

# RODD

Binôme 3 : Yuli Daune Funato, Thibaut Milhaud

Janvier-Mars 2021

## Table des matières

<b>1</b>	<b>TP 1 : Réserves tampons et centrales</b>	<b>2</b>
1.1	Problème à traiter . . . . .	2
1.2	Programme linéaire . . . . .	2
1.3	Instances à traiter . . . . .	3
1.4	Étude du comportement du modèle . . . . .	3
1.5	Avec l'espérance . . . . .	4
<b>2</b>	<b>Minimisation de la fragmentation</b>	<b>4</b>
2.1	Problème à traiter . . . . .	4
2.2	Programme linéaire . . . . .	4
2.3	Instances à traiter . . . . .	5
2.4	Étude du comportement du modèle . . . . .	6
<b>3</b>	<b>Protection de la diversité génétique</b>	<b>7</b>
3.1	Problème à traiter . . . . .	7
3.2	Programme linéaire . . . . .	7
3.3	Instances à traiter . . . . .	8
3.4	Étude du comportement du modèle . . . . .	9
<b>4</b>	<b>Exploitation durable de la forêt</b>	<b>10</b>
4.1	Problème à traiter . . . . .	10
4.2	Explication du PL du sujet . . . . .	10
4.3	Programme quadratique . . . . .	10
4.4	Linéarisation . . . . .	10
4.5	Instances à traiter . . . . .	11
4.6	Avec une contrainte sur le nombre des parcelles coupées . . . . .	12
4.7	Génération d'instances . . . . .	12

# Introduction

**Code** Tous nos fichiers, en particulier les .mod et les instances générées, sont disponibles à l'adresse suivante : <https://github.com/titiyeahti/rodd>.

## Matériel et logiciels utilisé pour les mesures

- processeur : Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz
- 16Go de RAM
- OS : Debian 10
- Version CPLEX : Version IBM ILOG CPLEX Optimization Studio 12.5

## 1 TP 1 : Réserves tampons et centrales

### 1.1 Problème à traiter

On dispose d'un terrain divisé en parcelles où habitent des espèces d'animaux. On veut déterminer un sous-ensemble de ces parcelles de coût minimal tel que sa préservation permet de garantir la survie de chaque espèce à un certain seuil. Les espèces sont divisées en espèces communes, qui peuvent survivre sur n'importe quelle parcelle protégée, et en espèces rares, dont la survie n'est possible que dans des parcelles centrales (i.e. entourées de parcelles protégées).

### 1.2 Programme linéaire

On a modélisé le problème par le PL suivant :

$$\min z = \sum_{i \leq m} \sum_{j \leq n} x_{ij}^p c_{ij}$$

s.c.

$$\begin{aligned} \sum_i x_{i1}^c &= 0 \\ \sum_i x_{in}^c &= 0 \\ \sum_j x_{1j}^c &= 0 \\ \sum_j x_{mj}^c &= 0 \\ \forall 2 \leq i \leq m-1, \forall 2 \leq j \leq n-1, 9x_{ij}^c &\leq \sum_{a=i-1}^{i+1} \sum_{b=j-1}^{j+1} x_{ab}^p \\ \forall 1 \leq k \leq p, \sum_{i \leq m} \sum_{j \leq n} \log(1 - p_{kij}) x_{ij}^c &\leq \log(1 - \alpha_k) \\ \forall p+1 \leq k \leq p+q, \sum_{i \leq m} \sum_{j \leq n} \log(1 - p_{kij}) x_{ij}^p &\leq \log(1 - \alpha_k) \end{aligned}$$

Les variables de décision sont :

- les  $x_{ij}^c$ , qui valent 1 ssi la parcelle ij est centrale
- les  $x_{ij}^p$ , qui valent 1 ssi la parcelle ij est protégée.

Les quatre premières contraintes expriment le fait qu’une parcelle sur un côté du terrain ne peut pas être centrale. La cinquième contrainte impose le fait qu’une centrale doit être entourée de parcelles protégées pour être centrale (et être elle-même protégée). Les deux dernières contraintes assurent que la probabilité de survie des espèces rares puis communes est suffisamment élevée.

### 1.3 Instances à traiter

Les sorties de CPLEX pour les quatre instances données dans l’énoncé sont disponibles dans les fichiers i1.res à i4.res.

Instance	Temps	Noeuds	Coût	Pr(1)	Pr(2)	Pr(3)	Pr(4)	Pr(5)	Pr(6)
1	0,01s	12	119	0.58	0.52	0.64	0.86	0.52	0.76
2	0,01s	5	327	0.91	0.91	0.92	0.98	0.89	0.98
3	0,02s	15	130	0.58	0.52	0.64	0.93	0.91	0.93
4	0,01s	9	211	0.82	0.81	0.82	0.97	0.78	0.88

TABLE 1 – Résultats sur les instance 1 à 4

### 1.4 Étude du comportement du modèle

L’objectif est ici d’étudier le comportement du modèle en fonction de la taille des instances. Les paramètres que nous avons fait varier sont la taille du terrain considéré et le nombre d’espèces.

**Instances de test** Le jeu d’instances que nous avons testé à été construit de la façon suivante :

- $n \in \{10, 12, \dots, 40\}$  ;
- $n = m$ , terrain carré ;
- $p \in \{2, 3, 4\}$  ;
- $p = q$ , autant d’espèces en danger que d’espèces communes.

Les paramètres  $c$ ,  $proba$  et  $\alpha$  ont été tirés selon des lois uniformes pour chaque instances.

- $c_{ij} \sim U(2, 10)$  ;
- $\alpha_i \sim U(0.4, 0.6)$  ;
- $proba_{kij} \sim U(0, 0.6)$ .

Afin de limiter l’impact de de l’aléatoire dans nos mesures, nous avons généré 9 instances différentes pour chaque couple  $(n, p)$ . Au total, on obtient donc 432 instances.

**Description de l’expérience** Nous avons ensuite lancé la résolution de ces instances en mesurant le temps de résolution.

**Résultats** Pour les petites instances,  $n < 20$ , le temps de résolution est raisonnable et croit avec  $n$  de façon régulière. Ce comportement n’est plus observé pour les instances plus grande surtout lorsque  $p > 2$ . En effet pour certaines  $(n, p) = (26, 4)$  certaines instances ont été résolues en moins de 2 secondes tandis qu’il à fallu plus de 1000 secondes pour d’autres. Ces grands écarts de temps de résolution nous ont empêché d’arriver au bout des instances de tests.

Cela dit, on peut observer sur la figure1 que le temps d’exécution augmente fortement lorsque  $n$  augmente. Mais aussi que plus  $n$  est grand plus le temps d’exécution peut être variable (on observe des différences de plus de 4 ordre de grandeur entre 2 instances pour les mêmes  $n, p$ ). Il semblerait également qu’augmenter  $p$  augmente à la fois le temps de résolution moyen et la disparité des performances du modèle pour un  $n$  donné.

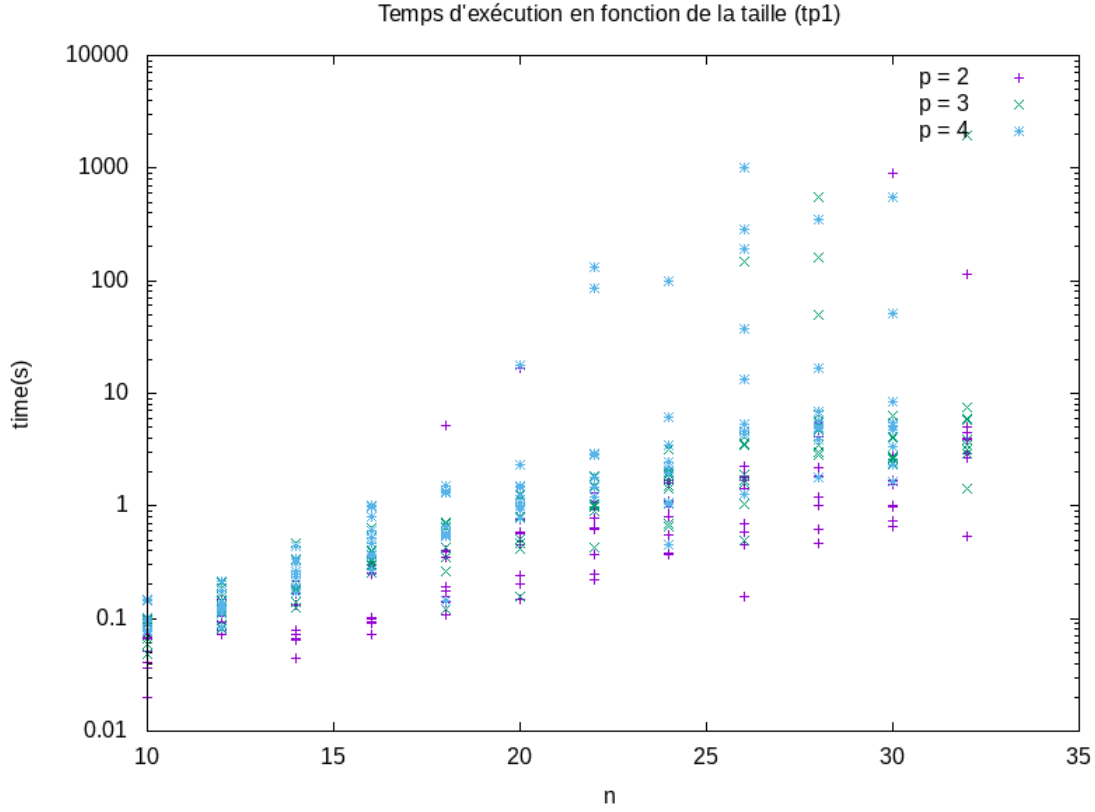


FIGURE 1 – Mesure de performance du modèle utilisé pour le tp1

### 1.5 Avec l'espérance

La fonction objectif devient :  $\max z = \sum_{i=1}^{p+q} \gamma(i)$ , où  $\gamma(i)$  est la probabilité que  $i$  espèces exactement survivent. On remplace les contraintes de probabilité dans la modélisation précédente par  $\sum_{ij} c_{ij} x_{ij}^p \leq B$ , avec  $B$  le budget.

On a bien une formulation en variables 0-1, mais l'objectif n'est pas du tout linéaire en les  $x_{ij}^p$ , car la probabilité que l'espèce  $k$  survive est  $(1 - \prod_{ij} 1 - p_{kij}) \times \prod_{ij} \prod_{k' \neq k} (1 - p_{k'ij})$ . Nous n'avons pas trouvé de linéarisation.

## 2 Minimisation de la fragmentation

### 2.1 Problème à traiter

On souhaite minimiser la fragmentation du paysage, i.e. l'éclatement de grandes superficies en petites parcelles. On choisit comme indicateur la distance moyenne au plus proche voisin (DMPPV), sur un ensemble de parcelles réparties sur un territoire d'aire  $A$ . On cherche à sélectionner un ensemble de parcelles de coût et d'aire raisonnables minimisant cette distance.

### 2.2 Programme linéaire

$$\min f(x, y) - \lambda g(x, y)$$

S.C.

$$\begin{aligned}
\sum_{i \leq m} \sum_j j \leq n x_{ij} &\leq A_{max} \\
\sum_{i \leq m} \sum_{j \leq n} x_{ij} &\geq A_{min} \\
\sum_{i \leq m} \sum_{j \leq n} 10 x_{ij} c_{ij} &\leq B \\
\forall 1 \leq i1, i2 \leq m, \forall 1 \leq j1, j2 \leq n, &y_{i1j1i2j2} \leq x_{i2j2} \\
\sum_{i \leq m} \sum_{j \leq n} y_{ijij} &= 0 \\
\forall 1 \leq i \leq m, \forall 1 \leq j \leq n, \sum_{i2 \leq m} \sum_{j2 \leq n} &y_{i1j1i2j2} \leq x_{i1j1}
\end{aligned}$$

Les variables de décision binaires sont :

- les  $x_{ij}$ , qui valent 1 ssi la parcelle  $ij$  est choisie.
- les  $y_{i1j1i2j2}$ , qui valent 1 ssi la parcelle  $i2j2$  est le voisin le plus proche de la parcelle  $i1j1$ .

Les trois premières contraintes imposent une aire et un budgt raisonnable, la quatrième contrainte impose que le voisin le plus proche d'une parcelle doit être une parcelle choisie, la cinquième qu'une parcelle ne peut pas être son plus proche voisin. La sixième contrainte impose qu'on ne peut avoir qu'un seul plus proche voisin (et qu'une parcelle avec un plus proche voisin est sélectionnée).

## 2.3 Instances à traiter

**Instance 1**  $A_{min} = 30, A_{max} = 35, B = 920$

temps de calcul : 0.65 s  
 2 itérations  
 f=34.66  
 g=30  
 DMPPV = 1.155  
 v=0

x =  
 0 1 0 0 1 0 0 1 0 0  
 0 0 0 0 1 0 0 1 0 0  
 0 1 1 0 1 1 1 0 0 0  
 0 1 0 0 1 0 0 0 0 0  
 1 1 1 1 0 0 1 0 0 1  
 0 0 1 0 0 0 0 0 0 0  
 0 1 1 0 0 0 0 0 0 1  
 0 1 0 1 1 0 0 0 1 1  
 0 0 0 1 0 0 0 0 0 0  
 0 0 0 0 1 0 0 1 0 0

**Instance 2**  $A_{min} = 20, A_{max} = 21, B = 520$

temps de calcul : 0.6 s  
 2 itérations  
 f= 25.47870866461908  
 g= 20

DMPPV = 1.273

v= 0

x=

```
0 1 0 0 1 0 0 1 0 0
0 0 0 0 1 0 0 1 0 0
0 1 0 0 1 1 0 0 0 0
0 1 0 0 0 0 0 0 0 0
1 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 1
0 1 0 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 0
```

**Instance 3**  $A_{\min} = 70, A_{\max} = 75, B = 3500$

temps de calcul : 0.5 s

2 itérations

f=70

g=70

DMPPV = 1

v=0

x=

```
0 1 1 0 1 1 1 1 1 1
0 0 0 1 1 1 1 1 0 0
1 1 1 1 1 1 1 0 1 0
1 1 1 1 1 1 1 0 1 1
1 1 1 1 0 1 1 0 1 1
0 1 1 0 0 0 1 1 0 0
1 1 1 1 0 0 1 0 1 1
1 1 0 1 1 1 1 0 1 1
0 1 1 1 1 1 1 0 0 1
1 1 0 0 1 1 1 1 0 0
```

## 2.4 Étude du comportement du modèle

L'objectif est ici d'étudier le comportement du modèle en fonction de la taille des instances. Ici le seul paramètre que nous avons fait varier est la taille de la parcelle considérée.

**Instances de test** Nous avons fait varier  $n$  entre 10 et 40 par pas de 2. Nous considérons encore une fois des parcelles carrés ( $n = m$ ) et nous avons généré 10 instances différentes pour chaque  $n$ .

Le coût de chaque case (matrice  $C$ ) est tiré uniformément entre 2 et 10. Donc

$$\forall(i, j), \mathbb{E}(c_{ij}) = \mathbb{E}(c_{1,1}) = 6$$

Enfin, pour un  $n$  donné, les paramètres suivant sont choisis de façon déterministe et indépendamment de l'instance :

- $A_{\min} = \frac{2n^2}{5}$  ;
- $A_{\max} = \frac{n^2}{2}$  ;
- $B = 10 \times \mathbb{E}(c_{1,1}) \times \frac{4}{5} A_{\min} = 48 \times A_{\min}$ .

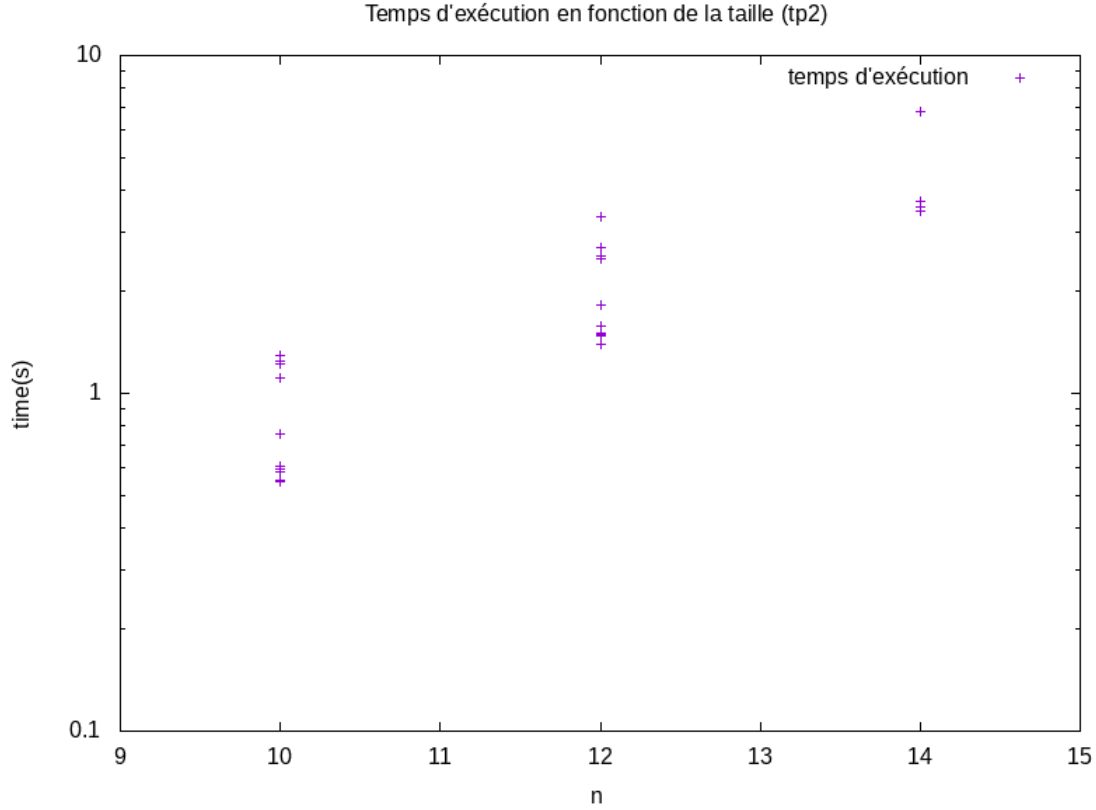


FIGURE 2 – Mesure de performance du modèle utilisé pour le tp2

**Résultats** Là encore, il y avait de grandes disparités au niveau des temps de résolution entre deux instances de même taille à tel point que CPLEX a bloqué indéfiniment (nous avons arrêté le programme au bout de quelques heures) sur une instance de taille ( $n = 14$ ) mettant ainsi fin à la campagne de test.

Le peu de points présent sur la figure2 nous renseigne quand même légèrement sur le comportement du modèle :

- la complexité du problème semble exponentielle en  $n$  (comme attendu) ;
- beaucoup de disparité entre des instances de même taille.

### 3 Protection de la diversité génétique

#### 3.1 Problème à traiter

On veut déterminer la contribution optimale de chaque individu d'une population pour minimiser la perte d'allèles. On cherche donc à minimiser l'espérance du nombre d'allèles disparus d'une génération à la suivante, en maintenant une population constante.

#### 3.2 Programme linéaire

$$\min z = \sum_{j \leq G} \sum_{j \leq A} p_{jk}$$

S.C.

$$\begin{aligned}
& \forall i, x_i \leq 3 \\
& \forall i, x_i \geq 0 \\
& \forall c, \forall g, \forall a, p_{jk} \geq t_{jk} - \sum_{ind_{i1j1}=ind_{i2j2}=a} x_i \\
& \forall 1 \leq r \leq T, \forall j, \forall k, \ln(\theta_r) + \frac{t_{jk} - \theta_r}{\theta_r} \geq \sum_{ind_{i1j1} \neq ind_{i1j2}} -x_i \times \ln(2) \\
& \forall j, k, t_{jk} \geq 0 \wedge p_{jk} \geq 0
\end{aligned}$$

Les variables de décision sont :

- les  $x_i$  : nombre d'enfants de i.
- les  $t_{jk}$  : probabilité que l'allèle k sur le gène j disparaisse
- les  $z_{jk}$  : variables permettant de modéliser tous les tuples possibles

### 3.3 Instances à traiter

#### Instance 1 : 3 enfants ou moins

temps de calcul : 0.01 s

Enfants pour chaque individus

1 3 3 1 3 0 3 2

Proba de disparitions des allèles

gène: 1

0 0

gène: 2

0 0.015588

gène: 3

0 0

gène: 4

0 0

gène: 5

0 0

#### Instance 2 : 2 précisément 2 enfants

temps de calcul : 0.01 s

Enfants pour chaque individus

2 2 2 2 2 2 2 2

Proba de disparitions des allèles

gène: 1

0 0

gène: 2

0 0.06243336502896749

gène: 3

0 0



gène: 4  
0 0  
gène: 5  
0 0

### 3.4 Étude du comportement du modèle

**Instances de tests** Ici les paramètres que nous avons fait varier sont le nombre d'individus ( $N = N_f + N_m$  avec  $N_m = N_f$ ), le nombre de gènes considérés ( $G$ ), le nombre d'allèles par gène ( $A$ ) et le nombre de pas pour l'approximation du logarithme ( $T$ ).

- $4 \leq N_f \leq 14$ ;
- $2 \leq A \leq 4$ ;
- $5 \leq G \leq 8$ ;
- $T \in \{50 + 100i, i \in \{0 \dots 5\}\}$

Étant donné le nombre de paramètres que nous faisons varier et le nombre important d'instances que cela induit ( $11 \times 3 \times 4 \times 6 = 792$ ), nous nous sommes pour cette fois contents d'une seule instance par quadruplé ( $N_f, G, A, T$ ).

Le côté aléatoire des instances porte ici sur la répartition des allèles de chaque gène pour chaque individu.

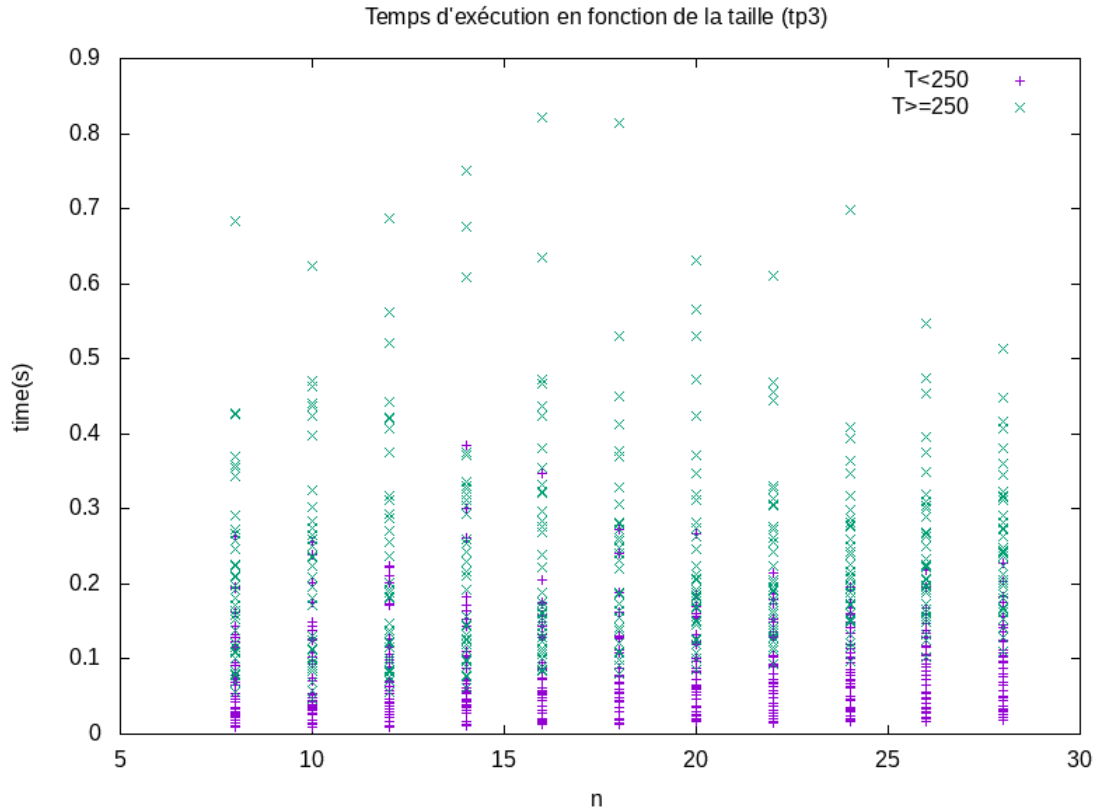


FIGURE 3 – Mesure de performance du modèle pour le tp3

**Résultats** Sur la figure3, on ne voit pas une augmentation majeure du temps d'exécution lorsque  $N$  augmente, du moins sur l'intervalle considéré ( $N \in [8, 28]$ ). Il n'apparaît pas non plus de corrélation

évidente entre la rapidité d'exécution et les valeurs de  $G$  et  $A$ .

Sur les tests que nous avons effectués, le paramètre qui influe de manière notable sur le temps d'exécution est  $T$  qui définit le nombre de contraintes liées à l'approximation linéaire du logarithme. Nous avons tenté d'illustrer cela sur la figure 3 :

- en vert les instances pour lesquelles  $T \geq 250$  ;
- en violet celles telles que  $T < 250$ .

Pour observer l'impact du nombre d'individus et de gènes/allèles considérés il faudrait sûrement passer à l'ordre de grandeur suivant.

Une autre explication est que le fait de tirer les allèles de manière uniforme sur une population de plus en plus grande tend vers une situation où il est facile de garantir qu'aucun des allèles initiaux ne disparaîtra à la génération suivante.

## 4 Exploitation durable de la forêt

### 4.1 Problème à traiter

On considère des parcelles forestières et deux espèces y habitant. La première vit dans les parcelles coupées, l'autre dans les non coupées. On veut déterminer quelles parcelles couper de façon à ce que la somme pondérée des populations des deux espèces soit maximale.

### 4.2 Explication du PL du sujet

Les  $x_{ij}$  sont des v.d. qui valent 1 si  $s_{ij}$  est coupée. Les  $d_{ij}$  sont des variables intermédiaires correspondant au nombre de parcelles non coupées autour d'une parcelle. En effet, quand  $s_{ij}$  est coupée,  $x_{ij}$  est nul, donc  $d_{ij} = 0$  aussi (d est toujours aussi petit qu'il a le droit de l'être à cause de la fonction objectif). Quand  $s_{ij}$  est non coupée, le terme  $|A_{ij}|(1 - x_{ij})$  est nul donc  $d_{ij}$  est le nombre de parcelles non coupées adjacentes à  $s_{ij}$ .

### 4.3 Programme quadratique

$$\max z = w_1 \sum_{(i,j) \in M \times N} t_{ij}(1 - x_{ij}) + w_2 gl \sum_{(i,j) \in M \times N} \sum_{(k,l) \in A_{i,j}} x_{ij}(1 - x_{kl})$$

### 4.4 Linéarisation

$$\max z = w_1 \sum_{(i,j) \in M \times N} t_{ij}(1 - x_{ij}) + w_2 gl \sum_{(i,j) \in M \times N} \sum_{(k,l) \in A_{i,j}} (x_{ij} - y_{ijkl})$$

s.c.

$$\begin{aligned} \forall i, j, \forall (k, l) \in A_{ij}, y_{ijkl} &\geq x_i + x_j - 1 \\ \forall i, j, \forall (k, l) \in A_{ij}, y_{ijkl} &\geq 0 \end{aligned}$$

La première contrainte modélise la multiplication sur  $\{0,1\}$ . On remarque que la matrice des contraintes est TU. En effet, il s'agit de la matrice d'incidence d'un graphe biparti, qui est TU. On peut s'en convaincre en considérant la grille de parcelles comme un échiquier : tous les voisins d'une case blanche (resp. noires) sont des cases noires (resp. blanches).

## 4.5 Instances à traiter

### Instance 1 $n = 10$

Modèle Mixte :

temps de calcul : 0.033 s

Plan de coupe :

```
0 1 0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 1 0 0
0 0 1 0 0 1 0 0 0 0
0 1 0 0 1 0 0 0 0 0
1 0 1 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 1
0 1 0 0 1 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 0
```

Fonction économique: 8219.57819

Modèle TU :

temps de calcul : 0.003 s

Plan de coupe :

```
0 1 0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 1 0 0
0 0 1 0 0 1 0 0 0 0
0 1 0 0 1 0 0 0 0 0
1 0 1 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 1
0 1 0 0 1 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 0
```

Fonction économique: 8219.57819

### Instance 2 $n = 5$

Modèle Mixte :

temps de calcul : 0.007 s

Plan de coupe :

```
0 0 0 1 0
0 0 0 1 0
0 0 1 0 0
1 0 0 0 0
1 0 0 0 0
```

Fonction économique: 442.55536

Modèle TU :

temps de calcul : 0.0015 s

Plan de coupe :

```
0 0 0 1 0
0 0 0 1 0
```

```

0 0 1 0 0
1 0 0 0 0
1 0 0 0 0

```

Fonction économique: 442.55536

**Instance 3 (comme la 1 mais il faut laisser 60 non coupées au minimum)**

Modèle mixte:

temps de calcul : 0.03 s

Plan de coupe :

```

1 1 0 1 1 1 1 1 1 1
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 1
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 1 1 1 1 1

```

Fonction économique : 6753.93

Modèle TU (enfin plus maintenant -> résolution PLNE)

temps de calcul : 0.015 s

Plan de coupe :

```

1 1 0 1 1 1 1 1 1 1
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
1 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 1
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 1 1 1 1 1

```

Fonction économique: 6753.93

## 4.6 Avec une contrainte sur le nombre des parcelles coupées

On ajoute la contrainte suivante :

$$\sum_{(i,j) \in M \times N} x_{ij} \geq 60$$

. La matrice des contraintes perd son caractère TU.

## 4.7 Génération d'instances

**Instances de test** Nous avons généré les instances de tests en faisant varier la  $n$  c'est à dire la taille de la forêt entre 10 et 95 par pas de 5.

Ici encore nous avons considéré des forêts carrés et nous avons conservée les valeurs données dans le sujet pour la plupart des paramètres :

- $w_1 = 1$
- $w_2 = 5$
- $L = 3$
- $g = 1.26$

Enfin pour que les instances varient, chaque coordonnée de la matrice  $t$  a été tirée uniformément entre 60 et 100. Pour chaque  $n$  considéré, nous avons généré 10 instances.

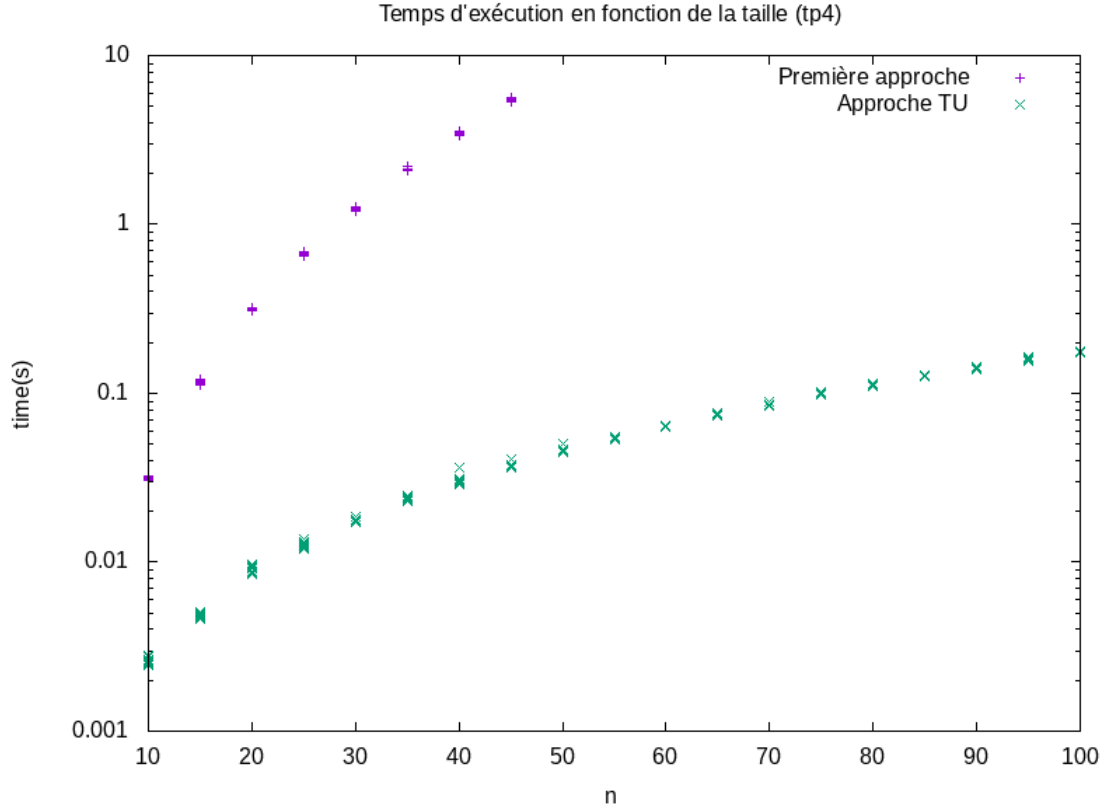


FIGURE 4 – Mesure de performance pour des modèles du tp4

**Résultats** Dans ce dernier TP, nous avons deux approches à étudier, la première en variables mixtes (en violet sur fig4) et la seconde avec une matrice des contraintes TU et donc en avec des variables réelles (en vert).

On observe bien sur la figure4 que la seconde approche est polynomiale en  $n$  ( $n$  défini seul la taille de l'instance).

Enfin, nous n'avons pas pu traiter les instances de taille  $n > 45$  avec la première méthode pour cause de manque de mémoire (CPLEX saturait les 16Go de RAM à notre disposition avant de trouver une réponse).