



**INSTITUT
POLYTECHNIQUE
DE PARIS**

LE CNAM
INSTITUT POLYTECHNIQUE DE PARIS

Projet RODD

Auteur :
Justine DE SOUSA
Vincent MARON

Superviseurs:
Amélie LAMBERT

le cnam

27 mars 2022

Table des matières

1	Sélection de réserves naturelles	1
1.1	Modélisation	1
1.2	Résolution numérique	2
1.3	Modèle alternatif	3
2	Maîtrise des effets néfastes engendrés par la fragmentation du paysage	5
2.1	Modélisation en variables entières	5
2.2	Modélisation du programme d'optimisation fractionnaire	6
2.3	Résolution numérique	7
3	Protection de la diversité génétique	9
3.1	Modélisation du problème	9
3.2	Approximation du logarithme par une fonction linéaire par morceaux	10
3.3	Résolution numérique	11
4	Exploitation durable de la forêt	15
4.1	Formulation par un programme linéaire en variables mixtes	15
4.2	Formulation par un programme quadratique en variables 01	15
4.3	Linéarisation TU	16
4.3.1	Linéarisation	16
4.3.2	Matrice des contraintes TU	16
4.4	Résolution numérique	16
4.4.1	Résultats sur les deux instances données	16
4.5	Ajout d'une contrainte	17
4.6	Génération d'instances aléatoires	17
	Annexe Projet 1	19

Projet 1

Sélection de réserves naturelles

1.1 Modélisation

Variables

- $x_i \in \{0, 1\} \quad \forall i \in \mathcal{N}$ vaut 1 si la parcelle i est protégée
- $y_i \in \{0, 1\} \quad \forall i \in \mathcal{N}$ vaut 1 si la parcelle i est centrale

Formulation en variables 01

$$\left\{ \begin{array}{ll} \min & \sum_{i \in \mathcal{N}} c_i \cdot x_i \\ \text{s.c. :} & \\ & p_k \geq \alpha_k \quad \forall k \in P \quad (\text{Probabilité de présence}) \\ & y_i \leq x_k \quad \forall i \in \mathcal{N}, k \in V(i) \quad (s_i \text{ est centrale si ses parcelles voisines sont protégées}) \\ & y_i \geq \sum_{k \in V(i)} x_k - |V(i)| + 1 \quad \forall i \in \mathcal{N} \quad (\text{Une case dont toutes les voisines sont protégées est centrale}) \\ & y_i = 0 \quad \forall i \in \mathcal{N} : i \text{ est sur le bord} \quad (\text{Les parcelles du bord ne peuvent être centrales}) \\ & x_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \\ & y_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \end{array} \right.$$

Probabilité de présence p_k dénote la probabilité de présence de l'espèce k dans la réserve. Elle s'écrit de la façon suivante :

Soit $k \in P$,

$$\begin{aligned} p_k &= 1 - \mathbb{P}(k \text{ non présente}) = 1 - \prod_{i \in \mathcal{N}} \mathbb{P}(k \text{ non présente dans la parcelle } i) \\ &= 1 - \prod_{i \in \mathcal{N}} (1 - \mathbb{P}(k \text{ présente dans la parcelle } i)) \end{aligned}$$

Or,

$$\mathbb{P}(k \text{ présente dans la parcelle } i) = \begin{cases} p_{ki} \cdot y_i & \text{si l'espèce } k \text{ est rare} \\ p_{ki} \cdot x_i & \text{si l'espèce } k \text{ est commune} \end{cases}$$

La contrainte de probabilité de présence peut alors s'écrire :

$$\left\{ \begin{array}{ll} 1 - \prod_{i \in \mathcal{N}} (1 - p_{ki} \cdot y_i) \geq \alpha_k & \forall k \in P : k \text{ rare} \\ 1 - \prod_{i \in \mathcal{N}} (1 - p_{ki} \cdot x_i) \geq \alpha_k & \forall k \in P : k \text{ commune} \end{array} \right.$$

Afin de rendre ces contraintes linéaires, on applique la fonction croissante \log aux deux inégalités, on obtient :

$$\begin{cases} \log(1 - \alpha_k) \geq \sum_{i \in \mathcal{N}} \log(1 - p_{ki} \cdot y_i) & \forall k \in P : k \text{ rare} \\ \log(1 - \alpha_k) \geq \sum_{i \in \mathcal{N}} \log(1 - p_{ki} \cdot x_i) & \forall k \in P : k \text{ commune} \end{cases}$$

On remarque que

$$\begin{cases} x_i = 0 \Rightarrow \log(1 - p_{ki} \cdot x_i) = 0 & = x_i \log(1 - p_{ki}) \\ x_i = 1 \Rightarrow \log(1 - p_{ki} \cdot x_i) = \log(1 - p_{ki}) & = x_i \log(1 - p_{ki}) \end{cases}$$

De même pour les y_i . Les contraintes deviennent alors

$$\begin{cases} \log(1 - \alpha_k) \geq \sum_{i \in \mathcal{N}} y_i \cdot \log(1 - p_{ki}) & \forall k \in P : k \text{ rare} \\ \log(1 - \alpha_k) \geq \sum_{i \in \mathcal{N}} x_i \cdot \log(1 - p_{ki}) & \forall k \in P : k \text{ commune} \end{cases}$$

Formulation linéaire 01 :

$$\left\{ \begin{array}{ll} \min & \sum_{i \in \mathcal{N}} c_i \cdot x_i \\ \text{s.c. :} & \\ & \log(1 - \alpha_k) \geq \sum_{i \in \mathcal{N}} y_i \cdot \log(1 - p_{ki}) \quad \forall k \in P : k \text{ rare} \quad (1) \\ & \log(1 - \alpha_k) \geq \sum_{i \in \mathcal{N}} x_i \cdot \log(1 - p_{ki}) \quad \forall k \in P : k \text{ commune} \quad (2) \\ & y_i \leq x_k \quad \forall i \in \mathcal{N}, k \in V(i) \\ & y_i \geq \sum_{k \in V(i)} x_k - |V(i)| + 1 \quad \forall i \in \mathcal{N} \\ & y_i = 0 \quad \forall i \in \mathcal{N} : i \text{ est sur le bord} \\ & x_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \\ & y_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \end{array} \right.$$

où les contraintes (1) et (2) correspondent à la probabilité de présence de l'espèce k .

1.2 Résolution numérique

Résultats sur les 4 instances données On teste le modèle obtenu sur les 4 instances données et on affiche les résultats 1.1. On obtient des solution réalisables qui correspondent aux solutions données dans l'énoncé.

TABLE 1.1 – Résultats obtenus sur les 4 instances données

Instance	Espèce	1	2	3	4	5	6	Temps(s)	Noeuds	Coût
1	α	0.5	0.5	0.5	0.5	0.5	0.5	0.13	0	119.0
	Probabilités de survie	0.58	0.52	0.7	0.93	0.64	0.5			
2	α	0.9	0.9	0.9	0.5	0.5	0.5	0.03	0	327.0
	Probabilités de survie	0.92	0.91	0.92	0.98	0.89	0.98			
3	α	0.5	0.5	0.5	0.9	0.9	0.9	0.21	0	130.0
	Probabilités de survie	0.58	0.52	0.64	0.93	0.91	0.93			
4	α	0.8	0.8	0.8	0.6	0.6	0.6	0.06	0	211.0
	Probabilités de survie	0.82	0.81	0.82	0.97	0.78	0.88			

Génération aléatoire d'instances Pour chacun des 4 vecteurs α utilisés précédemment, on génère aléatoirement des grilles de taille 10 à 50. Chacune des instance est générée de la façon suivante :

- Pour chaque espèce $k \in K$, on tire aléatoirement le nombre de parcelle ayant une probabilité non nulle de survie pour cette espèce.
- On tire aléatoirement un nombre entre 0.1 et 0.5 qui représente la probabilité de survie de l'espèce k sur cette parcelle.

Les résultats sur les instances générées aléatoirement sont présentés en annexes . Observe que pour la plupart des instances, il existe une solution réalisable et celle-ci est trouvée en temps raisonnables : de 0s pour les instances de taille 10×10 à quelques secondes pour les instances de taille 50. Les instances qui prennent le plus de temps sont pour $\alpha = (0.8, 0.8, 0.8, 0.6, 0.6, 0.6)$, ce qui paraît raisonnable puisque c'est ce α qui contraint le plus le problème et ce, à la fois pour les espèces communes et les espèces rares. Contrairement à $\alpha = (0.9, 0.9, 0.9, 0.5, 0.5, 0.5)$ et $\alpha = (0.5, 0.5, 0.5, 0.9, 0.9, 0.9)$ qui ne contraignent qu'un type d'espèce. On peut remarquer, néanmoins, que lorsque les espèces rares sont les plus contraintes, alors la valeur du coût augmente considérablement. Ce qui est également assez logique puisque pour protéger les espèces rares, il est nécessaire de protéger plus de parcelles afin d'avoir des parcelles centrales.

1.3 Modèle alternatif

On souhaite maximiser l'espérance du nombre d'espèces qui survivent sous une contrainte budgétaire. Pour cela, on introduit les variables aléatoires $X_k \sim \mathcal{B}(p_k)$ où p_k est la probabilité de présence de l'espèce k dans la réserve.

La variable aléatoire $X = \sum_{k=1}^K X_k$ compte alors le nombre d'espèces qui survivent dans la réserve. On a donc

$$\mathbb{E}[X] = \sum_{k=1}^K \mathbb{E}[X_k] = \sum_{k=1}^K p_k = \sum_{k=1}^K 1 - \prod_{i \in \mathcal{N}} \left(1 - p_{ki} \cdot \begin{cases} y_i & \text{si l'espèce est rare} \\ x_i & \text{si l'espèce est commune} \end{cases} \right)$$

Formulation alternative :

$$\left\{ \begin{array}{ll} \max & \sum_{k \in P: k \text{ rare}} 1 - \prod_{i \in \mathcal{N}} (1 - p_{ki} \cdot y_i) + \sum_{k \in P: k \text{ commune}} 1 - \prod_{i \in \mathcal{N}} (1 - p_{ki} \cdot x_i) \\ \text{s.c. :} & \\ & \sum_{i \in \mathcal{N}} c_i \cdot x_i \leq C \quad \text{(Contrainte budgétaire)} \\ & y_i \leq x_k \quad \forall i \in \mathcal{N}, k \in V(i) \\ & y_i \geq \sum_{k \in V(i)} x_k - |V(i)| + 1 \quad \forall i \in \mathcal{N} \\ & y_i = 0 \quad \forall i \in \mathcal{N} : i \text{ est sur le bord} \\ & x_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \\ & y_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \end{array} \right.$$

Programme linéaire? On pourrait faire une linéarisation classique en remplaçant chaque produit de variables

$\prod_{i \in S \subset \mathcal{N}} x_i$ par une variable $z_{i \in S}$ en ajoutant les contraintes suivantes :

- $z_{i_1, \dots, i_{|S|}} \leq x_i \quad \forall i \in S$

Or dans la formulation alternative, il existe $\sum_{k=1}^N \binom{N}{k} = 2^N$ produits de x_i . Cela introduirait donc un nombre exponentiel de variables. Le PLNE ne serait alors plus compact.

Projet 2

Maîtrise des effets néfastes engendrés par la fragmentation du paysage

2.1 Modélisation en variables entières

Variables

- $x_i = 1$ si on sélectionne la parcelle i
- $y_{i,j} = 1$ si la parcelle j est la plus proche voisine de i

Fonction objectif On cherche à minimiser $DMPPV = \frac{1}{n} \sum_{i=1}^n \min_{j \neq i} \{d_{ij}\}$

Où $n = \sum_{i=1}^N x_i$ est le nombre de parcelles sélectionnées et d_{ij} est la distance entre les parcelles i et j . Or cette distance est atteinte lorsque j est la plus proche voisine de i , c'est-à-dire lorsque $y_{ij} = 1$. On peut alors réécrire

$$DMPPV = \frac{1}{\sum_{i=1}^N x_i} \sum_{i=1}^N \sum_{j=1}^N d_{ij} y_{ij}$$

Formulation en variables 01

$$\left\{ \begin{array}{ll} \min & \frac{\sum_{i,j=1}^N d_{ij} y_{ij}}{\sum_{i=1}^N x_i} \\ \text{s.c. :} & \\ & A_{min} \leq \sum_{i=1}^N x_i \leq A_{max} \quad (\text{Aire totale des parcelles sélectionnées}) \\ & \sum_{i=1}^N c_i \cdot x_i \leq B \quad (\text{Coût total}) \\ & \sum_{j=1}^N y_{ij} = x_i \quad \forall i \in \{1, \dots, N\} \quad (\text{Chaque parcelle sélectionnée a exactement une plus proche voisine}) \\ & y_{ij} \leq x_j \quad \forall i, j \in \{1, \dots, N\} \quad (\text{Une parcelle non sélectionnée ne peut être plus proche voisine}) \\ & x_i, y_i \in \{0, 1\} \quad \forall i \in \{1, \dots, N\} \end{array} \right.$$

Notons X l'espace des solutions réalisables relâchées.

2.2 Modélisation du programme d'optimisation fractionnaire

On note f et g les fonctions linéaires suivantes :

- $f(x, y) = \sum_{i,j=1}^N d_{i,j} \cdot y_{i,j}$
- $g(x, y) = \sum_{i=1}^N x_i$

On a alors le programme d'optimisation fractionnaire suivant :

$$\begin{cases} \min \frac{f(x, y)}{g(x, y)} \\ \text{s.c. :} \\ (x, y) \in X \end{cases} \quad (P)$$

On a les propriétés suivantes :

- $f(x, y) = \sum_{i,j=1}^N d_{i,j} \cdot y_{i,j} \geq 0$ et $f(x, y) = 0 \Leftrightarrow y = 0 \Rightarrow x = 0 \Rightarrow \sum_{i=1}^N x_i = 0 < A_{min}$ donc $\boxed{f(x,y) > 0}$
- $g(x, y) \geq A_{min} > 0 \forall (x, y) \in X$

On considère le problème (différent de celui donné dans le sujet de projet) :

$$\begin{cases} \max \lambda \\ \text{s.c. :} \\ f(x, y) - \lambda g(x, y) \geq 0 \quad \forall (x, y) \in X \\ \lambda \in \mathbb{R} \end{cases} \quad (Q)$$

$(\lambda^*, x^*, y^*) \in \mathbb{R} \times X$ est une solution optimale de (Q) si et seulement si

$$\forall (x, y) \in X, f(x, y) - \lambda^* g(x, y) \geq f(x^*, y^*) - \lambda^* g(x^*, y^*) = 0$$

Ce qui est équivalent à

$$\forall (x, y) \in X, \frac{f(x, y)}{g(x, y)} \geq \lambda^* = \frac{f(x^*, y^*)}{g(x^*, y^*)}$$

Ce qui signifie que λ^* est la valeur optimale de (P) et que (x^*, y^*) est la solution optimale de (P). On va utiliser le problème paramétrique suivant :

$$\begin{cases} \min v(\lambda) = f(x, y) - \lambda g(x, y) \\ \text{s.c. :} \\ (x, y) \in X \end{cases} \quad (P_\lambda)$$

Soit $(x_\lambda^*, y_\lambda^*)$ une solution optimale de (P_λ) . On a $v(\lambda) = 0$ si et seulement si $(\lambda, x_\lambda^*, y_\lambda^*)$ est une solution optimale de (Q) si et seulement si $(x_\lambda^*, y_\lambda^*)$ est solution de (P) de valeur λ . Une nouvelle formulation du problème (P) est alors la suivante :

$$\boxed{\text{Trouver } \lambda \in \mathbb{R} \text{ tel que } \min \{f(x, y) - \lambda g(x, y) : (x, y) \in X\} = 0}$$

On a les propriétés suivantes :

- $v(\lambda)$ est strictement décroissante :
Soit $\lambda, h > 0$ et (x^*, y^*) tel que $v(\lambda) = f(x^*, y^*) - \lambda g(x^*, y^*)$.

$$v(\lambda + h) \leq f(x^*, y^*) - \lambda g(x^*, y^*) - \underbrace{h g(x^*, y^*)}_{>0} < v(\lambda)$$

- $v(0) > 0$ car $f(x, y) > 0$
- v a une seule racine
- v est concave

On en déduit l'algorithme de *Dinkelbach* donné dans l'énoncé dans lequel on remplace la maximisation par la minimisation.

2.3 Résolution numérique

Résultats sur les 3 instances données On affiche les résultats obtenus sur les 3 instances données Figure 2.5. Pour l'instance 2 (Figure 2.3), on obtient la même solution que dans l'énoncé mais pour les instances 1 et 3 (Figure 2.2, 2.4), on obtient une solution différente mais proche avec la même valeur de DMPPV et autant de parcelles protégées pour l'instance 1. Dans les 3 cas, il n'y a aucun noeud dans le Branch&Bound et une seule itération de l'algorithme de *Dinkelbach*.

FIGURE 2.1 – Résultats obtenus sur les 3 instances

Taille	A_{min}, A_{max}, B	Temps(s)	Noeuds	Itérations	DMPPV	Nb parcelles
10×10	30, 35, 920	5.31	0	1	1.16	30
10×10	20, 21, 520	5.83	0	1	1.27	20
10×10	70, 75, 3500	5.37	0	1	1.0	70

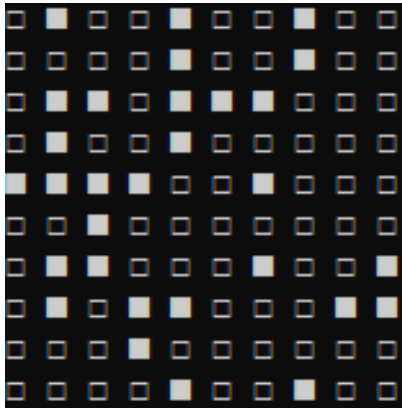


FIGURE 2.2 – Instance 1

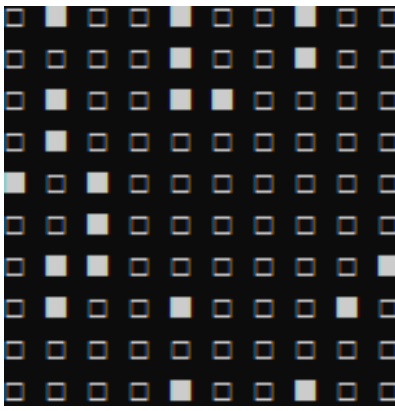


FIGURE 2.3 – Instance 2

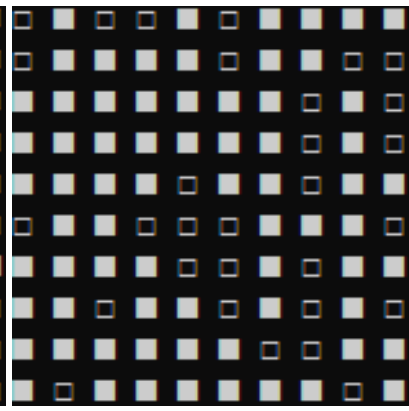


FIGURE 2.4 – Instance 3

FIGURE 2.5 – Solution obtenue pour chacune des instances : les cases blanche correspondent aux parcelles non coupées

Génération d'instances aléatoires On génère maintenant des instances aléatoires : pour chaque instance à générer, dans un premier temps, on tire aléatoirement $A \sim \mathbb{U}_{[1,100]}$, puis, pour un problème de taille n , on prend :

$$\begin{aligned}
 \bullet B &= \frac{n^2}{100} \times 20 \times A \\
 \bullet A_{min} &\sim \frac{n^2}{100} \times \mathbb{U}_{[1,A]} \\
 \bullet A_{max} &\sim \frac{n^2}{100} \times \mathbb{U}_{[A,100]} \\
 \bullet c_{i,j} &\sim \mathbb{U}_{[1,n]}
 \end{aligned}$$

Les résultats sont ensuite affichés dans le Tableau 2.3. On notera que certaines instances n'ont pas de solution, leur DMPPV est alors laissé vide. Il semble que les paramètres B, A_{min}, A_{max} ont peu d'influence sur le temps de résolution. Néanmoins, cela est peut-être dû à la façon dont ils sont générés. En effet, on a tenté de générer des instances réalisables.

Notons que le nombre d'itérations de l'algorithme devrait être dans le pire des cas en $O(\log(nM))$ où $M = \max\{1, \max|c_{i,j}|, \max|d_{i,j}|\}$. Dans notre cas, $|c_{i,j}| \leq n$ et $|d_{i,j}| \leq \sqrt{2}n$, donc le nombre d'itérations est au pire en $O(\log(n))$, ce qui semble être vérifié dans la pratique.

TABLE 2.1 – Résultats obtenus sur les instances générées aléatoirement

Taille	A_{min}, A_{max}, B	Temps(s)	Noeuds	Itérations	DMPPV
10×10	51, 92, 1720	7.44	0	1	1.0
10×10	18, 87, 920	9.32	0	2	1.0
10×10	31, 86, 1060	9.71	0	2	1.0
10×10	9, 95, 420	9.58	0	2	1.0
10×10	6, 92, 1300	9.85	0	2	1.0
11×11	39, 117, 1065	19.45	7439	2	1.0
11×11	44, 93, 1186	4.1	0	0	-
11×11	35, 109, 1621	14.1	0	2	1.0
11×11	33, 119, 1863	14.22	0	2	1.0
11×11	67, 100, 1936	4.64	0	0	-
12×12	10, 82, 317	33.44	0	3	1.0
12×12	12, 134, 720	25.57	0	2	1.0
12×12	1, 73, 230	32.1	0	2	1.0
12×12	23, 81, 1498	26.23	0	2	1.0
12×12	66, 137, 2592	31.56	0	2	1.0
13×13	32, 169, 642	56.36	0	2	1.29
13×13	15, 159, 1048	53.44	0	2	1.0
13×13	49, 90, 1622	53.78	0	2	1.0
13×13	41, 149, 2467	40.72	0	2	1.0
13×13	79, 162, 3008	45.97	0	2	1.0
14×14	8, 96, 1137	78.47	0	2	1.0
14×14	41, 125, 1686	84.31	0	2	1.0
14×14	65, 176, 3332	91.03	27624	2	1.0
14×14	14, 192, 1019	90.55	0	2	1.0
14×14	65, 165, 2548	99.24	0	2	1.0
15×15	104, 187, 2610	55.32	0	0	-
15×15	47, 198, 2835	157.08	4165	2	1.0
15×15	4, 218, 495	128.03	0	2	1.0
15×15	74, 162, 3240	143.66	987	2	1.0
15×15	2, 214, 3240	141.67	0	2	1.0
16×16	28, 223, 1331	255.32	0	2	1.0
16×16	41, 169, 2611	268.63	0	2	1.0
16×16	13, 38, 717	225.55	0	2	1.0
16×16	28, 184, 1382	232.14	0	2	1.0
16×16	15, 59, 666	230.36	0	2	1.0

Projet 3

Protection de la diversité génétique

3.1 Modélisation du problème

Variables

- $\forall i \in \{1, \dots, N\} x_i \in \{1, \dots, N\}$ = nombre d'enfants de l'individu i
- $\forall k \in \{1, \dots, m\} p_k \in [0, 1]$ = probabilité de disparition de l'allèle k
- $\forall k \in \{1, \dots, m\} t_k \in [0, 1]$ = probabilité de disparition de l'allèle k si aucun individu sélectionné ne possède deux fois l'allèle k

Fonction objectif On souhaite minimiser l'espérance du nombre d'allèles disparus. La fonction objectif s'écrit donc :

$$\min \sum_k p_k$$

On utilisera les notations suivantes qui permettent de distinguer deux ensembles d'individus : ceux qui ont deux copies de l'allèle k et ceux qui ont en une.

- $I_k^1 = \{i \in \{1, \dots, N\} : i \text{ possède une fois l'allèle } k\}$
- $I_k^2 = \{i \in \{1, \dots, N\} : x_i > 0 \text{ et } i \text{ possède deux fois l'allèle } k\}$

Contraintes

- La taille de la population reste constante : $\sum_{i \text{ femelle}} x_i = N = \sum_{i \text{ mâle}} x_i$
- Calcul de la probabilité de disparition : $p_k = \underbrace{\prod_{i \in I_k^1} 0.5^{x_i}}_{t_k} \prod_{i \in I_k^2} 0^{x_i}$
- Probabilité de disparition si $I_k^2 = \emptyset$: $t_k \geq \prod_{i \in I_k^1} 0.5^{x_i}$

S'il existe un individu de la population qui a deux fois l'allèle k et qui fait au moins un enfant ($x_i > 0$) alors on a $p_k = 0$. De plus, on sait que $\prod_{i \in I_k^1} 0.5^{x_i} \in [0, 1]$ et comme on cherche à minimiser les p_k , une façon d'écrire la deuxième contrainte est la suivante :

$$p_k \geq \prod_{i \in I_k^1} 0.5^{x_i} - \sum_{i \in I_k^2} x_i$$

En effet, dans le cas où I_k^2 est vide, alors on retrouve bien le calcul de p_k qui se collera à sa borne car on est en minimisation. Dans le cas où I_k^2 est non vide alors le membre de droite est négatif, donc $p_k = 0$.

La troisième contrainte peut également s'écrire $\log(t_k) \geq \log(0.5) \sum_{i \in I_k^1} x_i$.

Le problème se modélise donc par le programme suivant :

$$\left\{ \begin{array}{ll} \min \sum_k p_k & \\ \sum_{i \text{ femelle}} x_i = N = \sum_{i \text{ mâle}} x_i & (1) \\ p_k \geq t_k - \sum_{i \in I_k^2} x_i & \forall k \in \{1, \dots, m\} \quad (2) \\ \log(t_k) \geq \log(0.5) \sum_{i \in I_k^1} x_i & \forall k \in \{1, \dots, m\} \quad (3) \\ x_i \in \{1, \dots, N\} & \forall i \in \{1, \dots, N\} \\ p_k, t_k \in [0, 1] & \forall k \in \{1, \dots, m\} \end{array} \right. \quad (P)$$

3.2 Approximation du logarithme par une fonction linéaire par morceaux

En approximant la fonction \log par une fonction linéaire par morceaux par le dessus, on peut relâcher la contrainte (3) par l'ensemble de contraintes suivant :

$$\log(\theta_r) + \frac{1}{\theta_r}(t_k - \theta_r) \geq \log(0.5) \sum_{i \in I_k^1} x_i \quad \forall r \in \{1, \dots, h\}$$

où $\theta \in \mathbb{R}^h$ est tel que $0 < \theta_1 < \theta_2 < \dots < \theta_h = 1$.

Le programme en variables mixtes P_R suivant correspond donc à une relaxation du problème P précédent.

$$\left\{ \begin{array}{ll} \min \sum_k p_k & \\ \sum_{i \text{ femelle}} x_i = N = \sum_{i \text{ mâle}} x_i & (1) \\ p_k \geq t_k - \sum_{i \in I_k^2} x_i & \forall k \in \{1, \dots, m\} \quad (2) \\ \log(\theta_r) + \frac{1}{\theta_r}(t_k - \theta_r) \geq \log(0.5) \sum_{i \in I_k^1} x_i & \forall r \in \{1, \dots, h\} \quad \forall k \in \{1, \dots, m\} \quad (\bar{3}) \\ x_i \in \{1, \dots, N\} & \forall i \in \{1, \dots, N\} \\ p_k, t_k \in [0, 1] & \forall k \in \{1, \dots, m\} \end{array} \right. \quad (P_R)$$

Comme la seule contrainte qui assure que la solution est admissible est la contrainte (1), la solution x renvoyée par P_R sera également admissible pour le problème d'origine. La valeur de la fonction objectif retournée peut, en revanche, être inférieure à la valeur réelle et donne donc une borne inférieure sur la valeur du problème d'origine. On peut retrouver la valeur du problème d'origine facilement en calculant les p_k à partir des x_i .

3.3 Résolution numérique

Résultats sur les deux instances données On résout l'instance donnée en limitant le nombre d'enfants de chaque individu à 3 puis à 2. On trouve les solutions Figure 3.1. La solution de droite est la même que dans l'énoncé mais celle de gauche est légèrement différente. Il semble qu'il y ait une permutation dans la solution.

Solution obtenue avec la contrainte $x_i \leq 3 \quad \forall i \in \{1, \dots, N\}$							
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	3	3	1	3	2	2	0
Espérance du nombre d'allèles perdus 0.015625							
Borne inférieure 0.015621							
Probabilité de disparition des allèles A,a,B,C,c,D,d,E,e = 0							
Probabilité de disparition de l'allèle b = 0.015621							

Solution obtenue avec la contrainte $x_i \leq 2 \quad \forall i \in \{1, \dots, N\}$							
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
2	2	2	2	2	2	2	2
Espérance du nombre d'allèles perdus 0.062500							
Borne inférieure 0.062433							
Probabilité de disparition des allèles A,a,B,C,c,D,d,E,e = 0							
Probabilité de disparition de l'allèle b = 0.062500							

Instance	Morceaux	Temps(s)	Noeuds	Borne inférieure	Espérance
$x_i \leq 3$	50	1.0	0	0.015621	0.015625
$x_i \leq 2$	50	0.0	0	0.062433	0.062500

FIGURE 3.1 – Résultats obtenus sur l'instance donnée avec $\theta_1 = 0.0001$

Génération d'instances aléatoires On génère des instances aléatoires.

Le génotype des individus : pour chacun des gènes, on tire selon une loi de Bernoulli $\mathcal{B}\left(\frac{g + \alpha m}{2(1 + \alpha)m}\right)$ l'allèle possédé où

- g correspond au numéro du gène considéré
- m est le nombre d'allèles total ($= 2g$)
- α est un paramètre qui détermine la rareté d'un allèle sur les premiers gènes : l'allèle numéro 2 sera présent en proportion $\frac{1 + \alpha m}{2(1 + \alpha)m} < \frac{1}{2}$, tandis que sur le dernier gène, l'allèle numéro 2 sera présent en proportion $\frac{(1 + \alpha)m}{2(1 + \alpha)m} = \frac{1}{2}$. On a donc la proportion du premier allèle du gène numéro 2 dans l'intervalle $\left[\underbrace{\frac{1}{2m}}_{\alpha=0}, \underbrace{\frac{1+m}{4m}}_{\alpha=1}\right]$.

Ainsi, plus α est petit, plus il y a d'allèles rares dans l'instance.

Dans un premier temps, on fait varier le paramètre α . Le nombre d'enfants par individu n'est pas borné. On affiche les solutions pour $\alpha = 0$ et $\alpha = 1$ Tableau 3.1, 3.3. On observe les résultats dans le Tableau 3.5. La colonne "Nb allèles disparus" correspond à l'espérance de cette quantité. On remarque qu'elle diminue à mesure que α augmente. La borne inférieure est également plus serrée lorsque α est grand, cela est peut-être dû au fait que l'espérance se rapproche de 0 et que 0 est une borne triviale. Le problème avec $\alpha = 1$ semble néanmoins plus facile puisque la solution obtenue est d'espérance proche de 0. Dans la suite, afin d'avoir des nombres suffisamment éloignés de 0, on prendra $\alpha = 0$.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
3	0	5	0	4	4	0	0

TABLE 3.1 – Solution pour $\alpha = 0$

p_2	p_4	p_6	p_8	p_{10}
0.03125	0.00391	0.00781	0.00781	0.00781

TABLE 3.2 – Probabilités non nulles pour $\alpha = 0$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
8	0	0	0	4	4	0	0

TABLE 3.3 – Solution pour $\alpha = 1$

p_2	p_4	p_6
0.00024	0.00024	0.00024

TABLE 3.4 – Probabilités non nulles pour $\alpha = 1$

Taille	Morceaux	α	Temps(s)	Noeuds	Borne inférieure	Allèles disparus	Parents
8	9	0.0	0.02	0	0.057239	0.058594	4
8	9	0.1	0.01	0	0.007149	0.008545	3
8	9	0.2	0.01	0	0.003574	0.004425	3
8	9	0.3	0.01	0	0.003574	0.004425	3
8	9	0.4	0.01	0	0.003574	0.003967	3
8	9	0.5	0.01	0	0.0	0.000519	3
8	9	0.6	0.01	0	0.0	0.000519	3
8	9	0.7	0.01	0	0.0	0.000519	3
8	9	0.8	0.01	0	0.0	0.000732	3
8	9	0.9	0.01	0	0.0	0.000732	3
8	9	1.0	0.01	0	0.0	0.000732	3

TABLE 3.5 – Résultats obtenus pour une population de taille $N = 8$ en faisant varier le paramètre de rareté α

Dans un deuxième temps, on fait varier le nombre de morceaux considérés dans l'approximation linéaire par morceaux du log. Les résultats sont présentés Tableau 3.6. Sur la même instance, on observe que la valeur de la solution est la meilleure dès lors qu'on a plus de 6 morceaux dans l'approximation linéaire par morceaux du log. De plus, la borne inférieure est la meilleure lorsque $h = 7$ ou 9.

Taille	Morceaux	α	Temps(s)	Noeuds	Borne inférieure	Allèles disparus	Parents
0	0.01027	1.004639	3				
8	2	0.0	0.01	0	0.01112	0.256836	4
8	3	0.0	0.01	0	0.039677	0.066406	4
8	4	0.0	0.01	0	0.042945	0.058594	4
8	5	0.0	0.01	0	0.049055	0.066406	4
8	6	0.0	0.01	0	0.05051	0.058594	4
8	7	0.0	0.01	0	0.057202	0.058594	4
8	8	0.0	0.02	0	0.054898	0.058594	4
8	9	0.0	0.02	0	0.057239	0.058594	4
8	10	0.0	0.02	0	0.055703	0.058594	4

TABLE 3.6 – Résultats obtenus pour une population de taille $N = 8$ en faisant varier le nombre h de morceaux

Ensuite, on fait varier la taille de la population. Les résultats obtenus sont présentés dans le Tableau 3.7. On observe que lorsque la taille de la population augmente, l'espérance du nombre d'allèles disparus diminue. Ce qui est assez logique puisque plus il y a d'individus qui peuvent transmettre leur patrimoine génétique, plus les allèles ont de chance d'être conservés d'une génération à une autre. Notons que quoiqu'il arrive, le nombre de parents est

Taille	Morceaux	α	Temps(s)	Noeuds	Borne inférieure	Allèles disparus	Parents
8	9	0.0	0.44	0	0.057239	0.058594	4
10	9	0.0	0.04	0	0.002929	0.002934	2
12	9	0.0	0.07	32	0.0	0.000795	4
14	9	0.0	0.06	0	0.0	0.000366	3
16	9	0.0	0.05	0	0.0	7.6e-5	2
18	9	0.0	0.04	0	0.0	1.1e-5	2
20	9	0.0	0.05	0	0.0	0.000489	4

TABLE 3.7 – Résultats obtenus pour $\alpha = 0$, $h = 9$ en faisant varier la taille de la population

toujours au nombre de 4. C'est peut-être lié au nombre de gènes qui vaut 5. Faisons donc varier ce paramètre. On observe alors dans le Tableau 3.8 qu'effectivement le nombre d'individus faisant des enfants (nombre de parents) peut aller jusqu'à 5. Ce qu'il est intéressant de constater, c'est que le nombre d'allèles disparus augmente lorsqu'on augmente de nombre de gènes. Cela est dû à la façon dont sont générées les instances puisque la probabilité que l'allèle 2 soit présent sur le gène g pour un individu donné vaut $\frac{g + \alpha m}{2(1 + \alpha)m}$ avec $\alpha = 0$. Ce qui signifie que sur les premiers gènes, la probabilité de présence de l'allèle 2 diminue avec m . On fait le même test avec $\alpha = 0.5$ et

Taille	Gènes	Morceaux	α	Temps(s)	Noeuds	Borne inférieure	Allèles disparus	Parents
8	5	9	0.0	0.01	0	0.057239	0.058594	4
8	6	9	0.0	0.01	0	0.018196	0.019562	3
8	7	9	0.0	0.01	0	0.022094	0.023499	4
8	8	9	0.0	0.01	0	1.061317	1.0625	5
8	9	9	0.0	0.01	0	1.483918	1.516235	5
8	10	9	0.0	0.02	0	1.139546	1.141663	4

TABLE 3.8 – Résultats obtenus pour $N = 8$, $\alpha = 0$ et $h = 9$ en faisant varier le nombre de gènes

on reporte les résultats dans le Tableau 3.9. On observe la même tendance dans le nombre de gènes disparus qui augmente. Ce qui encore une fois paraît logique puisqu'on garde une population de même taille et on augmente le nombre de gènes donc il est clair que tous les gènes ne pourront pas être transmis, il faut forcément en favoriser certains plutôt que d'autres. On remarque également qu'à partir de 50 gènes, toute la population fait des enfants. On a donc remarqué qu'un grand nombre de gènes présents dans la population entraîne un grand nombre de gènes

Taille	Gènes	Morceaux	α	Temps(s)	Noeuds	Borne inférieure	Allèles disparus	Parents
8	10	9	0.5	0.02	0	0.000849	0.00174	5
8	20	9	0.5	0.04	0	0.129015	0.131104	6
8	30	9	0.5	0.06	0	2.160467	2.162231	6
8	40	9	0.5	0.07	0	0.410948	0.426025	7
8	50	9	0.5	0.07	0	0.846952	0.885254	8
8	60	9	0.5	0.07	0	0.767458	0.789459	8
8	110	9	0.5	0.17	0	1.802935	1.877197	8
8	160	9	0.5	0.13	0	5.225666	5.372162	8
8	210	9	0.5	0.34	0	6.371545	6.540466	8
8	260	9	0.5	0.35	0	7.576498	7.766418	8
8	310	9	0.5	0.43	0	11.852477	12.183197	8
8	360	9	0.5	0.36	0	11.427156	11.752808	8
8	410	9	0.5	0.39	0	11.557024	11.986572	8
8	460	9	0.5	0.65	0	15.842572	16.346313	8

TABLE 3.9 – Résultats obtenus pour $N = 8$, $\alpha = 0.5$ et $h = 9$ en faisant varier le nombre de gènes

disparus, ceci est grandement dû au fait que la population est de petite taille. Observons dans le Tableau 3.3 le comportement des solutions lorsque la population devient bien plus grande. On remarque que la borne inférieure tombe vite à 0 à partir de 28 individus dans la population. Le nombre d'allèles disparus quant à lui passe également à 0 mais bien plus tard : à partir de 93 individus. Le nombre de parents augmente avec la taille de la population mais il n'y a que pour les populations très petites (8 et 13 individus) que tous les individus font des enfants. Le temps de résolution est fonction du nombre de noeuds parcourus qui sont en nombre important pour une population de taille 23 mais il n'y a, à priori, pas de lien direct entre le nombre de noeuds et la taille de la population.

Taille	Gènes	Morceaux	α	Temps(s)	Noeuds	Borne inférieure	Allèles disparus	Parents
8	500	9	0.5	0.89	0	17.597224	18.124939	8
13	500	9	0.5	2.71	11	2.422798	2.546721	13
18	500	9	0.5	5.36	181	0.285802	0.300039	15
23	500	9	0.5	25.59	4037	0.009385	0.016154	18
28	500	9	0.5	5.21	180	0.0	0.003891	21
33	500	9	0.5	1.68	0	0.0	0.001856	20
38	500	9	0.5	2.83	61	0.0	0.001572	17
43	500	9	0.5	2.65	77	0.0	0.001552	18
48	500	9	0.5	1.77	0	0.0	0.001136	15
53	500	9	0.5	1.78	0	0.0	0.000724	17
58	500	9	0.5	2.39	0	0.0	0.000608	17
63	500	9	0.5	2.58	0	0.0	0.000345	18
68	500	9	0.5	1.53	0	0.0	0.000285	19
73	500	9	0.5	1.45	0	0.0	8.0e-6	20
78	500	9	0.5	1.67	0	0.0	7.7e-5	21
83	500	9	0.5	0.83	0	0.0	1.6e-5	22
88	500	9	0.5	0.69	0	0.0	1.6e-5	22
93	500	9	0.5	0.54	0	0.0	0.0	24
98	500	9	0.5	0.58	0	0.0	0.0	26
110	500	9	0.5	0.82	0	0.0	0.0	28
160	500	9	0.5	0.95	0	0.0	0.0	40
210	500	9	0.5	0.96	0	0.0	0.0	54
260	500	9	0.5	1.45	0	0.0	0.0	66
310	500	9	0.5	1.36	0	0.0	0.0	78
360	500	9	0.5	1.53	0	0.0	0.0	90
410	500	9	0.5	1.76	0	0.0	0.0	104
460	500	9	0.5	1.94	0	0.0	0.0	116

TABLE 3.10 – Résultats obtenus pour $h = 9$, $\alpha = 0.5$, $G = 500$ en faisant varier la taille de la population

Projet 4

Exploitation durable de la forêt

4.1 Formulation par un programme linéaire en variables mixtes

On utilise les notations suivantes :

- $M = \{1, \dots, m\}$, $N = \{1, \dots, n\}$
- $A_{ij} = \{\text{parcelles } (k, l) \text{ adjacentes à } (i, j)\}$

On modélise le problème de la façon suivante :

$$\left\{ \begin{array}{ll} \max & w_1 \sum_{(i,j) \in M \times N} t_{ij}(1 - x_{ij}) + w_2 gl \sum_{(i,j) \in M \times N} 4x_{ij} - d_{ij} \\ \text{s.c. :} & \\ & d_{ij} \geq \sum_{(k,l) \in A_{ij}} x_{kl} - |A_{ij}|(1 - x_{ij}) \quad \forall (i, j) \in M \times N \\ & d \in \mathbb{R}_+^{M \times N} \\ & x \in \{0, 1\}^{M \times N} \end{array} \right. \quad (\text{P1})$$

Variables

- $x_{ij} = 1$ si la parcelle s_{ij} n'est pas coupée
- d_{ij} = nombre de parcelles non coupées autour de la parcelle s_{ij}

Fonction objectif

- $\sum_{(i,j) \in M \times N} t_{ij}(1 - x_{ij})$ donne la population attendue dans les parcelles non coupées
- $gl \sum_{(i,j) \in M \times N} 4x_{ij} - d_{ij}$ donne la population attendue dans les lisières

Contraintes

- $d_{ij} \geq \sum_{(k,l) \in A_{ij}} x_{kl} - |A_{ij}|(1 - x_{ij})$ fixe la variable d_{ij} comme étant le nombre de parcelles non coupées autour de la parcelle s_{ij} si elle est non coupée et 0 sinon. En effet, comme d_{ij} a un coefficient négatif dans l'objectif, on cherche à minimiser les d_{ij} et donc elle va se coller à sa borne inférieure.

4.2 Formulation par un programme quadratique en variables 01

La deuxième partie de la fonction objectif peut également s'écrire :

$$\sum_{(i,j) \in M \times N} 4x_{ij} - d_{ij} = \sum_{(i,j) \in M \times N} x_{ij} \left(\sum_{(k,l) \in A_{ij}} (1 - x_{kl}) \right)$$

En effet, il s'agit de compter le nombre de parcelles coupées autour de s_{ij} non coupée, cela donne donc bien le nombre de lisières autour de s_{ij} . Les variables d_{ij} ne sont alors plus nécessaires. Le programme obtenu est alors le suivant :

$$\left\{ \begin{array}{l} \max \quad w_1 \sum_{(i,j) \in M \times N} t_{ij}(1 - x_{ij}) + w_2 gl \sum_{(i,j) \in M \times N} x_{ij} \left(\sum_{(k,l) \in A_{ij}} (1 - x_{kl}) \right) \\ \text{s.c. :} \\ x \in \{0, 1\}^{M \times N} \end{array} \right. \quad (P2)$$

4.3 Linéarisation TU

4.3.1 Linéarisation

On ajoute une variable y_{ijkl} par produit $x_{ij}x_{kl}$ ainsi que les contraintes $\begin{cases} y_{ijkl} \leq x_{ij} \\ y_{ijkl} \leq x_{kl} \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \end{cases}$. La fonction

objectif s'écrit alors

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

On remarque que les deux premières contraintes ne sont pas nécessaires car on cherche à minimiser les y_{ijkl} . Le programme s'écrit alors

$$\left\{ \begin{array}{l} \max \quad w_1 \sum_{(i,j) \in M \times N} t_{ij}(1 - x_{ij}) + w_2 gl \sum_{(i,j) \in M \times N} \sum_{(k,l) \in A_{ij}} x_{ij} - y_{ijkl} \\ \text{s.c. :} \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad \forall (i,j), (k,l) \\ x \in \{0, 1\}^{M \times N} \\ y_{ijkl} \geq 0 \quad \forall (i,j), (k,l) \end{array} \right. \quad (P3)$$

4.3.2 Matrice des contraintes TU

La matrice des contraintes a donc autant de lignes que de couples $(i,j), (k,l)$ et sur chacune des lignes, on a un 1 sous les variables x_{ij} et x_{kl} ainsi qu'un -1 sous la variable y_{ijkl} . La matrice est donc la concaténation d'une matrice d'incidence d'un graphe biparti et de $-I_h$. Or la matrice d'incidence d'un graphe biparti est TU et elle le reste si on lui accole I_h . Par exemple pour $m = n = 2$, on aurait la matrice suivante :

$$\begin{array}{cccc} & 11 & 12 & 21 & 22 & & 1112 & 1121 & 1122 & 1221 & 1222 & 2122 \\ \begin{array}{l} 1112 \\ 1121 \\ 1122 \\ 1221 \\ 1222 \\ 2122 \end{array} & \left(\begin{array}{cccc|cccccc} 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{array} \right) \end{array}$$

4.4 Résolution numérique

4.4.1 Résultats sur les deux instances données

On affiche les résultats obtenus avec les deux méthodes dans le Tableau 4.4.1. La méthode par linéarisation du programme quadratique semble être plus rapide que la méthode directe.

Taille	Quadratique		PLNE	
	Temps(s)	Noeuds	Temps(s)	Noeuds
10×10	0.029	0	0.062	0
5×5	0.03	0	0.051	0

TABLE 4.1 – Résultats obtenus sur les instances données

4.5 Ajout d'une contrainte

L'ajout de la contrainte $\sum_i x_i \leq 60$ ne permet pas de conserver une matrice des contraintes TU. En effet, on peut calculer le déterminant de la sous matrice suivante :

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{vmatrix} = -2$$

Les résultats obtenus avec les deux méthodes de résolution après l'ajout de la contraintes sont affichés dans le Tableau 4.5. La méthode par linéarisation du programme quadratique semble toujours être plus rapide. (L'instance de taille 5×5 ne comporte que 25 parcelles, la contrainte ne peut donc pas s'appliquer à cette instance).

Taille	Quadratique		PLNE	
	Temps(s)	Noeuds	Temps(s)	Noeuds
10×10	0.124	0	0.247	0

TABLE 4.2 – Résultats obtenus avec contrainte

4.6 Génération d'instances aléatoires

On génère des instances aléatoires et on observe les résultats obtenus sans la contrainte sur le nombre de parcelles dans le Tableau 4.6. La méthode directe utilise le même temps de calcul pour des instances de taille 10×10 à 50×50 tandis que le temps pris par la méthode par linéarisation d'un programme quadratique augmente avec la taille des instances. Ensuite, on observe les résultats avec la contrainte dans le Tableau 4.6. Avec les deux

Taille	Quadratique		PLNE	
	Temps(s)	Noeuds	Temps(s)	Noeuds
10×10	0.003	0	0.005	0
15×15	0.035	0	0.003	0
20×20	0.015	0	0.005	0
25×25	0.046	0	0.004	0
30×30	0.067	0	0.008	0
35×35	0.133	0	0.006	0
40×40	0.216	0	0.005	0
45×45	0.346	0	0.005	0
50×50	0.781	0	0.006	0

TABLE 4.3 – Résultats obtenus sur les instances aléatoires sans contrainte

méthodes, les temps d'exécution sont plus grands d'un facteur d'ordre de grandeur 100. La méthode directe semble néanmoins avoir un temps qui ne dépend pas de la taille des instances tout du moins pour des instances de cette taille. Quant à la méthode par linéarisation, on observe nettement un allongement du temps de calcul avec la taille des instances. On peut également constater que la taille des modèles pour ces deux méthodes diffère. En effet, la méthode directe comporte nm variables binaires et nm contraintes. La taille du modèle est donc linéaire en fonction du nombre de parcelles. Tandis que le modèle par linéarisation comporte nm variables binaires et $(nm)^2$ contraintes, donc une taille du programme quadratique en fonction du nombre de parcelles. Ce qui peut expliquer les différences dans les temps calculs observés.

Taille	Quadratique		PLNE	
	Temps(s)	Noeuds	Temps(s)	Noeuds
10×10	0.035	0	0.522	0
15×15	0.337	0	0.43	0
20×20	0.302	0	0.071	0
25×25	1.348	0	0.04	0
30×30	3.084	0	0.08	0
35×35	5.181	0	0.059	0
40×40	8.589	0	0.088	0
45×45	18.808	0	0.117	0
50×50	27.932	0	0.17	0

TABLE 4.4 – Résultats obtenus sur les instances aléatoires avec contrainte

Annexe Projet 1

TABLE A1 – Résultats obtenus pour $\alpha = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$

Taille instance	Proba de survie (par espèce)						Temps(s)	Noeuds	Coût
	1	2	3	4	5	6			
10× 10	0.51	0.5	0.7	0.91	0.52	0.86	0.03	0	128
11× 11	0.5	0.5	0.5	0.64	0.82	0.52	0.05	0	73
12× 12	0.5	0.51	0.5	0.52	0.51	0.82	0.08	0	182
13× 13	0.5	0.5	0.52	0.68	0.55	0.6	0.07	0	145
14× 14	0.5	0.57	0.58	0.58	0.71	0.5	0.08	0	129
15× 15	0.5	0.5	0.58	0.78	0.5	0.92	0.17	0	151
16× 16	0.58	0.5	0.58	0.78	0.96	0.75	0.19	0	209
17× 17	0.5	0.5	0.5	0.6	0.58	0.55	0.1	0	161
18× 18	0.6	0.55	0.58	0.6	0.65	0.52	0.52	0	201
19× 19	0.55	0.5	0.5	0.65	0.6	0.69	0.24	0	180
20× 20	0.58	0.5	0.5	0.82	0.62	0.58	0.58	0	175
21× 21	0.5	0.5	0.5	0.84	0.65	0.55	0.31	0	209
22× 22	0.5	0.5	0.5	0.73	0.5	0.86	0.73	0	237
23× 23	0.5	0.7	0.5	0.75	0.68	0.88	1.05	0	202
24× 24	0.58	0.52	0.65	0.5	0.87	0.68	0.28	0	195
25× 25	0.65	0.52	0.5	0.52	0.64	0.61	1.08	0	231
26× 26	0.5	0.5	0.5	0.68	0.57	0.66	0.33	0	156
27× 27	0.5	0.5	0.5	0.5	0.5	0.64	0.76	0	211
28× 28	0.5	0.5	0.5	0.55	0.55	0.5	0.47	0	271
29× 29	0.5	0.5	0.5	0.57	0.93	0.66	0.33	0	179
30× 30	0.7	0.5	0.5	0.73	0.72	0.74	0.9	0	277
31× 31	0.5	0.5	0.64	0.75	0.62	0.51	1.17	0	228
32× 32	0.58	0.5	0.5	0.52	0.7	0.6	2.33	0	288
33× 33	0.5	0.5	0.5	0.55	0.66	0.7	0.69	0	227
34× 34	0.5	0.5	0.5	0.5	0.58	0.64	1.62	0	247
35× 35	0.5	0.55	0.5	0.8	0.61	0.85	1.33	0	262
36× 36	0.52	0.5	0.5	0.57	0.5	0.7	1.34	0	317
37× 37	0.5	0.5	0.5	0.72	0.72	0.74	1.18	0	286
38× 38	0.5	0.55	0.5	0.73	0.75	0.7	1.19	0	268
39× 39	0.5	0.5	0.5	0.51	0.64	0.6	1.6	0	290
40× 40	0.5	0.5	0.5	0.55	0.65	0.78	1.02	0	234
41× 41	0.5	0.5	0.5	0.58	0.56	0.68	2.01	0	305
42× 42	0.58	0.5	0.65	0.8	0.71	0.52	1.48	0	236
43× 43	0.6	0.5	0.5	0.69	0.66	0.51	0.75	0	241
44× 44	0.5	0.5	0.5	0.64	0.84	0.68	1.59	0	269
45× 45	0.5	0.64	0.5	0.55	0.5	0.77	2.85	0	316
46× 46	0.5	0.5	0.5	0.52	0.87	0.6	3.19	0	313
47× 47	0.5	0.5	0.5	0.77	0.7	0.5	2.91	0	359
48× 48	0.5	0.5	0.5	0.6	0.62	0.7	2.93	0	403
49× 49	0.5	0.5	0.5	0.55	0.5	0.62	2.99	0	333
50× 50	0.5	0.5	0.5	0.78	0.68	0.58	1.41	0	328

TABLE A2 – Résultats obtenus pour $\alpha = [0.9, 0.9, 0.9, 0.5, 0.5, 0.5]$

Taille instance	Proba de survie (par espèce)						Temps(s)	Noeuds	Coût
	1	2	3	4	5	6			
10× 10	-	-	-	-	-	-	0.01	0	-
11× 11	-	-	-	-	-	-	0.01	0	-
12× 12	-	-	-	-	-	-	0.01	0	-
13× 13	-	-	-	-	-	-	0.01	0	-
14× 14	0.9	0.9	0.9	0.98	0.93	0.9	0.08	0	606
15× 15	0.92	0.91	0.91	0.98	0.98	0.96	0.18	0	407
16× 16	0.91	0.91	0.91	0.95	0.92	0.96	0.28	0	679
17× 17	0.91	0.91	0.91	0.97	0.98	0.98	0.11	0	738
18× 18	0.91	0.92	0.91	0.93	0.84	0.97	0.53	0	530
19× 19	0.91	0.91	0.91	0.96	0.91	0.89	2.85	0	659
20× 20	0.92	0.91	0.91	0.97	0.97	0.83	0.29	0	516
21× 21	0.91	0.92	0.92	0.83	0.96	1.0	1.01	0	599
22× 22	0.92	0.92	0.91	0.89	0.93	0.83	0.6	0	567
23× 23	0.91	0.9	0.91	0.84	0.93	0.88	3.61	264	754
24× 24	0.91	0.92	0.92	0.87	0.74	0.99	2.76	78	632
25× 25	0.91	0.9	0.91	0.99	0.92	0.94	1.44	0	598
26× 26	0.92	0.91	0.91	0.91	0.75	0.91	3.21	0	650
27× 27	0.91	0.91	0.91	0.98	0.98	0.83	3.64	0	811
28× 28	0.91	0.91	0.91	0.97	0.96	0.96	3.07	60	695
29× 29	0.92	0.91	0.9	0.95	0.95	0.74	2.69	0	718
30× 30	0.91	0.91	0.92	0.91	0.96	0.89	4.33	205	879
31× 31	0.91	0.91	0.9	0.85	0.74	0.68	5.32	215	756
32× 32	0.91	0.92	0.92	0.97	0.87	0.9	5.98	67	689
33× 33	0.92	0.91	0.91	0.51	0.85	0.6	0.96	0	704
34× 34	0.93	0.92	0.91	0.84	0.76	0.86	3.72	0	787
35× 35	0.91	0.91	0.94	0.74	0.85	0.9	2.44	0	880
36× 36	0.91	0.91	0.9	0.68	0.89	1.0	8.87	354	951
37× 37	0.91	0.91	0.91	0.91	0.97	0.94	2.46	0	882
38× 38	0.91	0.91	0.92	0.83	0.99	0.89	3.27	0	846
39× 39	0.91	0.9	0.9	0.98	1.0	0.96	1.66	0	1023
40× 40	0.91	0.93	0.91	0.85	0.98	0.62	1.25	0	822
41× 41	0.91	0.91	0.91	0.95	0.61	0.93	1.71	0	881
42× 42	0.91	0.91	0.91	0.84	0.95	1.0	3.1	0	995
43× 43	0.91	0.91	0.91	0.81	0.87	0.82	2.37	0	1046
44× 44	0.91	0.91	0.9	0.85	0.94	0.87	7.86	129	969
45× 45	0.91	0.91	0.91	0.82	0.74	0.96	4.33	0	1186
46× 46	0.91	0.9	0.91	0.99	0.98	0.98	2.79	0	951
47× 47	0.9	0.91	0.91	0.91	0.84	0.94	4.03	0	1224
48× 48	0.91	0.92	0.91	0.94	0.99	0.85	4.47	0	1056
49× 49	0.91	0.9	0.92	0.91	0.98	0.82	7.75	184	1109
50× 50	0.91	0.9	0.91	0.91	0.99	0.78	2.31	0	1043

TABLE A3 – Résultats obtenus pour $\alpha = [0.5, 0.5, 0.5, 0.9, 0.9, 0.9]$

Taille instance	Proba de survie (par espèce)						Temps(s)	Noeuds	Coût
	1	2	3	4	5	6			
10×10	-	-	-	-	-	-	0.01	0	-
11×11	0.51	0.5	0.5	0.92	0.9	0.93	0.04	0	160
12×12	-	-	-	-	-	-	0.01	0	-
13×13	0.5	0.5	0.5	0.93	0.94	0.91	0.07	0	178
14×14	0.5	0.5	0.5	0.92	0.9	0.93	0.17	0	159
15×15	0.52	0.58	0.52	0.92	0.92	0.94	0.34	0	203
16×16	0.58	0.5	0.5	0.92	0.91	0.91	0.3	0	185
17×17	0.5	0.5	0.64	0.92	0.92	0.91	0.7	0	207
18×18	0.52	0.58	0.5	0.91	0.93	0.92	0.36	0	225
19×19	0.5	0.5	0.5	0.93	0.92	0.9	0.24	0	183
20×20	0.5	0.5	0.5	0.91	0.97	0.9	0.33	0	183
21×21	0.5	0.5	0.5	0.91	0.91	0.91	0.25	0	187
22×22	0.5	0.52	0.5	0.91	0.91	0.97	0.83	0	243
23×23	0.65	0.65	0.51	0.91	0.91	0.93	0.41	0	174
24×24	0.5	0.5	0.5	0.9	0.91	0.91	1.05	0	257
25×25	0.51	0.5	0.65	0.91	0.91	0.93	0.32	0	170
26×26	0.5	0.5	0.5	0.91	0.95	0.93	4.23	81	257
27×27	0.5	0.5	0.5	0.92	0.91	0.95	0.28	0	203
28×28	0.6	0.5	0.5	0.92	0.92	0.91	1.21	0	247
29×29	0.5	0.5	0.5	0.91	0.92	0.93	0.67	0	280
30×30	0.5	0.5	0.5	0.94	0.93	0.92	0.74	0	217
31×31	0.5	0.5	0.7	0.91	0.91	0.9	1.22	0	251
32×32	0.5	0.5	0.5	0.9	0.9	0.93	1.77	0	270
33×33	0.5	0.6	0.5	0.91	0.91	0.92	3.21	0	288
34×34	0.64	0.5	0.5	0.91	0.92	0.91	2.61	0	306
35×35	0.7	0.5	0.5	0.91	0.92	0.94	1.97	0	207
36×36	0.5	0.5	0.65	0.92	0.91	0.92	1.84	0	296
37×37	0.5	0.58	0.64	0.91	0.91	0.93	2.01	0	257
38×38	0.5	0.5	0.6	0.91	0.94	0.91	4.77	0	337
39×39	0.5	0.5	0.5	0.92	0.92	0.91	2.64	0	288
40×40	0.5	0.5	0.5	0.9	0.91	0.92	2.21	0	320
41×41	0.5	0.64	0.5	0.91	0.9	0.91	3.74	0	394
42×42	0.55	0.51	0.5	0.92	0.91	0.92	9.79	0	319
43×43	0.58	0.5	0.5	0.92	0.9	0.91	2.83	0	266
44×44	0.5	0.52	0.5	0.91	0.91	0.93	2.78	0	302
45×45	0.5	0.5	0.5	0.9	0.9	0.92	2.4	0	319
46×46	0.5	0.5	0.5	0.9	0.9	0.91	1.63	0	289
47×47	0.5	0.64	0.5	0.91	0.91	0.9	4.44	0	347
48×48	0.5	0.5	0.5	0.94	0.91	0.92	5.64	0	400
49×49	0.5	0.5	0.5	0.92	0.95	0.91	5.2	0	369
50×50	0.5	0.5	0.5	0.9	0.92	0.92	3.82	0	344

TABLE A4 – Résultats obtenus pour $\alpha = [0.8, 0.8, 0.8, 0.6, 0.6, 0.6]$

Taille instance	Proba de survie (par espèce)						Temps(s)	Noeuds	Coût
	1	2	3	4	5	6			
10×10	-	-	-	-	-	-	0.0	0	-
11×11	0.82	0.8	0.9	0.9	0.82	0.85	0.2	0	304
12×12	0.81	0.81	0.81	0.99	0.89	0.97	0.05	0	459
13×13	0.82	0.81	0.82	0.73	0.93	0.91	0.42	0	379
14×14	0.81	0.8	0.81	0.81	0.96	0.98	0.15	0	386
15×15	0.8	0.81	0.82	1.0	0.75	0.98	0.42	0	590
16×16	0.82	0.82	0.81	0.97	0.95	0.86	1.26	48	364
17×17	0.83	0.8	0.8	0.92	0.86	0.9	0.77	0	385
18×18	0.8	0.85	0.84	0.93	0.78	0.95	0.99	0	405
19×19	0.85	0.82	0.82	0.84	0.85	0.89	1.93	0	421
20×20	0.82	0.85	0.82	0.64	0.84	0.77	1.1	0	475
21×21	0.84	0.8	0.8	0.98	0.71	0.73	2.79	0	429
22×22	0.85	0.8	0.8	0.98	0.83	0.8	1.13	0	441
23×23	0.83	0.81	0.82	0.75	0.8	0.76	1.35	0	536
24×24	0.85	0.82	0.82	0.6	0.7	0.71	0.89	0	410
25×25	0.82	0.9	0.8	0.76	0.95	0.76	4.25	0	487
26×26	0.82	0.8	0.82	0.92	0.72	0.81	3.31	0	547
27×27	0.8	0.82	0.81	0.94	0.73	0.84	1.36	0	563
28×28	0.8	0.82	0.85	0.62	0.68	0.61	1.68	0	461
29×29	0.82	0.81	0.8	0.81	0.73	0.9	2.77	29	583
30×30	0.8	0.85	0.83	0.64	0.83	0.84	2.91	0	590
31×31	0.82	0.8	0.85	0.79	0.88	0.94	5.98	0	641
32×32	0.82	0.82	0.8	0.92	0.99	0.6	3.25	0	584
33×33	0.85	0.82	0.8	0.81	0.86	0.78	2.5	0	545
34×34	0.8	0.8	0.81	0.96	0.8	0.99	6.11	39	651
35×35	0.82	0.82	0.82	0.87	0.89	0.88	1.68	0	539
36×36	0.82	0.82	0.8	0.7	0.85	0.8	1.51	0	648
37×37	0.82	0.82	0.8	0.95	0.89	0.83	3.36	0	592
38×38	0.82	0.82	0.85	0.95	0.74	0.94	4.13	56	690
39×39	0.8	0.85	0.82	0.65	0.94	0.87	2.06	0	594
40×40	0.84	0.82	0.82	0.84	0.97	0.99	3.62	0	724
41×41	0.8	0.82	0.85	0.78	0.91	0.98	3.77	0	820
42×42	0.83	0.82	0.82	0.82	0.78	0.69	15.69	220	741
43×43	0.82	0.83	0.85	0.78	0.83	0.73	4.43	0	644
44×44	0.85	0.88	0.82	0.94	0.87	0.7	7.49	0	701
45×45	0.85	0.82	0.82	0.82	0.9	0.9	6.96	0	772
46×46	0.82	0.8	0.85	0.6	0.9	0.75	15.03	139	764
47×47	0.8	0.82	0.82	0.95	0.81	0.76	8.38	0	710
48×48	0.82	0.85	0.82	0.96	0.89	0.76	7.28	0	729
49×49	0.8	0.83	0.8	0.92	0.78	0.94	7.02	0	794
50×50	0.85	0.82	0.82	0.81	0.82	0.69	5.35	0	741