

Un problème de tomographie discrète

MOGPL

Master d'informatique M1

Université Pierre et Marie Curie

Baptiste JARRY
Magdaléna TYDRICHOVA

10 décembre 2017

Introduction

En cadre de ce projet, on s'intéresse au problème de tomographie discrète.

Dans la première partie, on résoud le problème à l'aide de la programmation dynamique, on teste l'algorithme sur plusieurs instance et essaie d'observer et expliquer ses limites.

Souhaitant pouvoir résoudre des grilles plus complexes, on met en oeuvre la résolution à l'aide d'un PLNE. On modélise d'abord le problème comme un PLNE, puis on le test et on compare les solutions avec celles de la méthode précédente.

Dans la dernière section du projet, une analyse brève des résultats ainsi que certaines améliorations possibles sont proposées.

1 Raisonnement par la programmation dynamique

1.1 Première étape

Q1

Si on a calculé tous les $T(i, j)$, la case $T(m-1, k)$ nous indique si on peut colorier les m premières cases de la ligne l_i (donc la ligne entière) avec k premiers blocs (i.e. la séquence entière).

Q2

1. Si $l = 0$ et $j \in \{0, \dots, m-1\}$, alors $T(j, l) = 0$. En effet, on peut "colorier" n'importe quel nombre de cases avec aucun bloc.
2. On suppose maintenant $l \geq 1$.
 - (a) $j < s_l - 1 \Rightarrow T(j, l) = FALSE$. En effet, cette inégalité signifie que le nombre de cases à colorier ($j+1$) est strictement plus petit que la longueur du dernier bloc. On peut pas donc colorier les $j+1$ premières cases avec le bloc s_l , et alors non plus avec la sous-séquence des blocs (s_1, \dots, s_l) .
 - (b) $j = s_l - 1 \Leftrightarrow j+1 = s_l$, ce qui signifie que la longueur du dernier bloc est exactement égale au nombre de cases à colorier. On en déduit que $T(j, l) = TRUE$ et $T(j, l) = FALSE$ pour $l > 1$.

Q3

On considère dans cette question le dernier cas non traité, c'est à dire le cas où $l \geq 1, j > s_l - 1$. Il y a deux possibilités :

- Soit la case (i, j) restera blanche après la coloration, dans quel cas $T(j, l) = T(j-1, l)$.
- Soit la case (i, j) sera noire après la coloration, ce qui signifie que le bloc s_l se termine à la case (i, j) . On en déduit qu'il commence à la case $(i, j - (s_l - 1))$. Les blocs étant séparés par au moins une case blanche, la case $(i, j - s_l)$ sera blanche. Si $j - s_l > 0$, alors $T(j, l) = T(j - s_l - 1, l - 1)$. Si $j - s_l = 0$, alors $T(j, l) = TRUE$ si et seulement si $l = 1$.

1.2 Généralisation

Q5

1. Si $l = 0$ et $j \in \{0, \dots, m-1\}$: On peut colorier $j+1$ premières cases avec 0 blocs si aucune de cases $0, \dots, j$ n'est pas déjà coloriée à noir (ce qui imposerait une présence d'un bloc, ou au moins de sa partie).
2. On suppose maintenant $l \geq 1$.
 - (a) $j < s_l - 1 \Rightarrow T(j, l) = FALSE$ pour le même raison que précédemment.
 - (b) $j = s_l - 1$ Comme dans la question 2, $T(j, l) = FALSE$ pour $l > 1$. Si $l = 1$, $T(j, l) = TRUE$ si et seulement si il n'y a pas de case déjà coloriée à noir avant la case (i, j) .
 - (c) Supposons $j > s_l - 1$. On distingue toujours deux cas.
 - Soit le l -ième bloc finit sur la case (i, j) . Dans ce cas là, il faut que la case $(i, j+1)$ n'est pas coloriée à noir (si elle existe). De plus, il

- faut qu'aucune case entre $(i, j - s_l + 1)$ n'est pas coloriée à blanc, et que la case $(i, j - s_l)$ qui précède le bloc n'est pas coloriée à noir. $T(j, l) = TRUE$ si toutes ces conditions sont vérifiées et on a de plus $T(j - s_l, l - 1) = TRUE$.
- Soit le l -ième bloc ne finit pas sur la case (i, j) . Dans ce cas là, on a $T(j, l) = TRUE$ si et seulement si on a $T(j - 1, l) = TRUE$ et la case (i, j) n'est pas coloriée à noir (dans quel cas on ne pourrait pas colorier les $j + 1$ premières cases uniquement avec l premières blocs, il y aurait au moins une case du bloc suivant).

Dans les deux cas, on calcule en plus le nombre de cases noires entre les positions $(i, 0)$ et (i, j) . Si ce nombre dépasse la somme des longueurs des l premiers blocs, alors on ne peut pas colorier $j + 1$ premières cases uniquement avec l blocs, en conséquence, $T(j, l) = FALSE$.

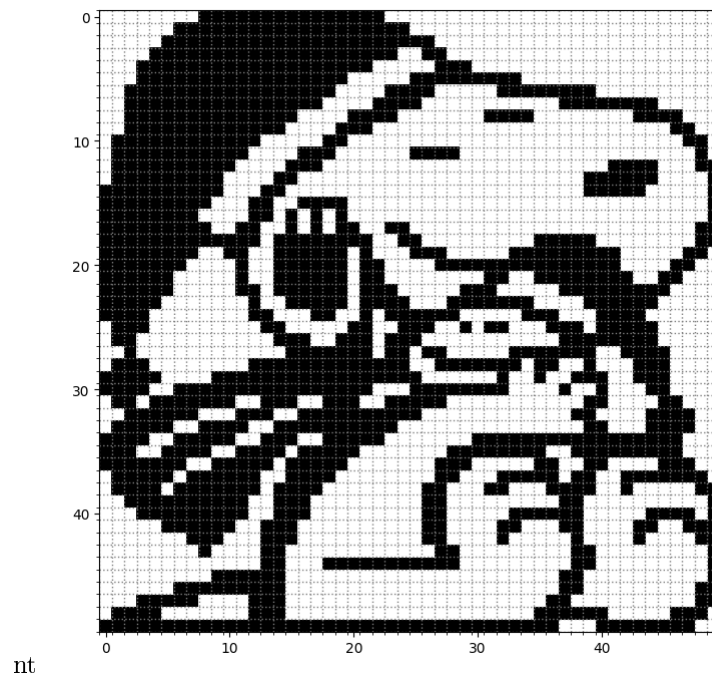
De la même manière, si la somme des cases noires entre $(i, j + 1)$ et $(i, N - 1)$ dépasse la somme des longueurs des blocs s_{l+1}, \dots, s_k , on en déduit que $T(j, l) = FALSE$.

Q8

Voici le tableau des temps de résolution pour les instances 1-10 :

numéro d'instance	temps de résolution [s]
1	0.013
2	5.8
3	4.2
4	10.7
5	7.7
6	22.7
7	10.7
8	18.5
9	342.7
10	349.2

La grille obtenue pour le fichier *9.txt* est la suivante :



Q9

Si on applique l'algorithme sur l'instance *11.txt*, on observe que aucune case n'a pas pu être coloriée.

2 La PLNE

2.1 Modélisation

Q10

Soit l_i la i -ième ligne avec une séquence associée (s_1, \dots, s_k) . Si le bloc t de longueur s_t commence par la case (i, j) , alors les cases (i, j) à $(i, j + s_t - 1)$ doivent être noires, ce qui s'exprime comme :¹

$$y_{ij}^t \leq \frac{\sum_{l=j}^{j+s_t-1} x_{il}}{s_t}$$

De manière analogue, on a pour la j -ième colonne c_j possédant la séquence $(s'_1, \dots, s'_{k'})$:

$$z_{ij}^t \leq \frac{\sum_{l=i}^{i+s'_t-1} x_{lj}}{s_t}$$

Q11

Avec les notations de la question précédente, on souhaite d'exprimer le fait que si le bloc t de la i -ième ligne commence à la case (i, j) , alors le $(t+1)$ -ième bloc ne peut pas commencer avant la case $(i, j + s_t + 1)$. Ce qui se formule par :

$$y_{ij}^t \leq \sum_{l=j+s_t+1}^{N-1} y_{il}^{t+1}, t \in \{1, \dots, k-1\}$$

De manière analogue, on obtient pour les colonnes :

$$z_{ij}^t \leq \sum_{l=j+s_t+1}^{M-1} z_{lj}^{t+1}, t \in \{1, \dots, k'-1\}$$

Q12

Pour formuler le PLNE, il faut exprimer à l'aide des variables définies précédemment les contraintes suivantes :

- (1) Un bloc ne doit apparaître qu'une seule fois sur une ligne.
- (2) Un bloc ne doit apparaître qu'une seule fois sur une colonne.
- (3) Il doit y avoir un bon nombre de cases noires par ligne.
- (4) Il doit y avoir un bon nombre de cases noires par colonne.
- (5) Le bloc $t+1$ d'une ligne doit se trouver après le bloc t .
- (6) Le bloc $t+1$ d'une colonne doit se trouver après le bloc t .
- (7) Si le bloc t d'une ligne démarre à la colonne j , alors les cases qui suivent doivent être noires sur la longueur du bloc.
- (8) Si le bloc t d'une colonne démarre à la colonne i , alors les cases qui suivent doivent être noires sur la longueur du bloc.

1. Cette formulation n'est pas exacte. En fait, elle ne prend pas en compte le fait qu'on peut potentiellement dépasser les bords dans la somme. Ce problème est réglé dans la question 13 en limitant le nombre de variables.

On en déduit alors le PLNE suivant :

$$\min \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{ij}$$

$$\sum_{j=0}^{N-1} y_{ij} = 1 \quad \forall i \in \{0, \dots, M-1\} \quad (1)$$

$$\sum_{i=0}^{M-1} z_{ij} = 1 \quad \forall j \in \{0, \dots, N-1\} \quad (2)$$

$$\sum_{j=0}^{N-1} x_{ij} - \sum_{l=1}^k s_l = 0 \quad \forall i \in \{0, \dots, M-1\} \quad (3)$$

$$\sum_{i=0}^{M-1} x_{ij} - \sum_{l=1}^{k'} s_l = 0 \quad \forall j \in \{0, \dots, N-1\} \quad (4)$$

$$y_{ij}^t \leq \sum_{l=j+s_t+1}^{N-1} y_{il}^{t+1}, t \in \{1, \dots, k-1\}, i \in \{0, \dots, M-1\} \quad (5)$$

$$z_{ij}^t \leq \sum_{l=j+s_t+1}^{M-1} z_{lj}^{t+1}, t \in \{1, \dots, k'-1\}, j \in \{0, \dots, N-1\} \quad (6)$$

$$y_{ij}^t \leq \frac{\sum_{l=j}^{j+s_t-1} x_{il}}{s_t}, i \in \{0, \dots, M-1\}, t \in \{1, \dots, k-1\} \quad (7)$$

$$z_{ij}^t \leq \frac{\sum_{l=i}^{i+s'_t-1} x_{lj}}{s_t}, j \in \{0, \dots, N-1\}, t \in \{1, \dots, k'-1\} \quad (8)$$

2.2 Implantations et tests

Q13

Soit l_i la i -ième ligne avec la séquence associée (s_1, \dots, s_k) . Le l -ième bloc peut commencer au plus tôt sur la case (i, j_d) avec

$$j_d = l + \sum_{t=1}^{l-1} s_t$$

Ceci correspond au cas dans lequel le premier bloc commence par la première case de l_i et tous les prédécesseurs du l -ième bloc sont séparés entre eux par exactement une case.

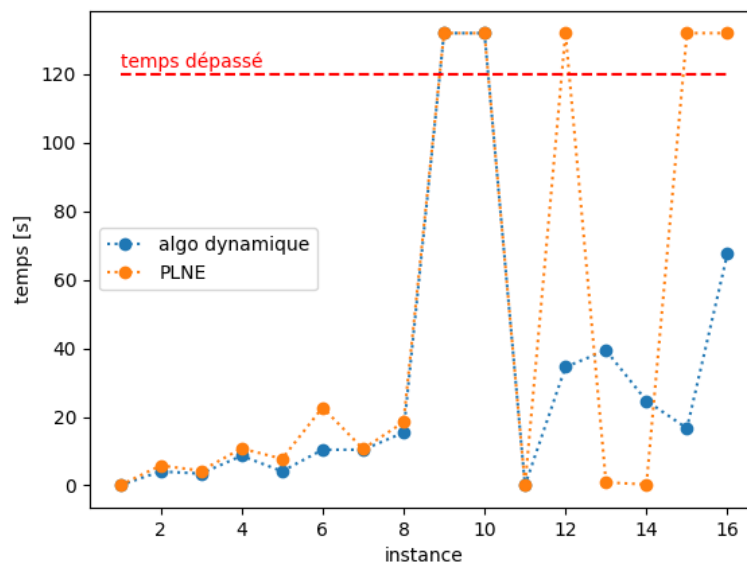
De manière analogue, le bloc doit finir strictement avant la case (i, j_f) avec

$$j_f = (N-1) - (k-l) - \sum_{t=l+1}^k s_t$$

Voici le tableau des temps de calcul avec la méthode PLNE (sans le timeout de 2 minutes) :

numéro d'instance	temps de résolution [s]
1	0.01
2	28.47
3	0.05
4	29.45
5	9.54
6	756
7	0.27
8	1.36
9	?
10	5.64
12	216.41
13	0.88
14	0.26
15	176.52
16	536.6

En employant l'algorithme dynamique sur les instances *12.txt* - *16.txt*, on a pu observer qu'aucune d'instances n'a pas été complètement résolue. La comparaison des temps d'exécution (cette fois-ci avec un timeout de 2 minutes) se lit dans un graphe ci-dessous. Les deux grilles obtenues pour l'instance *15.txt* ainsi que toutes les grilles autres obtenues pour différentes méthodes sont fournies dans l'annexe.



3 Pour aller plus loin

Dans les sections précédentes, on a pu observer qu'en général, l'algorithme dynamique semble d'être plus efficace en termes de la complexité temporelle que celui du PLNE. Pourtant, l'algorithme dynamique n'assure pas la résolution complète d'une grille. On a alors codé une méthode globale proposée dans le sujet - on a appliqué d'abord l'algorithme dynamique sur la grille. Puis, on a utilisé le PLNE de la section précédente sur les cases non déterminées. Pour exprimer que la case (i, j) est déjà coloriée, on a ajouté une contrainte de la forme

$$x_{ij} = 0, \text{ resp. } x_{ij} = 1$$

Conclusion

Annexe