École Centrale Lyon

Projet option
Modélisation du problème d'optimisation d'une exploitation forestière

Optimisation exploitation forestière

Étudiants :
Baptiste Desarnauts
Maxime Peter
Théo Malka-Lacombe
Julien Verdun

Commanditaire :
Charles De Rambuteau

Tuteurs :
Stéphane Derrode
Alexandre Saidi

25 mars 2021

Table des matières

1 Modèle 1

1.1 Version 0

Équation à résoudre :

$$Max\left(\sum_{t=1}^{T}\left(\sum_{z=1}^{Z} x_{zt}(R_{zt}-C_{zt})\right)\right) \tag{1}$$

Liste des conditions :

$$\forall t, \sum_{z=1}^{Z} x_{zt} v_{zt} = V_t \tag{2}$$

$$V_t \leq G_t + \delta_t \tag{3}$$

$$V_t \geq G_t - \delta_t \tag{4}$$

$$x_{zt} = y_{zt} \mathbb{1}_{[M-\lambda;M+\lambda]}(a_{zt}) \tag{5}$$

Avec :

- t : chaque période de temps
- z : chaque parcelle
- x_{zt} : la décision de couper ou non la parcelle z à la période t
- R_{zt} : le revenu engendré par la coupe de la parcelle z à la période t
- C_{zt} : le coût de coupe de la parcelle z à la période t
- v_{zt} : le volume produit par la parcelle s à la période t
- G_{pt} et δ_{pt} : les bornes du flux de production
- M : la maturité des Douglas (50 ans)
- λ : le lambda toléré pour la maturité des Douglas
- a_{zt} : l'âge de la parcelle z à la période t

Matrice des décisions :

$$X = [x_{zt}]_{zt} \tag{6}$$

1.2 Version 1

Une perspective d'amélioration du modèle est de découper les différentes parcelles (z) en Ω sous parcelles. Chaque sous parcelle à une taille minimale de m.

Si on a le choix de replanter une sous parcelle seulement quand toutes les sous parcelles d'une parcelle sont plantées alors on peut modifier la matrice X précédente en remplaçant les 0 et les 1 par des pourcentages de coupes.

Sachant qu'on a les contraintes additionnelles suivantes, en notant S_z la taille de la parcelle z

$$x_{zt} \geq \frac{S_z}{m} = m_z \tag{7}$$

$$x_{zt} \leq 1 - \frac{S_z}{m} = M_z \tag{8}$$

2 Modèle 2

Optimisation sous contraintes : minimisation de la variance de génération en génération avec contraintes sur les années de décalage de coupe.

$$\mathrm{X} = \begin{pmatrix} x_{1,1} & x_{2,1} & x_{3,1} & x_{4,1} & .... & x_{n+3,1} & x_{n+4,1} \\ x_{1,2} & x_{2,2} & x_{3,2} & x_{4,2} & .... & x_{n+3,2} & x_{n+4,2} \\ & & & ... & & & \\ x_{1,j} & x_{2,j} & x_{3,j} & x_{4,j} & .... & x_{n+3,j} & x_{n+4,j} \\ & & & ... & & & \\ x_{1,n} & x_{2,n} & x_{3,n} & x_{4,n} & .... & x_{n+3,n} & x_{n+4,n} \end{pmatrix}$$

$$X = (X_1, X_2, .., X_{n+2}, X_{n+3}, X_{n+4}) \text{ et } X_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \\ ... \\ x_{i,n-1} \\ x_{i,n} \end{pmatrix}$$

X_i la répartition des quantités d'arbre à couper l'année i.
$\sum_{j=1}^{n} X_{i,j}$ la quantité d'arbre effectivement coupée l'année i.
$x_{i,j}$: quantité d'arbre (en hectare) d'âge nominal de coupe à blanc d'année j coupé avec |i-j-2| année d'avance / retard.
Ainsi on a les propriétés suivantes :

- somme sur les lignes : quantité d'arbre arrivant à maturité à l'année j (année de début de simulation + j)

$$\forall j \in [\![1, n]\!] : \sum_{i=1}^{n+4} x_{i,j} = M_j$$

- somme sur les colonnes : quantité d'arbre effectivement coupé à l'année i (année de début de simulation -2 + i)

$$\forall i \in [\![1, n+4]\!] : \sum_{j=1}^{n} x_{i,j} = X_i$$

2.1 Fonction objectif

X définit précédemment, vecteur de taille (1,n+4), quantité à couper par année, on le limite à la prochaine génération (3 à n+2).

$$\boxed{\mathrm{f} : \mathrm{X} \to \frac{1}{n} \sum_{i=3}^{n+2} (X_i - \bar{X})^2}$$

f représente la variance de la distribution des quantités d'arbre à couper chaque année de la génération suivante. On cherche à **minimiser cette grandeur** puisque l'on souhaite obtenir une répartition constante de la quantité d'arbre par année. On cherche donc X qui minimise cette quantité :

$$X^* = argmin_X(f(X))$$

2.2 Fonctions contraintes

De plus : On peut décaler la coupe d'une parcelle de plus ou moins deux ans.

$$\forall j \in [\![1, n]\!] : x_{i,j} = 0 \Leftrightarrow j \leq i - 3 \cup j \geq i + 3$$

On optimise génération par génération :

$$\forall i, j \in \{1, 2, n + 3, n + 4\} \times [\![1, n]\!] : x_{i,j} = 0$$

X devient :

$$\mathrm{X} = \begin{bmatrix} 0 & 0 & x_{3:5,1} & & & & 0 & 0 & 0 \\ & & & x_{2:6,1} & & & & & \\ \vdots & \vdots & & & \ddots & & & \vdots & \vdots \\ & & & & & x_{n-1:n+3,1} & & & \\ 0 & 0 & 0 & & & & x_{n:n+2,n} & 0 & 0 \end{bmatrix}$$

On connaît les quantités d'arbre de la génération précédente, notons les Q.

$$Q = (Q_1, Q_2, ..., Q_{n-1}, Q_n)$$

La quantité d'arbre arrivant à maturité à l'année j (année de début de simulation + j) M_j est égale à la quantité d'arbre disponible une génération avant l'année j Q_j.

$$\forall j \in [\![1, n]\!] : \sum_{i=1}^{n+4} x_{i,j} = M_j = Q_j$$

$$\forall j \in [\![1, n]\!] : G(X) = \sum_{i=1}^{n+4} x_{i,j} - Q_j = M_j - Q_j = 0$$

2.3 Résolution avec Scipy.Optimize.Minimize

Contraintes sur les valeurs nulles Contraintes sur la somme des lignes Encadrement de toutes les valeurs entre 0 et valeur maximale.

3 Modèle 3

3.1 Modèle simplifié

Ce modèle est une version simplifié du modèle numéro 1. En effet, dans ce cas nous nous intéressons seulement aux bénéfices qui sont principalement effectués lors de la coupe

à blanc des arbres. Cette coupe à blanc peut être effectué lorsque l'arbre a de 48 à 52. Pour ce modèle, on va chercher à minimiser la fonction objectif suivante :

$$\sum(y_i x_i) - Moyenne \tag{9}$$

Avec :

— y_i : La surface d'arbres d'âge i

— x_i : la proportion de coupe d'arbre d'âge i

dans le cas simplifié la fonction objectif devient :

$$\sum_{i=48}^{52}(y_i * x_i) - moyenne \tag{10}$$

3.2 Résolution

La résolution utilise la librairie Scipy de Python qui permet de faire de l'optimisation sous contrainte. La résolution est faite de façon itérative. Chaque itération correspond à une année et chaque année on calcule la surface, en hectare, à couper et à planter à l'année suivante. Pour cela, nous avons défini les contraintes suivantes :

Contraintes

— Fonction objectif > 0

— $x_{52} = 1$ afin d'être sur de ne pas avoir du bois de plus de 52 ans.

3.3 Limites

Comme nous l'avons vu, ce modèle est une version simplifié de la réalité. Une des principales difficultés rencontrés est due aux contraintes fixées pour la résolution. En effet, le but à terme est d'avoir la même quantité d'arbre à couper chaque année. Cependant, pour la résolution du modèle on doit fixer une contrainte qui oblige l'algorithme à couper tous les arbres qui ont 52 ans. Donc lorsque la courbe est stabilisée à la moyenne, tous les arbres sont coupés à 52 ans bien que l'exploitant souhaite couper ses arbres à 50 ans.

4 Génération du plan de coupe

4.1 Nécessité de générer un plan de coupe

Comme nous l'avons vu précédemment, les différents modèles étudiés et mis en place fournissent une surface par âge d'essence et par année. Il est donc à ce stade nécessaire de récupérer la sortie de ces algorithmes afin de déterminer quelles surfaces (ou quelles portion de surfaces) il faut effectivement couper pour que ces modèles soit utilisable par l'exploitation.

Le but est donc de récupérer à partir d'un json de la forme suivante un nouveau json donnant pour chaque année une liste de parcelles a traiter avec une certaine surface a traiter pour chacune.

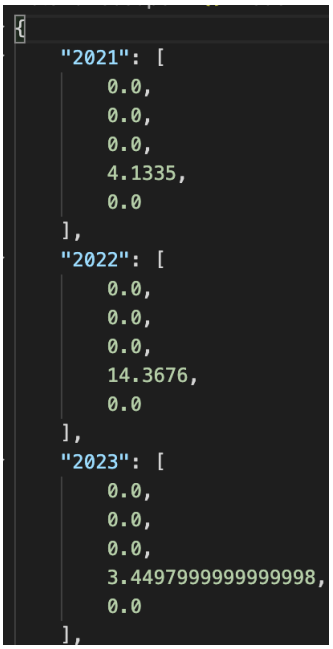

FIGURE 1 – Sortie des modèles (.json)

On notera qu'a une année a, on associe un tableau de longueur 5, cela correspond à la surface totale d'essences âgées de $[48ans, 49ans, 50ans, 51ans, 52ans]$. Cela est due aux normes légales quand à l'âge des arbres.

4.2 Définition des contraintes

Afin que cet algorithme soit efficace, on fait le choix d'imposer les contraintes suivantes :

1. Couper des parcelles entières tant que cela est possible.
2. Couper les parcelles les plus grandes en premier.
3. Si il est nécessaire de couper une partie seulement d'une parcelle, on ne traite qu'une seule parcelle .

Plus précisément, il n'est pas rentable pour l'exploitation d'aller couper une parcelle qui est trop petite, c'est la raison pour laquelle la règle 3 est mise en place. Cela veut dire que si il faut couper une parcelle, on essaye de garder une surface raisonnable telle qu'il est intéressant financièrement d'aller couper. Il n'est donc pas possible de couper plus d'une parcelle d'un âge donné à une année donnée pour des raisons de rentabilité.

4.3 Choix d'implémentation

Afin de commencer à implémenter ce code indépendamment des modèles (pour des raisons d'efficiences), il a été fait sous la forme d'un notebook **jupyter**.

De plus, les modèles donnent la surface à couper sur plusieurs centaines d'années, néanmoins, on a fait le choix de ne donner le plan de coupe que sur cinquante ans, ce qui simplifie grandement l'implémentation et n'est pas un problème pour notre commanditaire.

4.4 Détail du code

4.4.1 Récupération des données et formatage

On commence par récupérer les deux jeux de données dont on a besoin pour générer le plan de coupe :

1. Les résultats du modèle choisi...
2. Les données de l'exploitation forestière **simu.json**.

On utilise ensuite une fonction **data_format()** qui va transformer les données de **simu.json** en un json dont la clé est l'année de plante et la valeur est un autre json avec la surface totale des parcelles plantée cette année là ainsi que le détail des surfaces des parcelles. Cela donne finalement un json de la forme suivante :

```
'1988': {'total': 2.1852, 9: 1.4195, 170: 0.3232, 244: 0.1395, 353: 0.303},
'2019': {'total': 0.38039999999999996, 13: 0.1475, 326: 0.2329},
'2000': {'total': 2.4942, 14: 0.7299, 55: 1.634, 230: 0.0905, 262: 0.0398},
```

FIGURE 2 – Format des données de l'exploitation (.json)

4.4.2 Génération

La fonction **generate_plan()** génère les fonctions souhaitées, on stocke dans un fichier texte les logs de cette fonction afin d'avoir un suivi du plan de coupe établi. Le nom de ce fichier est **logs_plan_de_coupe.txt** et est de la forme suivante :

```
2  ----------------------------------------
3  Annee courante : 2021
4  Annee de plante : 1969
5  Surface a couper : 4.1335
6  Parcelles dispo : {'total': 4.1335, 35: 1.4711, 40: 0.6817, 92: 0.4284, 116: 1.1662, 186: 0.3861}
7  Plan de coupe pour l'annee  : {35: 1.4711, 116: 1.1662, 40: 0.6817, 92: 0.4284, 186: 0.3861}
8  ----------------------------------------
9  Annee courante : 2022
10 Annee de plante : 1970
11 Surface a couper : 14.3676
12 Parcelles dispo : {'total': 14.3676, 180: 0.4481, 193: 12.1005, 196: 0.0542, 275: 1.7648}
13 Plan de coupe pour l'annee  : {193: 12.1005, 275: 1.7648, 180: 0.4481, 196: 0.0542}
14 ----------------------------------------
15 Annee courante : 2023
16 Annee de plante : 1971
17 Surface a couper : 3.4498
18 Parcelles dispo : {'total': 3.4497999999999998, 27: 1.4658, 178: 0.6625, 337: 1.3215}
19 Plan de coupe pour l'annee  : {27: 1.4658, 337: 1.3215, 178: 0.6625}
20 ----------------------------------------
21 Annee courante : 2024
22 Annee de plante : 1972
23 Surface a couper : 3.0272
24 Parcelles dispo : {'total': 3.0271999999999997, 86: 1.1861, 258: 1.1589, 259: 0.6822}
25 Plan de coupe pour l'annee  : {86: 1.1861, 258: 1.1589, 259: 0.6822}
26 ----------------------------------------
27 Annee courante : 2025
28 ----------------------------------------
29 Annee courante : 2026
30 Annee de plante : 1974
31 Surface a couper : 14.1431
32 Parcelles dispo : {'total': 14.143099999999999, 32: 6.8175, 209: 0.4948, 210: 0.0281, 292: 1.4328, 294: 0.0943, 311:
   5.0066, 314: 0.269}
33 Plan de coupe pour l'annee  : {32: 6.8175, 311: 5.0066, 292: 1.4328, 209: 0.4948, 314: 0.269, 294: 0.0943, 210: 0.0281}
```

FIGURE 3 – Extrait de **logs_plan_de_coupe.txt**

On stocke enfin le json contenant les résultats sous le nom **resultat_plan_coupe.json** qui est de la forme suivante :

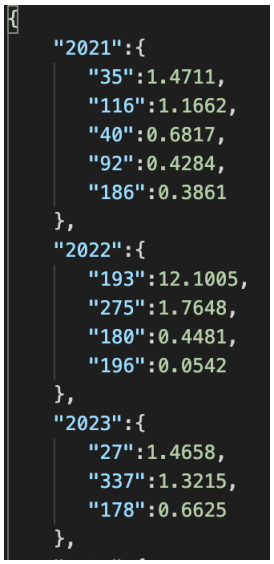

FIGURE 4 – Extrait de **resultat_plan_coupe.json**

4.4.3 Notes

L'algorithme tel quel a été terminé de coder pour le modèle 2 qui, jusqu'à 2070 ne nécessite pas que l'on coupe une partie seulement de parcelles. Si il s'avère que l'on choisit un autre modèle, il faudra faire attention à modifier la fonction **order_parcelles()** (appelée par **generate_plan()**) afin de gérer la coupe partielle des fonctions. Cela n'a pas été traité ici pour des raisons de temps.

On donne quelques pistes pour modifier cette fonction :

- Si variable **total** n'est pas nulle.
- Parcourir les parcelles qui n'ont pas encore été coupées.
- Sélectionner la plus grande.
- Stocker le fait qu'on coupe la surface **total** de cette parcelle dans le json résultat.
- Mettre à jour la surface restant dans le json contenant les informations des parcelles.