Rappel de la dernière fois

- On avait discuté sur l’emplacement réel de l’insertion d’un sommet dans l’ensemble P

Algorithm 3 ReduceBranches(G, D, U, D^*), algorithm to identify a branching set B

Input: A graph G=(V,E), a partial solution D, the set of undominated vertices U and the best solution D^* found so far
Output: A branching set B

Let P=$\emptyset$, C=$\mathbb{S}_1 \cup \mathbb{S}_2$, U=$\mathbb{S}_1 \cup \mathbb{S}_3$=$\{u_1, u_2, \ldots, u_{|U|}\}$;
Let $\Pi = \{S_1, S_2, \ldots, S_k\}$ be the set of k ISs of $G[P]^2$, each S_i is initialized to $\emptyset$ at the beginning;
for $i = |U|$ to 1 **do**
  Let $totalScore = 0$;
  for each nonempty IS $S_j \in \Pi$ **do**
    $totalScore \leftarrow totalScore + \delta(S_j, u_i)$;
  end for
  if $totalScore \geq 0$ **then**
    $P \leftarrow P \cup \{u_i\}$;
    for each nonempty $S_j \in \Pi$ **do**
      if $\delta(S_j, u_i) > 0$ **then**
        insert u_i into S_j;
      end if
      if $\delta(S_j, u_i) < 0$ **then**
        remove $|N(u_i) \cap S_j|$-1 conf. vertices from S_j
      end if
    end for
    if u_i hasn’t been inserted into any nonempty IS **then**
      insert u_i into the first empty IS S_i;
    end if
  end if
end for
Let $lb = \max\{|S_j| \mid S_j \in \Pi\}$;
if $lb < |D^*| - |D|$ **then return** C;

Comment les IS sont remplis ?

for $i = |U|$ to 1 **do**
    Let $totalScore = 0$;
    for each nonempty IS $S_j \in \Pi$ **do**
        $totalScore \leftarrow totalScore + \delta(S_j, u_i)$;
    end for
    if $totalScore \geq 0$ **then**
        $P \leftarrow P \cup \{u_i\}$;
        for each nonempty $S_j \in \Pi$ **do**
            if $\delta(S_j, u_i) > 0$ **then**
                insert u_i into S_j;

Dans un premier temps, les seuls sommets pouvant intégrer un IS sont des sommets de U, à savoir les sommets de S1 U S2.

Condition d’insertion dans l’IS

Comme on travaille pour le moment avec les sommets de U (S1 U S2), seules les lignes encadrées nous intéressent.

(S1 = non branché non dominé)

$$\delta(S_j, u) = \begin{cases} 1 & u \in \mathbb{S}_1 \wedge N_1 = \emptyset \wedge N_2 = \emptyset \\ 0 & u \in \mathbb{S}_1 \wedge N_1 = \emptyset \wedge N_2 \neq \emptyset \\ 1 - |N_1| & u \in \mathbb{S}_1 \cup \mathbb{S}_2 \wedge N_1 \neq \emptyset \\ 0 & u \in \mathbb{S}_2 \wedge N_1 = \emptyset \\ 0 & u \in \mathbb{S}_3 \wedge N_2 \neq \emptyset \\ 1 & u \in \mathbb{S}_3 \wedge N_2 = \emptyset \end{cases}$$

$$N_1 = N_G(u) \cap S_j$$
$$N_2 = N_G[u] \cap N_G[S_j] \cap P \cap C$$

N1 est vide si le sommet que l’on insère dans IS n’a pas de voisin dans celui-ci

N2 est vide si u et l’IS dans lequel on veut l’insérer n’ont pas de voisins en commun, qui soit dans P et dans C

Conclusion

La réduction des branches commence à se faire quand on a déjà branché beaucoup de sommets donc vers la fin des branches…

Tentative d’amélioration en cours :

Actuellement

On part d’un graphe G’, graphe réduit de G par les règles d’Alber et on applique l’algorithme de branch & bound et de réduction de branche de Hua jusqu’à son terme

Prochainement

Même chose, sauf qu’à chaque fois qu’on branche sur un élément, on réapplique les règles d’Alber pour réduire le graphe à nouveau et avoir un plus petit graphe -> donc moins de branche

Passer du problème de MDS au problème de pcentre

Première approche : Dichotomique

Algorithm 1 MDS to solving pcenter problem

Require: $pcentre$; $instance$; $borneInf \leftarrow$ Plus petite valeur de la matrice de distance
$borneSup \leftarrow$ Plus grande valeur de la matrice de distance

while $borneInf \neq borneSup$ **do**
    $element \leftarrow (borneSup + borneInf)/2$
    if Card(EMOS(G, $element$)) $\leq pcentre$ **then**
        $borneInf \leftarrow element$
    else
        $borneSup \leftarrow element + 1$
    end if
end while
return $borneInf$

Distance	1	2	3	4	5	6	7	8	9	10
Taille MDS	9	8	7	7	6	5	4	4	3	2

On cherche la plus petite distance qui donne une taille de MDS = 4 (nombre de centre)

borneinf=1 ; bornesup=10 ; pcentre = 4

element = (1+10)/2 = 5

6>4 donc on travaille sur la partie droite de l’indice 5

Distance	1	2	3	4	5	6	7	8	9	10
Taille MDS	9	8	7	7	6	5	4	4	3	2

borneinf=6 ; bornesup=10 ; pcentre = 4

element = (6+10)/2 = 8

4=4 donc on travaille sur la partie gauche de l’indice 8, avec 8 inclus.

Distance	1	2	3	4	5	6	7	8	9	10
Taille MDS	9	8	7	7	6	5	4	4	3	2

borneinf=6 ; bornesup=8 ; pcentre = 4

element = (6+8)/2 = 7

4=4 donc on travaille sur la partie gauche de l’indice 7, avec 7 inclus.

Distance	1	2	3	4	5	6	7	8	9	10
Taille MDS	9	8	7	7	6	5	4	4	3	2

borneinf=6 ; bornesup=7 ; pcentre = 4

element = (6+7)/2 = 6

5>4 donc on travaille sur la partie droite de l'indice 6

Distance	1	2	3	4	5	6	7	8	9	10
Taille MDS	9	8	7	7	6	5	4	4	3	2

borneinf=7 ; bornesup=7 ; pcentre = 4

borneinf=bornesup

On a notre plus petite distance qui donne une taille de MDS de 4.