Un problème de tomographie discrète

MOGPL

Master d'informatique M1

Université Pierre et Marie Curie

Baptiste JARRY
Magdaléna TYDRICHOVA

10 décembre 2017

Introduction

Dans le 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 instances 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 teste et on compare les solutions avec celles de la méthode précédente.

Dans la dernière section du projet, une brève analyse 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 les k premiers blocs (i.e. la séquence entière).

Q2

1. Si $l = 0$ et $j \in \{0, \ldots, m-1\}$, alors $T(j,l) = 0$. En effet, on peut "colorier" n'importe quel nombre de cases avec zéro 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 ne peut donc pas colorier les $j+1$ premières cases avec le bloc s_l, et on ne peut donc pas non plus les colorier avec la sous-séquence des blocs $(s_1, \ldots, s_l)$.
   (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 $T(j,1) = TRUE$ 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) sera blanche après la coloration : dans ce cas on aura $T(j,l) = T(j-1,l)$.

— 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 $j \in \{0, \ldots, m-1\}$: On peut colorier les $j+1$ premières cases avec zéro bloc si aucune des cases $0, \ldots, j$ n'est pas déjà coloriée en noir (ce qui impliquerait la présence d'un bloc ou au moins d'une partie d'un bloc).

2. On suppose maintenant $l \geq 1$.
   (a) $j < s_l - 1 \Rightarrow T(j,l) = FALSE$ pour la 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 en 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, il faut que

la case $(i, j+1)$ ne soit pas en noir (si elle existe). De plus, il ne faut aucune case blanche jusqu'à la case $(i, j - s_l + 1)$; en outre, la case $(i, j - s_l)$ qui précède le bloc ne doit pas etre coloriée en noir. $T(j, l) = TRUE$ si toutes ces conditions sont vérifiées et si on a également $T(j - s_l, l - 1) = TRUE$.

— Soit le l-ième bloc ne finit pas sur la case (i, j). Dans ce cas, on a $T(j, l) = TRUE$ si et seulement si on a $T(j - 1, l) = TRUE$ et si la case (i, j) n'est pas déjà noire (auquel cas on ne pourrait pas colorier les $j + 1$ premières cases uniquement avec les l premiers blocs, car 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 les $j + 1$ premières cases uniquement avec ces l blocs, et on aura donc $T(j, l) = FALSE$.

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 $s_{l+1}, \ldots s_k$, on en déduit que $T(j, l) = FALSE$.

Q8

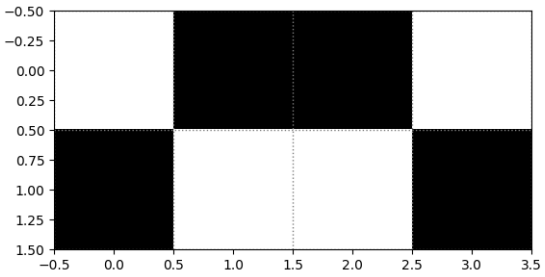
Voici le tableau des temps de résolution pour les instances 1-10 :

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 qu'aucune case n'a pu être coloriée. Si on essaie de résoudre cette instance "de tête", on voit que pour satisfaire les conditions il y a exactement une case noire dans la colonne, et que les blocs sont séparés par au moins une case blanche, il y a une unique solution qui est la suivante :



Pourtant, l'algorithme dynamique ne regarde qu'une seule ligne ou colonne en même temps (il ne regarde pas ce qui se passe "dans le futur"). Pour cette raison, d'après lui, chacune des cases peut être coloriée en noir ou en blanc au début. Il ne peut donc rien colorier dans la figure.

2 La PLNE

2.1 Modélisation

Q10

Soit l_i la i-ième ligne avec une séquence associée $(s_1, \ldots, s_k)$. 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 $(s'_1, \ldots, s'_{k'})$:

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

Q11

Avec les notations de la question précédente, on souhaite 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 :

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

De manière analogue, on obtient pour les colonnes :

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

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 le bon nombre de cases noires par ligne.
(4) Il doit y avoir le 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 demarre 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 demarre 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 reglé dans la question 13 en limitant le nombre de variables.

On en déduit alors le PLNE suivant :

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

$$\begin{aligned}
\sum_{j=0}^{N-1} y_{ij} &= 1 \quad \forall i \in \{0,\dots,M-1\} && (1)\\
\sum_{i=0}^{M-1} z_{ij} &= 1 \quad \forall j \in \{0,\dots,N-1\} && (2)\\
\sum_{j=0}^{N-1} x_{ij} - \sum_{l=1}^{k} s_l &= 0 \quad \forall i \in \{0,\dots,M-1\} && (3)\\
\sum_{i=0}^{M-1} x_{ij} - \sum_{l=1}^{k'} s_l &= 0 \quad \forall j \in \{0,\dots,N-1\} && (4)\\
y_{ij}^t &\leq \sum_{l=j+s_t+1}^{N-1} y_{il}^{t+1}, t \in \{1,\dots,k-1\}, i \in \{0,\dots,M-1\} && (5)\\
z_{ij}^t &\leq \sum_{l=j+s_t+1}^{M-1} z_{lj}^{t+1}, t \in \{1,\dots,k'-1\}, j \in \{0,\dots,N-1\} && (6)\\
y_{ij}^t &\leq \frac{\sum_{l=j}^{j+s_t-1} x_{il}}{s_t}, i \in \{0,\dots,M-1\}, t \in \{1,\dots,k-1\} && (7)\\
z_{ij}^t &\leq \frac{\sum_{l=i}^{i+s'_t-1} x_{lj}}{s_t}, j \in \{0,\dots,N-1\}, t \in \{1,\dots,k'-1\} && (8)
\end{aligned}$$

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 separés entre eux par exactement une case.

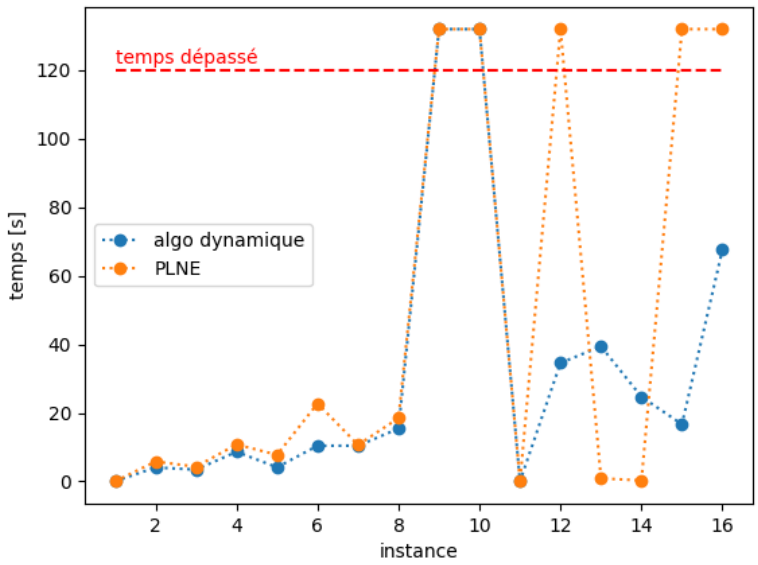
De manière analogue, le bloc doit finir strictement avant la case (i, j_f) avec :

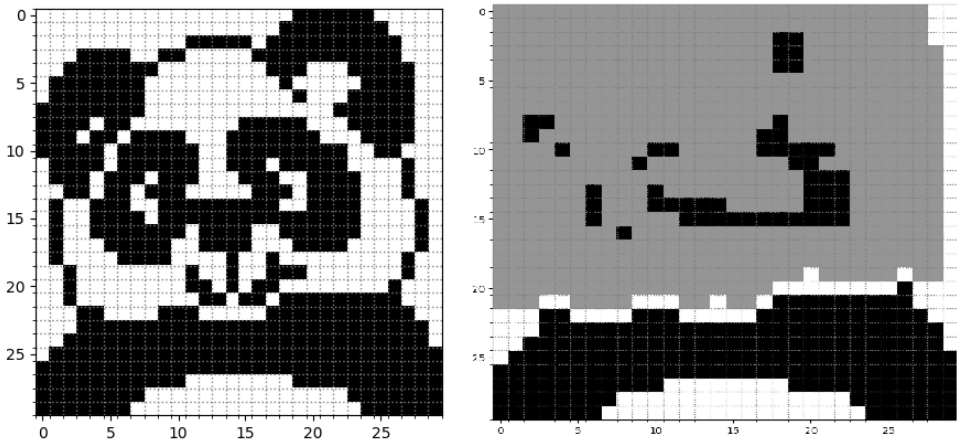
$$j_f = (N-1) - (k-l) - \sum_{t=l+1}^{k} s_t$$

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*, nous avons observé qu'aucune des instances n'a été complétement résolue. La comparaison des temps d'exécution (cette fois-ci avec un timeout de 2 minutes) se lit dans le graphe ci-dessous. Les deux grilles obtenues pour l'instance *15.txt* sont fournies ci-dessous.

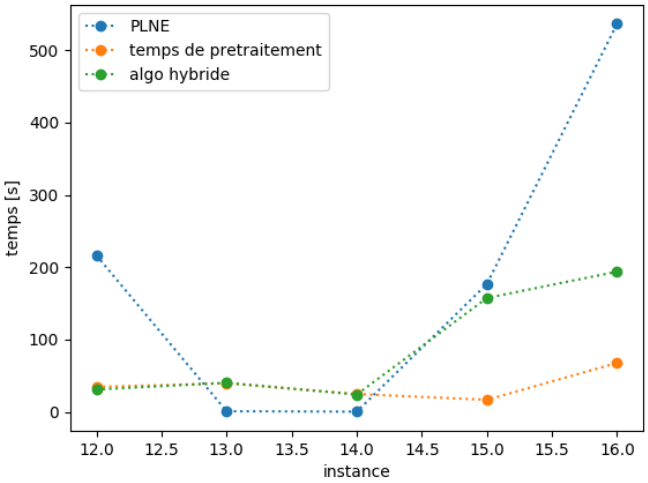




3 Pour aller plus loin

Dans les sections précédentes, on a pu observer qu'en général, l'algorithme dynamique semble 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 d'abord appliqué 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 fixe la borne inférieure de la variable x_{ij} à 1 si la case était noire, ou bien la borne supérieure à 0 si la case était blanche.

Dans le graphique suivant, on donne les temps d'exécution en employant l'algorithme hybride ou bien le PLNE. Pour avoir une idée sur l'impact de l'étape de prétraitement sur la vitesse de résolution par le PLNE, on trace également la durée de l'étape du pretraitement. Pour bien voir les différences temporelles entre algorithmes, on ne considère pas le timeout de 2 minutes :



On peut constater qu'en général, la durée de l'étape du PLNE est plus courte si on fait l'étape de prétraitement. Cette tendance est logique parce que l'algorithme *Branch & Bound* utilisé pour la résolution du PLNE est de complexité exponentielle. Fixer la valeur de certaines variables peut considérablement réduire l'arbre de recherche. Pourtant, on voit que sur les instances 13 et 14, l'algorithme hybride est moins efficace que la résolution par PLNE pure. Ceci est dû au fait que la durée d'exécution du PLNE est négligeable devant la durée de l'étape du prétraitement.

Conclusion

On a codé plusieurs méthodes de résolution d'un problème de tomographie discrète.

On a vu pas mal d'instances résolvables complètement par l'algorithme dynamique. En revanche, nous avons constaté que cette méthode n'est pas suffisante pour la résolution des grilles plus complexes. On peut en conclure qu'il n'est pas souhaitable d'utiliser la méthode dynamique pure.

Puis, on a pu comparer expérimentalement l'algorithme de PLNE avec l'algorithme hybride (avec l'étape de pretraitement). Sur la plupart des grilles complexes, l'algorithme hybride avait de meilleures performances. Sur les grilles plus simples (grilles *1.txt* - *10.txt*), l'algorithme hybride ne diffère pas de l'algorithme dynamique pur - l'étape de prétraitement est suffisante pour résoudre la grille. Etant donné que sur cette classe d'instances l'algorithme dynamique performait légèrement mieux que l'algorithme de PLNE, il semble être souhaitable de préférer l'algorithme hybride à celui de PLNE.

Pourtant, il y a aussi plusieurs instances sur lesquelles l'algorithme de PLNE

fonctionne beaucoup mieux que l'algorithme hybride. Idéalement, il faudrait être capable de classifier (facilement) les instances avant la résolution pour savoir s'il s'agit d'une instance facilement résolvable par la méthode dynamique, par la méthode de PLNE ou bien par l'algorithme dynamique.

Pour conclure, il nous semble souhaitable d'utiliser la méthode hybride qui donne en général des résultats solides. Il faudrait faire plus de tests et analyses sur d'autres instances pour avoir des conclusions plus précises.