1. Unique regions identification

We proposed a method to count and identify the regions defined by the KN hyperplane arrangement A with sign vectors. In our problem, all the hyperplanes pass one common point. Here we define two concepts about strict hyperplane and adjacent region. Strict hyperplane can be taken as non-redundant bounding hyperplane of some region. If two regions are existed when their sign vectors differ in only one hyperplane, then this hyperplane is a strict hyperplane. Adjacent region of region r is the neighbor region of r if only one strict hyperplane seperates them. The general idea of identifying unique regions is based on partial order set that we first initialize a root region and then find out all the adjacent regions of each found region. This guarantees that we can enumerate every unique region without missing one. The procedure includes two main components. One component is initializing a region defined by the hyperplanes using an interior point method. The other component is finding out the sign vectors of adjacent regions by finding the set of strict hyperplanes. The unique regions are represented by θ^B matrix, where the rows are regions and there are KN columns. Each element of this matrix is either 0 or 1.

In the procedure, first we remove all the duplicate and all-zero coefficients hyperplanes to get unique hyperplanes. Then we start from a specific region r and put it into a open set. Open set is used to maintain a region list which need to be explored. Each time we pick one region from the open set for finding adjacent regions. Once finishing the step of finding adjacent regions, region r will be moved into a closed set. Closed set is used to maintain a region list which already be explored. Also, if the adjacent region is a newly found one, it also need to be put into the open set for exploring. Finally, once the open set is empty, regions in the closed set are all the unique regions, and the number of the unique regions is the length of the closed set. This procedure begins from one region and expands to all the neighbors until no new neighbor is existed.

The overview of this algorithm including two main components are shown below. $Ax = b$ are the equations of unique hyperplanes.

Algorithm 1 Unique regions identification Algorithm

Sort the rows of the KN x M qualifying constraint coefficient matrix.
Compare adjacent rows of the qualifying constraint coefficient matrix and eliminate duplicate rows.
Eliminate rows of the qualifying constraint coefficient matrix with all-zero coefficients.
Determine the list of unique qualifying constraints by pairwise test.
Set S and *num_cuts* to the set of unique, non-trivial qualifying constraints and the number of them.
Initialize a region *root* using an interior point method (Component 1).
Put region *root* into the open set.
if open set is not empty **then**
    Get a region R from the open set
    Calculate the adjacent regions set *R_adj* (Component 2).
    Put region R into the closed set.
    for each region r in *R_adj* **do**
        if r is not in open set *and* not in closed set **then**
            Put region r into open set.
        end if
    end for
end if
Reflect the sign of the regions in the close set
Get all the regions represented by string of 0 and 1

1.1. **Hyperplane filtering.** Assuming there are two different hyperplanes H_i and H_j represented by $A_i = \{a_{i,0}, ..., a_{i,MK}\}$ and $A_j = \{a_{j,0}, ..., a_{j,MK}\}$. We take these two hyperplanes duplicated when

$$\frac{a_{i,0}}{a_{j,0}} = \frac{a_{i,1}}{a_{j,1}} = ... = \frac{a_{i,MK}}{a_{j,MK}} = \frac{\sum_{l=0}^{MK} a_{i,l}}{\sum_{l=0}^{MK} a_{j,l}}, a_{j,l}! = 0 \tag{1}$$

This can be converted to

$$|\sum_{l=0}^{MK} a_{i,l} \cdot a_{j,n} - \sum_{l=0}^{MK} a_{j,l} \cdot a_{i,n}| \leqslant \tau, \forall n \epsilon [0, MK] \tag{2}$$

where threshold τ is a very small positive value.

We eliminate a hyperplane H_i represented by $A_i = \{a_{i,0}, ..., a_{i,MK}\}$ from hyperplane arrangement A if the coefficients of A_i are all zero,

$$|a_{i,j}| \leqslant \tau, \ \forall a_{i,j} \epsilon A_i, j \epsilon [0, MK] \tag{3}$$

1.2. **Interior point method.** An interior point is found by solving the following optimization problem:

$$\text{maximize } z$$

$$\text{subject to } -A_i x + z \leqslant b_i, if \theta_i^B = 0 \tag{4}$$

$$A_i x + z \leqslant -b_i, if \theta_i^B = 1 \tag{5}$$

$$z > 0 \tag{6}$$

Algorithm 2 Interior Point Method (Component 1)

```
Generate 2^{num_cuts} different strings using 0, 1
for each s in the strings do
    Solve an optimization problem to get an interior point.
    if Get a interior point then
        Get the root region represented by 0 and 1.
    end if
end for
```

1.3. **Get adjacent regions.**

Algorithm 3 Get adjacent regions (Component 2)

Initialize an empty set SH for strict hyperplanes.
Initialize an adjacent region set ADJ
# Find out all the strict hyperplanes for region R.
for each hyperplane H of num_cuts hyperplanes **do**
    Pick one hyperplane H from all the hyperplanes definging region R.
    Flip the sign of H to get H'
    Form a new hyperplane arrangement A' with H'.
    Solve the problem to get an interior point constrained by A'.
    if The interior point is not Non **then**
        H is a strict hyperplane and put into set SH.
    else
        H is a redundant hyperplane.
    end if
end for
# Find out all the adjacent regions for region R.
for each strict hyperplane sh in set SH **do**
    Take the opposite sign sh' of sh.
    Form a adjacent region adj based on sh' and all the else hyperplanes.
    Put adj into set ADJ.
end for