UPMC
SORBONNE UNIVERSITÉS

4I200 – Modélisation, Optimisation, Graphes et Programmation Linéaire

Rapport de projet

Un problème de tomographie discrète

Celia Kherfallah, 3605205

Stieban Fernandez, 3407186

Encadré par M. Patrice Perny

Table des matières

Raisonnement par programmation dynamique

Première étape

Q1) T(j,l) est vrai s'il est possible de colorier les j + 1 premières cases de la ligne li avec la sous-séquence des l premiers blocs de la ligne li. Colorier la ligne li entière avec la séquence entière revient donc à savoir si T(M-1, k) est vrai, avec M la taille de la ligne li, et k la taille de la séquence (nombre de blocs).

Q2)

Cas l = 0 :

Le nombre de blocs étant à 0, il y a une infinité de coloriage possible avec cette séquence. Par conséquent, T(j, l) est vrai quelque soit la valeur de j.

Cas l >= 0 :

Cas j < sl – 1 :

La taille du bloc étant supérieure au nombre de cases disponibles, le coloriage de cette séquence est impossible. T(j, l) est donc faux.

Cas j = sl – 1 :

La taille du bloc correspond parfaitement au nombre de cases disponibles. Il y a ici deux possibilités :

- Si l = 1, il s'agit du dernier bloc donc T(j, l) est vrai.
- Si l >= 0, il ne s'agit pas du dernier bloc et il en reste d'autres. Il sera impossible de placer les autres blocs par la suite, T(j, l) est donc faux.

Q3) Relation de récurrence pour le cas j > sl – 1 :

Si la case (i, j) est blanche : T(j - 1, l)
Sinon T(j – st – 1, l – 1)

Q4) On considère dans cet algorithme que les cases ne sont pas coloriées :

Algorithme 1 Calcul de T(j, l, seq) :

Entrées :

- j : indice de la dernière case (de la partie) de la ligne li à considérer
- l : indice du bloc dans la séquence à placer
- seq : séquence de blocs associée à la ligne li (s1,…, sm)

Sortie : Booléen (vrai ou faux)

```
Si l = 0 alors :
    Renvoyer vrai
Si l >= 0  alors :
    Si j < sl – 1 :
        Renvoyer faux
    Sinon si j = sl – 1 :
        Si l = 1 :
            Renvoyer vrai
        Sinon :
            Renvoyer faux
    Sinon :
        Renvoyer T(j – sl – 1, l – 1, seq)
```

Généralisation

Q5-6)

On considère dans cet algorithme que certaines cases sont cette fois-ci déjà coloriées :

Algorithme 2 Calcul de T(j, l, li, seq) :

Entrées :

- j : indice de la dernière case (de la partie) de la ligne li à considérer
- l : indice du bloc dans la séquence à placer
- li : ligne li contenant les cases déjà coloriées
- seq : séquence de blocs associée à la ligne li (s1,…, sm)

Sortie : Booléen (vrai ou faux)

```
Si l = 0 alors :
    S'il y a une case noire parmi les j + 1 premières cases :
        Renvoyer faux
    Sinon :
        Renvoyer vrai
Si l >= 0  alors :
    Si j < sl – 1 :
        Renvoyer faux
    Sinon si j = sl – 1 :
        Si l = 1 :
```

```
                S'il y a une case blanche parmi les j + 1 premières cases :
                    Renvoyer faux
                Sinon :
                    Renvoyer vrai
            Sinon :
                Renvoyer faux
        Sinon :
            S'il y a une case blanche entre la case(i, j – sl + 1) et la case(i, j) :
                b <- indice de la case blanche
                S'il y a une case noire entre la case (i, b) et la case (i, j) :
                    Renvoyer faux
                Sinon :
                    Renvoyer T(b – 1, l, li, seq)
            Sinon :
                Si la case(i, j – sl ) est noire :
                    Si la case(i, j) n'est pas noire :
                        Renvoyer T(j – 1, l, li, seq)
                    Sinon :
                        Renvoyer faux
                Sinon :
                    Renvoyer T(j – sl – 1, l – 1, seq)
```

Propagation

Voir le code.

Tests

Q7-8) Temps de résolution (avec application de la mémoïsation) :

Instances	***Temps de résolution***
0.txt	0.001366166789647337
1.txt	0.002520462260696294
2.txt	0.2684402275453975
3.txt	0.15642518101618325
4.txt	0.28734516073811867
5.txt	0.46785512547144126
6.txt	1.4384391022081169
7.txt	0.49193587703545827
8.txt	0.8686320847221349
9.txt	9.597671028356666
10.txt	8.587654793441962

Grille de l'instance 9.txt :

Q9) Le programme ne se termine pas sur l'instance 11.txt, on remarque une certaine boucle infinie.

Explication :

Notre algorithme est limité en termes de contraintes. Nous avons défini précédemment que si pour une ligne li, il est possible de la colorier entièrement avec la séquence complète en coloriant une case(i, j) en blanc ou en noir, alors on ne peut rien dire sur la couleur de cette case. On l'explore alors par la suite en ajoutant la colonne j à la liste des colonnes à voir. Ces cases restant ainsi indéterminées, le programme ne s'arrête pas.

Alternative : Pour terminer tout de même le programme (instances 11.txt à 16.txt), nous avons ajouté une variable k arbitraire qui définit un nombre d'itérations limite, car la grille n'évolue plus au-delà d'un nombre d'itérations. Nous remarquons à la fin des itérations la présence cases indéterminées (pour l'instance 11.txt, toutes les cases sont indéterminées).

La PLNE à la rescousse

Modélisation

Q10) Si $y_{(i,j)}^{t}$=1 et que la taille du bloc t est égale à s_t

Alors $x_{(i,j)} + x_{(i,j+1)} + \cdots + x_{(i,j+st-1)} = s_t$

$$s_t * y_{(i,j)}^{t} - \sum_{k=j}^{j+sl-1} x(i,j) <= 0$$

Si $z_{(i,j)}^{t}$=1 et que la taille du bloc t est égale à s_t

Alors $x_{(i,j)} + x_{(i,j+1)} + \cdots + x_{(i,j+st-1)} = s_t$

$$s_t * z_{(i,j)}^{t} - \sum_{k=j}^{j+sl-1} z(i,j) <= 0$$

Q11) Si $y_{(i,j)}^{t}$=1 et que la taille du bloc t est égale à s_t

Alors $y_{(i,j)}^{t} + \sum_{k=0}^{j+sl} y_{(i,k)}^{t+1}$ <= 1, avec t+1 < longueur_sequence

Si $z_{(i,j)}^{t}$=1 et que la taille du bloc t est égale à s_t

Alors $z_{(i,j)}^{t} + \sum_{k=0}^{j+sl} z_{(i,k)}^{t+1}$ <= 1, avec t+1 < longueur_sequence

Q12) Contraintes supplémentaires :

La contrainte ci-dessous nous permet d'assure que pour chaque ligne (respectivement pour chaque colonne), qu'un bloc t est représenté une et une seule fois c-a-d que si le bloc t commence à la case (i,j) pour tous les yi,k associés au bloc t avec k {j,j+st}, uniquement yi,j_t est égale à 1 ce qui correspond à la case de début du bloc.

Si $y_{(i,j)}^{t}$= 1 et que la taille du bloc t est égale à s_t

Alors $y_{(i,j+1)}^{t} + y_{(i,j+2)}^{t} + \cdots + y_{(i,j+sl-1)}^{t}$ = 0

Donc $y_{(i,j)}^{t} + \sum_{k=j}^{j+sl-1} y_{(i,k)}^{t}$ == 1

Si $z_{(i,j)}^{t}$=1 et que la taille du bloc t est égale à s_t

Alors $z_{(i,j+1)}^{t} + z_{(i,j+2)}^{t} + \cdots + z_{(i,j+sl-1)}^{t}$ = 0

Donc $z_{(i,j)}^{t} + \sum_{k=j}^{j+sl-1} z_{(i,k)}^{t}$ == 1

La contrainte ci-dessous nous permet d'assure que pour chaque ligne i (respectivement pour chaque colonne j) la somme des cases coloriées en noir est égale à la somme des blocs de toute la séquence.

$\sum_{j=0}^{M-1} x_{(i,j)}$ - $\sum_{k=0}^{longueur-1} sequence\_ligne[i][k]$ = 0

$\sum_{i=0}^{N-1} x_{(i,j)}$ - $\sum_{k=0}^{longueur-1} sequence\_colonne[j][k]$ = 0

La contrainte ci-dessus stipule qu'après un bloc t de taille St qui débute à la case (i,j) alors la case $x_{(i,j+St)}$ qui suit le bloc t est forcément blanche.

$y_{(i,j)}^{t} + x(i, j + St)$ <= 1, avec $j + St < M$

$z_{(i,j)}^{t} + x(i + St, j)$ <= 1, avec $i + St < N$

La contrainte ci-dessous impose que le nombre de cases blanches présentent dans une ligne (respectivement dans une colonne) est supérieur ou égale à la taille de la séquence -1

M - $\sum_{j=0}^{M-1} x_{(i,j)}$ >= longueur_sequence_ligne -1

N - $\sum_{j=0}^{N-1} x_{(i,j)}$ >= longueur_sequence_colonne -1

La contrainte ci-dessus impose que si un un bloc t commence à la case (i,j) alors yt_(i,j) est égale à 1 donc forcément x(i,j) est égale à 1

$x_{(i,j)} - y_{(i,j)}^{t}$ <=1

$x_{(i,j)} - z_{(i,j)}^{t}$ <=1

Implantation et tests

Q13) Pour une ligne li, l'intervalle hors duquel le bloc d'indice l ne peut commencer se définit ainsi :

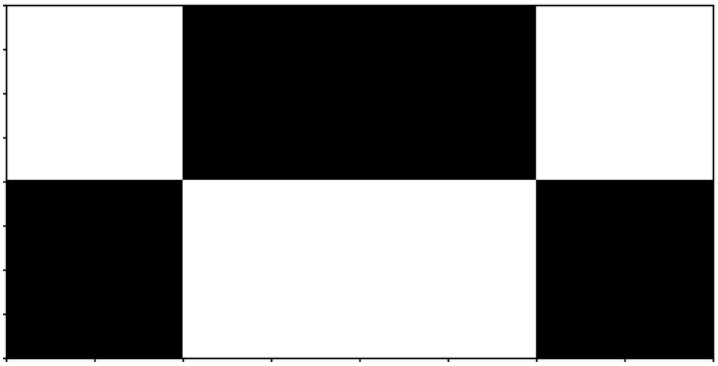
[inf ; sup]

inf = $\sum_{k=0}^{k=t-1} sequence\_ligne[i][k]$ + longueurSequence[i][:t-1],

avec t-1 >= 0

sup = M – ($\sum_{k=t}^{k=longueurSequence-1} sequenceligne[i][k]$ + longueurSequence[i][t+1 :]), avec t+1 < longueurSequence[i] – 1

Q14-15) Grille de l'instance 11.txt :



Temps de résolution avec la PLNE (Gurobi) :

Instances	***Temps de résolution***
0.txt	0.00
1.txt	0.00
2.txt	1.53
3.txt	0.25
4.txt	5.63
5.txt	1.38
6.txt	22.68
7.txt	1.06
8.txt	1.91
9.txt	time over
10.txt	35.02
11.txt	0.00

	PLNE	***Prog. Dynamique (k=100)***
12.txt	36.20	4.3637328353990785
13.txt	2.97	1.3385399426921074
14.txt	1.94	2.2175819470974183
15.txt	23.66	12.765473845805332
16.txt	535.09 (time over)	46.369106229273356

Grille de l'instance 15.txt avec la programmation dynamique (gauche) et PLNE (droite) :

Pour aller plus loin

Temps de résolution avec le prétraitement puis la PLNE sur les cases non déterminées :

Instances	***Temps de résolution***
0.txt	0.02328861675520908
1.txt	0.01961715805783647
2.txt	0.6323336159506112
3.txt	0.4351913861479412
4.txt	0.977987009868878
5.txt	0.7866070934370843
6.txt	1.9741854225934736
7.txt	1.1065254533331428
8.txt	1.7061462110040393
9.txt	15.489136829649961
10.txt	18.9487879897554
11.txt	0.05213390707206667
12.txt	3.762876039516567
13.txt	2.87002977561801
14.txt	2.3427315491422327
15.txt	12.781607223565663
16.txt	476.2743982836224