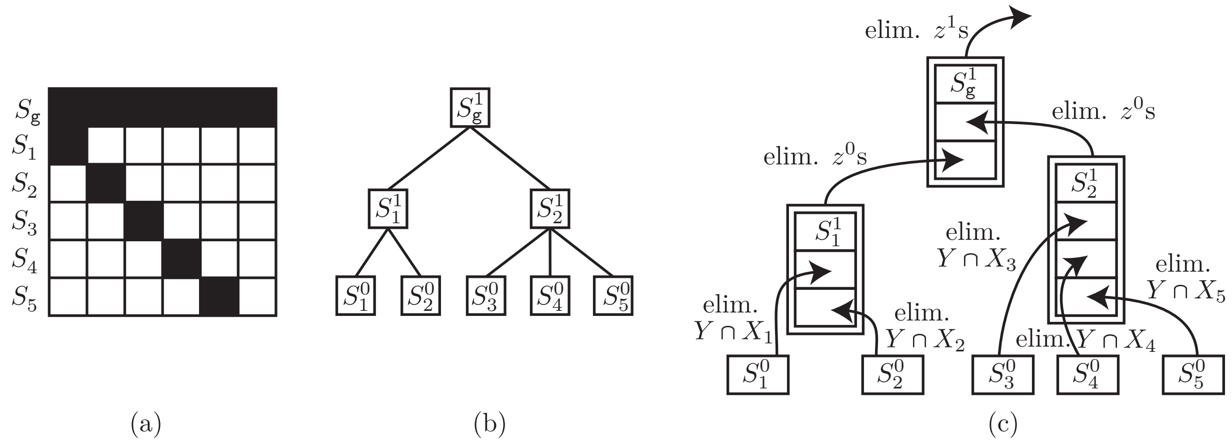

Figure 1: An illustration of (a) the coefficient matrix for a block-structured inequality system S (non-zero sections coloured black). (b) S decomposed into a tree stucture of subsystems. (c) Recursive projection of the tree-structure in (b).

1 Decomposing system

In the following we will describe how the structure of the problem can be exploited to make our projection framework perform better on large systems. Detailed pseudocode and proofs can be found in [**?**].

The considered problem has a *(primal) block angular structure* [**?**]. That is, the inequality system S can be divided into k local subsystems, $S_1, \ldots, S_k$, using smaller, disjoint sets of variables, $X_1, \ldots, X_k$, and a global system, S_{g}, whose constraints "connect" the otherwise independent subsystem by using variables from several local subsystems (see Figure 1(a)). If there were no global constraints, each subsystem could be projected separately and the resulting systems could then be combined to give the projection of the original system. Unfortunately, the global constraints cause the subsystems to get "mixed" when variables are eliminated and hence result in an increasing number of global and dense constraints, which again makes FME perform worse.

However, we can still exploit the structure of the problem. We will define and use auxiliary variables to ensure that we can project the subsystems separately without producing global constraints, before we combine the projected subsystems and eliminate the auxiliary variables.

Using auxiliary variables to separate blocks To remove local variables from the global inequalities, we do as follows for all subsystems S_i:

- For each global inequality c using variables in S_i, we define an auxiliary variable $z^0_{c,i}$ that equals the variables in S_i's contribution to c. We add the equality defining $z^0_{c,i}$ to S_i, and we substitute with $z^0_{c,i}$ in c. For example, if $c : a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_kx_k \le b$ is an inequality in S_{g} and $X_i = \{x_1, x_2\}$, then we add $-z^0_{c,i} + a_1x_1 + a_2x_2 = 0$ to S_i and rephrase c as $z^0_{c,i} + a_2x_3 + \ldots + a_kx_k \le b$.

  We name the thus expanded subsystem S^0_i.
- Then we project S^0_i w.r.t. all variables from Y that are present in S^0_i. We do keep the auxiliary z^0-variables. Because of these auxiliary variables this only produces inequalities with variables not present in other subsystems S_j.

After projecting each S^0_i we can then combine the results with the rephrased, global inequalities, S^0_{g}, to create the system $\mathcal{S}$. We can then eliminate all the auxiliary z^0-variables, Z^0, plus the variables in Y not occurring in any S_i from $\mathcal{S}$. This results in a system equivalent to the projection of S w.r.t. Y.

Example As an example, consider the block-structured problem S illustrated in Figure 1(a), where $X_i = \{u_i, w_i\}$ for $i \in \{1, 2, 3, 4, 5\}$, and $S_{\mathrm{t}} = \{e, c\}$ consists of an equality defining the weighted sum, $e : -wSum + 1u_1 + 1w_1 + 2u_2 + 2w_2 + 3u_3 + 3w_3 + 4u_4 + 4w_4 + 5u_5 + 5w_5 = 0$, and another inequality limiting the ws, $c : w_1 + w_2 + w_3 + w_4 + w_5 \le 20$. Assume we want to eliminate all variables but $wSum$, that is $Y = \bigcup_{i \in \{1,2,3,4,5\}} \{u_i, w_i\}$.

We then create the subsystems $S_1^0, \dots, S_5^0$ and S_g^0, where $S_i^0 = S_i \cup \{-z_{e,i}^0 + iu_i + iw_i = 0\} \cup \{-z_{c,i}^0 + w_i = 0\}$, while the rephrased global inequalities are $S_g^0 = \{-wSum + z_{e,1}^0 + z_{e,2}^0 + z_{e,3}^0 + z_{e,4}^0 + z_{e,5}^0 = 0\} \cup \{z_{c,1}^0 + z_{c,2}^0 + z_{c,3}^0 + z_{c,4}^0 + z_{c,5}^0 \le 20\}$.

To procure the projection of S w.r.t. Y we can instead project S_i^0 w.r.t. $Y \cap X_i$ for all $1 \le i \le 5$, join the projections together with S_g^0 in the system $\mathcal{S}$, and project $\mathcal{S}$ w.r.t. $\{ z_{c,i}^0, z_{e,i}^0 \mid 1 \le i \le 5 \}$.

Equivalent projections Comparing the union of the new (unprojected) subsystems, $\mathfrak{S} := S_1^0 \cup \dots \cup S_k^0 \cup S_g^0$, with the original system S it is clear that all we have done is defining auxiliary variables and substituted them in the system. Intuitively, it is therefore clear, that eliminating the variables in Y from S is equivalent to eliminating Y and the defined z^0-variables from $\mathfrak{S}$.

This, in turn, is equivalent to projecting the subsystems S_i^0 separately w.r.t. $X_i \cap Y$, combining the resulting systems with S_g^0, and then eliminate the z^0-variables and the remaining Y-variables: When eliminating $Y \cup Z^0$ from $\mathfrak{S}$, we can choose to first eliminate $X_1 \cap Y$, then $X_2 \cap Y$ up to $X_k \cap Y$, and finally $Z^0 \cup Y \setminus (X_1 \cup \dots \cup X_k)$. Any variable in $X_1 \cap Y$ has a zero-coefficient in all (in)equalities not in S_1^0, and the inequalities in $\mathfrak{S} \setminus S_1^0$ will therefore not be changed by the FME procedure when $X_1 \cap Y$ is eliminated. We can therefore set aside these inequalities until FME is done eliminating $X_1 \cap Y$. Likewise, when eliminating variables in $X_i \cap Y$, no (in)equality from either $S_{i+1}^0 \cup \dots \cup S_k^0 \cup S_g^0$ or the already projected systems contains any variables from $X_i \cap Y$ and can hence be "put aside" and only included later when $Z^0 \cup Y \setminus (X_1 \cup \dots \cup X_k)$ is eliminated.

Thus, the following holds.

Proposition 1. *The projection of S w.r.t. Y defines the same feasible area as the projection of $S_1^0 \cup \dots \cup S_k^0 \cup S_g^0$ w.r.t. $Y \cup Z^0$.*

Further decomposition $\mathcal{S}$ is by construction block structured, and instead of eliminating $Z^0 \cup Y \setminus (X_1 \cup \dots \cup X_k)$ immediately, if necessary we can use further auxiliary variables to postpone "mixing" blocks when eliminating the remaining variables. To do so, we collect all subsystems into k_1 small groups; we have mostly used groups of size 2, 3 or 1[1]. For each group i we do as follows:

- We join the systems in the group into a new system, S_i^1. For each (rephrased) global inequality c using variables occurring in S_i^1, we then define a variable, $z_{c,i}^1$, that equals S_i^1's contribution to c. We then add the defining equality to S_i^1 and rephrase c using $z_{c,i}^1$.
- Then we project the resultant system, S_i^1, w.r.t. the previous, auxiliary z^0-variables, while we do keep the newly created z^1-variables.

Subsequently we can then join the projected S^1-systems with the (rephrased) transverse inequalities, S_g^1, and finally project the last auxiliary variables, or we can repeat the step above step until the final projection can be done.

Tree structure Each time we further decompose a system, we use a partitioning of the subsystems $S_1^l, \dots, S_{k_l}^l$ to create a new "level" of k_{l+1} subsystems, $S_1^{l+1}, \dots, S_{k_{l+1}}^{l+1}$. This effectively creates a tree structure of smaller inequality systems, where each node is associated with a subsystem and a set of variables that should be eliminated from the system. Using this tree, an inequality system associated with a node is projected by recursively projecting the systems associated with the children of the node, and as expected, projecting the root of the tree constructed from S and Y as explained creates a system equivalent to the projection of S w.r.t. Y.

Proposition 2. *The projection of the system associated with the root of the tree constructed from S and Y w.r.t. the Y- and Z-variables as described corresponds to projecting S w.r.t. Y*

The intuition is as before; projecting all Y and Z-variables from the union of all constructed (unprojected) subsystems corresponds to projecting the Y variables from S, and because we can choose the elimination order of the variables, we only need to project the subsystems in the tree in the correct order; a rigorous proof can be found in [**?**].

Using our projection framework we obtain the projection of S w.r.t. Y by calling PROJECTNODE(root of T), where T is the tree structure constructed from S and Y and PROJECTNODE is as below.

[1]"Combining" a single sybsystem corresponds to substituting a variable with a new variable, but can be done for convenience of notation

```
function PROJECTNODERECURSIVELY(Node $n$)
    $(S, Y) \leftarrow$ the system and variable set associated with $n$
    if $n$ is a leaf then
        return FM-PROJECTIONFRAMEWORK($S$, $Y$)
    else
        for all children $m$ of $n$ do
            $S \leftarrow S \cup$ PROJECTNODE$(m)$
        return FM-PROJECTIONFRAMEWORK$(S, Y)$
```

We note that due to the boundary coarsening this only approximates $proj_Y(S)$.

Example Consider the system from the previous example, which was decomposed into the subsystems $S_1^0, \ldots, S_5^0$ and S_g^0. Instead of projecting these subsystems as described in the example we insert an additional level in the decomposition, and we choose to group the five subsystems into two groups, namely $\{S_1^0, S_2^0\}$ and $\{S_3^0, S_4^0, S_5^0\}$. That is, we construct systems $S_1^1 = \{-z_{e,1}^1 + z_{e,1}^0 + z_{e,2}^0 = 0\} \cup \{-z_{c,1}^1 + z_{c,1}^0 + z_{c,2}^0 = 0\}$ and $S_2^1 = \{-z_{e,2}^1 + z_{e,3}^0 + z_{e,4}^0 + z_{e,5}^0 = 0\} \cup \{-z_{c,2}^1 + z_{c,3}^0 + z_{c,4}^0 + z_{c,5}^0 = 0\}$, and make the tree structure as in Figure 1(b). The rephrased global inequalities are now $S_\mathrm{t}^1 = \{-wSum + z_{e,1}^1 + z_{e,2}^1 = 0\} \cup \{z_{c,1}^1 + z_{e,2}^1 \leq 20\}$. The projection of S w.r.t. Y is now made by projecting the root of the tree recursively as shown in Figure 1(c).

Other block structures Using this decomposition, it is of course also possible to project nested block structured problems, that is a system that on the top-level can be divided into a transverse part and a number of local parts that in themselves can be divided into further local parts and a transverse part, and so on. Other block structured problems such as staircase problems can also be decomposed into a tree structure and projected using this structure.

Parallelization When the system S is decomposed into subsystems in a tree structure as described above, the projection itself can be parallelized by maintaining a queue of not yet projected subsystems whose children have all been projected; this queue thus initially contains all leafs. The members of the queue are then solved independently by multiple solvers in parallel, who also add systems to the queue when the membership condition is met. The pseudocode for this is presented below. Besides a system S and a variable set Y, each node n is associated with a number, n_{count}, corresponding to the number of projected children (initially set to 0), and a system, n_{proj}, corresponding to the projection $proj_Y(S)$ when it is done (initially set to ∅).

```
function PARALLELTREEPROJECTION($S$, $Y$)
    Construct tree structure $T$ from $S$ and $Y$
    $n_{count} \leftarrow 0$ and $n_{proj} \leftarrow \emptyset$ for all nodes $n$ in $T$
    Create a set $W$ of workers, initially idle
    Initialize a queue $Q$ with all leaves in $T$
    while $root(T)_{count} = \emptyset$ do
        if $Q$ is non-empty and a worker $w \in W$ is idle then
            Remove first node $n$ from $Q$
            Call PROJECTSINGLENODE($n$) on $w$
    return the projection of the root of $T$

function PROJECTSINGLENODE(Node $n$)
    $(S', Y') \leftarrow$ the system and the variable set associated with $n$
    $S' \leftarrow S' \cup_{m \in children(n)} m_{proj}$
    $n_{proj} \leftarrow$ FM-PROJECTIONFRAMEWORK($S'$,$Y'$)
    Increase $parent(n)_{count}$ by one
    if $parent(n)_{count} = |children(parent(n))|$ then
        Add $parent(n)$ to $Q$
    return
```

All workers project a system using the FM-based projection framework, which again involves parallel redundancy removal, and thus it is important to only use as many resources (threads) in total as there are available. Then, when at some point there are less nodes left to project than there are workers in W, the superfluous workers can pass on their resources to the remaining workers.