Supplementary Materials for Paper "Uncovering Hidden Structure through Parallel Problem Decomposition for the Set Basis Problem: Application to Materials Discovery "

Yexiang Xue
Cornell University, USA
yexiang@cs.cornell.edu

Stefano Ermon
Stanford Univerity, USA
ermon@cs.stanford.edu

Carla P. Gomes
Cornell University, USA
gomes@cs.cornell.edu

Bart Selman
Cornell University, USA
selman@cs.cornell.edu

Proof of Theorem 2.1

Proof. We need show $B'_1, \ldots, B'_K$ collectively cover $\mathcal{C}$. First of all, we prove $B_i \subseteq B'_i$ for all $i \in \{1, \ldots, K\}$. This is because by definition, $B_i \subseteq C_j$ for all $C_j \in \mathcal{C}_i$. Hence, $B_i \subseteq \cap_{C_j \in \mathcal{C}_i} C_j$. On the other hand, $B'_i = \cap_{C_j \in \mathcal{C}_i} C_j$. Therefore, $B_i \subseteq B'_i$.

For a set C_i in $\mathcal{C}$, because $\{B_1, \ldots, B_K\}$ covers C_i, we can write C_i in terms of its sponsors. Let $C_i = B_{i,1} \cup \ldots \cup B_{i,m(i)}$, in which $B_{i,1}, \ldots, B_{i,m(i)} \in \{B_1, \ldots, B_K\}$. For notation purposes, we use $B'_{i,j}$ to mean the counter-part of $B_{i,j}$ in the basis set $\{B'_1, \ldots, B'_K\}$ (for example, if $B_{i,j}$ is B_1, then $B'_{i,j}$ is B'_1). We will prove $C_i = B'_{i,1} \cup \ldots \cup B'_{i,m(i)}$. Notice this completes the proof of the Theorem that $B'_1, \ldots, B'_K$ collectively cover $\mathcal{C}$ as well.

First $C_i \subseteq B'_{i,1} \cup \ldots \cup B'_{i,m(i)}$, because we just proved $B_{i,j} \subseteq B'_{i,j}$ for every j and $C_i = B_{i,1} \cup \ldots \cup B_{i,m(i)}$. Second, because $C_i = B_{i,1} \cup \ldots \cup B_{i,m(i)}$, we must have $B_{i,j} \subseteq C_i$ for $j \in \{1, \ldots, m(i)\}$. $B'_{i,j}$ is made up from the intersection of those sets $C_k \in \mathcal{C}$ who are supersets of $B_{i,j}$, which includes C_i. Hence $B'_{i,j} \subseteq C_i$ for all $j \in \{1, \ldots, m(i)\}$. This implies $B'_{i,1} \cup \ldots \cup B'_{i,m(i)} \subseteq C_i$. Based on the previous two points, $B'_{i,1} \cup \ldots \cup B'_{i,m(i)} = C_i$. □

Mixed Integer Programming Formulation

MIP Formulation For the Set Basis Problem

This MIP formulation determines if there are K basis sets to cover $C_1, \ldots, C_m$ from a finite universe U. We indexes elements in U as element 1 to n. Below are all the variables:

- For the element i ($1 \le i \le n$) and the k-th basis set B_k ($1 \le k \le K$), denote a binary variable $y_{i,k}$, which is 1 if and only if B_k contains element i.
- For B_k and C_j, denote a binary variable $z_{k,j}$, which is 1 if and only if the B_k is a contributor to set C_j.
- For element i in C_j, define a variable $u_2(i,j)$, ($0 \le u_2(i,j) \le 1$). $u_2(i,j) \ge 1$ implies element i in C_j is a false negative element (In other words, it is not covered by any basis set that is a contributor to C_j).
- For element i not included in C_j, define a variable $t_2(i,j)$ ($0 \le t_2(i,j) \le 1$). $t_2(i,j) \ge 1$ implies element i outside of C_j is a false positive element (In other words, it is contained in a basis set that is a contributor to C_j, but C_j does not have element i).

Below are all the constraints:

- For element i in set C_j, there must exist at least a basis set k, such that both $y_{i,k}$ and $z_{k,j}$ are true. Otherwise, this element counts as a false negative element. When represented using logic, for element i in set C_j,
$$(u_2(i,j) \ge 1) \vee \left(\vee_{k=1}^{K} (y_{i,k} \wedge z_{k,j})\right).$$
This constraint can be translated into a set of linear constraints, by introducing auxiliary variables $u_3(i,j,k)$ ($0 \le u_3(i,j,k) \le 1$) in the following way: For every $k \in \{1, \ldots, K\}$,
$$y_{i,k} - u_3(i,j,k) \ge 0,$$
and
$$z_{k,j} - u_3(i,j,k) \ge 0.$$
and
$$u_2(i,j) + \sum_{k=1}^{K} u_3(i,j,k) \ge 1.$$
- For element i that is outside of C_j, for the k-th basis set B_k that is a contributor to set C_j, B_k must not cover element i. Otherwise, this element counts as a false positive element. When represented using logic, for element i outside of C_j,
$$(t_2(i,j) \ge 1) \vee \left(\wedge_{k=1}^{K} (\neg y_{i,k} \vee \neg z_{k,j})\right).$$
It can be translated into linear constraints as:
$$-y_{i,k} - z_{k,j} + t_2(i,j) \ge -1,$$
for $k \in \{1, \ldots, K\}$.
- The total number of false positives and false negatives are bounded.
$$\sum_{j=1}^{m} \sum_{i \in C_j} u_2(i,j) \le FF,$$
and
$$\sum_{j=1}^{m} \sum_{i \notin C_j} t_2(i,j) \le FT.$$
In both cases to solve the global problem and the exploration phase, we would like an exact solution, hence FF and FT are set to zero.

- (Symmetry Breaking) The k-th basis set is a contributor to the 1st set, unless the $(k-1)$-th basis set is a contributor to the 1st set:

$$z_{k,1} \Rightarrow z_{k-1,1}.$$

Moreover, if the k_1-th basis set does not exist on the 1st till the $(m-1)$-th set, then the k-th basis set exists on m-th set, unless the $(k-1)$-th basis set exists on the m-th set (for $k > k_1$).

$$(\wedge_{j=1}^{m-1} \neg z_{k_1,j}) \Rightarrow (\wedge_{k=k_1+1}^{K} (z_{k,m} \Rightarrow z_{k-1,m})).$$

In our experiment, we insert these type of constraints until $m = 4$.

- (Redundant Constraint) This constraint is redundant. It is used to trigger more propagation: if all elements in the k-th basis set are all contained in set C_j, then $z_{k,j} = 1$. In the form of linear constraints,

$$z_{k,j} + \sum_{i \notin C_j} y_{i,k} \geq 1.$$

MIP Formulation In the Pre-solving Step

We detail the MIP formulation for the selection sub-step in the pre-solving step, in which K basis sets are selected from $\mathcal{U}$ which minimize the number of uncovered and falsely covered elements.

Below are all the variables:

- For the i-th set from $\mathcal{U}$, introduce binary variable $b_{i,k}$, which is 1 if and only if the i-th set is selected to be the k-th final basis set B_k^* $(1 \leq k \leq K)$.
- Introduce binary variable $I_{k,j}$, which is 1 if and only if the k-th final basis set B_k^* is a contributor to the j-th set C_j.
- Real variable $u_{l,j,k}$: $u_{l,j,k} \geq 0$. $u_{l,j,k} \geq 1$ implies the element l from C_j is covered by the k-th final basis set.
- For element l in set C_j, define a real variable $t_{l,j}$, $(0 \leq t_{l,j} \leq 1)$, $t_{l,j} \geq 1$ implies element l in the set C_j is a false negative element (In other words, it is not covered by any final basis sets that is a contributor in C_j).
- For element l outside of set C_j, define a real variable $f_{l,j}$, $(0 \leq f_{l,j} \leq 1)$, $f_{l,j} \geq 1$ implies element l outside of the set C_j is a false positive element (In other words, it is contained in a basis set that is a contributor for set C_j, but C_j does not have element l).

Below are all the constraints:

- Every set from $\mathcal{U}$ is selected at most once:

$$\sum_k b_{i,k} \leq 1.$$

- The k-th final basis set can only pick at most one set from $\mathcal{U}$:

$$\sum_i b_{i,k} \leq 1.$$

- By definition, $u_{l,j,k} \geq 1$ implies the element l from C_j is covered by the k-th final basis set. Represented in logic:

$$u_{l,j,k} \Rightarrow \left(\vee_{(i' \in \mathcal{U}) \wedge (l \in i')} b_{i',k}\right) \wedge I_{k,j},$$

$l \in i'$ means element l is in the i'-th set from $\mathcal{U}$. It can be translated to linear equations as:

$$-u_{l,j,k} + \sum_{(i' \in \mathcal{U}) \wedge (l \in i')} b_{i',k} \geq 0,$$

and

$$-u_{l,j,k} + I_{k,j} \geq 0.$$

- For element l in set C_j, l is covered by at least one basis set, otherwise l counts as a false negative element; which is:

$$\sum_k u_{l,j,k} + t_{l,j} \geq 1.$$

- For every element l that does not exist at set C_j, for every $k \in \{1, \ldots, K\}$, for the i_1-th set in $\mathcal{U}$ that contains l, either $b_{i_1,k}$ is not true, or $I_{k,j}$ is not true, or l counts as a false positive element, which is:

$$-b_{i_1,k} - I_{k,j} + f_{l,j} \geq -1.$$

The goal of the pre-solving step is to find K basis sets that minimizes the total number of uncovered elements and falsely covered elements in $\mathcal{C}$. So the objective function is,

$$\text{minimize} \sum_j \sum_{l \in C_j} t_{l,j} + \sum_j \sum_{l \notin C_j} f_{l,j}.$$

Pseudocode

This is the incomplete algorithm to form $\mathcal{U}$ within the space of $\mathcal{B}_0$ in the pre-solving step. In our experiment, p is set to 0.95, c is set to 0.5.

Algorithm 1: The incomplete algorithm to form $\mathcal{U}$ within the space of $\mathcal{B}_0$.

```
U ← ∅;
while |U| < T_U do
    B ← randomly chosen from B_0;
    b_0 ← |B|;
    while |U| < T_U and |B| ≥ c · b_0 do
        if with probability p then
            C ← arg max_{C ∈ B_0, C ⊉ B} |B ∩ C|;
        else
            C ← randomly chosen from B_0;
        end
        B ← B ∩ C;
        U ← U ∪ {B};
    end
end
return U
```