

Une introduction à un modèle de gestion de forêt

Térence Bayen*

April 27, 2012

Ce document est un bref résumé d'un modèle de gestion de forêt tiré de la publication [6]. Il peut servir de base aux étudiants ayant suivi ce module d'introduction à la modélisation. La question que l'on se pose est de trouver une stratégie pour exploiter une forêt de façon à optimiser des intérêts économiques tout en préservant la ressource (cad. sans l'épuiser).

Le modèle présenté ci-dessous est relativement classique pour modéliser la gestion d'une forêt et a fait l'objet de nombreux travaux (voir par exemple [4],[6],[5]). Les propriétés du modèle sont tirés principalement de l'article [6]. Le modèle ci-dessous est déterministe, et il existe également de nombreuses extensions au cas stochastique, voir par exemple la publication [5]. Enfin, le livre suivant [2] recense un grand nombre d'exemples de modèle de gestion de ressources naturelles : pêche, forêt, émission du carbone.

Un dénominateur commun à ces problèmes est la formulation mathématique que l'on peut effectuer, notamment grâce à l'utilisation de la **programmation dynamique** (il s'agit d'une théorie mathématique inventée par Bellman dans les années 1960, voir [1]). Ce module a pour but de travailler sur le modèle présenté dans [6] et également d'effectuer une petite introduction à ce concept mathématique qui permet de répondre à la question posée (cad à montrer que certaines stratégies de coupe sont optimales et meilleures que d'autres).

1 Modélisation de la croissance d'une forêt

1.1 Modèle discret

La première question que l'on se pose est la suivante : comment modéliser mathématiquement la croissance, le développement d'une forêt? Quels sont les paramètres importants à prendre en compte?

On aimerait décrire la forêt par un système du type:

$$x(t+1) = H(x(t)) - u(t),$$

où $x(t)$ est le nombre d'arbres à l'instant t , $H(x(t))$ est la croissance des arbres entre le début de l'année t et la fin de l'année t , et $u(t)$ est le nombre d'arbres exploités dans cette période. Mathématiquement, il existe des outils pour étudier correctement ce type de systèmes (comportement...). On va écrire un tel système plus en détail car il semble pertinent d'introduire l'âge des arbres dans le modèle (les arbres trop jeunes ne peuvent être coupés).

On divise la forêt en parcelles : une parcelle contient des arbres du même âge entre i et $i+1$. On note N_i le nombre de ces arbres dans chaque parcelle. Ce nombre $N_i(t)$ dépend de l'année t . Ainsi : $N_i(t)$ = nombre d'arbres dans une parcelle dont l'âge est compris entre i et $i+1$. Il y a (au moins) deux facteurs à prendre en compte:

- la mortalité des arbres,
- la régénérescence des arbres (chaque arbre peut engendrer de nouveaux arbres)

Il existe ainsi une relation entre toutes les quantités $N_i(t)$ à une année t , et toutes les quantités $N_j(t+1)$ à l'année suivante. Cette relation s'écrit sous la forme d'un système dynamique discret entre ces quantités $N_i(t)$

*Université Montpellier 2, Case courrier 051, 34095 Montpellier cedex 5, France (tbayen@math.univ-montp2.fr)

à une année t , et toutes les quantités $N_j(t+1)$ à l'année suivante. Une façon d'écrire l'évolution des arbres est de considérer :

$$N(t+1) = AN(t)$$

où

$$N(t) = \begin{pmatrix} N_n(t) \\ \vdots \\ N_1(t) \end{pmatrix}, \quad A = \begin{pmatrix} 1 - m_n & 1 - m_{n-1} & 0 & \cdots & 0 \\ 0 & 0 & 1 - m_{n-2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 1 - m_1 \\ \gamma_n & \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 \end{pmatrix},$$

Ici m_i représente la mortalité des arbres d'âge entre $i-1$ qui meurent avant d'atteindre l'âge i : $N_i(t+1) = (1 - m_i)N_i(t)$. De plus, les arbres se régénèrent : γ_j est la proportion est arbres nouveaux (d'âge 0) générés par les arbres d'âge $i-1$.

Question mathématique: on a un système dynamique dans \mathbb{R}^n :

$$N(t+1) = AN(t), \quad N(0) = N_0 \in \mathbb{N}_+^n.$$

On peut étudier le comportement de ce système.

- Le système est bien posé? A-t-on un nombre d'arbres entiers, un nombre positif? Que dire sur les coefficients de A pour que $N(t)$ reste toujours un vecteur positif?
- Etudier la convergence de ce système dynamique discret, i.e. la croissance des arbres. Y-a-t-il des cycles (trajectoires périodiques)?

Outils mathématiques : trigonaliser, algèbre linéaire...

Remark 1.1. *Il s'agit d'un modèle qui ne décrit pas forcément de façon exacte la réalité. En particulier, le choix du paramètre de temps (un an) peut être discuté.*

1.2 Modèle continu

Il existe d'autres modèles de gestion de forêt. Le modèle ci-dessous est tiré d'un article mathématique de 1988, voir [3]. Il s'agit d'un modèle continue dont les équations peuvent rappeler des modèles simples en dynamique des populations. On effectue un très bref aperçu de ce modèle. Les hypothèses effectuées dans [3] sont les suivantes.

- Au temps t , on suppose qu'il y a $x(t, \tau)$ arbres par unité de surface dont l'âge est entre τ et $\tau + dt$
- Mortalité : $\mu(\tau)dt$ arbres parmi les arbres d'âge τ meurent entre t et $t + dt$.
- Les arbres d'âge plus grand que τ accroissent la mortalité des arbres d'âge τ par le facteur $\int_{\tau}^T n(\sigma)x(t, \sigma)d\sigma dt$ (on peut penser à une contamination : arbre malade) ; et $T = +\infty$ éventuellement (on ne connaît pas l'âge maximum...on impose $n = 0$ à partir d'un certain âge $\bar{\tau}$). Ainsi il faudra multiplier ce nombre par $x(t, \tau)$ dans le bilan.
- Régénérescence : les arbres d'âge τ produisent entre t et $t + dt$ une quantité $\lambda(\tau)dt$ graines qui commence à pousser. Ainsi dans l'intervalle $[t, t + dt]$ il y a $\int_0^T \lambda(\sigma)x(t, \sigma)d\sigma dt$ arbres d'âge $\tau = 0$. Même remarque qu'avant pour T . D'où si on plante $v(t)$ graines:

$$x(t, 0) = \int_0^t \lambda(\tau)x(t, \tau)d\tau + v(t),$$

Par ailleurs, un bilan dans un petit intervalle de temps fournit la variation d'arbre suivante:

$$x(t + \Delta t, \tau + \Delta \tau) = x(t, \tau) - [\mu(\tau) + \int_{\tau}^T n(\sigma)x(t, \sigma)d\sigma]x(t, \tau)\Delta t \Delta \tau$$

d'où une équation pour la variable inconnue $x(t, \tau)$:

$$\frac{\partial x}{\partial t}(\tau, t) + \frac{\partial x}{\partial \tau}(\tau, t) = -[\mu(\tau) + \int_{\tau}^T n(\sigma)x(t, \sigma)d\sigma]x(t, \tau)$$

L'équation précédente décrit la variation de $x(t, \tau)$ au cours du temps, par ailleurs l'action de l'homme est représentée par le contrôle $v(t)$ qui intervient dans l'équation $x(t, 0)$. La question que l'on se pose est la suivante: comment choisir le contrôle $v(t)$ (i.e. la stratégie de coupe de la forêt à chaque instant t) de façon à maximiser notre profit:

$$\int_0^{T_0} \ell(t, x(t), v(t))dt.$$

Ici T_0 est par exemple un an, deux ans...et $\ell(t, x(t), v(t))$ représente un coût instantané (une utilité qui prend en compte le coût de l'exploitation et le profit).

Remark 1.2. *On étudie en général les liens entre un modèle continu et un modèle discret pour justifier les approximations.*

2 Exploitation par l'homme

On retourne à l'étude du modèle discret. Le fait d'exploiter les arbres d'un certain âge (âge mur pour les arbres que l'on a le droit de couper) et de prendre en compte une éventuelle replantation d'arbres d'âge 0 dans les parcelles qui ont été coupées) modifie le système dynamique précédent:

- On exploite les arbres d'âges mur qui ont n années au moins.
- Lorsqu'une parcelle est récoltée, on replante des arbres d'âge 0.

Soit $h(t)$ la quantité de bois coupée durant l'année t , et $i(t)$ la quantité des nouveaux arbres replantés. Le nouveau système dynamique est:

$$N(t+1) = AN(t) + h(t)b + i(t)c$$

où

$$b = \begin{pmatrix} -1 \\ \vdots \\ 0 \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix}.$$

On ne peut récolter que des arbres d'âge mur à la fin de l'année t . Ainsi, on considère que l'on ne peut couper à la fin de l'année t , uniquement les arbres d'âge n et les arbres d'âge $n-1$ (qui sont coupables à la fin de l'année t car ils ont quasiment un an de plus). Ainsi, la contrainte à prendre en compte dans toute la suite est la suivante:

$$0 \leq h(t) \leq N_n(t) + N_{n-1}(t).$$

Elle traduit que l'on ne peut couper les arbres trop jeunes. Cette contrainte est très importante dans notre modèle.

3 Critère économique

Couper du bois coûte de l'argent. On en gagne en vendant le bois coupé. Cela se modélise à l'aide d'une fonction d'utilité.

- A la fin de l'année t , on vend une proportion $h(t)$ de bois: couper une quantité de bois $h(t)$ coûte $\alpha h(t)$ (c'est proportionnel).
- Le profit est $C(h(t))$ où $C: \mathbb{R} \rightarrow \mathbb{R}$ est une fonction.

A la fin de l'année t , le profit est : $C(h(t)) - \alpha h(t)$. On peut considérer plus généralement que le profit à la fin de l'année t est égal à une fonction $U(t) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ concave, croissante (lorsque le coût de la coupe et le profit sont des fonctions plus compliquées).

On va étaler le profit sur un horizon lointain (p années, voire un horizon infini). En effet, on aimerait trouver une stratégie de coupe qui puisse garantir chaque année un certain revenu, tout en préservant la ressource.

a) **Définition de la fonction d'utilité.**

- On introduit une fonction $U : X \rightarrow \mathbb{R}_+$ appelée fonction d'utilité qui va de l'espace des biens X (ici c'est la quantité de bois, de ressource disponible) vers \mathbb{R}_+ . La fonction d'utilité dépend de chaque individu (ici on suppose qu'on ne change pas de fonction d'utilité).
- Si $x \leq y$ alors $U(x) \leq U(y)$: plus on a de biens, plus on augmente l'utilité qu'on peut avoir de ce bien. La fonction U est croissante: $U' \geq 0$ si U est de classe C^1 .
- On a $U'' \leq 0$. Cela traduit l'aversion à l'égard du risque. L'espérance d'utilité de la richesse est inférieure à sa richesse initiale. On n'est pas sûr de gagner en investissant. Par exemple, si on dispose de $100000F$, l'utilité des $50000F$ premiers euros est plus importante que les autres $50000F$. Ainsi l'utilité de 100000 est inférieure au double de l'utilité de $50000F$: $U(100000) \leq 2U(50000)$. Traduire cette constatation (justifiée économiquement) conduit à considérer une fonction U concave qui vérifie:

$$U(tx + (1 - t)y) \geq tU(x) + (1 - t)U(y),$$

Remark 3.1. Une fonction concave est l'opposée d'une fonction convexe (pour laquelle la corde est au-dessus de la fonction). Par ailleurs, dire que U est concave est équivalent à supposer que sa dérivée seconde U'' est négative.

Une façon équivalente de considérer U concave est de voir que l'individu cherche à maximiser l'espérance de sa richesse, et non sa richesse.

b) **Définition du facteur d'escompte.** Pour résumer, à la fin de l'année t , la quantité $h(t)$ de bois donne lieu à l'utilité $U(h(t))$. On espère l'année suivante $U(h(t+1))$. A la date t , il faut maximiser $U(h(t)) + \beta U(h(t+1))$, puis ainsi de suite on maximise

$$\sum_{k=t}^{t+p} \beta^k U(h(t+k)) \rightarrow \max_{h(\cdot)}$$

Plus généralement, on considère un problème en horizon infini: on cherche à maximiser la fonctionnelle

$$J(h) = \sum_{k=t}^{+\infty} \beta^k U(h(t+k))$$

Explication du facteur d'escompte. On a un bien qui vaut $p(t)$ à une date t . Ce bien s'apprécie chaque année. Son prix est multiplié par $(1 + \delta)$ chaque année (on suppose que δ est fixe ce qui n'est pas le cas en général!). Ainsi au bout de n années, le prix espéré du bien est relié au prix $p(t)$ de départ par :

$$E(p(t+n)) = p(t)(1 + \delta)^n.$$

Par exemple avec $\frac{1}{1+\delta}$ euros aujourd'hui j'aurai demain 1 euros!

L'inconnue est la suite $(h(t))_{t=0,1,\dots}$ à déterminer. On maximise notre profit dans un espace de suite.

4 Modèle final à étudier

Mathématiquement, on a un problème de contrôle optimal discret. On cherche à maximiser un profit économique par rapport à une séquence de décision $(h(t))_{t \in \mathbb{N}}$:

$$\sup_{(h(t))_{t \in \mathbb{N}}} J(h) := \sum_{t=0}^{+\infty} \beta^t U(h(t)),$$

sous la contrainte qu'à chaque année:

$$N(t+1) = AN(t) + h(t)b + i(t)C, \quad N(0) = N_0 \in \mathbb{N}^n, i(t) = h(t)$$

et on a la contrainte $0 \leq h(t) \leq N_n(t) + N_{n-1}(t)$. Ainsi, la variable de décision est la suite $h(t)$ qui est autorisée à prendre ses valeurs dans l'intervalle $[0, N_n(t) + N_{n-1}(t)]$.

Présentation du modèle final. Dans toute la suite, on prend $i(t) = h(t)$: lorsque l'on coupe une parcelle on replante des jeunes arbres. On prendra dans la suite pour simplifier $\gamma_i = 0$ et $m_i = 0$. On admet que ce type de problèmes possède une solution (un moyen assez simple de régler le problème de l'existence d'une solution est de supposer que U est bornée).

Questions :

1) Ecrire le système avec $n = 4$ ou $n = 5$. Montrer que le système vérifie pour tout t $\sum_{1 \leq i \leq n} x_i(t) = S$ où S est le nombre total d'arbres. Montrer que l'ensemble des états possible du système (i.e. l'ensemble des vecteurs $N(t)$ possible est fini).

2) On considère une politique $h : \mathbb{R}^n \rightarrow \mathbb{R}$ qui à un vecteur $N \in \mathbb{R}^n$ associe $h(N)$. Autrement dit, on suppose que h varie en fonction de l'état (ce que l'on appelle un feedback). A chaque instant, on dispose donc du vecteur $N(t)$ et on applique la stratégie de coupe $h(x(t))$. Montrer que le vecteur est périodique : il existe un nombre $T \in \mathbb{N}$ tel que $N(t+T) = N(t)$ pour $t \in \mathbb{N}$.

3) Calculer le cardinal de l'ensemble des états possible. Pour cela, on notera tout d'abord que le nombre de parties à p éléments dans un ensemble à n éléments est C_n^p (avec $p \leq n$). Ensuite, on veut trouver le cardinal de l'ensemble suivant :

$$E = \{(x_1, \dots, x_n) \in \mathbb{N}^p \mid \sum_{1 \leq i \leq n} x_i = p\}$$

On veut montrer que le cardinal de E est C_{n+p-1}^{n-1} . Montrer que le problème revient à trouver le nombre de façons de ranger p chiffons dans n tiroirs, autrement dit de trouver le nombre de mots $t_1 \cdots t_1 t_2 \cdots t_2 \cdots t_n \cdots t_n$ (cad le nombre de combinaisons avec répétition de p éléments parmi n éléments). Pour compter le nombre de ces mots, montrer que l'on doit considérer le nombre de séparateurs entre les t_i , à savoir $n-1$.

Remark 4.1. *Voici une formulation équivalente de ce nombre. On considère $n+p-1$ emplacements pour y placer $n-1$ marqueurs. Le nombre d'emplacements vides entre les i -ème et $i+1$ -ème marqueurs indique le nombre de répétitions de l'objet $i+1$. Celui avant le premier marqueur indique le nombre de répétitions de l'objet 1. Celui après le $(n-1)$ -ème et dernier marqueur indique le nombre de répétitions de l'objet n .*

5 Politique gloutonne

On coupe $x_n(t) + x_{n-1}(t)$ pour chaque t . Calculer cette stratégie lorsque $n = 4$ (système dynamique). Montrer que la stratégie est périodique de période $n-1$ (à partir d'une itération). Calculer sa période. Calculer le coût de cette politique en fonction de la condition initiale. Etant donnée une condition initiale $x^0(x_1, \dots, x_n)$ telle que $\sum_{1 \leq i \leq n} x_i = S$ on peut montrer que le coût de cette stratégie est:

$$J_g(x^0) = \frac{1}{1-\delta^{n-1}} [U(x_n + x_{n-1}) + \sum_{1 \leq j \leq n-2} \delta^j U(x_{n-1-j})].$$

Quel est l'avantage de cette politique? Pour quelle valeur de δ serait-il préférable d'avoir recours à cette stratégie?

6 Politique durable

Il s'agit de trouver la stratégie pour rejoindre en temps fini le vecteur $(0, \sigma, \dots, \sigma)$ où $\sigma = S/(n-1)$. Expliquer sur un exemple ($n = 4$ par ex.) comment atteindre cet équilibre. Il faut discuter si $x_n > \sigma$ (sinon laisser croître la forêt). On admet que l'on a laisser croître suffisamment la forêt pour pouvoir considérer $\sigma = S/(n-1)$. Calculer le coût de cette stratégie. Si on prend $h(t) = e \in [0, \sigma]$, alors le coût de cette stratégie est :

$$J_{dur} = \frac{1}{1-\delta} U(e)$$

Si on prend $e = S/(n - 1)$, quelle est la différence avec la stratégie gloutonne? Quel est l'avantage de cette stratégie? Pour quelle valeur de δ serait-il préférable d'avoir recours à cette stratégie?

7 La programmation dynamique

Dans cette partie, on fait une introduction très brève à la théorie mathématique qui s'appelle la programmation dynamique. Le but est d'étudier l'optimalité des stratégies ci-dessus. On s'intéresse donc à une façon de caractériser mathématiquement les stratégies optimales de coupe pour le problème. L'équation de Bellmann en programmation dynamique permet de répondre exactement à cette question.

7.1 Retour sur le système

On écrit le système dynamique sous la forme

$$x(t + 1) = Ax(t) + h(t)b, \quad 0 \leq h(t) \leq x_n(t) + x_{n-1}(t),$$

où

$$A = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ \vdots \\ 1 \end{pmatrix}.$$

La fonction valeur du problème est définie par:

$$v(x^0) = \sup_{(h(t))_{t \in \mathbb{N}}} J(h) := \sum_{t=0}^{+\infty} \beta^t U(h(t)).$$

Autrement dit on se fixe une condition initiale x^0 , i.e. une population d'arbres dans une forêt. On définit $v(x^0)$ comme le supremum par rapport à toutes les stratégies de coupe du critère évalué sur cette stratégie. La fonction $v(x^0)$ n'est pas simple d'usage et ne se calcule pas a priori. Elle possède cependant de nombreuses propriétés théoriques. Si U est borné par exemple, on peut facilement montrer qu'elle est bien définie. Il faut bien avoir à l'esprit qu'elle dépend uniquement de la condition initiale.

L'équation de Bellmann pour le problème est:

$$w(x) = \max_{e \in [0, x_n + x_{n-1}]} (U(e) + \delta w(Ax + eb)), \quad x \in \mathbb{N}^n.$$

On a le résultat suivant qui caractérise la fonction valeur.

Theorem 7.1. *La fonction valeur du problème est l'unique solution de l'équation de Bellmann.*

Remark 7.1. *On peut illustrer très facilement cette équation sur un graphe pour se rendre d'un point A à un point B : si un chemin optimal de A à B passe par un point intermédiaire M , alors il reste optimal de A à M .*

L'idée dans la suite est la suivante. Supposons que l'on dispose d'une stratégie h de coupe que l'on suppose optimale (par exemple la stratégie gloutonne ou bien la stratégie durable). Si cette stratégie vérifie l'équation de Bellmann et que les autres stratégies ont un coût strictement inférieur, alors le théorème permet de conclure que cette stratégie est optimale.

- 1) Montrer que la stratégie durable vérifie cette équation (on rappelle que le meilleur choix de e constant chaque année pour la stratégie durable est $S/(n - 1)$).
- 2) Montrer que la stratégie gloutonne vérifie également cette équation (le calcul est un peu plus compliqué).
- 3) On admet que la fonction valeur est l'unique solution de l'équation de Bellmann. Comment faudrait-il procéder pour montrer que la stratégie gloutonne est optimale pour certaines valeurs du paramètre δ ?

7.2 Dérivation de l'équation d'Hamilton-Jacobi-Bellmann en général

On appelle parfois équation de Bellmann équation d'Hamilton-Jacobi-Bellmann. Formalisons cette équation en introduisant un problème de contrôle optimal discret. Soit X un espace abstrait (par exemple $X = \mathbb{N}^n$ pour ce qui nous intéresse car le vecteur d'état qui nous intéresse est le nombre d'arbres d'âge entre 1 et n). On se fixe une donnée initiale $x^0 \in X$, et l'on considère un système dynamique discret où à chaque temps t , le successeur dépend d'une stratégie $h(t)$:

$$x(t+1) = f(x(t), h(t)), \quad x(0) = x^0 \in X.$$

Cette équation signifie que lorsque le système est dans l'état $x(t)$, $x(t+1)$ dépend de $x(t)$ et de notre décision à l'instant t donnée par $h(t)$. On suppose ici que la fonction f est de classe C^1 et $f : X \times \mathbb{R} \rightarrow \mathbb{R}$. On suppose également que nos décisions $h(t)$ à chaque année t prennent leur valeur dans un ensemble borné de \mathbb{R} (typiquement on s'autorise à retirer une quantité bornée à l'avance de bois). Plus généralement, le contrôle $h(t)$ peut prendre ses valeurs dans un espace H , par exemple \mathbb{R}^m . Pour simplifier, on a choisi de se restreindre à $H = \mathbb{R}$. La question qui nous intéresse est de trouver une suite $(h(t))_{t \in \mathbb{N}}$ de façon à maximiser un critère économique. Ainsi, on se donne une fonction U de classe C^2 , croissante, concave et bornée (ce qui permet de résoudre le problème de l'existence d'une solution), et on considère le profit économique suivant à maximiser:

$$J(h(\cdot)) := \sum_{t \in \mathbb{N}} \beta^t U(h(t)) \rightarrow \max.$$

On maximise ce critère dans l'espace des suites $(h(t))_{t \in \mathbb{N}}$ qui satisfont les contraintes du problème (par exemple, on a imposé précédemment que $0 \leq h(t) \leq x_n(t) + x_{n-1}(t)$ pour tout t . Dans notre exemple précédent, la suite $h(t)$ représente combien d'arbres murs on coupe chaque année. On va supposer que $h(t)$ prend ses valeurs dans un ensemble $C(x(t)) \subset \mathbb{R}$:

$$\forall t \in \mathbb{N}, \quad h(t) \in C(x(t)).$$

Pour l'exemple des arbres, on a : $C(x) = [0, x_n + x_{n-1}]$.

Dérivation de l'équation de Bellmann. La fonction importante à définir s'appelle la fonction valeur. Elle est définie par:

$$v(x^0) := \max_{h(\cdot)} \sum_{t \in \mathbb{N}} \beta^t U(h(t)) \quad h(t) \in C(x(t)), \quad \forall t \in \mathbb{N},$$

et sous la contrainte $x(t+1) = f(x(t), h(t))$, $x(0) = x^0$ fixé. La fonction valeur dépend de la condition initiale.

Proposition 7.1. *La fonction valeur vérifie le principe de la programmation dynamique:*

$$v(x^0) = \sup_{h \in C(x^0)} [U(h) + \beta v(f(x^0, h))],$$

et c'est l'unique solution de cette équation.

Supposons que l'on sache résoudre cette équation. On connaît ainsi $v(x)$. Pour calculer la stratégie optimale, il faut trouver h qui réalise le maximum dans l'équation précédente.

Remark 7.2. *Attention : dans l'équation précédente, h est un réel qui prend ses valeurs dans $C(x^0)$; pour le problème de forêt, $C(x^0) = [0, x_n^0 + x_{n-1}^0]$.*

La signification de l'équation de Bellmann est la suivante. Le terme $U(h) + \beta v(f(x^0, h))$ représente la somme du coût partant de x^0 lorsque l'on prélève une ressource h et de la fonction valeur évalué sur le successeur de x^0 en prélevant h . L'équation signifie que partant de x^0 , la fonction $v(x^0)$ est précisément égale au maximum de cette somme parmi tous les successeurs possibles depuis x^0 !

Esquisse de preuve de l'équation de Bellmann. Soit x^0 donné dans X . Si $h(t)$ est une stratégie donnée, on a par définition du maximum dans la fonction valeur:

$$v(x^0) \geq U(h(0)) + \sum_{t \geq 1} \beta^t U(h(t)).$$

En passant au sup par rapport à $(h(t))_{t \geq 1}$, on obtient:

$$v(x^0) \geq U(h(0)) + v(f(x^0, h(0))),$$

et on obtient l'inégalité $v(x^0) \geq \sup_{h \in [0, C(x^0)]} [U(h) + \beta v(f(x^0, h))]$ en passant au sup par rapport à $h(0)$. L'autre inégalité se fait de façon analogue en utilisant la définition de la borne supérieure. Soit $\varepsilon > 0$. Il existe une suite $(h(t))$ telle que

$$v(x^0) - \varepsilon \leq \sum_{t \geq 0} \beta^t U(h(t)) = U(h(0)) + \sum_{t \geq 1} \beta^t U(h(t)) \leq \sup_{h \in C(x^0)} [U(h) + \beta v(f(x^0, h))].$$

Il y a deux passages au *sup* dans la seconde inégalité ci-dessus. Ainsi, on a bien $v(x^0) \leq \sup_{h \in C(x^0)} [U(h) + \beta v(f(x^0, h))]$ en prenant $\varepsilon > 0$ arbitraire.

8 Réalisation du TP sous Matlab (matrice de transition, opérateur de Bellman)

1) Créer un premier fichier td.m sous matlab (le script). Affecter les variables **S** et **n**:

```
-->S=9; n=4;
```

et construire les matrices *A*, *B*, et $CA = [1 \ 0 \ \dots \ 0]A$:

```
-->A=
```

```
-->B=
```

puis le vecteur ligne $CA = [1 \ 0 \ \dots \ 0]A$

```
-->CA=
```

Indication : utiliser les fonctions `zeros(,)` et `eye(,)` de Matlab.

2) Construire une fonction

```
function X=comb(S,n)
```

qui à **S** et **n** associe une matrice **X** dont les colonnes représentent l'ensemble des vecteurs possibles de dimension **n** dont la somme vaut **S** (**X** est la matrice de tous les états possible)

```
-->X=comb(S,n)
```

```
X =
```

```
column 1 to 11
```

```
! 9.  8.  8.  8.  7.  7.  7.  7.  7.  7.  6. !
! 0.  1.  0.  0.  2.  1.  1.  0.  0.  0.  3. !
! 0.  0.  1.  0.  0.  1.  0.  2.  1.  0.  0. !
! 0.  0.  0.  1.  0.  0.  1.  0.  1.  2.  0. !
```

```
column 12 to 22
```

Indication : procéder récursivement

3) On va coder chaque vecteur $X(:, i)$ par un nombre entier (cad qu'on va associer à chaque état $X(:, i)$ un unique entier $\sum_{1 \leq k \leq n} X(k, i)S^{k-1}$ (penser à la décomposition en base de *S*):

$$X(1, i) + S * X(2, i) + S^2 * X(3, i) + \dots$$

Pour cela, définir un vecteur ligne de composantes 1, *S*, *S*², ...

```
-->PuissancesS=
```

puis créer une fonction


```
function IndX=cree_index(X)
```

qui à la matrice X fait correspondre un vecteur ligne d'indexation IndX de dimension S^n tel que $\text{IndX}(m)=i$ si $m=X(1,i)+S*X(2,i)+S^2*X(3,i)+\dots$ et 0 sinon (autrement dit IndX est un vecteur tel que si sa coordonnée m est non nul, alors il fournit l'unique état i tel que $m = \sum_{1 \leq k \leq n} X(k,i)S^{k-1}$, et sinon il rend 0).

```
IndX=cree_index(X);
```

Vérifier que si *etat* est un vecteur possible, par exemple

```
-->etat=[7;0;1;1]
```

```
etat =
```

```
! 7. !
! 0. !
! 1. !
! 1. !
```

l'instruction $X(:,\text{IndX}(\text{PuissancesS} * \text{etat}))$ redonne bien le vecteur *etat*:

$$\text{etat} \mapsto \text{IndX}(\text{PuissanceS} * \text{etat}) \mapsto X(:, \text{IndX}(\text{PuissanceS} * \text{etat})) = \text{etat}.$$

4) Création de la matrice de transition. Ecrire en matlab une fonction

```
function [T,NC]=transition(X,A,B,CA)
```

qui calcule la matrice de transition T et le vecteur NC tels que

- la i -ème ligne de la matrice T représente la liste des numéros d'états accessibles en un pas de temps à partir de l'état numéro i , en utilisant respectivement les commandes $e = 0, 1, \dots$ (par convention, un zéro en j -ème colonne signifie que la commande $e = j - 1$ n'est pas admissible pour l'état correspondant).
- le vecteur NC représente le nombre de commandes admissibles (en nombres entiers) pour chaque colonne de X (c'est-à-dire $CAx + 1$ pour chaque vecteur x).

```
-->[T,NC]=transition(X,A,B,CA);
```

Tester la bonne construction, en prenant un état au hasard, par exemple l'état numéroté 50 :

```
-->etat=50;
```

```
-->X(:,etat)
```

```
ans =
```

```
! 4. !
! 1. !
! 0. !
! 4. !
```

```
-->NC(etat)
```

```
ans =
```

```
6.
```

```
-->T(etat,:)
```

```
ans =
```

```
! 31.    52.    80.    116.    161.    216.    0.    0.    0.    0. !
```

```
-->X(:,T(etat,1))
```

```

ans =

! 5. !
! 0. !
! 4. !
! 0. !

-->X(:,T(etat,2))
ans =

! 5. !
! 0. !
! 4. !
! 0. !

```

etc...

5) Vérifier par matlab le nombre d'états du système

```

-->nb_etats=size(X,2)
nb_etats =

220.

```

6) Définir un vecteur F ("feedback") de dimension `nb_etats` (où `nb_etats` désigne le nombre d'états possible) qui à chaque état associe la plus grande commande admissible. Ce feedback correspond à la stratégie gloutonne.

7) On note $\mathcal{S}(S, n)$ l'ensemble des vecteurs en nombres entiers de dimension n et dont les sommes des éléments est égale à S , et on considère une fonction d'utilité de la forme

$$U(e) = e^\beta, \quad \beta \in]0, 1[.$$

On rappelle que l'opérateur de Bellman \mathcal{O} appliqué à une fonction $g(\cdot)$ définie sur $\mathcal{S}(S, n)$ s'écrit

$$\mathcal{O}[g](x) = \max_{e \in]0, CAx]} U(e) + \delta g(Ax + Be), \quad \forall x \in \mathcal{S}(S, n) . \quad (8.1)$$

Définir des valeurs pour les paramètres β et δ , par exemple

```

--> beta=0.9; delta=0.1;

```

et affecter au vecteur U l'ensemble des valeurs possibles pour $U(e)$:

```

--> U=(0:S)^beta;

```

On représentera une fonction $g(\cdot)$ définie sur $\mathcal{S}(S, n)$ par un vecteur G de dimension `nb_etats`. Ainsi l'opérateur de Bellman sera représenté par une fonction qui à un vecteur G fait correspondre un vecteur NG de même dimension. Ecrire une telle fonction, en utilisant la matrice T , les vecteurs NC , U et le nombre `delta` définis précédemment :

```

function NG=Bellman(G)
NG=[];
for i=1:nb_etats
    NG=[NG; ]
end
%endfunction

```

8) Écrire une fonction

```
function V_iter=iterations(V_init,nb_iter)
```

qui à partir d'un vecteur V_init effectue nb_iter itérations de l'opérateur de Bellman.

Choisir un vecteur V_init quelconque, par exemple

```
--> V_init=zeros(nb_etats,1);
```

puis effectuer trente itérations

```
--> V=iterations(V_init,30);
```

et vérifier que le vecteur V ainsi obtenu est une bonne approximation du point fixe de l'équation (8.1), en effectuant une itération supplémentaire.

9) Écrire une fonction

```
function F=feedback(V)
```

qui détermine un vecteur de commandes de dimension nb_etats , qui pour chaque état donne une commande réalisant le max dans l'équation (8.1) pour la fonction donnée par le vecteur V .

Procéder à différentes simulations, et relancer les calculs pour de nouvelles valeurs de β et δ , par exemple

```
--> beta=0.1; delta=0.9;
```

Quelle caractéristique présentent désormais les trajectoires ?

References

- [1] R. BELLMANN, *Dynamic Programming*, Courier Dover Publications, 1957.
- [2] M. DELARA, L. DOYEN, *Sustainable management of natural resources*, Springer, 2008.
- [3] O. HELLMAN *On the Optimal Control of the Growth of a Forest in a Special Case*, *Journal of Optimization*, 1988.
- [4] L. DOYEN, A. RAPAPORT, J.-P. TERREAUX, *A viability analysis for the management of forest systems*, Document de Travail 2001-25, LAMETA, Montpellier.
- [5] A. PIAZZA *The optimal harvesting problem with a land market: a characterization of the asymptotic convergence*, *Economic Theory*, 40, Issue 1, pp. 113-138, 2009.
- [6] A. RAPAPORT, S. SRAIDI, J.-P. TERREAUX, *Optimality of greedy and sustainable policies in the management of renewable resources*, *Optimal Control Applications and Methods*, 2003, 24, pp.23-44.