In particular, if we extend all subgraphs associated with the decomposition to cliques, we will obtain a dense matrix with a complete, i.e., chordal sparsity graph; hence also Assumption 4 is satisfied. $\Diamond$



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

Fig. 2: An example of an arrow type matrix (a), its sparsity graph (b), and an example of its decomposition I_1, I_2, I_3 satisfying Assumptions 1–4 (c) and the associated decomposition $\widehat{I}_1, \widehat{I}_2, \widehat{I}_3$ (d).

Notice that the structure of the overlapping blocks can be more complicated and that, in general, A (the arrow "shaft") does not have to be a band matrix. Such matrices arise in the application introduced later in Section 3; see Figure 6 and 7. In this application, we will have $m = 1$, so that B will be an n-vector and $C \in \mathbb{R}$. However, in this section we consider the more general situation which may be useful in other applications.

Before going on, we need the following auxilliary result. It tells us that we do not need to consider all intersections of the extended sets $\widehat{I}_k$ but only those that are extensions of $I_{k,\ell}$, i.e., only $\widehat{I}_{k,\ell}$ as defined in (2).

Lemma 2 *Let $G(N_G, E_G)$ be a chordal graph with maximal cliques $G_k(I_k, E_k)$, $k = 1, \ldots, p$. Let $H(N_H, E_H)$ be a complete graph, disjoint with G. Define*

$$\widetilde{N} = N_G \cup N_H, \quad \widetilde{E} = E_G \cup E_H \cup E_{GH}\,,$$

where E_{GH} contains all edges with one vertex in N_G and one vertex in N_H. Define the extension of G as $\widetilde{G}(\widetilde{N}, \widetilde{E})$. Similarly, define $\widetilde{G}_k(\widetilde{I}_k, \widetilde{E}_k)$ with

$$\widetilde{I}_k = I_k \cup N_H, \quad \widetilde{E}_k = E_k \cup E_H \cup E_{kH}, \quad k = 1, \ldots, p\,,$$

where E_{kH} contains all edges with one vertex in I_k and one vertex in N_H. Then $\widetilde{G}(\widetilde{N}, \widetilde{E})$ is a chordal graph with maximal cliques $\widetilde{G}_k(\widetilde{I}_k, \widetilde{E}_k)$, $k = 1, \ldots, p$.

Proof Obviously, $\widetilde{G}_k(\widetilde{I}_k, \widetilde{E}_k)$ are cliques, as all vertices in $\widetilde{I}_k$ are adjacent, by assumption (G_k is a clique and H is complete) and by construction. Further, by construction, every maximal clique $G_k(I_k, E_k)$ is included in some clique in $\widetilde{G}(\widetilde{N}, \widetilde{E})$. Now suppose that $K(N_K, E_K)$ is a maximal clique in $\widetilde{G}$; then $N_H \subset N_K$. We want to show that $K \setminus H$ is a maximal clique in G. Assume, by contradiction, that it is not. Then there is a vertex $v \in G$ such that a graph with vertices $v \cup (N_K \setminus N_H)$ is a clique in G. Hence a graph with vertices $N_K \cup v$ is a clique in $\widetilde{G}$ which contradicts maximality of K. Finally, $\widetilde{G}$ is chordal, as any newly created cycle with vertices in N_H contains a chord in E_{GH}. □

We can now adapt Theorem 2 to the arrow type structure.

Corollary 1 *Let Assumptions 1–4 hold. Let M be defined as in (3). The following two statements are equivalent:*

(i) $M \succeq 0$.

(ii) There exist matrices $S_{k,\ell} \in \mathbb{S}^n(\widehat{I}_{k,\ell}),\ (k,\ell) \in \Theta_p$, *such that*

$$M = \sum_{k=1}^{p} \widetilde{M}_k \ \text{ with } \ \widetilde{M}_k = M_k - \sum_{\ell:\ell<k} S_{\ell,k} + \sum_{\ell:\ell>k} S_{k,\ell}$$

and

$$\widetilde{M}_k \succeq 0 \quad k = 1, \ldots, p\,.$$

If $I_{k,\ell} = \emptyset$ *or is not defined then* $S_{k,\ell}$ *is a zero matrix.*

Proof Let $H(N_H, E_H)$ be a sparsity graph of the matrix C; this is a complete graph. Recall that $\widehat{G}(N, \widehat{E})$ is the chordal extension of $G(N, E)$, the sparsity graph of the matrix A. The maximal cliques of $\widehat{G}$ are $\widehat{G}_k(I_k, \widehat{E}_k)$. The matrix B is dense and so all vertices from N_H are adjacent to all vertices in N. Then, by Lemma 2, the chordal extension of the sparsity graph of M has maximal cliques $\widehat{G}_k(\widehat{I}_k, \widehat{E}_k)$. The rest is a direct application of Theorem 2 with $Q_k = M_k$ for $k = 1, \ldots, p-1$, and $Q_p = \begin{bmatrix} A_p & B_p \\ B_p^\top & C \end{bmatrix}$. □

Under additional assumptions, but leaving out Assumption 4, we can strengthen the above corollary as follows.

Theorem 3 *Let Assumptions 1–3 hold. Assume that* $A_k \succeq 0$, $k = 1, \ldots, p$, $A \succ 0$ *and* $C \succ 0$. *Let* M *be defined as in (3). The following two statements are equivalent:*

(i) $M \succeq 0\,.$

(ii) There exist matrices $D_{k,\ell} \in \mathbb{R}^{n\times m}$ *such that* $(D_{k,\ell})_{i,j} = 0$ *for* $(i,j) \notin I_{k,\ell} \times \{1, \ldots m\}$, $(k,\ell) \in \Theta_p$, *and matrices* $C_k \in \mathbb{S}^m$, $k = 1, \ldots, p$, *such that*

$$M = \sum_{k=1}^{p} \widetilde{M}_k, \ \text{with } \widetilde{M}_k = M_k - \sum_{\ell:\ell<k} \begin{bmatrix} 0 & D_{\ell,k} \\ D_{\ell,k}^\top & 0 \end{bmatrix} + \sum_{\ell:\ell>k} \begin{bmatrix} 0 & D_{k,\ell} \\ D_{k,\ell}^\top & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & C_k \end{bmatrix}$$

and

$$\widetilde{M} \succeq 0, \quad k = 1, \ldots, p\,.$$

If $I_{k,\ell} = \emptyset$ *or is not defined then* $D_{k,\ell}$ *is a zero matrix.*

Proof We will prove the theorem by constructing matrices $D_{k,k+1}$ and C_k. By assumption, A is positive definite, so that we can define

$$X = A^{-1}B\,, \quad \text{i.e.,} \quad \sum_{k=1}^{p} A_k X = \sum_{k=1}^{p} B_k\,. \tag{5}$$

Then

$$(A_k X)_{i,j} = (B_k)_{i,j} \quad \text{for } i \in I_k \setminus \left(\bigcup_{\ell:\ell>k} (I_{k,\ell}) \cup \bigcup_{\ell:\ell<k} (I_{\ell,k}) \right),\ j = 1, \ldots, p\,. \tag{6}$$

We define $D_{k,k+1}$ and C_k as follows. For $k = 1$, we solve the equation

$$A_1 X - B_1 = \sum_{\substack{\ell:\ell>1 \\ I_{1,\ell}\neq\emptyset}} D_{1,\ell}\,.$$

As in the proof of Theorem 2, some elements of thus defined $D_{1,\ell}$ may not be unique; in this case, we just select a solution. Then, for any $1 < k < p$, we solve the equation

$$A_k X - B_k = -\sum_{\substack{\ell:\ell<k \\ I_{\ell,k}\neq\emptyset}} D_{\ell,k} + \sum_{\substack{\ell:\ell>k \\ I_{k,\ell}\neq\emptyset}} D_{k,\ell}$$

to define $D_{k,\ell}$, $\ell > k$, analogously to Theorem 2. Any selection of the non-unique elements of $D_{\bullet,\bullet}$ will be consistent with the last equation

$$A_p X - B_p = -\sum_{\substack{\ell:\ell<p \\ I_{\ell,p}\neq\emptyset}} D_{\ell,p}$$

because of (5). From (4) and (6) we see that $D_{k,\ell}$ is only non-zero on $I_{k,\ell}$, $(k,\ell) \in \Theta_p$, as required.

Define further

$$\widehat{C}_k = X^\top A_k X, \;\; k = 1, \ldots, p,$$
$$C_k = \widehat{C}_k, \;\; k = 1, \ldots, p-1 \quad \text{and} \quad C_p = C - \sum_{k=1}^{p-1} C_k \,.$$

Now the matrices defined for $k = 1, \ldots, p$ by

$$\widehat{M}_k = M_k - \sum_{\substack{\ell:\ell<k \\ I_{\ell,k}\neq\emptyset}} \begin{bmatrix} 0 & D_{\ell,k} \\ D_{\ell,k}^\top & 0 \end{bmatrix} + \sum_{\substack{\ell:\ell>k \\ I_{k,\ell}\neq\emptyset}} \begin{bmatrix} 0 & D_{k,\ell} \\ D_{k,\ell}^\top & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \widehat{C}_k \end{bmatrix} = \begin{bmatrix} A_k & A_k X \\ X^\top A_k & X^\top A_k X \end{bmatrix}$$

are clearly positive semidefinite with (at least) m zero eigenvalues. We set $\widetilde{M}_k = \widehat{M}_k$, $k = 1, \ldots, p-1$, and $\widetilde{M}_p = M_p - \sum_{\substack{\ell:\ell<p \\ I_{\ell,p}\neq\emptyset}} \begin{bmatrix} 0 & D_{\ell,p} \\ D_{\ell,p}^\top & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & C_p \end{bmatrix}$. By construction, $M = \sum_{k=1}^{p} \widetilde{M}_k$.

It remains to show that $\widetilde{M}_p = \begin{bmatrix} A_p & A_p X \\ X^\top A_p & C - \sum_{k=1}^{p-1} C_k \end{bmatrix} \succeq 0$ whenever $M \succeq 0$. As $A_p \succeq 0$ by assumption, positive semidefiniteness of $\widetilde{M}_p$ amounts to

$$C - \sum_{k=1}^{p-1} C_k - X^\top A_p A_p^{-1} A_p X = C - \sum_{k=1}^{p} X^\top A_k X \succeq 0$$

which, by (5), is the same as

$$C - B^\top X \succeq 0 \,.$$

By the Schur complement theorem, the last inequality is equivalent to $M \succeq 0$. This completes the proof. □

We will call the decomposition of arrow type matrices using Corollary 1 *chordal decomposition* and the one using Theorem 3 *arrow decomposition*.

Complexity remarks Let Assumptions 1–4 hold. Let r be the number of non-empty sets $I_{k,\ell}$, $(k,\ell) \in \Theta_p$. Comparing Corollary 1 with Theorem 3 we see that both provide us with a replacement of a "large" matrix inequality $M \succeq 0$ to a number of smaller ones $\widetilde{M}_k \succeq 0$, $k = 1, \ldots, p$. However, while in Corollary 1 we have to introduce r additional matrix variables of sizes $|\widehat{I}_{k,\ell}| \times |\widehat{I}_{k,\ell}|$, in Theorem 3 we only have r additional matrix variables of sizes $|I_{k,\ell}| \times m$ and p matrix variables of size $m \times m$. Recall that $m < \min_{\substack{k,\ell=1,\ldots,p \\ k<\ell}} |I_{k,\ell}|$ and, in our application below, $m = 1$, so the additional variables in Theorem 3 are vectors instead of matrices of the same dimension in Corollary 1, offering thus significant reduction in the dimension of the additional variables.

The example below shows that the arrow decomposition does not only lead to a problem of smaller dimension, it also allows us to use decompositions that do not satisfy Assumption 4. In particular, the arrow decompositions can be sparser, with smaller overlaps and hence leading to sparser and smaller SDO problem.

Example 3 Consider a 13×13 matrix as shown in Fig. 3a. The sparsity graph of this matrix is shown in Fig. 3b, where the central node corresponds to the last index in the matrix. Let us compare the arrow decomposition with the chordal decomposition.

Arrow decomposition As in Example 2, we decompose the 12×12 leading principal submatrix into six 4×4 principal submatrices, as shown in Fig. 3c; here $I_1^{\mathrm{A}} = \{1,2,3,4\}$, $I_2^{\mathrm{A}} = \{3,4,5,6\}$, $I_3^{\mathrm{A}} = \{5,6,7,8\}$, $I_4^{\mathrm{A}} = \{7,8,9,10\}$, $I_5^{\mathrm{A}} = \{9,10,11,12\}$, $I_6^{\mathrm{A}} = \{1,2,11,12\}$. Hence, with $\widehat{I}_k^{\mathrm{A}} = I_k^{\mathrm{A}} \cup \{13\}$, $k = 1, \ldots, 6$, we get six intersections $\widehat{I}_{k,\ell}^{\mathrm{A}}$, all of dimension 3. This decomposition satisfies Assumptions 1–3 but *not* Assumption 4. Indeed, if we extend the subgraphs associated with the decomposition to cliques, we obtain the graph shown in Fig. 3d. This graph is, however, not chordal: for instance, the cycle 1–4–6–7–9–11–1 does not have a chord. Therefore, we cannot apply neither of the theorems based on chordal decomposition (Theorems 1, 2, Corollary 1); however, we *can* apply Theorem 3 above. As a result, we get six additional vector variables of dimension 3 corresponding to $\widehat{I}_{k,\ell}^{\mathrm{A}}$.

Chordal decomposition Chordal decomposition must satisfy Assumption 4: the closest one to the above is $I_1^{\mathrm{C}} = \{1,2,3,4,12\}$, $I_2^{\mathrm{C}} = \{3,4,5,6,12\}$, $I_3^{\mathrm{C}} = \{5,6,7,8,12\}$, $I_4^{\mathrm{C}} = \{7,8,9,10,12\}$, $I_5^{\mathrm{C}} = \{9,10,11,12\}$ with $\widehat{I}_k^{\mathrm{C}} = I_k^{\mathrm{C}} \cup \{13\}$, $k = 1, \ldots, 5$, and thus with four intersections $\widehat{I}_{1,2}^{\mathrm{C}} = \{3,4,12,13\}$, $\widehat{I}_{2,3}^{\mathrm{C}} = \{5,6,12,13\}$, $\widehat{I}_{3,4}^{\mathrm{C}} = \{7,8,12,13\}$, $\widehat{I}_{4,5}^{\mathrm{C}} = \{9,10,12,13\}$; see Fig. 3e for the decomposition of the leading principal submatrix, Fig. 3f for the corresponding extended chordal graph and Fig. 3g for the matrix associated with the extended graph. In this case, we need five additional matrix variables of dimension 4×4. ◊

Two natural questions arise:

1. Are the additional assumptions of Theorem 3 too restrictive? Are there any applications satisfying them?
2. Is it worth reducing the dimension of the additional variables? Will it bring any significant savings of CPU time when solving the decomposed problem?

Both questions will be answered in the rest of the paper using a problem from structural optimization.

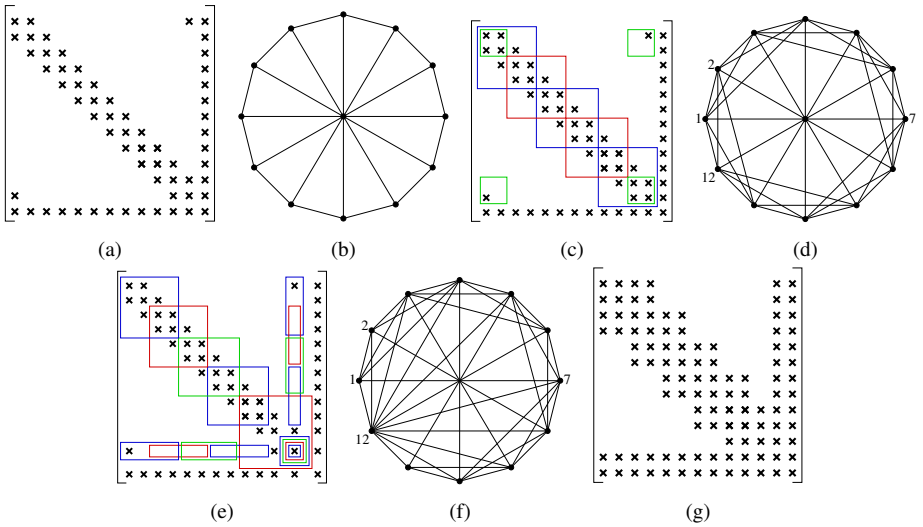


Fig. 3: An example of an arrow type matrix (a) its sparsity graph (b), its arrow decomposition satisfying Assumptions 1–3 (c) and the corresponding graph extension (d), its chordal decomposition (e), the corresponding extension of its sparsity graph to a chordal graph (f) and, finally, the completion of the matrix corresponding the chordal graph (g).

3 Application: Topology optimization problem, semidefinite formulation

Consider an elastic body occupying a d-dimensional bounded domain $\Omega \subset \mathbb{R}^d$ with a Lipschitz boundary $\partial\Omega$, where $d \in \{2, 3\}$. By $\boldsymbol{u}(\xi) \in \mathbb{R}^d$ we denote the displacement vector at a point ξ, and by

$$\boldsymbol{e}_{ij}(\boldsymbol{u}(\xi)) = \frac{1}{2}\left(\frac{\partial \boldsymbol{u}_i(\xi)}{\partial \xi_j} + \frac{\partial \boldsymbol{u}_j(\xi)}{\partial \xi_i}\right), \quad i, j = 1, \ldots, d$$

the (small-)strain tensor. We assume that our system is governed by linear Hooke's law, i.e., the stress is a linear function of the strain

$$\boldsymbol{\sigma}_{ij}(\xi) = \boldsymbol{E}_{ijk\ell}(\xi)\boldsymbol{e}_{k\ell}(\boldsymbol{u}(\xi)) \qquad \text{(in tensor notation)},$$

where $\boldsymbol{E}$ is the elastic (plane-stress for $d = 2$) stiffness tensor.

Assume that the boundary of Ω is partitioned as $\partial\Omega = \Gamma_u \cup \Gamma_f$, $\Gamma_u \cap \Gamma_f = \emptyset$ and that an external load function $\boldsymbol{f} \in [L_2(\Gamma_f)]^d$ is given. Define $\mathcal{V} = \{\boldsymbol{u} \in [H^1(\Omega)]^d \,|\, \boldsymbol{u} = 0 \text{ on } \Gamma_u\} \supset [H^1(\Omega)]^d$. The weak form of the linear elasticity problem reads as:

$$\begin{aligned} &\text{Find } \boldsymbol{u} \in \mathcal{V}, \text{ such that} \\ &\int_\Omega \boldsymbol{a}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_f} \boldsymbol{f}(\xi) \cdot \boldsymbol{v}(\xi)\, \mathrm{d}s, \quad \forall \boldsymbol{v} \in \mathcal{V}, \end{aligned} \tag{7}$$

where

$$\boldsymbol{a}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \langle \boldsymbol{x}(\xi)\boldsymbol{E}(\xi)\boldsymbol{e}(\boldsymbol{u}(\xi)), \boldsymbol{e}(\boldsymbol{v}(\xi))\rangle \, \mathrm{d}\xi\,. \tag{8}$$

In the basic topology optimization problem, the design variable is the multiplier $\boldsymbol{x} \in L_\infty(\Omega)$ of the elastic stiffness tensor $\boldsymbol{E}$ which is a function of the space variable ξ. We will consider the following constraints on $\boldsymbol{x}$:

$$\int_\Omega \boldsymbol{x}(\xi)\ \mathrm{d}\xi = V, \quad \underline{\boldsymbol{x}} \le \boldsymbol{x} \le \overline{\boldsymbol{x}} \ \text{ a.e. in } \Omega$$

with some given positive "volume" V and with $\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}} \in L_\infty(\Omega)$ satisfying $0 \le \underline{\boldsymbol{x}} \le \overline{\boldsymbol{x}}$ and $\int_\Omega \underline{\boldsymbol{x}}(\xi)\ \mathrm{d}\xi < V < \int_\Omega \overline{\boldsymbol{x}}(\xi)\ \mathrm{d}\xi$.

The *minimum compliance single-load topology optimization problem* reads as

$$\begin{aligned} &\inf_{\boldsymbol{x}\in L_\infty} \int_{\Gamma_f} \boldsymbol{f}(\xi)\cdot\boldsymbol{u}(\xi)\ \mathrm{d}\xi \qquad (9)\\ &\text{subject to}\\ &\quad \boldsymbol{u} \text{ solves (7)}\\ &\quad \int_\Omega \boldsymbol{x}(\xi)\ \mathrm{d}\xi = V\\ &\quad \underline{\boldsymbol{x}} \le \boldsymbol{x} \le \overline{\boldsymbol{x}} \ \text{ a.e. in } \Omega\,. \end{aligned}$$

The objective, the so called compliance functional, measures how well the structure can carry the load $\boldsymbol{f}$.

Problem (9) is now discretized using the standard finite element method; the details can be found, e.g., in [6,11]. In particular, we use quadrilateral elements, element-wise constant approximation of function $\boldsymbol{x}$ and element-wise bilinear approximation of the displacement field $\boldsymbol{u}$. After discretization, the variables will be vectors $x \in \mathbb{R}^m$ and $u \in \mathbb{R}^n$, where m is the number of finite elements and n the number of degrees of freedom (the number of finite element nodes times the spatial dimension). With every element we associate the local (symmetric and positive semidefinite) stiffness matrix K_i and (for elements including part of the boundary Γ_f) the discrete load vector f_i, $i = 1,\dots,m$. Now we can formulate the discretized version of the linear elasticity problem (7) as the following system of linear equations

$$K(x)u = f \qquad (10)$$

where $K(x) = \sum_{i=1}^m x_i K_i$ is the global stiffness matrix and $f = \sum_{i=1}^m f_i$ is the finite element assembly of the load vector.

The topology optimization problem (9) becomes

$$\begin{aligned} &\min_{u\in\mathbb{R}^n,\, x\in\mathbb{R}^m,\, \gamma\in\mathbb{R}} \ \gamma \qquad (11)\\ &\text{subject to}\\ &\quad K(x)u = f\\ &\quad f^\top u \le \gamma\\ &\quad \sum_{i=1}^m x_i \le V\\ &\quad \underline{x}_i \le x_i \le \overline{x}_i\,, \quad i = 1,\dots,m\,. \end{aligned}$$

Using the Schur complement theorem, the compliance constraint and the equilibrium equation can be written as one matrix inequality constraint:

$$Z(x) := \begin{bmatrix} K(x) & f \\ f^\top & \gamma \end{bmatrix} \succeq 0\,. \tag{12}$$

The minimum compliance problem can then be formulated as follows:

$$\begin{aligned} &\min_{x\in\mathbb{R}^m,\,\gamma\in\mathbb{R}} \ \gamma \\ &\text{subject to} \\ &\quad Z(x) \succeq 0 \\ &\quad \sum_{i=1}^{m} x_i \le V \\ &\quad \underline{x}_i \le x_i \le \overline{x}_i\,, \quad i = 1,\ldots,m\,. \end{aligned} \tag{13}$$

For ease of notation, in the rest of the paper we will restrict ourselves to the planar case $d = 2$. Generalization of all ideas to the three-dimensional case is straightforward.

4 Decomposition of the topology optimization problem (13)

Let $\Omega_h \subset \mathbb{R}^2$ be a polygonal approximation of Ω discretized by finite elements. Assume that Ω_h is partitioned into p non-overlapping subdomains D_k, $k = 1,\ldots,p$, whose boundaries coincide with finite element boundaries. In our examples $\Omega = \Omega_h$ is a rectangle, the underlying finite element mesh is regular and so is the partitioning into the subdomains. Confront Figure 4 that shows typical decomposition of Ω_h into $N_x \times N_y$ subdomains.

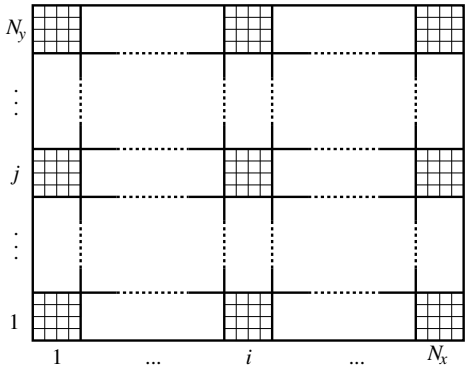


Fig. 4: Regular partitioning of the computational domain into subdomains coinciding with groups of finite elements.

Let I_k be the index set of all degrees of freedom associated with the subdomain D_k, $k = 1,\ldots,p$. The intersections of these index sets will include the degrees of freedom on the respective internal boundaries and will be again denoted by

$$I_{k,\ell} = I_k \cap I_\ell, \quad (k,\ell) \in \Theta_p\,.$$

Denote by $\mathcal{D}_k$ the index set of elements belonging to subdomain D_k and define

$$K^{(k)}(x) = \sum_{i \in \mathcal{D}_k} x_i \, K_i \, . \tag{14}$$

Matrix $K^{(k)}(x) = K^{(k)}$ can then be partitioned as follows

$$K^{(k)} = \begin{bmatrix} K^{(k)}_{\mathcal{I}\mathcal{I}} & K^{(k)}_{\mathcal{I}\Gamma} \\ K^{(k)}_{\Gamma\mathcal{I}} & K^{(k)}_{\Gamma\Gamma} \end{bmatrix}$$

where the set Γ collects indices of all degrees of freedom corresponding with indices in one of he sets $I_{\ell,k}$ or $I_{k,\ell}$, $\ell = 1, \ldots, p$; the set $\mathcal{I}$ then collects indices of all remaining "interior" degrees of freedom in D_k.

We are now in a position to apply the theorems from Section 2. Notice that Assumptions 1–3 are satisfied, as well as the additional assumptions of Theorem 3. Assumption 4 is discussed below.

Case A – Chordal decomposition Let us first apply Corollary 1. It says that the matrix inequality $Z(x) \succeq 0$ from (13) can be equivalently replaced by the following matrix inequalities

$$Z^{(k)}_A := \begin{bmatrix} K^{(k)}_{\mathcal{I}\mathcal{I}}(x) & K^{(k)}_{\mathcal{I}\Gamma}(x) & 0 \\ K^{(k)}_{\Gamma\mathcal{I}}(x) & K^{(k)}_{\Gamma\Gamma}(x) & f^{(k)} \\ 0 & (f^{(k)})^\top & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & S^{(k)} & \sigma^{(k)} \\ 0 & (\sigma^{(k)})^\top & s^{(k)} \end{bmatrix} \succeq 0 \tag{15}$$

where

$$S^{(k)} = - \sum_{\substack{\ell:\ell<k \\ I_{\ell,k} \neq \emptyset}} S_{\ell,k} + \sum_{\substack{\ell:\ell>k \\ I_{k,\ell} \neq \emptyset}} S_{k,\ell} \tag{16}$$

$$\sigma^{(k)} = - \sum_{\substack{\ell:\ell<k \\ I_{\ell,k} \neq \emptyset}} \sigma_{\ell,k} + \sum_{\substack{\ell:\ell>k \\ I_{k,\ell} \neq \emptyset}} \sigma_{k,\ell} \, . \tag{17}$$

The additional variables are the matrices, vectors and scalars

$$S_{k,\ell} \in \mathbb{S}^{|I_{k,\ell}|}, \quad \sigma_{k,\ell} \in \mathbb{R}^{|I_{k,\ell}|}, \quad s \in \mathbb{R}^p, \quad (k,\ell) \in \Theta_p \, .$$

Case B – Arrow decomposition Now we apply Theorem 3. In this case, the matrix inequality $Z(x) \succeq 0$ from (13) can be replaced by the following matrix inequalities

$$Z^{(k)}_B := \begin{bmatrix} K^{(k)}_{\mathcal{I}\mathcal{I}}(x) & K^{(k)}_{\mathcal{I}\Gamma}(x) & 0 \\ K^{(k)}_{\Gamma\mathcal{I}}(x) & K^{(k)}_{\Gamma\Gamma}(x) & f^{(k)} \\ 0 & (f^{(k)})^\top & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & g^{(k)} \\ 0 & (g^{(k)})^\top & \gamma^{(k)} \end{bmatrix} \succeq 0 \tag{18}$$

where

$$g^{(k)} = - \sum_{\substack{\ell:\ell<k \\ I_{\ell,k} \neq \emptyset}} g_{\ell,k} + \sum_{\substack{\ell:\ell>k \\ I_{k,\ell} \neq \emptyset}} g_{k,\ell} \, . \tag{19}$$

The additional variables $g_{\bullet,\bullet}$ and γ, respectively, have the same dimensions as the variables $\sigma_{\bullet,\bullet}$ and s in Case A.

Recall that Theorem 3 does not use the restrictive Assumption 4 from Section 2. This is important, because Assumption 4 is not satisfied when the domain Ω contains holes, and so the decomposition technique would not be applicable to some practical problems. Consider, for instance, the finite element mesh in Figure 4 and assume that the (i,j)th subdomain is not part of the domain Ω, it is a hole with no finite elements. Then, even if we assume all matrices $K^{(k)}$ to be dense, the sparsity graph of $K(x)$ is not chordal, as it contains the chordless cycle connecting (more than 3) nodes on the boundary of the internal hole.

Before formulating the decomposed version of problem (13) we notice that, according to Corollary 1 and Theorem 3, $Z = \sum_{k=1}^{p} Z_A^{(k)} = \sum_{k=1}^{p} Z_B^{(k)}$, which means, in particular, that

$$\gamma = \sum_{k=1}^{p} s_k = \sum_{k=1}^{p} \gamma_k \, .$$

We will therefore replace the variable γ in the decomposed problems by either s_k or γ_k and the objective function by one of the above sums.

Case A Using the chordal decomposition approach, the decomposed optimization problem in variables

$$\begin{aligned} & x \in \mathbb{R}^m, \quad s \in \mathbb{R}^p, \\ & \sigma = \{\sigma_{k,\ell}\}_{(k,\ell)\in\Theta_p}, \quad \sigma_{k,\ell} \in \mathbb{R}^{|I_{k,\ell}|} \\ & S = \{S_{k,\ell}\}_{(k,\ell)\in\Theta_p}, \quad S_{k,\ell} \in \mathbb{S}^{|I_{k,\ell}|} \end{aligned}$$

is formulated as follows

$$\begin{aligned} & \min_{x,\, s,\, \sigma,\, S} \ \sum_{k=1}^{p} s_k \\ & \text{subject to} \\ & \quad \sum_{i\in\mathcal{D}} x_i \le V \\ & \quad \underline{x} \le x \le \overline{x} \\ & \quad Z_A^{(k)} \succeq 0 \qquad k = 1, \ldots, p \end{aligned} \tag{20}$$

with $Z_A^{(k)}$ defined as in (15),(16),(17).

Case B Using the arrow decomposition approach, the decomposed optimization problem in variables

$$\begin{aligned} & x \in \mathbb{R}^m, \quad \gamma \in \mathbb{R}^p, \\ & g = \{g_{k,\ell}\}_{(k,\ell)\in\Theta_p}, \quad g_{k,\ell} \in \mathbb{R}^{|I_{k,\ell}|} \end{aligned}$$

reads as

$$\min_{x,\,\gamma,\,g} \sum_{k=1}^{p} \gamma_k \tag{21}$$

subject to

$$\sum_{i\in\mathcal{D}} x_i \le V$$

$$\underline{x} \le x \le \overline{x}$$

$$Z_B^{(k)} \succeq 0 \qquad k = 1, \ldots, p$$

with $Z_B^{(i,j)}$ defined as in (18),(19).

A versus B Consider now the finite element mesh and decomposition as in Figure 4 with $n_x \times n_y$ finite elements and $N_x \times N_y$ subdomains. Instead of (13) we can solve one of the decomposed problems (20) and (21). In Case A of the chordal decomposition the single matrix inequality of dimension $(n+1) \times (n+1)$ is replaced by $N_x \cdot N_y$ inequalities of dimension of order $2(n_x/N_x + 1)(n_y/N_y + 1) + 1$ and we have to add $N_x(N_y - 1) + (N_x - 1)N_y$ additional vectors $\sigma_{\bullet,\bullet}$ of a typical size $2(n_x/N_x + 1)$ or $2(n_y/N_y + 1)$, the same number of additional (dense) matrix variables $S_{\bullet,\bullet}$ of the same order and $N_x \cdot N_y$ scalar variables $s_\bullet$. (Recall that the factor 2 stems from the fact that there are two degrees of freedom at every finite element node.) In Case B of the arrow decomposition, the number and order of the new matrix constraints is the same as above but we only need the additional scalar and vector variables; the additional matrix variables are not needed.

Later in Section 6 we will see that these decompositions leads to enormous speed-up in computational time of a state-of-the-art SDO solver. We will also see that the omission of the additional matrix variables in the arrow decomposition can make a big difference.

Example 4 The notation used in the above decomposition approaches is rather cumbersome, so let us illustrate it using a simple example. Figure 5 presents a finite element mesh with 16

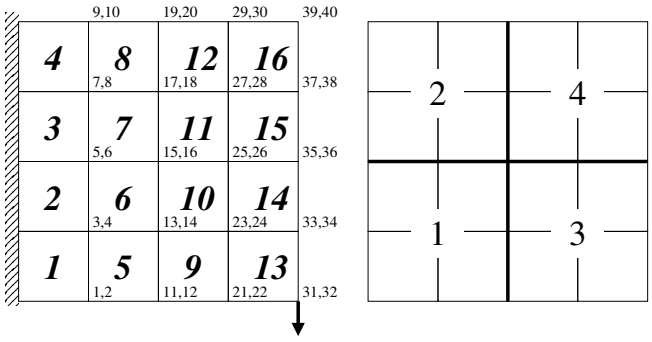


Fig. 5: Example 4: Problem setting, finite element mesh (left) and decomposition into four subdomains (right).

elements and 25 nodes. All nodes on the left-hand side are fixed and thus eliminated from

the stiffness matrix. Hence the corresponding stiffness matrix will have dimension 40×40 (two degrees of freedom associated with every free finite element node, as depicted in the figure). The structure of the corresponding stiffness matrix K is shown in Figure 6; here the elements corresponding to interior degrees of freedom (index sets $\mathcal{I}$) are denoted by circles, while elements associated with the the intersections $I_{k,\ell}$ are marked by full dots.

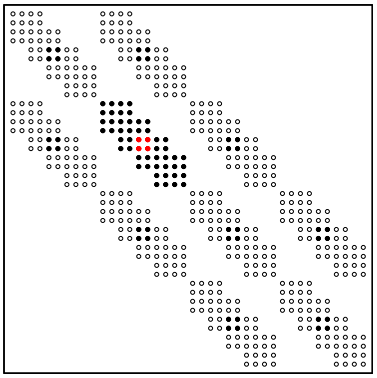


Fig. 6: Sparsity structure of stiffness matrix K in Example 4.

Thus in the original topology optimization problem (13) we have $n = 40$ and $m = 16$ and the matrix constraint $Z(x) \succeq 0$ is of dimension 41×41. We now decompose the problem into four subdomains, containing elements $\{1, 2, 5, 6\}$, $\{3, 4, 7, 8\}$, $\{9, 10, 13, 14\}$, $\{11, 12, 15, 16\}$; see Figure 5–right. Then

$$
\begin{aligned}
&I_1 = \{1, \ldots, 6, 11, \ldots, 16\}, \quad I_2 = \{5, \ldots, 10, 15, \ldots, 20\},\\
&I_3 = \{11, \ldots, 16, 21, \ldots, 26, 31, \ldots, 36\},\ I_4 = \{15, \ldots, 20, 25, \ldots, 30, 35, \ldots, 40\},\\
&I_{1,2} = \{5, 6, 15, 16\}, \quad I_{1,3} = \{11, \ldots, 16\}, \quad I_{1,4} = \{15, 16\},\\
&I_{2,3} = \{15, 16\}, \quad I_{2,4} = \{15, \ldots, 20\}, \quad I_{3,4} = \{15, 16, 25, 26, 35, 36\}.
\end{aligned}
$$

The structure of the stiffness matrices associated with domains 1–4 is shown, left-to-right, in Figure 7. Notice that indices 15,16 (marked by red dots in Figures 6,7) are contained in all six sets $I_{\bullet,\bullet}$.

The chordal decomposition problem (20) will have four matrix constraints, two of order 13 and two of order 19, and additional variables $s \in \mathbb{R}^6$, $\sigma_{1,4}, \sigma_{2,3} \in \mathbb{R}^2$, $\sigma_{1,2} \in \mathbb{R}^4$, $\sigma_{1,3}, \sigma_{2,4}, \sigma_{3,4} \in \mathbb{R}^6$ and $S_{1,4}, S_{2,3} \in \mathbb{S}^2$, $S_{1,2} \in \mathbb{S}^4$, $S_{1,3}, S_{2,4}, S_{3,4} \in \mathbb{S}^6$. The arrow decomposition problem (21) will have the same number of matrix constraints as (20) and additional variables $\gamma \in \mathbb{R}^6$, $g_{1,4}, g_{2,3} \in \mathbb{R}^2$, $g_{1,2} \in \mathbb{R}^4$, $g_{1,3}, g_{2,4}, g_{3,4} \in \mathbb{R}^6$. $\Diamond$

5 Decomposition by fictitious loads

So far, all the reasoning was purely algebraic. There is, however, an alternative, functional analytic view of the arrow decomposition in Theorem 3. We will present it in this section. The purpose is to illustrate a different viewpoint and so, to keep the notation simple, we will only consider the case of two subdomains.

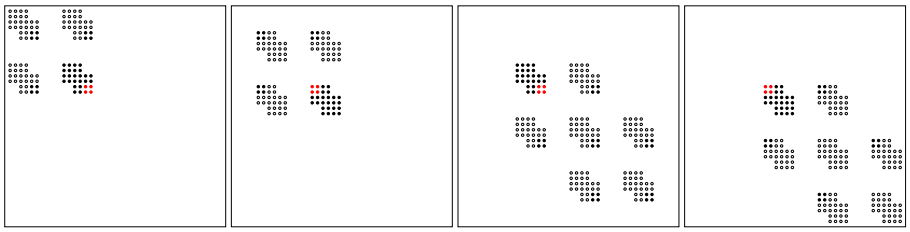


Fig. 7: Sparsity structure of stiffness matrices $K_1, \ldots, K_4$ associated with subdomains 1–4 in Example 4.

5.1 Infinite dimensional setting

Let us recall the weak formulation (7) of the elasticity problem depending on parameter x:

$$\boldsymbol{a}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_f} \boldsymbol{f}\boldsymbol{v} \, \mathrm{d}s \quad \forall \boldsymbol{v} \in \mathcal{V} \,. \tag{22}$$

Let Ω be partitioned into two mutually disjoint subdomains Ω_1 and Ω_2 such that $\Omega_1 \cup \Omega_2 = \Omega$. Denote the interface boundary between the two subdomains by Γ_I; see Figure 8. We consider the general situation when Γ_u and Γ_f may be a part of both, $\partial\Omega_1 \cap \partial\Omega$ and $\partial\Omega_2 \cap \partial\Omega$. Define $\boldsymbol{a}_i$ as a restriction of the bilinear form $\boldsymbol{a}$ to Ω_i (the integral in (8) is simply computed over Ω_i), $\boldsymbol{f}_i = \boldsymbol{f}|_{\partial\Omega_i}$,

$$\mathcal{V}(\Omega_i) = \{\boldsymbol{v} \in [H^1(\Omega_i)]^2 \mid \boldsymbol{v} = 0 \text{ on } \Gamma_u \cap \partial\Omega_i\}, \;\; i = 1, 2 \,,$$

and

$$\mathcal{V}_i = \{\boldsymbol{v} \in [H^1(\Omega_i)]^2 \mid \boldsymbol{v} = 0 \text{ on } (\Gamma_u \cap \partial\Omega_i) \cup \Gamma_I\}, \;\; i = 1, 2 \,.$$

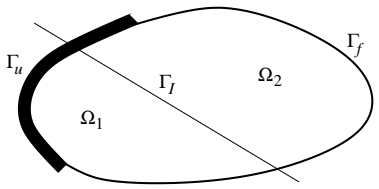


Fig. 8: Partitioning of domain Ω into two subdomains with interface boundary Γ_I.

Consider the following "restricted" problems:

$$\text{Find } \boldsymbol{u} \in [H^1(\Omega_1)]^2 \text{ such that } \quad \boldsymbol{u} - \boldsymbol{u}^* \in \mathcal{V}_1 \text{ and} \tag{23}$$
$$\boldsymbol{a}_1(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_f \cap \partial\Omega_1} \boldsymbol{f}_1 \boldsymbol{v} \, \mathrm{d}s \quad \forall \boldsymbol{v} \in \mathcal{V}_1 \,;$$

$$\text{Find } \boldsymbol{u} \in [H^1(\Omega_2)]^2 \text{ such that } \quad \boldsymbol{u} - \boldsymbol{u}^* \in \mathcal{V}_2 \text{ and} \tag{24}$$
$$\boldsymbol{a}_2(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_f \cap \partial\Omega_2} \boldsymbol{f}_2 \boldsymbol{v} \, \mathrm{d}s \quad \forall \boldsymbol{v} \in \mathcal{V}_2 \,.$$

The following theorem forms a basis of our approach.

Theorem 4 *Assume that $\boldsymbol{u}^*$ solves (22). For all $\boldsymbol{x} \in L_\infty(\Omega)$ there exists $\boldsymbol{g} \in [H^{-1/2}(\Gamma_I)]^2$ such that solutions to (23) and (24) are equal to respective solutions of the following problems*

$$\textit{find } \boldsymbol{u} \in \mathcal{V}(\Omega_1) \textit{ s.t. } \boldsymbol{a}_1(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_f \cap \partial\Omega_1} \boldsymbol{f}_1 \boldsymbol{v} \, \mathrm{d}s + \langle \boldsymbol{g}, \boldsymbol{v} \rangle_{\Gamma_I} \quad \forall \boldsymbol{v} \in \mathcal{V}(\Omega_1)\,, \quad (25)$$

$$\textit{find } \boldsymbol{u} \in \mathcal{V}(\Omega_2) \textit{ s.t. } \boldsymbol{a}_2(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \int_{\Gamma_f \cap \partial\Omega_2} \boldsymbol{f}_2 \boldsymbol{v} \, \mathrm{d}s - \langle \boldsymbol{g}, \boldsymbol{v} \rangle_{\Gamma_I} \quad \forall \boldsymbol{v} \in \mathcal{V}(\Omega_2)\,, \quad (26)$$

where $\langle \cdot, \cdot \rangle_{\Gamma_I}$ denotes the duality pairing between $[H^{-1/2}(\Gamma_I)]^2$ and $[H^{1/2}(\Gamma_I)]^2$.

Proof The requested function $\boldsymbol{g}$ is the outcome of the respective Steklov-Poincaré operator applied to $\boldsymbol{u}^*$; see, e.g., [12]. □

In the above theorem, function g can be interpreted as a fictitious load applied to either of the problems (25),(26). The theorem says that there exists such a g that the solutions of (25),(26) are equivalent to the solution of the "full" problem (22) restricted to the respective subdomain. Or, in other words, the solutions of (25),(26) can be "glued" to form the solution of (22).

5.2 Finite dimensional setting

Now assume that the discretization of Ω is such that the interface boundary Γ_I is a union of boundaries of some finite elements. More precisely, we assume that the index set of finite elements used to the discretization of Ω can be split into two disjoint subsets

$$\{1, 2, \ldots, m\} = \mathcal{D}_1 \cup \mathcal{D}_2, \quad \mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset,$$

such that Ω_i is discretized by elements with indices from $\mathcal{D}_i$, $i = 1, 2$. Define

$$f^{(1)} = \sum_{i \in \mathcal{D}_1} f_i, \qquad f^{(2)} = \sum_{i \in \mathcal{D}_2} f_i\,,$$

the restrictions of the load vector f on boundaries of Ω_1 and Ω_2, respectively.

Denote the index set of degrees of freedom associated with finite element nodes on Γ_I by $I_{1,2}$. Let n_Γ be the dimension of $I_{1,2}$.

Finally, for a vector in $z \in \mathbb{R}^{n_\Gamma}$ denote by $\overleftrightarrow{z}$ its extension to $\mathbb{R}^n$:

$$\overleftrightarrow{z}_i := \begin{cases} z_i \text{ if } i \in I_{1,2} \\ 0 \text{ if } i \in \mathbb{R}^n \setminus I_{1,2} \end{cases}.$$

The discrete version of Theorem 4 can then be formulated as follows. (The following corollary is, in fact, trivial in the finite dimension; however, we need the above theorem to understand the meaning of the fictitious load and its existence in the original setting of the problem.)

Corollary 2 *Assume that u^* solves (10). Then for all $x \in \mathbb{R}^m$ there exists $g \in \mathbb{R}^{n_\Gamma}$ such that*

$$(\sum_{i\in\mathcal{D}_1} x_i K_i)u^* = f^{(1)} + \overleftrightarrow{g} \tag{27}$$

$$(\sum_{i\in\mathcal{D}_2} x_i K_i)u^* = f^{(2)} - \overleftrightarrow{g}\,. \tag{28}$$

Notice that (27), (28) are still systems of dimension n; however, many rows and columns in the matrix and the right hand side are equal to zero, so they can be solved as systems of dimensions $|\mathcal{N}^{(1)}|$ and $|\mathcal{N}^{(2)}|$, respectively. Hence, if we knew the fictitious load g, we could replace the large system of equations (10) by two smaller ones which, numerically, would be more efficient. Of course, we do not know it. However, and this is the key idea of this section, the linear system (10) is a constraint in an optimization problem, hence we can add g among the variables and, instead of searching for the optimal design x and the corresponding u satisfying (10), search for optimal x and for a pair (u, g) satisfying two smaller equilibrium equations (27) and (28).

We can now formulate a result regarding the decomposition of the discretized topology optimization problem (11).

Theorem 5 *Problem (11) is equivalent to the following problem:*

$$\min_{x\in\mathbb{R}^m,\, u\in\mathbb{R}^n, \gamma_1\in\mathbb{R},\, \gamma_2\in\mathbb{R},\, g\in\mathbb{R}^{n_\Gamma}} \quad \gamma_1+\gamma_2 \tag{29}$$

subject to

$$\sum_{i=1}^m x_i \le V$$

$$\underline{x} \le x \le \overline{x}$$

$$(\sum_{i\in\mathcal{D}_1} x_i K_i)u = f^{(1)} + \overleftrightarrow{g}$$

$$(\sum_{i\in\mathcal{D}_2} x_i K_i)u = f^{(2)} - \overleftrightarrow{g}$$

$$(f^{(1)} + \overleftrightarrow{g})^\top u \le \gamma_1$$

$$(f^{(2)} - \overleftrightarrow{g})^\top u \le \gamma_2\,.$$

In particular, if $(\tilde{x}, \tilde{u}, \tilde{\gamma})$ is a solution of (11) then there is $\tilde{\gamma}_1 \in \mathbb{R}_+, \tilde{\gamma}_2 \in \mathbb{R}_+, \tilde{g} \in \mathbb{R}^{n_\Gamma}$ such that $\tilde{\gamma} = \tilde{\gamma}_1 + \tilde{\gamma}_2$ and $(\tilde{x}, \tilde{u}, \tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{g})$ is a solution of (29). Vice versa, if $(\hat{x}, \hat{u}, \hat{\gamma}_1, \hat{\gamma}_2, \hat{g})$ is a solution of (29) then $(\hat{x}, \hat{u}, \hat{\gamma}_1 + \hat{\gamma}_2)$ is a solution of (11).

Proof The theorem follows from the comparison of the KKT conditions of both problems. Assuming that $(\tilde{x}, \tilde{u}, \tilde{\gamma})$ solves (11), we define $\tilde{g} = (\sum_{i\in\mathcal{D}_1} \tilde{x}_i K_i)\tilde{u} - f^{(1)}$ and $\tilde{\gamma}_1 = (f^{(1)} + \tilde{g})^\top u$, $\tilde{\gamma}_2 = (f^{(2)} - \tilde{g})^\top u$. Then it is straightforward to check that $(\tilde{x}, \tilde{u}, \tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{g})$ satisfies the KKT conditions of (29). Now assume that $(\hat{x}, \hat{u}, \hat{\gamma}_1, \hat{\gamma}_2, \hat{g})$ is a solution of (29). Then $(\hat{x}, \hat{u}, \hat{\gamma}_1 + \hat{\gamma}_2)$ is feasible in (11). We know from above that $(\tilde{x}, \tilde{u}, \tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{g})$ is a solution of (29) with the optimal objective value $\tilde{\gamma}_1 + \tilde{\gamma}_2$. Because both problems are equivalent to convex problems (their semidefinite reformulations), then $\tilde{\gamma} = \tilde{\gamma}_1 + \tilde{\gamma}_2 = \hat{\gamma}_1 + \hat{\gamma}_2$ is also the optimal objective value of (11), hence $(\hat{x}, \hat{u}, \hat{\gamma}_1, \hat{\gamma}_2, \hat{g})$ is also optimal for (11). □

Using again the Shur complement theorem, we finally arrive at the decomposition of the SDO problem (13).

Corollary 3 *Problem (13) can be equivalently formulated as follows:*

$$\min_{x\in\mathbb{R}^m,\,\gamma_1\in\mathbb{R},\,\gamma_2\in\mathbb{R},\,g\in\mathbb{R}^{n_\Gamma}} \gamma_1+\gamma_2 \tag{30}$$

subject to

$$\sum_{i=1}^{m} x_i \le V$$

$$\underline{x} \le x \le \overline{x}$$

$$\begin{pmatrix} \gamma_1 & (f^{(1)}+\overleftrightarrow{g})^\top \\ f^{(1)}+\overleftrightarrow{g} & \sum\limits_{i\in\mathcal{D}_1} x_i K_i \end{pmatrix} \succeq 0$$

$$\begin{pmatrix} \gamma_2 & (f^{(2)}-\overleftrightarrow{g})^\top \\ f^{(2)}-\overleftrightarrow{g} & \sum\limits_{i\in\mathcal{D}_2} x_i K_i \end{pmatrix} \succeq 0\,.$$

Problem (30) is now exactly the same as problem (21) arising from arrow decomposition applied to two subdomains.

6 Numerical experiments

The decomposition techniques described in the article were applied to an example whose data (geometry, boundary conditions and forces) are shown in Figure 9–left. We always use regular decomposition of the rectangular domain; an example of a decomposition into 8 subdomains is shown in Figure 9–right. We have used finite element meshes with up to 160×80 elements.

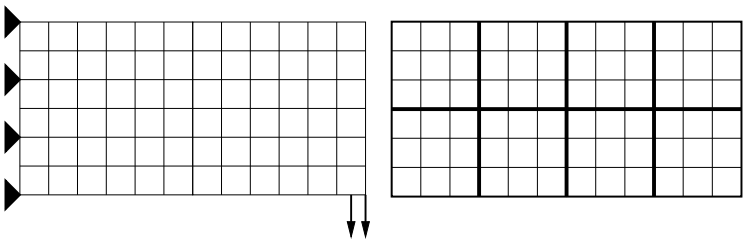


Fig. 9: Data of numerical examples: geometry, boundary condition and forces (left) and a sample decomposition into 4×2 subdomains (right)

We tested several codes to solve the SDO problems. Here we present results obtained by MOSEK, version 8.0 [9]. The reason for this is that MOSEK best demonstrated the decomposition idea; the speed-up achieved by the decomposition was most significant when using this software.

When solving the SDO problems, we used default MOSEK settings with the exception of duality gap parameter `MSK_DPAR_INTPNT_CO_TOL_REL_GAP` that was set to 10^{-9}, instead of the default value 10^{-8}. We will comment on the resulting accuracy of the solution later in the section.

We also tried to solve smaller problems by SparseCoLO [2], software that performs the decomposition of matrix constraints based on Theorem 1 automatically. In particular, the software checks whether the matrix in question has a chordal sparsity graph; if not, the graph is completed to be chordal. After that, maximal cliques are found and Theorem 1 is applied. Because the sparsity graph of the matrix in problem (13) is *not* chordal, a chordal completion is performed by SparseCoLO. Such a completion is not unique and may thus lead to different sets of maximal cliques. And here is the main difference to our approach: while we can steer the decomposition to result in smaller matrix constraints of the same size, matrix constraints resulting from application of SparseCoLO are of variable size, some small, some rather large. This fact has a big effect on the efficiency of SparseCoLO, as we will see in the examples below.

In all experiments we used a 2018 MacBook Pro with 2.3GHz dual-core Intel Core i5, Turbo Boost up to 3.6GHz and 16GB RAM, and MATLAB version 9.2.0 (2017a).

Remark 1 (Element-wise decomposition) The above text suggests that we always perform decomposition of the original finite element mesh into several (possibly uniform) sub-meshes, each of them having interior points; cf. Figures 4, 5, 9, and the notation used in Section 4. However, nothing prevents us from associating each subdomain with a finite element. When every subdomain consist of a single finite element, then the subdomains have no interior points, apart from those lying on the boundary of Ω and having no neighboring element. For instance, in Example 1, Figure 5, these would only be degrees of freedom number 31,32,39,40. In the numerical examples below, we will see that the big number of additional variables makes this option less attractive that other decompositions. However, while not the most effective of all decompositions, it is still much less computationally demanding than the original problem. The element-wise decomposition has one big advantage in simplicity of data preparation: the user can use any standard finite-element mesh generator and does not have to worry about definition of subdomains. This may be particularly advantageous in case of highly irregular meshes.

6.1 Computational results

In the following tables, we present results of the $N_x \times N_y$ examples using the chordal and arrow decomposition. In these tables the first row of numbers shows data for the original problem (13), the remaining rows are for the decomposed problems. The first column shows the number of subdomains, the next two ones the number of variables and the size of the largest matrix inequality. After that, we present the total number of iterations needed by MOSEK before it terminated. The next two columns show the total CPU time and CPU time per one iteration and are followed by columns reporting speed-up relative to the original problem formulation, both total and per iteration.

In the final column we see the MOSEK constant `MSK_DINF_INTPNT_OPT_STATUS`, a number that is supposed to converge to 1. Let us call this constant μ, for brevity. In our experience, MOSEK delivers acceptable solution reporting "Solution status: OPTIMAL" when

$$0.999 \leq \mu \leq 1.0009\,.$$

When μ is farther away from 1, MOSEK, typically in these examples, announces "Solution status: NEAR_OPTIMAL." For instance, in the 120×60 example with chordal decomposition with 800 subdomains, MOSEK finished with $\mu = 0.9946$ and the final objective value was correct to 3 digits, while with 1800 subdomains MOSEK reported $\mu = 0.9865$ and we only got 2 correct digits in the objective function.

We first present results for the 40×20 example using the chordal decomposition; see Table 1. The table shows that while we increase the number of the subdomains (refine the

Table 1: Results obtained by MOSEK for the 40×20 example using chordal decomposition.

no of doms	no of vars	size of matrix	no of iters	CPU (sec) total	CPU (sec) per iter	opt status
1	801	1681	69	1045	15	0.9999
8	3523	243	58	31	0.53	0.9996
32	5489	73	44	9.7	0.22	0.9997
50	6376	51	46	8.8	0.19	0.9995
200	11243	19	37	6.9	0.19	0.9987
800	24529	9	35	12	0.34	0.9980

decomposition), the number of variables increases (those are the additional matrix variables in chordal decomposition) and the size of the constraints decreases. We can further see from Table 1 that the total number of iterations needed to solve any of the problem formulations is almost constant. The main message of Table 1 is in columns 5 and 6; here we can see tremendous decrease in the CPU time when solving the decomposed problems.

We now solve the same 40x20 example using the arrow decomposition. The results are presented in Table 2. We have added two more columns showing the speed-up relative to the undecomposed problem, both total and per iteration. In all examples presented in

Table 2: Results obtained by MOSEK for the 40×20 example using arrow decomposition.

no of doms	no of vars	size of matrix	no of iters	CPU total	CPU per iter	speed-up total	speed-up per iter	opt status
1	801	1681	69	1045	15	1	1	0.9999
8	1032	243	70	28	0.40	37	38	0.9999
32	1492	73	63	7.6	0.12	138	126	1.0003
50	1764	51	64	7.1	0.11	147	137	0.9999
200	3544	19	51	5.1	0.10	204	151	0.9999
800	9204	9	46	6.9	0.15	150	100	0.9992

Table 2, MOSEK reported Optimal solution status. Comparing result in Table 1 and Table 2, we can see that the arrow decomposition is not only more efficient than the chordal one, due to smaller number of variables, but also delivers more accurate solution, i.e., a better conditioned SDO problem.

For a comparison, In Table 3 we present result for example 40×20 obtained by solving problems decomposed by the automatic decomposition software SparseCoLO. In this case, the size of the 34 matrix constraints varied from 11 to 260. The decomposed problem is

Table 3: Results obtained by MOSEK for the 40×20 example using SparseCoLO decomposition.

no of doms	no of vars	size of matrix	no of iters	CPU total	CPU per iter	speed-up total	speed-up per iter
34	22997	11...260	42	301	7	3	2

still solved more efficiently that the original one but that speed-up is negligible, compared to either the chordal or the arrow decomposition from Tables 1 and 2.

The next Table 4 presents results for the 80×40 discretization and chordal decomposition, while Table 5 present the results for the same problem using arrow decomposition. This was the largest problem we could solve by MOSEK in the original formulation (13), due to memory limitations. As we can see, for a larger problem the speed-up obtained by

Table 4: Results obtained by MOSEK for the 80×40 example using chordal decomposition.

no of doms	no of vars	size of matrix	no of iters	CPU (sec) total	CPU (sec) per iter	opt status
1	3201	6561	104	78813	758	0.9999
8	12583	883	74	1302	18	0.9992
32	17449	243	56	173	3.1	0.9993
128	24265	73	51	62	1.2	0.9990
200	27631	51	46	53	1.2	0.9993
800	46873	19	40	41	1.0	0.9986
3200	100249	9	32	52	1.6	0.9975

Table 5: Results obtained by MOSEK for the 80×40 example using arrow decomposition.

no of doms	no of vars	size of matrix	no of iters	CPU (sec) total	CPU (sec) per iter	speed-up total	speed-up per iter	opt status
1	3201	6561	104	78813	758	1	1	0.9999
8	3632	883	88	1098	12.5	72	61	0.9999
32	4412	243	83	121	1.5	651	520	0.9999
128	6308	73	69	25	0.4	3153	2092	0.9999
200	7424	51	65	18	0.3	4379	2737	0.9999
800	14864	19	62	17	0.3	4636	2764	0.9999
3200	37604	9	44	25	0.6	3153	1334	0.9999

arrow decomposition is even more significant.

Examples with finer discretization cannot be solved by MOSEK in the original formulation (13)(on the laptop we used for the experiments). They can, however, easily be solved in the decomposed setting. The results are presented in the next tables. In these tables, we also show estimated number of iterations and CPU time for the original problem; these numbers

are extrapolated from the lower-dimensional problems (also those that are not presented here).

Table 6 presents results for the 120×60 discretization and chordal decomposition, while Table 7 shows the results for the same example, this time using arrow decomposition. When

Table 6: Results obtained by MOSEK for the 120×60 example using chordal decomposition. Iteration count and CPU time in the first row are estimated and marked by the † symbol.

no of doms	no of vars	size of matrix	no of iters	CPU (sec) total	CPU (sec) per iter	opt status
1	7200	14641	139†	1045932†	7524	0.9999
200	51539	19	60	236	3.9	0.9950
800	76977	19	50	129	2.6	0.9946
1800	106903	19	47	114	2.4	0.9865

Table 7: Results obtained by MOSEK for the 120×60 example using arrow decomposition. Iteration count and CPU time in the first row are estimated and marked by the † symbol.

no of doms	no of vars	size of matrix	no of iters	CPU (sec) total	CPU (sec) per iter	speed-up total	speed-up per iter	opt status
1	7200	14641	†139	†1045932	7525	1	1	0.9999
50	9524	339	96	524	5.5	1996	1379	0.9996
200	12904	99	82	89	1.1	11752	6933	0.9997
450	16984	51	82	55	0.67	19017	11219	0.9997
800	21764	33	71	37	0.52	28268	14439	0.9997
1800	33424	19	65	42	0.65	24903	11645	0.9998
7200	85204	9	55	90	1.6	11621	4598	0.9997

using the chordal decomposition (Table 6), MOSEK had difficulties with convergence to the optimal solution. In case of 800 subdomains, the final objective value was correct to 3 digits, while for the 1800 subdomains only to 2 digits. In both cases, the solution status of MOSEK was “Nearly optimal”. In case of arrow decomposition, all problems finished with ”Optimal” solution status. Again, the arrow decomposition outperforms the chordal one, so from now on we will only focus on the arrow decomposition.

From the results presented so far, it seems that the most efficient decomposition is either the finest or the second-finest one (not counting the element-wise decomposition); in the first case, each subdomain contains four finite elements, in the second case 16 finite elements. To get a clearer idea about the relation of the problem size and speed-up, we present the next Table 8 of results for examples with dimension increasing from 40×20 to 160×80 elements. For each example we only consider the finest decomposition with four finite elements per subdomain. So the size of every matrix inequality is always at most 19. The CPU times for original formulation of the larger problems have been extrapolated and are denoted by the † symbol.

The last row of Table 8 presents the estimate of computational complexity of each approach, as a function $c\nu^q$ of problem size ν; in this case, ν is the number of variables of the

Table 8: Results obtained by MOSEK using arrow decomposition. Symbol † denotes extrapolated CPU times.

problem	ORIGINAL no of vars	size of matrix	CPU total	DECOMPOSED no of vars	size of matrix	CPU total	opt status	speed-up
40×20	801	1681	1045	3544	19	5	0.9999	204
60×30	1801	3721	12468	8164	19	9	0.9999	1370
80×40	3201	6561	78813	14684	19	17	0.9999	4636
100×50	5001	10201	†312560	23104	19	25	0.9999	12502
120×60	7201	14641	†1045932	33424	19	42	0.9998	24903
140×70	9801	19881	†2900382	45664	19	59	0.9994	49159
160×80	12801	25921	†7003213	59764	19	74	0.9984	94638
complexity $c{\cdot}\text{size}^q$			$q = 3.18$			$q = 1.0006$		

SDO problem, as reported in the table. The exponent q is estimated from the CPU times. In case of the original, undecomposed problem, we calculated $q \approx 3.18$ which slightly underestimates the theoretical complexity of interior point methods for SDO. The decomposed problem, on the other hand, exhibits linear complexity with $q \approx 1.0006$. See also Figure 10 for graphical representation of the complexity of the original problem (top line), single iteration of the original problem (middle line) and of the decomposed problem (bottom line). *This, in our opinion, is the principal contribution of the arrow decomposition method.*

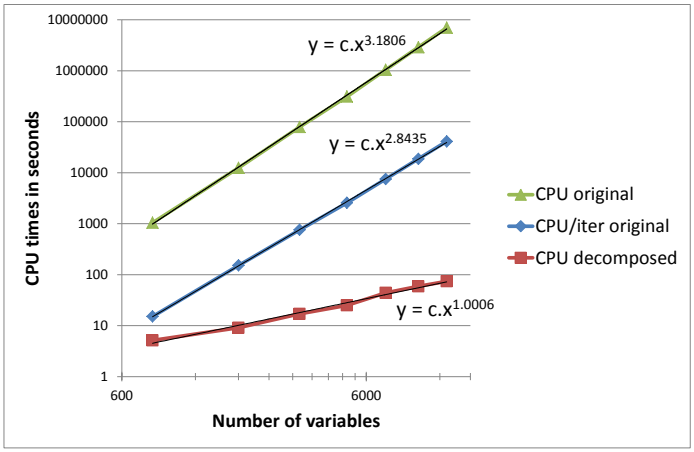


Fig. 10: Complexity of the original problem (top graph), of a single iteration in the original problem (middle) and of the decomposed problem (bottom).

Acknowledgment

The author would like to thank Masakazu Kojima for initiating the discussions on chordal decomposition of the topology optimization problem. The work on this article was initiated while the author was visiting the Institute for Pure and Applied Mathematics, UCLA. The support and friendly atmosphere of the Institute are acknowledged with gratitude. Thanks go also to Allan Lo, University of Birmingham, and two anonymous referees; their comments lead to significant improvements of the paper. Last but not least, the author wishes to thank MOSEK ApS for providing him with the academic version of their software.

References

1. Agler, J., Helton, W., McCullough, S., Rodman, L.: Positive semidefinite matrices with a given sparsity pattern. Linear Algebra and its Applications **107**, 101–149 (1988)
2. Fujisawa, K., Kim, S., Kojima, M., Okamoto, Y., Yamashita, M.: User's manual for SparseCoLO: Conversion methods for SPARSE COnic-form Linear Optimization. Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Tech. Rep (2009)
3. Fukuda, M., Kojima, M., Murota, K., Nakata, K.: Exploiting sparsity in semidefinite programming via matrix completion i: General framework. SIAM Journal on Optimization **11**(3), 647–674 (2001)
4. Griewank, A., Toint, P.L.: On the existence of convex decompositions of partially separable functions. Mathematical Programming **28**(1), 25–49 (1984)
5. Grone, R., Johnson, C., Sà, E., Wolkowitz, H.: Positive definite completions of partial Hermitian matrices. Linear Algebra and its Applications **58**, 109–124 (1984)
6. Haslinger, J., Kočvara, M., Leugering, G., Stingl, M.: Multidisciplinary free material optimization. SIAM Journal on Applied Mathematics **70**(7), 2709–2728 (2010)
7. Kakimura, N.: A direct proof for the matrix decomposition of chordal-structured positive semidefinite matrices. Linear Algebra and its Applications **433**(4), 819–823 (2010)
8. Kim, S., Kojima, M., Mevissen, M., Yamashita, M.: Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. Mathematical Programming **129**(1), 33–68 (2011)
9. MOSEK ApS: The MOSEK optimization toolbox for MATLAB manual. Version 8.0 (2016)
10. Nakata, K., Fujisawa, K., Fukuda, M., Kojima, M., Murota, K.: Exploiting sparsity in semidefinite programming via matrix completion ii: Implementation and numerical results. Mathematical Programming **95**(2), 303–327 (2003)
11. Petersson, J.: A finite element analysis of optimal variable thickness sheets. SIAM Journal on Numerical Analysis **36**(6), 1759–1778 (1999)
12. Quarteroni, A., Valli, A.: Theory and application of Steklov-Poincaré operators for boundary-value problems. In: R. Spigler (ed.) Applied and Industrial Mathematics: Venice–1, 1989, pp. 179–203. Springer Netherlands, Dordrecht (1991)
13. Vandenberghe, L., Andersen, M.S.: Chordal graphs and semidefinite optimization. Foundations and Trends in Optimization **1**(4), 241–433 (2015)