

RODD - Compte-rendu des projets 1 à 4

Quentin RAPILLY, Yohan BORNACHOT

23 mars 2022

Résumé

Ce document présente les différentes démarches de modélisation ainsi que les principaux résultats des projets de Recherche Opérationnelle et Développement Durable proposés en cours.

1 Projet 1 - Sélection de réserves naturelles

1.1 Contexte, Zone centrale et Zone tampon

On s'intéresse à un ensemble d'espèces à protéger, $E = \{e_1, e_2, \dots, e_p\}$, vivant sur un ensemble de parcelles - ou sites -, $S = \{s_1, s_2, \dots, s_n\}$, répartis sur un territoire. Pour chaque parcelle, on connaît toutes les espèces qui y vivent et l'on suppose que ces espèces survivront dans cette parcelle si celle-ci est protégée.

Le problème de base consiste à déterminer le plus petit sous-ensemble de parcelles permettant de protéger toutes les espèces considérées. Associons à chaque parcelle s_i une variable booléenne x_i qui vaut 1 si et seulement si cette parcelle est retenue pour être protégée. Dans notre cas, la fonction objectif vise à minimiser le nombre de parcelles retenues et les p contraintes d'inégalité expriment que, pour chaque espèce considérée, au moins une parcelle la contenant doit être retenue. On cherche ici à résoudre un problème de recouvrement (Set Covering Problem).

On considère deux types d'espèces à protéger, des espèces en danger et des espèces communes, et l'on distingue deux types de zones protégées, des zones centrales et des zones tampons qui protègent la zone centrale des nuisances entraînées par les activités humaines extérieures à la réserve.

Pour être protégée (avec une certaine probabilité), une espèce en danger doit être présente dans une zone centrale et une espèce commune doit être présente soit dans une zone centrale soit dans une zone tampon.

On considère qu'une parcelle sélectionnée fait partie d'une zone centrale si toutes les parcelles qui l'entourent sont elles aussi sélectionnées soit dans la zone centrale soit dans la zone tampon (le décideur peut définir une zone tampon plus ou moins importante). On connaît la probabilité p_{ki} que l'espèce e_k , présente dans la parcelle s_i , survive dans cette parcelle si celle-ci est protégée. On suppose que ces probabilités sont indépendantes et que $0 \leq p_{ki} \leq 1$. Le problème consiste à déterminer un sous-ensemble de parcelles de coût minimal et tel que, pour chaque espèce e_k , la probabilité de présence dans la réserve, c'est-à-dire dans ce sous-ensemble de parcelles, soit supérieure ou égale à une valeur donnée α_k .

1.2 Modélisation du problème de la détermination d'une réserve optimale avec zone centrale et zone tampon

Pour rappel, la variable $x = (x_{ij})_{(ij) \in [[1, n]]^2}$ témoigne de la sélection ou non des parcelles situées en (i, j) dans la réserve, de taille $n \times n$. De plus, on dispose des coûts $(c_{ij})_{(ij) \in [[1, n]]^2}$ correspondant au coût de sélection de chaque parcelle en unité de coût. On cherche donc à minimiser le coût de protection de la réserve tout en garantissant que toutes les espèces survivent avec une probabilité plus ou moins élevée α_k , comme introduit plus haut.

L'objectif étant la minimisation des coûts de protection de la réserve naturelle, il se formule comme suit :

$$\min_x \sum_{(i,j)} c_{ij} \cdot x_{ij} \quad (1)$$

On introduit une variable annexe y_{ij} qui vaut 1 si (i, j) fait partie d'une zone centrale, i.e. si l'ensemble des parcelles voisines sont sélectionnées. Les variables y_{ij} sont donc définies par :

$$9y_{ij} \leq \sum_{(k,l) \in V(i,j)} x_{kl}, \quad \forall (i, j) \in [[1, n]]^2 \quad (2)$$

où $V(i, j) = \{(k, l), k \in [i - 1, i + 1], l \in [j - 1, j + 1]\}$ désigne le voisinage de (i, j) , i.e. l'ensemble des cases adjacentes à (i, j) .

Il faut également prendre en compte le fait que les espèces, discriminées selon le danger de leur disparition (communes ou rares) doivent survivre avec une probabilité supérieures à α_k . Pour calculer la probabilité globale de survie d'une espèce k sur l'ensemble de la réserve naturelle, il est plus pratique de calculer la probabilité de la mort de l'espèce pour la configuration des parcelles retenues.

La probabilité pour l'espèce k (ici commune, voir l'explication ci-dessous pour une espèce rare) de mourir en (i, j) est $1 - p_{kij}$, donc la probabilité de l'espèce k de mourir sur l'ensemble de la réserve est (les p_{kij} étant indépendantes par hypothèse) :

$$\prod_{i,j} (1 - x_{ij} \cdot p_{kij}) \quad (3)$$

Ensuite, pour la probabilité qu'a l'espèce k de survivre sur l'ensemble de la réserve, il vient (événement contraire) :

$$1 - \prod_{i,j} (1 - x_{ij} \cdot p_{kij}) \quad (4)$$

Il faut donc que la probabilité ci-dessus soit supérieure à α_k , soit :

$$1 - \prod_{i,j} (1 - x_{ij} \cdot p_{kij}) \geq \alpha_k, \quad \forall k \in C \quad (5)$$

où C désigne l'ensemble des espèces communes.

Par ailleurs, une espèce rare ne survit avec la probabilité p_{kij} que si la parcelle (i, j) se trouve dans une zone centrale. Par le même raisonnement que précédemment, on obtient (il suffit juste de remplacer x par y) :

$$1 - \prod_{i,j} (1 - y_{ij} \cdot p_{kij}) \geq \alpha_k, \quad \forall k \in R \quad (6)$$

où R désigne l'ensemble des espèces rares.

A ce stade, on se heurte à un problème, car ici ces contraintes ne sont absolument pas linéaires. On pense alors à passer au logarithme, afin d'obtenir des sommes :

$$\sum_{i,j} \log(1 - x_{ij} \cdot p_{kij}) \leq \log(1 - \alpha_k), \quad \forall k \in C \quad (7)$$

Dernier problème, on a encore x à l'intérieur du logarithme, et donc cette contrainte n'est toujours pas linéaire... Mais on remarque ici que le rôle de x_{ij} est le même qu'il soit à l'intérieur du logarithme ou en dehors ! En effet, si $x_{ij} = 0$, alors le terme $\log(1 - x_{ij} \cdot p_{kij})$ vaudra $\log(1) = 0$, ce qui est équivalent au comportement du terme $\log(1 - p_{kij}) \cdot x_{ij}$. De même si $x_{ij} = 1$, on comptera $\log(1 - p_{kij})$ dans les deux cas. On peut donc réécrire la contrainte précédente de la manière suivante :

$$\sum_{i,j} \log(1 - p_{kij}) \cdot x_{ij} \leq \log(1 - \alpha_k), \quad \forall k \in C \quad (8)$$

Ce qui vient d'être dit vaut également pour les espèces rares, à ceci près qu'il faut remplacer x par y dans les inégalités.

Le problème final s'exprime donc comme suit :

$$(P2) \begin{cases} \min \sum_{(i,j)} c_{ij} x_{ij} \\ \text{s.t.} \\ 9y_{ij} \leq \sum_{(k,l) \in V(i,j)} x_{kl}, \quad \forall (i,j) \in [[1,n]]^2 \\ \sum_{i,j} \log(1 - p_{kij}) \cdot x_{ij} \leq \log(1 - \alpha_k), \quad \forall k \in C \\ \sum_{i,j} \log(1 - p_{kij}) \cdot y_{ij} \leq \log(1 - \alpha_k), \quad \forall k \in R \end{cases} \quad (9)$$

1.3 Résolution d'instances et études des résultats

1.3.1 cas n°1 : $\alpha_k = 0.5$ ($k = 1, \dots, 6$)

- Temps de resolution : 0.045005083084106445 s
- Valeur de l'objectif : 119.0
- Taux de survie des especes : (0.748 0.616 0.8200000000000001 0.86176 0.52 0.755)
- Tableau des zones protegees :

0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	2	1	0	0
0	0	0	0	0	1	2	1	0	0
0	1	1	1	0	1	1	1	0	0
0	1	2	1	0	0	0	0	0	0
0	1	2	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

1.3.2 cas n°2 : $\alpha_k = 0.9$ ($k = 1, \dots, 3$), et $\alpha_k = 0.5$ ($k = 4, \dots, 6$)

- Temps de résolution : 0.006296873092651367 s
- Valeur de l'objectif : 339.0
- Taux de survie des espèces : (0.9322624 0.9483904 0.957664 0.9804915712 0.9568 0.9802432)
- Tableau des zones protégées :

0	0	0	1	1	1	1	1	0	0
0	0	0	1	2	1	2	1	0	0
0	1	1	1	1	1	2	1	0	0
0	1	2	1	0	1	2	1	1	1
1	1	2	1	0	1	1	1	2	1
1	2	1	1	0	0	0	1	1	1
1	2	1	0	1	1	1	1	0	0
1	1	1	0	1	2	2	1	0	0
1	2	1	0	1	2	1	1	0	0
1	1	1	0	1	1	1	0	0	0

1.3.3 cas n°3 : $\alpha_k = 0.5$ ($k = 1, \dots, 3$), et $\alpha_k = 0.9$ ($k = 4, \dots, 6$)

- Temps de résolution : 0.0750131607055664 s
- Valeur de l'objectif : 130.0
- Taux de survie des espèces : (0.748 0.616 0.874 0.934 0.910 0.912)
- Tableau des zones protégées :

0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	2	1	0	0
0	1	1	1	1	1	2	1	0	0
0	1	2	1	0	1	1	1	0	0
0	1	2	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

1.3.4 cas n°4 : $\alpha_k = 0.8$ ($k = 1, \dots, 3$), et $\alpha_k = 0.6$ ($k = 4, \dots, 6$)

- Temps de résolution : 0.04032301902770996 s
- Valeur de l'objectif : 211.0
- Taux de survie des espèces : (0.932 0.877 0.820 0.972 0.784 0.878)
- Tableau des zones protégées :

0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	2	1	0	0
0	1	1	1	0	1	2	1	0	0
0	1	2	1	0	1	2	1	0	0
1	1	2	1	0	1	1	1	0	0
1	2	1	1	0	0	0	0	0	0
1	2	1	0	1	1	1	0	0	0
1	1	1	0	1	2	1	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

1.4 Étude du comportement du programme linéaire en fonction de la taille des instances

Nous avons implémenté une fonction permettant de générer des instances aléatoirement. Pour ce faire la fonction instancie une grille de taille voulue et pour chaque espèce k tire au hasard les cases sur lesquelles celle-ci a une probabilité $p_{k,i,j}$ de survivre (valeur elle aussi déterminé au hasard). A l'aide de celle-ci, nous avons généré 10 instances de chaque taille pour des valeurs entre $n = 5$ et $n = 50$.

Nous présentons ainsi l'évolution du temps mis par l'algorithme pour retourner une solution en fonction de la taille de l'instance en figure 1.4 :

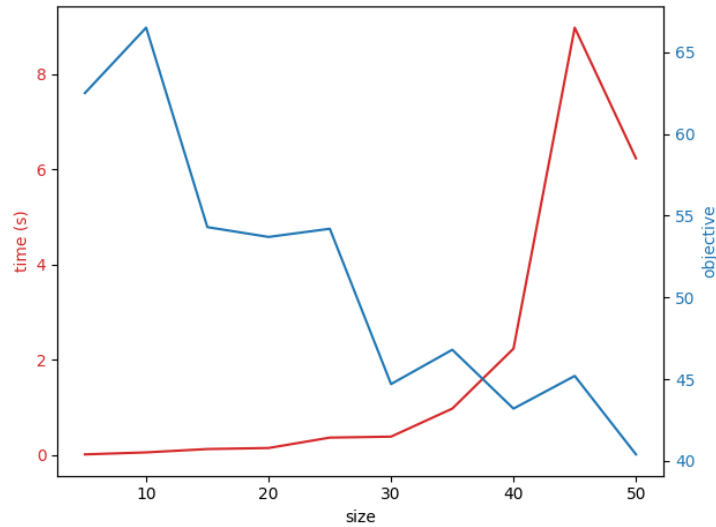


FIGURE 1 – Évolution du temps de résolution en fonction de la taille de l'instance

On remarque tout d'abord qu'à mesure que la taille augmente, le temps de résolution moyen pour une instance augmente également, ce qui est cohérent. De plus, la valeur moyenne de l'objectif retournée sur ces instances décroît à mesure que la taille de l'instance augmente, ce qui semble indiquer que l'algorithme parvient de plus en plus à trouver une solution dont le coût est minimal, ou du moins cela indique peut-être que plus la zone est grande, plus il est aisé de protéger des espèces du fait du nombre de possibilité pour créer des zones centrales et tampons.

A titre d'information, nous avons également souhaité tracer l'évolution du temps de résolution, ainsi que l'objectif moyen atteint pour des instances où le nombre d'espèces augmente, tout autre paramètre étant fixé par ailleurs (figure 2). Là encore, on génère aléatoirement 10 instances (de taille $n=10$) dont le nombre d'espèces varie entre 2 et 36 :

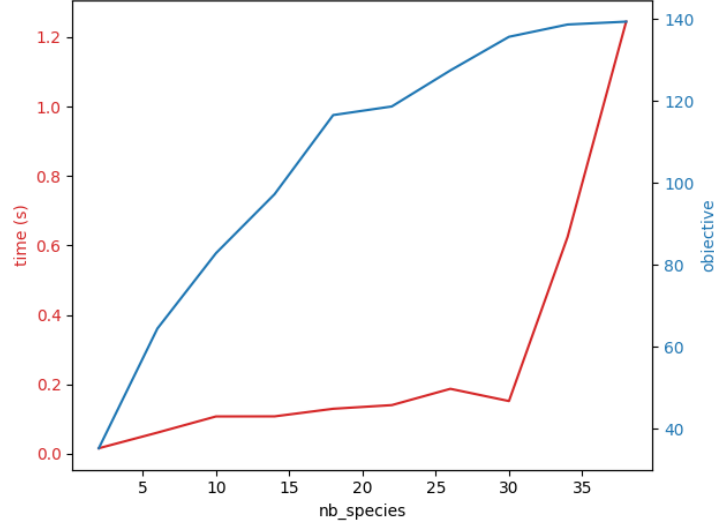


FIGURE 2 – Évolution du temps de résolution en fonction du nombre d'espèces

Dans ce cas, nous remarquons qu'à mesure que le nombre d'espèces augmente, le temps mis pour résoudre une instance semble croître faiblement jusqu'à 30 espèces environ. Ensuite, le temps augmente de manière plus drastique. De plus, le coût moyen par instance augmente lui aussi avec le nombre d'espèces : cela peut s'expliquer par le fait que pour protéger davantage d'espèces, il est nécessaire de créer plus de zones centrales et tampons, ce qui coûte nécessairement plus cher.

1.5 Modification du problème : contrainte budgétaire

On souhaite modifier légèrement le problème afin de déterminer un sous-ensemble de parcelles tel que l'espérance mathématique du nombre d'espèces qui survivront dans la réserve formée soit maximale. Pour ce faire, on écrit le problème comme suit :

$$(P3) \begin{cases} \max \sum_{i,j} \log(1 - p_{k,i,j}) \cdot x_{i,j} \\ \text{s.t.} \\ \sum_{(i,j)} c_{ij} x_{ij} \leq C_{max} \\ 9y_{ij} \leq \sum_{(k,l) \in V(i,j)} x_{kl}, \quad \forall (i,j) \in [[1,n]]^2 \\ \sum_{i,j} \log(1 - p_{kij}) \cdot x_{ij} \leq \log(1 - \alpha_k), \quad \forall k \in C \\ \sum_{i,j} \log(1 - p_{kij}) \cdot y_{ij} \leq \log(1 - \alpha_k), \quad \forall k \in R \end{cases} \quad (10)$$

2 Projet 2 - Maîtrise des effets néfastes engendrés par la fragmentation du paysage

La fragmentation du paysage, c'est-à-dire l'éclatement de grandes superficies en petites parcelles doit être évaluée et contrôlée puisqu'elle influence de nombreux processus écologiques. Elle peut être mesurée par de multiples indicateurs. Ici nous considérons, à titre d'exemple, un indicateur courant, la distance moyenne au plus proche voisin (dmppv). Les problèmes d'optimisation associés à cette notion consistent par exemple à sélectionner, sous certaines contraintes, un sous-ensemble de parcelles optimal vis-à-vis des indicateurs retenus. Les parcelles sélectionnées seront, par exemple, conservées ou restaurées en tant que parcelles habitats et les autres seront affectés à d'autres usages (développement urbain, zones agricoles, coupes forestières, etc.). Voyons maintenant précisément la mesure de fragmentation que nous considérons ici, pour un ensemble de parcelles $S = \{s_1, s_2, \dots, s_n\}$:

$$\text{DMPPV} = \frac{1}{N} \sum_{i=1}^n \min \{d_{ij} : j = 1, \dots, n; j \neq i\} \quad (11)$$

où d_{ij} est la distance entre les deux parcelles s_i et s_j .

Nous considérons ici le problème suivant : soit un ensemble de parcelles $S = \{s_1, s_2, \dots, s_n\}$ d'aires respectives $\{a_1, \dots, a_n\}$ et d'aire totale A . Chaque parcelle a un coût de conservation $\{c_1, \dots, c_n\}$. On souhaite sélectionner un sous-ensemble de S , d'aire totale comprise entre A_{\min} et A_{\max} , de coût total inférieur ou égal à une certaine valeur B et qui minimise DMPPV.

2.1 Modélisation du problème

Dans notre cas on s'intéresse à des instances carrées de côté n . Ainsi toutes les grandeurs (s, a, c) seront doublement indicées par $(i, j) \in [1, n]^2$.

Par ailleurs, on introduit les variables :

- $x_{i,j} \in \{0, 1\} \quad \forall (i, j) \in [1, n]^2$ telles que $x_{i,j} = 1$ si et seulement si la parcelle (i, j) est retenue.
- $z_{i,j,k,l} \in \{0, 1\} \quad \forall (i, j, k, l) \in [1, n]^4$ telles que $z_{i,j,k,l} = 1$ si et seulement si la parcelle (i, j) est retenue et la parcelle (k, l) est la parcelle non-coupée la plus proche de (i, j)

Etant donné qu'on cherche à minimiser l'indicateur DMPPV le problème peut se mettre sous la forme :

$$\min_{x \in \mathcal{X}} \frac{1}{N} \sum_{(i,j,k,l) \in [1,n]^4} d_{i,j,k,l} \cdot z_{i,j,k,l} \quad (12a)$$

Or N est le nombre de parcelles, autrement dit : $N = \sum_{(i,j) \in [1,n] \times [1,n]} x_{ij}$, on réécrit donc l'objectif comme :

$$\min_{x \in \mathcal{X}} \frac{\sum_{(i,j,k,l) \in [1,n]^4} d_{i,j,k,l} \cdot z_{i,j,k,l}}{\sum_{(i,j) \in [1,n]^2} x_{i,j}}$$

Sous les contraintes :

$$\sum_{ij} x_{ij} \cdot a_{ij} \geq A_{\min} \quad (12b)$$

$$\sum_{ij} x_{ij} \cdot a_{ij} \leq A_{\max} \quad (12c)$$

$$\sum_{ij} x_{ij} \cdot c_{ij} \leq B \quad (12d)$$

$$\forall (i, j) \in [1, n]^2, \quad \sum_{(k,l) \in [1,n] \times [1,n]} z_{i,j,k,l} = x_{ij} \quad (12e)$$

$$\forall (i, j, k, l) \in [1, n]^4, \quad z_{i,j,k,l} \leq x_{k,l} \quad (12f)$$

$$\forall (i, j) \in [1, n]^2, \quad z_{ij,ij} = 0 \quad (12g)$$

La contrainte (11d) porte sur le coût de fragmentation du paysage qui ne doit pas dépasser la valeur B, et les contraintes (11b) et (11c) réfèrent à la plage d'aire totale disponible pour les parcelles retenues, respectivement supérieure à A_{min} , et inférieure à A_{max} .

Les contraintes (11e) à (11g) garantissent la cohérence des $z_{i,j,k,l}$:

- (11e) : si $x_{i,j} = 1$ un unique $z_{i,j,k,l}$ est pris en compte pour le calcul de la distance, si $x_{i,j} = 0$ aucun est pris en compte.
- (11f) : pour que $z_{i,j,k,l}$ puisse être retenu il faut que $x_{k,l} = 1$
- les deux contraintes précédentes et le fait que l'on minimise imposeront le choix du $z_{i,j,k,l}$ correspondant à la plus petite distance.
- (11g) : impose que on ne peut pas tenir compte la partielle elle même pour calculer la distance minimale.

2.2 Réécriture du problème sous forme de PO combinatoire fractionnaire

Le problème peut alors se mettre sous la forme d'un problème d'optimisation combinatoire fractionnaire, c'est à dire s'exprimant comme le ratio de deux fonctions linéaires à valeurs dans $[0, 1]^n$:

$$\begin{cases} \max \frac{f(z)}{g(z)} \\ \text{s.t.} \\ z \in Z \subseteq [0, 1]^n \end{cases} \quad (13)$$

En ajoutant les contraintes définies précédemment.

Dans notre cas, on a :

$$\begin{aligned} \text{— } f(z) &= \sum_{(i,j,k,l) \in [1,n]^4} d_{i,j,k,l} \cdot z_{i,j,k,l} \\ \text{— } g(z) &= \sum_{(i,j) \in [1,n]^2} x_{ij} = \sum_{(i,j,k,l) \in [1,n]^4} z_{i,j,k,l} \end{aligned}$$

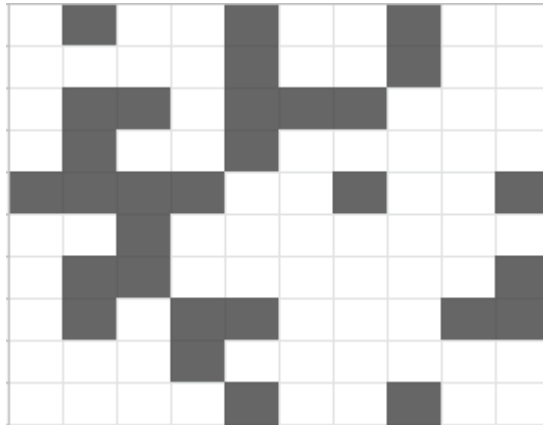
On peut donc appliquer l'algorithme de Dinkebach pour résoudre notre problème.

2.3 Résolution par algorithme de Dinkelbach

Dans cette partie nous présentons les résultats des 3 instances proposées lorsqu'on les résout avec l'algorithme de Dinkelbach :

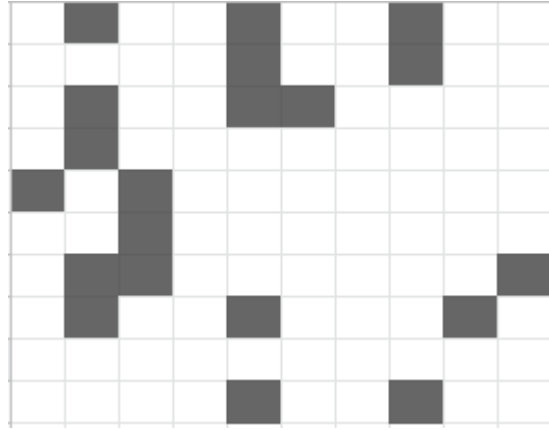
— **instance 1 : A min = 30, A max = 35, B = 920**

- Temps de résolution : 1.86835s
- Nombre d'itération de l'algorithme : 2
- DDPMV : 1.15501
- Parcelles retenues :



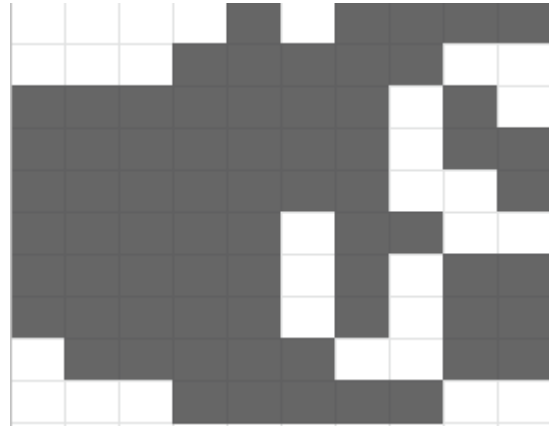
— **instance 2 : A min = 20, A max = 21, B = 520**

- Temps de résolution : 1.28064s
- Nombre d'itération de l'algorithme : 2
- DDPMV : 1.27394
- Parcelles retenues :



— **instance 3 : A min = 70, A max = 75, B = 3500**

- Temps de résolution : 1.49244s
- Nombre d'itération de l'algorithme : 2
- DDPMV : 1.0
- Parcelles retenues :



2.4 Analyse des résultats et comportement de la méthode en fonction de la taille des instances

Nous avons implémenté une fonction permettant de générer des instances aléatoirement. A l'aide de celle-ci, nous avons généré 10 instances de chaque taille, dont la taille varie de 4 à 16.

Nous présentons ainsi l'évolution du temps mis par l'algorithme pour retourner une solution et le nombre d'itération de l'algorithme de Dinkelbach nécessaire pour trouver cette solution en fonction de la taille de l'instance en figure 3 :

On remarque que le temps de recherche de la solution augmente au fur et à mesure que la taille de l'instance augmente, ce qui est cohérent. De plus, le nombre d'itération moyen (sur les 10 instances résolues par taille) qu'il a fallu via l'algorithme de Dinkelbach pour retourner une solution vaut en général 2 (sauf pour quelques tailles d'instances, où certaines ont nécessité 3 itérations, d'où une valeur moyenne à 2.1, ou 2.2).

Par ailleurs, on a aussi relevé les valeurs de DDPMV pour chaque taille d'instance, et pour toutes

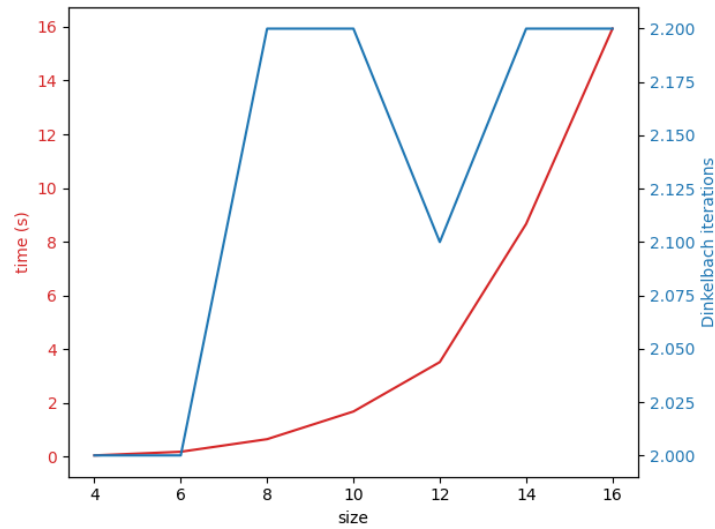


FIGURE 3 – Evolution du comportement de la méthode en fonction de la taille de la grille

les instances où l'algorithme a rendu une solution, cette dernière présentait une DDPMV de 1.0, soit la valeur optimale. Enfin, pour toutes ces mêmes instances, aucune n'a dépassé la `time.limit`, i.e. n'a nécessité plus de 30s pour converger vers une solution.

3 Projet 3 - Protection de la diversité génétique

On considère un problème de conservation de la diversité génétique. Il consiste à déterminer la contribution (nombre de progénitures) optimale de chaque individu de façon à minimiser la perte d'allèles dans la population engendrée. On considère une population formée de P individus, I_1, I_2, \dots, I_P , appartenant à une espèce donnée et un ensemble de M gènes polyalléliques à prendre en compte, g_1, g_2, \dots, g_M . Pour chaque gène g_i , les allèles possibles sont $a_{i,1}, a_{i,2}, \dots, a_{i,t(i)}$. On se place dans le cas de cellules diploïdes (être humain et la plupart des animaux) où, pour chaque gène, chaque individu a deux allèles, chaque allèle correspondant à un chromosome de la paire considérée. Ces allèles peuvent être identiques ou non. Chaque individu reçoit au hasard un des deux allèles de chacun de ses parents. On cherche à déterminer le nombre optimal de progénitures de chaque individu de façon à minimiser l'espérance mathématique du nombre d'allèles disparus dans la génération suivante tout en conservant une population de taille constante.

3.1 Modélisation du problème

On introduit N la variable aléatoire comptant le nombre d'allèles disparus dans la prochaine génération d'individus, et $\forall i \in [[1, M]]$, $X_{i,t}$ la variable aléatoire booléenne valant 1 si l'allèle t du gène i disparaît dans la prochaine génération d'individus. De plus, on note $n = (n_1, n_2, \dots, n_p)$ le vecteur de \mathbb{N}^p tel que n_j est le nombre de progénitures/enfants de l'individu j . Chaque $X_{i,t}$ dépend de n , puisque la probabilité que l'allèle t du gène i disparaisse dépend directement du phénotype des individus et donc du nombre d'enfants qu'ils ont.

Le problème peut donc se mettre sous la forme :

$$\min_{n \in \mathbb{N}^p} \mathbf{E}[N] = \min_{n \in \mathbb{N}^p} \sum_{i=1}^M \sum_{t=1}^{t(i)} \mathbf{E}[X_{i,t}(n)] \quad (14)$$

sous certaines contraintes que nous exprimerons plus tard.

Tout d'abord, on exprime la probabilité que l'allèle $a_{i,t}$ disparaisse. Pour un individu j , la probabilité que l'allèle s'exprime chez sa progéniture est fonction du nombre d'allèle $a_{i,t}$ qu'il possède. Posons n_j le nombre d'enfants (progénitures) de l'individu j et $k_{i,t,j}$ le nombre d'allèle $a_{i,t}$ que possède l'individu j . Alors :

$$\mathbf{P}(\text{allèle } a_{i,t} \text{ disparaît chez la progéniture de } j) = (1 - 0.5k_{i,t,j})^{n_j} \quad (15)$$

Alors, la probabilité que l'allèle $a_{i,t}$ disparaisse lors de la prochaine génération vaut :

$$\mathbf{P}(\text{Allèle } a_{i,t} \text{ disparaît de la prochaine génération}) = \prod_{j=1}^p (1 - 0.5k_{i,t,j})^{n_j} \quad (16)$$

On a ensuite envie de passer au logarithme, afin de faire tomber la puissance n_j . Le problème d'une telle modélisation réside dans le fait que l'on obtiendrait $1 - 0.5k_{i,t,j} = 0$ dès lors que $k_{i,t,j}$ vaut 2, soit lorsque l'individu j possède deux allèles $a_{i,t}$ ce qui nous empêche de passer au log.

Etudions la question autrement : on effectue une disjonction de cas selon le nombre d'allèles $a_{i,t}$ du gène g_i dont dispose l'individu j :

Si l'individu j dispose de :

- 0 allèle $a_{i,t}$ ($k_{i,t,j} = 0$) : la probabilité que l'allèle disparaisse vaut 1
- 1 allèle $a_{i,t}$ ($k_{i,t,j} = 1$) : la probabilité vaut 0.5^{n_j}
- 2 allèles $a_{i,t}$ ($k_{i,t,j} = 2$) : la probabilité vaut 0

Ainsi, l'on est tenté d'introduire les variables $Z_{i,t} \geq 0$, $\forall (i, t)$ telle que :

$$\forall j, \quad Z_{i,t} = \begin{cases} 0 & \text{si } j \in I_t^2 \text{ et } n_j > 0 \\ \prod_{j \in I_t^1} 0.5^{n_j} & \text{sinon} \end{cases} \quad (17)$$

où $I_{i,t}^m$ désigne l'ensemble des individus tels qu'ils possèdent m allèle(s) $a_{i,t}$ du gène g_i , pour $m \in 0, 1, 2$.

Cependant, cette disjonction doit être linéarisé. Pour ce faire on pose :

$$\forall(i, t), \quad Z_{i,t} \geq \prod_{j \in I_t^1} 0.5^{n_j} - \sum_{j \in I_t^2} n_j \quad (18)$$

Dans cette formulation, si aucun parent possédant deux fois l'allèle n'a d'enfant, le second membre se réduit à $\prod_{j \in I_t^1} 0.5^{n_j}$ comme on le souhaite, si un parent possédant deux fois l'allèle a un enfant, ce second membre sera négatif, et donc la contrainte de positivité sur les $Z_{i,t}$ et le fait que l'on minimise fixera la variable considérée à 0, ce qui est bien l'effet recherché.

L'objectif peut alors se réécrire :

$$(P2) \begin{cases} \min_{n \in \mathbf{N}^p} \sum_{i,t} Z_{i,t} \\ \text{s.t.} \\ \forall(i, t), \quad Z_{i,t} \geq 0 \\ \forall(i, t), \quad Z_{i,t} \geq \prod_{j \in I_t^1} 0.5^{n_j} - \sum_{j \in I_t^2} n_j \end{cases} \quad (19a)$$

La dernière contrainte n'est cependant pas linéaire. On procède donc comme suit.

On introduit une nouvelle variable $y_{i,t}$ $\forall i, t$ telle que :

$$y_{i,t} = \prod_{j \in I_t^1} 0.5^{n_j} \quad \forall i, t \quad (19b)$$

Pour l'inclure dans le problème, et puisque l'on est dans le cadre d'une minimisation, $y_{i,t}$ prendra sa valeur même si l'on remplace le signe "=" par ">=" dans la contrainte si dessus :

$$\forall(i, t) \quad y_{i,t} \geq \prod_{j \in I_t^1} 0.5^{n_j} \quad (19c)$$

Le produit pose problème, on passe donc au logarithme :

$$\forall(i, t) \quad \ln(y_{i,t}) \geq -\ln(2) \sum_{j \in I_t^1} n_j \quad (19d)$$

On obtient alors une nouvelle formulation du problème que voici :

$$(P3) \begin{cases} \min_{n \in \mathbf{N}^p} \sum_{i,t} Z_{i,t} \\ \text{s.t.} \\ \forall(i, t), \quad Z_{i,t} \geq 0 \\ \forall(i, t), \quad Z_{i,t} \geq y_{i,t} - \sum_{j \in I_t^2} n_j \\ \forall(i, t) \quad \ln(y_{i,t}) \geq -\ln(2) \sum_{j \in I_t^1} n_j \end{cases} \quad (19e)$$

La dernière contrainte posée dans le problème (P3) pose encore un soucis elle n'est toujours pas linéaire. Pour s'en "débarrasser", on procède comme expliqué dans le sujet :

$$\log(y_{i,t}) \geq -\log(2) \sum_{j \in I_t^1} n_j \quad (20a)$$

On pose $f(n_1, \dots, n_N) = -\log(2) \sum_{j \in I_t^1} n_j$

Alors, l'ensemble de contraintes (C) constitue une relaxation de cette dernière contrainte :

$$(C) : \log(\theta_r) + \frac{1}{\theta_r} (y_{i,t} - \theta_r) \geq f(n_1, \dots, n_N) \quad (r \in H) \quad (20b)$$

où $H = \{1, 2, \dots, h\}$ et θ est un vecteur donné de \mathbb{R}^h tel que $0 < \theta_1 < \theta_2 < \dots < \theta_h = 1$.

Ainsi, la relaxation de ces contraintes permet d'approximer par des tangentes le logarithme de $y_{i,t}$, pour tout i, t , et d'obtenir par la même occasion un problème linéaire en variables mixtes.

3.2 Résolution de l'instance, analyse des résultats

Nous résolvons l'instance du problème proposée, dont les paramètres sont les suivants :

- Nombre d'individus dans la population : 8 (4 mâles/4 femelles)
- Nombre de paires de chromosomes (par individu) considérées : 1
- Nombre de locus considérés par chromosome : 5
- Nombre d'allèles par gène : 2
- Nombre de morceaux pour l'approximation du logarithme : 50
- $\theta_1 = 0.001$

Dans chacun des cas (limitation de la progéniture à 3, puis à 2), nous donnons le temps mis pour résoudre l'instance du problème, la valeur de l'objectif obtenue (correspondant à l'espérance du nombre d'allèles perdus lors de la prochaine génération), et la probabilité de perte des différents allèles.

3.2.1 Cas n°1 - Limitation de la progéniture à 3

$X = (0.0 \quad 3.0 \quad 3.0 \quad 2.0 \quad 3.0 \quad 3.0 \quad 1.0 \quad 1.0)$

Valeur de l'objectif : 0.01558809645330513

Probabilité de perte des allèles (dans l'ordre) : (0.0 0.0 0.0 0.0156 0.0 0.0 0.0 0.0 0.0 0.0)

3.2.2 Cas n°2 - Limitation de la progéniture à 2

$X = (2.0 \quad 2.0 \quad 2.0 \quad 2.0 \quad 2.0 \quad 2.0 \quad 2.0 \quad 2.0)$

Valeur de l'objectif : 0.06243336502896749

Probabilité de perte des allèles (dans l'ordre) : (0.0 0.0 0.0 0.0624 0.0 0.0 0.0 0.0 0.0 0.0)

3.3 Comportement du PL en fonction de la taille des instances

Dans cette section, on étudie le comportement du problème en fonction de divers paramètres. Nous avons fait varier un paramètre à la fois, tous les autres paramètres restant égaux par ailleurs : le nombre d'individus dans la population (figure 4), le nombre de gènes présents dans la population (figure 5), le nombre d'allèles présents dans la population (figure 6), ainsi que le nombre de morceaux pris pour approximer le logarithme dans la contrainte et effectuer la relaxation (figure 7). Pour chacune des figures, on a généré de manière aléatoire 10 instances, les résultats présentés sont moyennés sur ces instances.

On observe que lorsque l'on fait varier le nombre d'individus dans la population, l'espérance du nombre d'allèles perdus diminue jusqu'à atteindre 0, ce qui peut se comprendre aisément. En effet, si une population est très grande pour un nombre d'allèles constant, il sera plus simple de conserver ces allèles au cours de la prochaine génération. De plus, le temps pris pour résoudre les instances croît avec la taille de celles-ci, ce qui là encore semble logique car on augmente la dimension du problème.

On observe que lorsque l'on fait varier le nombre de gènes présents dans la population sans changer le reste des paramètres, l'espérance du nombre d'allèles perdus varie de manière non-monotone. On observe que les valeurs d'objectif sont globalement assez faibles quand bien même elles varient beaucoup. En effet, si le nombre de gènes augmente drastiquement, et si la répartition de la progéniture est effectuée correctement parmi les individus, pour un nombre d'allèles par gène constant, le nombre d'allèles perdus sera faible. De plus, le temps pris pour résoudre les instances croît avec la taille de celles-ci, ce qui là encore semble logique car on augmente la dimension du problème.

On observe que lorsque l'on fait varier le nombre d'allèles par gène présents dans la population sans changer le reste des paramètres, l'espérance du nombre d'allèles perdus croît beaucoup. En effet, pour 9 allèles différents par gène, on perdra visiblement en moyenne 20 allèles dans la génération future. Cela se comprend bien : contrairement au cas où un gène supplémentaire se retrouve dans la population avec 2 allèles disponibles et où les allèles peuvent tous deux être exprimés, un allèle supplémentaire

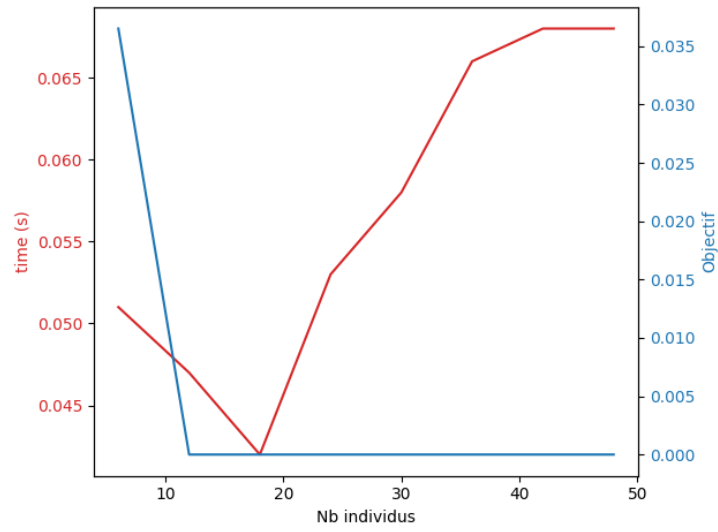


FIGURE 4 – Evolution du comportement de la méthode en fonction de la taille de la population

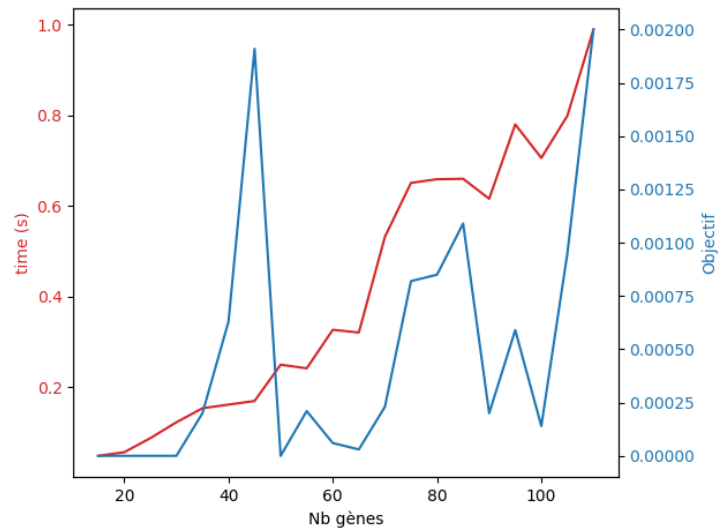


FIGURE 5 – Evolution du comportement de la méthode en fonction du nombre de gènes présents dans la population

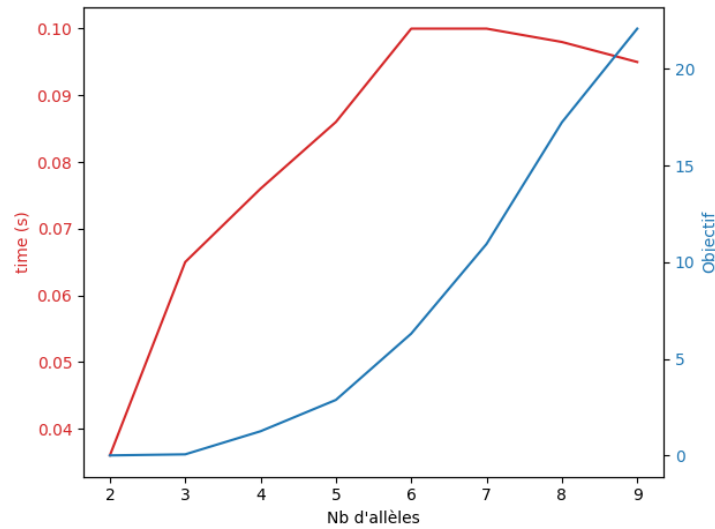


FIGURE 6 – Evolution du comportement de la méthode en fonction du nombre d'allèles présents dans la population

dans la population pour un gène ne sera pas forcément exprimé, et donc on est plus susceptible de perdre ces allèles dans la génération future. De plus, le temps pris pour résoudre les instances croît avec la taille de celles-ci, ce qui là encore semble logique car on augmente la dimension du problème.

On observe que lorsque l'on fait varier le nombre de morceaux de droites pour approximer le logarithme présent dans la dernière contrainte, tout paramètre égal par ailleurs. On assiste donc à la résolution de la même instance, mais en raffinant au fur et à mesure l'approximation que l'on fait du logarithme. Ainsi, on observe une augmentation de l'objectif au fur et à mesure que l'on augmente le nombre de morceaux de droites. Cela s'explique par le fait que lorsque l'approximation est grossière, on obtient une contrainte moins restrictive, et donc on peut obtenir de "meilleures solutions", sans pour autant que celles-ci soient réellement admissibles pour le problème (cela nécessiterait vérification de la solution a posteriori). Ainsi, on obtient des solutions moins bonnes à mesure que l'on s'approche de la véritable contrainte.

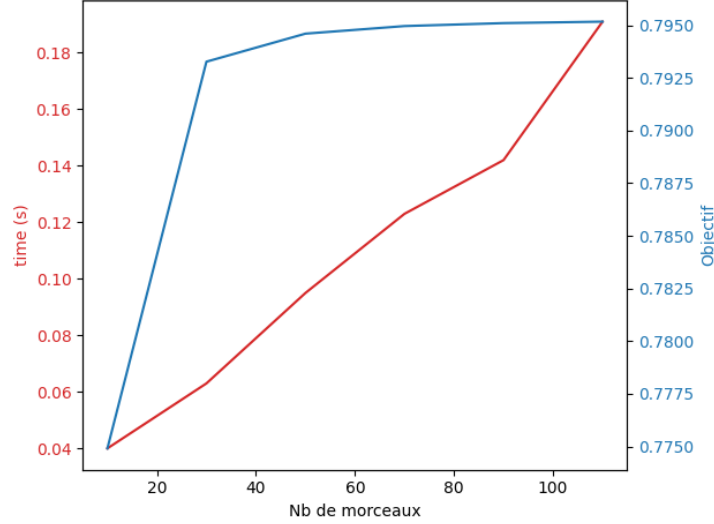


FIGURE 7 – Evolution du comportement de la méthode en fonction du nombre de morceaux pour l'approximation du logarithme dans la contrainte n°3

4 Projet 4 - Exploitation durable de la forêt

Nous présentons dans ce paragraphe un modèle d'exploitation de la forêt visant à protéger le mieux possible certaines espèces. On considère un ensemble de parcelles forestières carrées et identiques représenté par une matrice $m \times n$ et deux types d'espèces e_1 et e_2 . On note l la longueur du côté des parcelles. L'habitat de l'espèce e_1 se situe principalement dans les parcelles coupées et l'habitat de l'espèce e_2 est constitué essentiellement des lisières entre parcelles coupées et parcelles non coupées. On considère la zone représentée par la matrice comme une zone non coupée et la zone extérieure à la matrice comme une zone coupée. La population attendue de l'espèce e_1 dans chaque parcelle s_{ij} coupée (resp. non coupée) est égale à t_{ij} (resp. 0) et la population attendue de l'espèce e_2 est égale à gl où g désigne la population attendue de l'espèce e_2 pour chaque kilomètre de lisière et L , la longueur totale de lisière compte tenu des coupes réalisées.

4.1 Explication du problème linéaire en variables mixtes

Le problème posé est le suivant :

$$(P1) \begin{cases} \max w_1 \sum_{(i,j) \in M \times N} t_{ij} (1 - x_{ij}) + w_2 gl \sum_{(i,j) \in M \times N} (4x_{ij} - d_{ij}) \\ \text{s.t.} \\ d_{ij} \geq \sum_{(k,l) \in A_{ij}} x_{kl} - |A_{ij}| (1 - x_{ij}) \\ d \in \mathbb{R}_+^{M \times N} \\ x \in \{0, 1\}^{M \times N} \end{cases} \quad (i, j) \in M \times N \quad (21)$$

où $M = 1, \dots, m$, $N = 1, \dots, n$, w_1 et w_2 sont les coefficients de pondération, l est la longueur du côté de chaque parcelle et A_{ij} désigne l'ensemble des couples (k, l) tels que la parcelle s_{kl} est adjacente à la parcelle s_{ij} .

- $d_{i,j}$ correspond au nombre de parcelles coupées autour de (i, j) .
- L'objectif consiste à maximiser les populations des deux espèces en pondérant le rôle joué par chaque espèce dans l'objectif par les coefficients w_1 et w_2 .
- La première contrainte impose la bonne valeur de $d_{i,j}$ en fonction des cases adjacentes à (i, j) .

4.2 Proposition d'un modèle quadratique en variables 0-1

Dans l'objectif du problème (P1), on peut s'arranger pour compter autrement les lisières entre les zones coupées et non-coupées. En effet, on peut compter chaque lisière en exprimant le fait que la zone adjacente est coupée et l'autre non-coupée, ou l'inverse. De plus, on discrimine selon si la lisière est horizontale ou verticale. Ainsi :

$$\sum_{(i,j) \in N \times M} 4x_{ij} - d_{ij} = S_1 + S_2 + S_3 + S_4 \quad (22)$$

où

$$S_1 = \sum_{i=1}^n x_{i,0} + x_{i,m} \quad (23a)$$

$$S_2 = \sum_{j=1}^m x_{0,j} + x_{n,j} \quad (23b)$$

$$S_3 = \sum_{i=1}^n \sum_{j=1}^{m-1} x_{i,j}(1 - x_{i,j+1}) + (1 - x_{i,j})x_{i,j+1} = \sum_{i=1}^n \sum_{j=1}^{m-1} x_{i,j} + x_{i,j+1} - 2x_{i,j}x_{i,j+1} \quad (23c)$$

$$S_4 = \sum_{i=1}^{n-1} \sum_{j=1}^m x_{i,j}(1 - x_{i+1,j}) + (1 - x_{i,j})x_{i+1,j} = \sum_{i=1}^{n-1} \sum_{j=1}^m x_{i,j} + x_{i+1,j} - 2x_{i,j}x_{i+1,j} \quad (23d)$$

S_1 représente le nombre de lisières verticales sur l'adhérence (bords extérieurs) de la forêt, S_2 représente le nombre de parcelles horizontales sur l'adhérence de la forêt, S_3 représente les lisières intérieures verticales de la forêt, et S_4 les lisières intérieures horizontales de la forêt.

Le problème s'écrit alors sous la forme d'un modèle avec des variables 0-1, mais ce dernier est quadratique.

4.3 Linéarisation du programme

Il faut donc linéariser le problème pour que celui-ci comporte des variables 0-1 et n'ait plus de termes quadratiques. Chaque produit de variables peut être remplacé par une nouvelle variable binaires égale au produit des deux variables. On augmente ainsi la dimension du problème, mais on enlève son caractère quadratique.

Ainsi, on introduit des variables y_{ij}^h (produits concernant les lisières horizontales, autrement dit $y_{ij,(i+1)j}$) et y_{ij}^v (produits concernant les lisières verticales, autrement dit $y_{ij,i(j+1)}$) pour remplacer les produits de variables binaires dans les expressions (17c) et (17d) précédentes. On obtient alors :

$$\sum_{(i,j) \in N \times M} 4x_{ij} - d_{ij} = S_1 + S_2 + S_3 + S_4 \quad (24)$$

où

$$S_1 = \sum_{i=1}^n x_{i,0} + x_{i,M} \quad (25a)$$

$$S_2 = \sum_{j=1}^m x_{0,j} + x_{N,j} \quad (25b)$$

$$S_3 = \sum_{i=1}^n \sum_{j=1}^{m-1} x_{i,j} + x_{i,j+1} - 2y_{ij}^h \quad (25c)$$

$$S_4 = \sum_{i=1}^{n-1} \sum_{j=1}^m x_{i,j} + x_{i+1,j} - 2y_{ij}^v \quad (25d)$$

Ensuite, il convient d'introduire des contraintes portant sur les nouvelles variables afin de linéariser complètement notre problème. Pour chaque $y_{i,j}^h$, et chaque $y_{i,j}^v$, on pose les contraintes suivantes :

$\forall (i, j) \in N \times M \setminus \{m\}$:

$$y_{i,j}^h \geq x_{i,j} + x_{i+1,j} - 1 \quad (26a)$$

$$y_{i,j}^h \leq x_{i,j} \quad (26b)$$

$$y_{i,j}^h \leq x_{i+1,j} \quad (26c)$$

$$y_{i,j}^h \geq 0 \quad (26d)$$

$\forall (i, j) \in N \setminus \{n\} \times M$:

$$y_{i,j}^v \geq x_{i,j} + x_{i,j+1} - 1 \quad (26e)$$

$$y_{i,j}^v \leq x_{i,j} \quad (26f)$$

$$y_{i,j}^v \leq x_{i,j+1} \quad (26g)$$

$$y_{i,j}^v \geq 0 \quad (26h)$$

Reste à garantir la convexité du problème. Le coefficient w_2gl dans l'objectif étant positif, et les coefficients devant les y_{ij}^h et y_{ij}^v , étant négatifs (i.e. -2), la maximisation de l'objectif sera possible si l'on minimise les valeurs des y_{ij} . Cela permet notamment de supprimer de notre problème les contraintes (20b),(20c),(20f), et (20g).

Exprimons maintenant les contraintes restantes sous forme matricielle : On introduit Z la matrice concaténant les variables y_{ij}^h , y_{ij}^v , et les variables x_{ij} . Alors, le système des contraintes se résume sous la forme :

$$CZ \leq b \quad (27)$$

où

$$C = \begin{pmatrix} A & -I_{n(m-1)+(n-1)m} \\ 0_{n(m-1)+(n-1)m, n^2} & -I_{n(m-1)+(n-1)m} \end{pmatrix} \quad (28)$$

Avec :

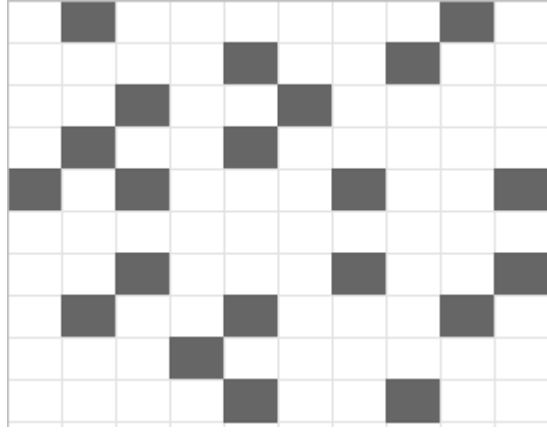
- A la matrice d'incidence (de taille $(n(m-1) + (n-1)m)$, n^2) du graphe composé des parcelles de forêt, reliées entre elles si elles sont adjacentes (grille). Ce graphe est biparti.
- I la matrice identité (de taille $(n(m-1) + (n-1)m)$ rendant compte des coefficients devant les y_{ij} dans les contraintes (20a) puis (20e).
- 0 la matrice nulle (de taille $(n(m-1) + (n-1)m, n^2)$)

4.4 Résolution, analyse des résultats

Dans cette partie, nous présentons les résultats obtenus avec CPLEX pour l'instance proposée avec les 2 méthodes différentes (méthode mixte et méthode quadratique avec linéarisation).

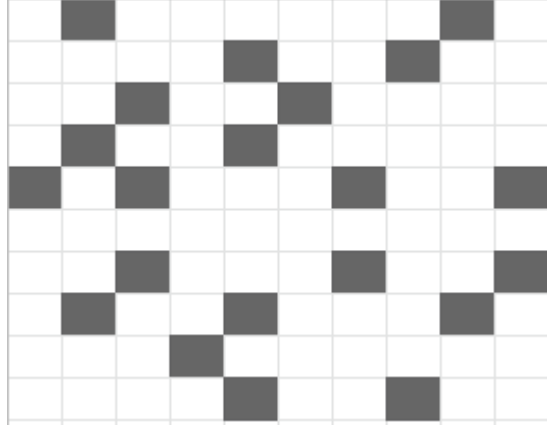
Méthode mixte

- Temps mis pour obtenir la solution : 0.0489s
- Valeur de l'objectif : 8219.5782
- Solution obtenue :



Méthode quadratique avec linéarisation

- Temps mis pour obtenir la solution : 0.0162s
- Valeur de l'objectif : 8219.5782
- Solution obtenue :



On remarque que les deux solutions rendues par l'une et l'autre des méthodes testées sont identiques, et ont donc la même valeur d'objectif. Toutefois, la deuxième méthode semble bien plus rapide, puisqu'elle a rendu la solution en 4 fois moins de temps que la première. Ceci reste à vérifier en effectuant la résolution sur un nombre plus important d'instances.

4.5 Ajout d'une contrainte sur le nombre de parcelles non-coupées

Dans cette partie, on rajoute une contrainte sur le nombre de parcelles non-coupées : ce nombre doit être supérieur à 60.

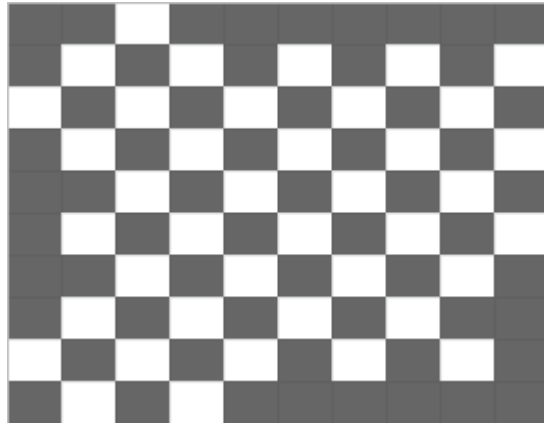
La contrainte se modélise comme suit :

$$\sum_{i=1}^n \sum_{j=1}^n x_{i,j} \geq 60 \quad (29)$$

On présente les résultats obtenus ci-après :

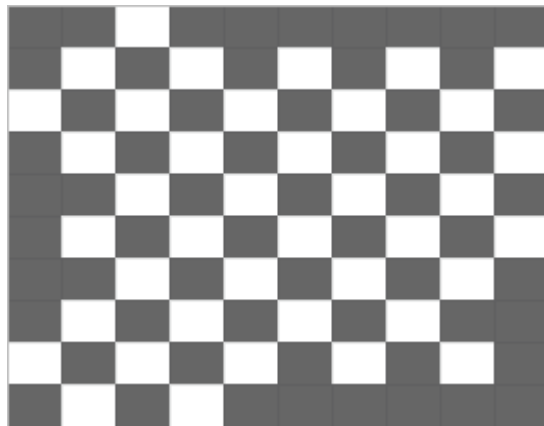
Méthode mixte

- Temps mis pour obtenir la solution : 0.229s
- Valeur de l'objectif : 6753.933
- Solution obtenue :



Méthode quadratique avec linéarisation

- Temps mis pour obtenir la solution : 0.025s
- Valeur de l'objectif : 6753.933
- Solution obtenue :



On remarque que les deux solutions rendues par l'une et l'autre des méthodes testées sont identiques, et ont donc la même valeur d'objectif. Toutefois, la deuxième méthode semble encore une fois plus rapide puisqu'elle a rendu la solution en environ 10 fois moins de temps que la première. Ceci reste à vérifier en effectuant la résolution sur un nombre plus important d'instances.

4.6 Etude du comportement des approches en fonction de la taille des instances

Dans cette partie, nous présentons les résultats obtenus pour les deux méthodes proposées sur 10 instances générées aléatoirement, pour chacune des tailles d'instance testées. La taille varie entre 10 et 55. Les résultats sont visibles figures 8 et 9).

Les résultats nous fournissent des informations sur le comportement de chacune des méthodes de résolution en fonction de la taille de l'instance. A première vue, on peut d'ores et déjà affirmer que les courbes obtenues semblent assez similaires : l'évolution des quantités mesurées, à savoir le temps moyen mis par l'algorithme pour obtenir la solution et la valeur d'objectif moyenne atteinte par ces solutions, est la même pour les deux méthodes. En effet, le temps de calcul et la valeur de l'objectif augmentent au fur et à mesure que la taille de l'instance augmente. Cela apparaît logique pour le temps car la dimension du problème augmente, et pour la valeur croissante de l'objectif aussi puisqu'avec une instance plus grande on peut naturellement augmenter les espaces où les deux espèces vivent. Il pourrait être intéressant de diviser l'objectif par le nombre de parcelles dans la grille pour connaître l'apport en objectif moyen par parcelle, ainsi on aurait une idée plus claire de l'évolution de l'objectif en fonction de la taille de l'instance.

En divisant chacune des valeurs d'objectifs par le nombre de cases composant l'instance, on obtient :

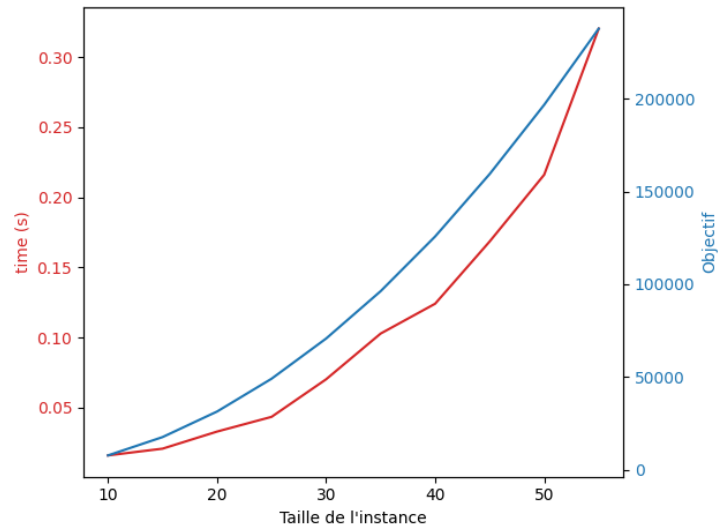


FIGURE 8 – Evolution du comportement de la méthode mixte en fonction de la taille de l'instance

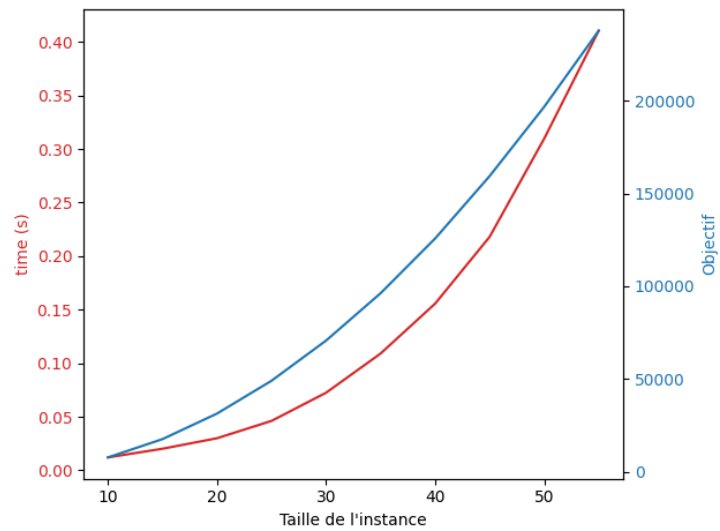


FIGURE 9 – Evolution du comportement de la méthode quadratique en fonction de la taille de l'instance

(78.438 78.707 78.678 78.642 78.654 78.606 78.604 78.716 78.674 78.575)

On remarque alors que l'objectif moyen apporté par chaque case est sensiblement le même quelle que soit la taille de l'instance.