UNIVERSITE DE YAOUNDE I
Faculté des Sciences
Département d'Informatique
BP 812 Yaoundé-Cameroun

UNIVERSITY OF YAOUNDE I
Faculty of Science
Department of Computer Science
P.O. Box 812 Yaounde-Cameroon

Mise au point d'un simulateur de gestions des ressources naturelles.

Mémoire en vue de l'obtention du
Diplôme de Master en Informatique

présenté et soutenu par

TENDJUI TAKAM ABDEL KARMELA
13R2188

Sous la co-direction de :

Pr Vivien Rossi
Cirad

Dr Moto Serge Mpong
Enseignant/UY1

Yaoundé, 2014

Résumé :

La prédiction forestière de nos jours est une discipline faite de façon récurrente par les chercheurs dans le domaine de foresterie ; Ceci dis, plusieurs modèles de prédiction forestière ont été mis sur pied pour résoudre ce problème de gestion de ressources naturelles. Toutefois, la plupart de ces modèles nécessite pour leurs tests des simulations pour pouvoir analyser les résultats. Certains outils informatiques ont été ainsi développés pour la prédiction mais la plupart le font de façon linéaire. Dans ce mémoire, nous nous proposons d'étudier dans un premier temps un modèle dit le modèle d'Usher et de mettre sur pied un simulateur qui se basera cette fois ci sur le modèle d'Usher stochastique pour pouvoir prédire l'évolution de toutes les espèces dans une forêt et ceci à travers le temps. Le simulateur proposé ici nous donnera à partir de l'effectif des espèces à l'initial l'évolution de ces effectifs pour ces espèces ; ceci en combinant à la fois le modèle en classe de taille et le modèle de Usher stochastique pour l'évolution de l'effectif de ces ressources. Nous développerons donc ce simulateur en JAVA et il devra en sortie nous donner des résultats qui seront stockés dans des fichiers et dont le principal est un fichier CSV. Ce fichier aura un récapitulatif des effectifs de toutes les espèces, dans toutes les classes et donc les résultats pourront être utilisés plus tard pour les interprétations graphiques.

Mots clés : Prédiction forestière, modèle d'Usher stochastique, modèle classe de taille, Simulateur, Ressources naturelles.

Abstract :

The forest prediction nowadays is a discipline made in a recurring way by the researchers in the field of forest prediction ; This say, several models of forest prediction were set up to solve this problem of management of natural resources. However, the majority of these models require simulations for their test, this to be able to analyze their results. Certain data-processing tools were thus developed for the prediction but the majority do it in a linear way. In this memory, we propose to study a model initially says the model of Usher and to set up a simulator which will be based this time on that Usher stochastic model to predict the evolution of all the species in a forest and this through time. The simulator proposed here will give us starting from the manpower of the species to initial the evolution of these manpower for these species ; this by combining at the same time the model in class of size and the Usher stochastic model for the evolution of the manpower of these resources.We will thus develop this simulator in JAVA and it will have at the exit to give us results which will be stored in a CSV file. That file will have a summary of manpower of all the species, in all the classes and thus the results could be used later for graphic interpretations.

Keywords : forest prediction ,usher stochastic model , Class size model , Simulator, Natural Resources

INTRODUCTION

De nos jours, nul ne peut contester le fait que l'informatique est une discipline qui est incontournable et qui s'applique dans presque tous les domaines de la vie. Dans cette dynamique, la prédiction forestière n'en est pas en reste car beaucoup de systèmes informatiques sont conçus pour résoudre des problèmes dans la prédiction de l'évolution des ressources. Des logiciels de simulation développés sur la base des modèles de prédiction ont été faits par la communauté scientifique, à but de faire de la simulation des écosystèmes forestiers. Ces outils informatiques ont permis une plus large diffusion de ces modèles.Il faut pouvoir prédire à long terme l'évolution des populations d'arbres exploitées .Pour ce faire, de nombreux modèles de dynamique des populations ont été développés dans le domaine de la recherche forestière. Ces modèles simulent la croissance, la mortalité et le recrutement des arbres en fonction de variables environnementales plus ou moins explicites et permettent de projeter sur long terme l'évolution du peuplement étudié. En ce sens, ils s'avèrent aujourd'hui des outils de gestion indispensables pour raisonner l'aménagement et la sylviculture des forêts de production. [8]

Leur mise en place devrait notamment permettre d'estimer le taux de reconstitution du stock de bois après exploitation avec une précision donnée.[1] Pour répondre aux attentes des exploitants forestiers. Cependant, l'étude de la dynamique forestière nécessite souvent des analyses à une échelle de temps et d'espace supérieure à celle qu'il est possible d'observer sur le terrain c'est pourquoi il est nécessaire d'avoir recours à la simulation. Nous nous retrouvons en fait dans un environnement qui fait face à de nombreuses catastrophes naturelles et donc beaucoup de système informatique ont été mis sur pied pour pouvoir faire des prédictions dans l'évolution forestière ; Toutefois, la plupart des outils se basent sur des modèles divers de foresterie et touchent ainsi plusieurs aspects dans cette discipline. Seulement, ces outils n'abordent pas la plupart du temps les prédictions dans un sens aléatoire ou stochastique. Rappelons que, non seulement la déforestation participe au réchauffement climatique, mais elle met également en péril la biodiversité animale et végétale. Les forêts concentrent en effet l'essentiel de cette biodiversité : la moitié des espèces terrestres de notre globe vivent dans les forêts tropicales. En détruisant leur milieu naturel, la déforestation les condamne à mort. Le rythme de disparition des espèces est de nos jours très important . A terme, cette éradication du vivant menace l'humanité même.Notons que La déforestation est une des causes du changement climatique, ceci pour deux raisons car d'une part, parce que moins de forêts, c'est moins de CO_2 [1] absorbé donc plus de CO_2 dans l'atmosphère et d'autre part parce qu'en coupant des arbres, l'industrie forestière libère le CO_2 emmagasiné par ces arbres.

Nous nous retrouvons de nos jours dans un environnement qui fait face à de nombreuses catastrophes naturelles et dont plusieurs systèmes informatiques ont été mis sur pied pour pouvoir faire des prédiction dans l'évolution forestière ; Toutefois, la plupart des outils se base sur des modèles divers de foresterie et touchent ainsi plusieurs aspect dans cette discipline.

1. www.greenpeace.org/france/fr/campagnes/forets/problemes/

Seulement, ces outils n'intègrent pas la dimension stochastique ou aléatoire dans les prédictions. Le problème pour nous ici est de concevoir un simulateur qui peut aider à prédire l'évolution de ressources naturelles (Des arbres, des espèces) dans une foret; ceci en se basant sur un modèle matriciel dit le modèle d'usher; A la différence qu'ici, notre outil intégrera la dimension stochastique dans les prédictions .

Nous utiliserons pour notre travail de conception du simulateur des modèles de dynamique de population structurés en classe, donc des modèles matriciels et plus particulièrement le modèle d'Usher qui est très utilisé en foresterie.(cf. Landry Clara [2]). Le choix de ce modèle est motivé par le fait qu'il donne une simplification assez simple du processus de gestion de ressources naturelles, mais aussi par le fait que la fiabilité des prédictions est assez importante. Notre simulateur aura donc comme fondement ce modèle d'Usher pour la prédiction dans la gestion de nos ressources naturelles. Bien que ce modèle nous donne une simplification du processus de gestion, notons qu'il peut être plus difficile et plus complexe si nous tenons compte de certains autres paramètres tel que l'état de peuplement [2] des espèces pour créer cette fois ci un nouveau modèle dis modèle densité-dépendant.[2].

Objectifs globaux

De façon générale, nous dirons que nos objectifs ici sont de plusieurs ordre à savoir : dans un premier temps implémenter grâce au langage Java un outil ou un simulateur basé sur des modèles forestiers de prédiction forestière ceci pour pouvoir garantir ou visualiser l'évolution des ressources naturelles dans un espace forestier. En plus l'outil devra l'ensemble de données issues de la simulation ceci dans l'intérêt de pouvoir faire des sorties et interprétations graphiques.

Objectifs spécifiques

De façon spécifique ici pour nous, il est question de partir initialement avec un effectif d'une ressource naturelle dans une forêt (arbre ou espèces qui seront rangés par classe de diamètre dans le cadre de cette simulation) et de pouvoir après un certains nombre d'année prédire l'évolution de cet effectif au travers de diverses simulations et ceci en respectant les différentes caractéristiques du modèle utilisé pour mener ce travail. Donc nous pouvons énumérez cela par :

a Utiliser un modèle de prédiction forestière pour concevoir le simulateur.

b Prédire grâce à ce simulateur l'évolution de l'effectif d'une ressource dans le temps.

c Tenir compte des différentes lois mathématiques et probabilités intervenant dans la prédiction forestière pour pouvoir faire évoluer cet effectif.

d Pouvoir après un certains nombre d'année déterminer si une espèce disparaîtra ou non.

2. Un peuplement forestier se décrit comme une superficie continue sur laquelle la forêt a des caractéristiques relativement homogènes en matière de composition et de dimension d'arbres.

e Déterminer l'intensité d'exploitation des différentes espèces pour qu'elles puissent se maintenir dans le peuplement.

Ceci dit, voila en quelque sorte les objectifs de cette application qui je précise est comme une interface entre les aménagistes et les ressources naturelles. Pour pouvoir atteindre ces objectifs, notre travail sera subdivisé en trois grandes parties :
La partie I, intitulée « État de l'art dans la simulation des ressources naturelles » nous permettra de comprendre le pourquoi même de notre simulateur et fera un tour sur un certains nombres d'outils faisant dans la simulation de ressources naturelles.
La partie II intitulée « Processus Markovien et modèle du simulateur » qui lui nous présentera le type de processus sur lequel nous nous somme basé pour pouvoir concevoir ce simulateur et nous présentera également le modèle que nous avons étudié, et sur lequel le simulateur tourne selon ses hypothèses. Ensuite la dernière partie « Présentation du simulateur et Résultats » nous présentera un certains nombres de résultats du simulateur et bien évidement les différentes perspectives et une conclusion qui nous fera un rappel sur les principaux résultats souhaités et ceux obtenus.

Première partie

État de l'art dans la simulation des ressources naturelles

Nous noterons ici qu'il n'existe pas beaucoup de simulateurs dans ce domaine de prédiction :Gestion des ressources naturelles. Toute fois nous avons pu ressortir après nos recherches un certains nombres d'outils tels que :

1 StoMat

C'est en fait un **Projet de mise à disposition des aménagistes et des exploitants forestiers d'un modèle matriciel robuste et simple pour estimer la reconstitution d'une ressource**.[7]

C'est un projet financé par le CIRAD et disposait comme membres (Guillaume Cornu, Sylvie Gourlet-Fleury, Nicolas Picard). Pour ce qui est de ses fonctionnalités, StoMat permet d'étudier la vitesse de reconstitution des effectifs d'une espèce ou d'un groupe d'espèces après exploitation forestière. Il met en œuvre un modèle matriciel à paramètres non régulés, mais prend en compte l'imprécision de ces paramètres. StoMat a comme avantage qu'il ne requiert aucune configuration particulière, on peut grâce a lui prédire l'évolution du peuplement et donner une idée de la fiabilité de la prédiction en effectuant des répétitions.Toutefois, il a comme inconvénient le fait que son domaine de prédiction efficace est restreint au données qui ont servis à calculer les paramètres de la prédiction ; en plus du fait qu'il est un logiciel qui fonctionne uniquement sous l'environnement Windows, notons également que l'imprécision de ses paramètres nous conduira à faire de ce modèle un modèle linéaire.

2 Capsis

Capsis [3] est une plate forme logicielle générique donnant accès à des modèles de croissance (croissance et mortalité des arbres), ou de dynamique forestière (qui intègrent, en plus, la dispersion et la régénération) pour des formations forestières variées, plus ou moins hétérogènes, pures ou en mélanges d'espèces, en zone tempérée, mais également subtropicale et tropicale, et pour divers types de gestion. Le logiciel intègre des modèles développés par des équipes de différents instituts de recherche et universités françaises et étrangères. Un tel modèle est un ensemble d'équations mathématiques qui relient les caractéristiques dendrométriques des arbres (hauteur, diamètre, volume) et du peuplement qu'ils constituent, leur nombre, leur âge, et qui

3. Croissance d'Arbres en Peuplement et Simulation d'Itinéraires Sylvicoles.

traduisent leur évolution selon la compétition qui s'exerce entre eux (régulée par les coupes sylvicoles) et selon la fertilité stationnelle (sol, climat, relief). [3]

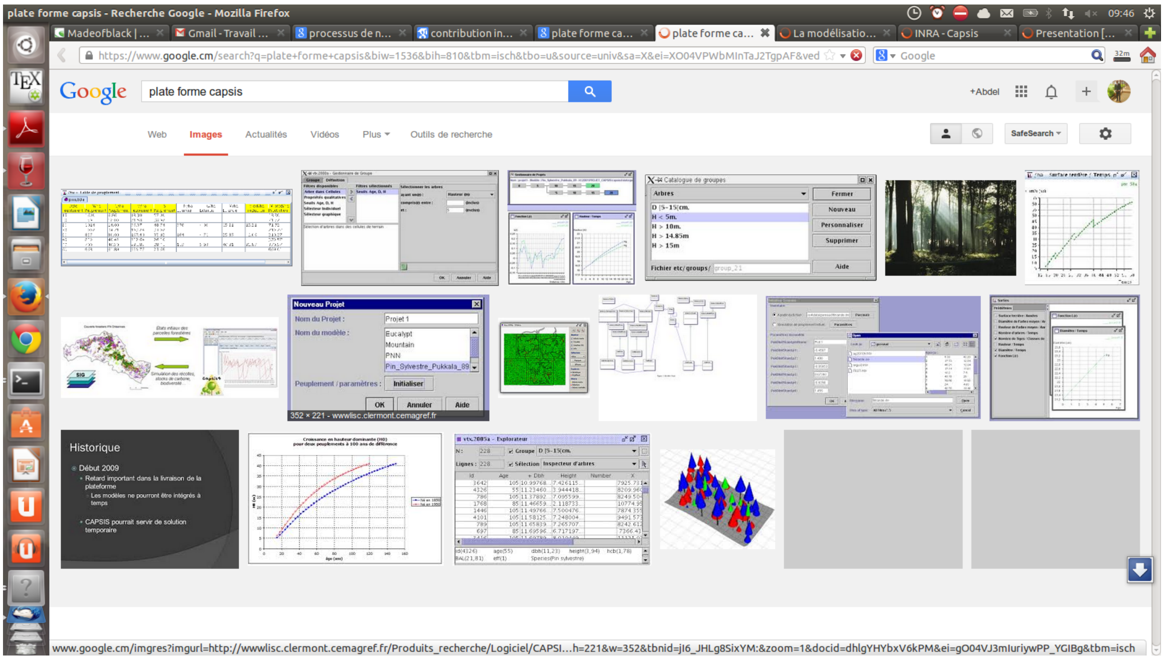


fig1 : présentation de certains résultat de capsis.

Solution proposée

Dans le cadre du déroulement de notre stage, pour notre travail de mémoire nous nous proposons ici de concevoir un simulateur de gestion de ressources naturelles qui aidera pour la prédiction forestière en développant un modèle d'Usher stochastique densité dépendant ; c'est à dire un outil qui pourra aider les analystes ou les exploitants de foret à pouvoir prédire l'évolution de cette dernière au fil du temps, de savoir en fait l'évolution des espèces présentes qui seront rangés par classe de diamètre, d'avoir à tout moment une trace des effectifs d'espèces, arbres etc... Ceci se fera en s'appuyant sur un certains nombres de critères, lois et probabilités mathématiques. Nous utiliserons le langage JAVA pour développer notre simulateur qui est en fait une implémentation basée sur du Usher stochastique. Notons que notre solution à un grand avantage par rapport a d'autre outils basé sous Usher en ceci que nous faisons évoluer tout un peuplement au même moment donc plusieurs espèces contrairement à ces outils qui faisaient évoluer une seule espèce.

Contribution du travail

L'outil conçu sera appliqué a la gestion des forets tropicales et permettra ainsi aux exploitants forestiers d'optimiser leurs pratiques par rapport à différents critères comme la préservation de la biomasse forestière ou la régénération du stock exploitable. Il n'existe pas de façon équivalente un outil pour aider à la gestion des forets tropicales donc ce travail s'inscrit dans le cadre du projet Dynaffor [4] http ://www.Dynaffor.org/ et sera diffusé aux exploitants forestiers du bassin du Congo, et en particulier à ceux du Cameroun par ce biais. Il apportera par la suite des éléments à la réflexion sur la valorisation énergétique des déchets d'exploitation pour l'électrification des zones rurales au Cameroun.

Deuxième partie

Processus Markovien et modèle du simulateur

3 Processus Markovien

Pour modéliser l'évolution au cours du temps (la dynamique) de la plupart des systèmes ,(substance, écosystème), on peut choisir des modèles aléatoires. Les plus simples de ces modèles aléatoires sont les chaînes de Markov que nous définissons de la manière suivante :

Une chaîne de Markov est une suite aléatoire (X_n)=$(X_1$,X_2,...,$X_n)$ avec $n \in N$, prenant des valeurs dans un ensemble E et qui jouit de la propriété de Markov qui stipule que connaissant (X_n), on peut oublier le passé pour prédire l'avenir de cette suite. Dans notre étude, nous nous limiterons au cas ou E est fini.On modélise avec des chaînes de Markov l'évolution au cours du temps de quantités (X) qui peuvent prendre un nombre fini d'états (X)= $(X_1$,X_2,...,$X_n)$ et qui passent de l'état X_i 'a l'instant t à l'état X_j 'à l'instant suivant t + 1 avec une probabilité P_{ij} donnée.Les nombres P_{ij} = P$(X_{t+1} = x_j/\ X_t = x_i\ )$ vérifient donc
0 ≤P$_{ij}$ ≤ 1 et

$$\sum_{j=0}^{n} P_{ij} = 1$$

. [5]

puisque si la chaîne est dans l'état x_i à un instant, elle sera nécessairement dans l'un des états possibles $x_1, \ . \ . \ . \ , x_n$ l'instant suivant et donc $p_{i1} + p_{i2} + ... + p_{in} = 1$. L'expression P$(X_{t+1} = x_j/\ X_t = x_i\ )$ s'appelle une probabilité conditionnelle et représente la probabilité que la quantité X vaille x_j a l instant t + 1 sachant qu'elle valait x_i à l'instant t. On défini ainsi notre matrice de transition Q par : P_{ij} = P$(X_n = j\ |$X$_{n-1} = i\ )$ avec les coefficients de cette matrice ne dependant pas de n.

4. Le projet DynAfFor intervient dans 5 pays de l'espace COMIFAC (Cameroun, Congo, Gabon, République centrafricaine, République Démocratique du Congo).Il repose sur un partenariat entre organismes de recherche, administrations forestières et exploitants forestiers ; il se déroulera sur une première phase de 5 ans, sous l'égide de la COMIFAC

Pour définir une chaîne de Markov il faut donc :

- Un espace des états S := $(x_1,x_2,...,x_n)$ connu que l'on supposera fini.
- La matrice de transition (ou de passage) (P_{ij}),1 $\leq$i$\leq$ n , 1 $\leq$j$\leq$ n. A noter que cette matrice est appelée matrice stochastique parce que ses coefficients sont tous compris entre 0 et 1 et la somme des coefficients de chaque ligne vaut 1 (ce qui n'est pas vrai en général pour les colonnes). [5]

4 Modèle Usher

4.1 Modèle Usher stochastique

Nous nous intéressons dans le cadre de ce travail au modèle d'Usher stochastique. Pour ces modèles matriciels qui sont dits déterministes, nous dirons notre vecteur d'effectif qui représente l'effectif de la ressources a gérer au temps t est lié a celui au temps t + 1 par une relation dite de récurrence de la forme : $N(t+1) = UN(t)$. Les taux de transitions entre états sont collectés dans la matrice de transition (L*L) notée U. On distinguera deux cas de figure ou scénario :

Le premier cas est celui ou les coefficients de la matrice de transition U ne dépendent pas des effectifs N, là on parlera alors de modèle linéaire :$N(t + 1) = UN(t)$. (Modèle de Usher Linéaire) Un second cas dans lequel les coefficients de croissance, mortalité, et la probabilité de rester dans une classe pour une ressource dépendent explicitement des effectifs $N(t)$ dans chaque classe, on parlera alors de modèle densité-dépendant.[2] .On notera $N(t+1) = U(N(t))N(t)$. Ils vont donc devoir être modifié ou mis a jour à chaque temps, on perdra dans ce cas l'hypothèse de stationnarité. Un des avantages majeurs de ce modèle est qu'il permet de modéliser une population avec des traits démographiques simples (fécondité, croissance, mortalité). Notons qu'il existe aussi d'autres modèles densité dépendant à l'instar du modèle de Favrichon qui est un modèle matriciel de dynamique de population de type modèle de Usher, c'est-à-dire pour des populations structurées en taille, avec les probabilités de transition et le taux de recrutement au temps t étant eux-mêmes fonction de l'état du peuplement.[4].

5 Processus stochastique et modèle du simulateur

5.1 Processus stochastique

On considère une suite de variables aléatoires $(\mathrm{X}_n)_{n\in N}$ à valeurs dans un ensemble E fini appelé espace d'état. Supposons que E = 1, 2,.., N. Nous appellerons notre suite de variables un processus stochastique. Car en effet chaque variable peut prendre une valeur de manière aléatoire. Si nous notons π_n la loi de variables aléatoires X_n, $\forall_{n\in N}$ on a :

π_n(j) = $(X_n =$ j), $\forall_{j\in E}$; donc π_n = loi(X_n)

Le vecteur suivant :

π_n = [π_n(1) π_n(2) π_n(3) .. π_n(N)] nous permet donc de représenter la mesure de probabilité π_n qui veut dire
π_n(j) ≥ 0 *et* $\sum_{j\in E}\pi_n$(j) $= 1; Laloiduprocessus(X_n)_{n\in N}$ est la donnée de la famille des probabilités jointes :
p_n (i_0, i_1, i_n) = P ($X_0 = i_0$, $i_1 = i_1$, $X_n = i_n$).[6]

Le processus stochastique $(X_n)_{n\in N}$ sera dit de Markov (d'ordre 1) lorsque :
P($X_{n+1} = i_{n+1}$|$X_0 = i_0$,...,$X_n = i_n$)= P($X_{n+1} = i_{n+1}$|$X_n = i_n$) $\forall_{n\in E}$ et $\forall$$i_0$,$i_1$,...,$i_{n+1} \in E$.

5.2 Modèle du simulateur

5.2.1 Modèle en classe de taille

Ici, nous modéliserons la dynamique d'un arbre au travers de différentes classes de tailles tout en gardant a l'esprit ici qu'une taille en fait égale a :DHP (Diamètre a Hauteur de Poitrine) Soit par exemple un ensemble de 4 classes :
Classe 1 : C1= [20,30[
Classe 2 : C2= [30,40[
Classe 3 : C1= [40,50[
Classe 4 : C1= [50,60]
Associées a 4 états 1,2,3,4 par exemple.
L'adjonction de l'état "0" a ces états représente en fait la mort d'un individu, d'un arbre. Dans ce genre de modèle que nous utiliserons par la suite, un arbre qui grandi passe d'une classe a la classe suivante.Le passage d'une classe a une autre signifie alors un changement d'état. A l'état 4 l'individu reste alors dans cet état ou alors peut disparaître.[6]

Ce modèle de croissance est utilisée, dans les modèles forestiers dits matriciels de Leslie et modèle de Usher.[11] Le modèle de Markov décrit ici est en fait le modèle de croissance d'un individu dans les modèles matriciels.

En résumé : Chaque individu dans notre simulateur appartiendra a une classe représenté ci dessus. Il peut passer d'une classe à la suivante ceci en fonction de sa croissance ou de sa taille.[9]

Pour prédire la dynamique avec le modèle matriciel, nous devons donc passer des trajectoires individuelles à l'évolution de la distribution des individus dans les classes de diamètre.Le fait qu'il n'y ai que trois transitions possibles, pas de sauts de deux classes et pas de retour en arrière le rend très utilisé en foresterie. Cependant, il présente une limitation majeure : il n'intègre aucune information spatiale, mais reste performant pour prédire la dynamique d'un peuplement forestier sur moyen terme.

5.2.2 Hypothèse du simulateur

Notre simulateur est fait un modèle utilisé pour prédire la dynamique forestière. C'est un modèle qui s'appuie sur un certains nombres d'hypothèses a savoir :

- Hypothèse de Markov :l'évolution du peuplement entre 2 temps t et t+1 ne dépend que de son état au temps t.
- Hypothèse indépendance : Les arbres sont indépendants.
- Hypothèse de stationnarité : Le passage d'un arbre d'un état vers un autre dépend du temps.
- Hypothèse d'USHER : A un temps t un arbre peut soit mourir, soit passer à la classe supérieure soit rester dans la même classe. (confert These Dakis [10]).

5.2.3 Présentation détaillée du modèle du simulateur

Notre simulateur décrit l'évolution d'une population d'arbres représentée par un vecteur $N^t = (N_1^t, ..., N_L^t)$ ordonné suivant différentes classes de diamètre croissant. Ainsi le composant l du vecteur,N^t est le nombre d'arbres de la classe de diamètre l au temps t. Chaque classe de diamètre est définie par un vecteur ordonné $(D_0, ..., D_L)$
On aurait pu commencer à (D_1) tout en sachant que (D_0) représente en fait le diamètre que toute recrue possède lors de son recrutement et $D_L = \infty$

Nous nous baserons tout au long de cette conception sur le modèle de classe détaillé plus haut et notre simulateur aura en effet 12 classes de diamètres et le nombre de groupes du modèle sera en fait donner par l'utilisateur ou l'aménagiste lors de l'exécution d'une simulation. Il devra aussi fournir le nombre d'années sur lequel la simulation se fera tout en sachant que nous ferons des simulations suivant les cycles de rotations normales. Notre simulateur comme chaîne de Markov reviens en fait à décrire l'Évolution de la population par une chaîne de Markov sur le vecteur d'effectif N^t c'est pourquoi nous définissons une chaîne de Markov et les processus stochastiques plus haut. L'état futur, en l'occurrence le nombre d'arbres présent par classe, ne dépend du passé que par l'intermédiaire du présent. Notre simulateur restreint les transitions entre les États, en effet entre t et t+1, un individu ne peut ni sauter de plusieurs classes en même temps, ni retomber dans la classe précédente. Donc le changement d'État se fait de façon graduelle. La simulation pour nous ici représentera en fait une fonction aléatoire qui modifie le vecteur N^t entre 2 temps t et t+1.
$Usher(N_1^t, ..., N_L^t | \theta_1, ..., \theta_L) = (N_1^{t+1}, ..., N_L^{t+1})$

Nous introduisons ici les effectifs suivant $N_{l,p}^t$,$N_{l,q}^t$ et $N_{l,d}^t$ qui représenterons respectivement le nombre d'individu dans notre vecteur d'effectif qui respectivement vont rester dans la classe l, sauter pour la classe l+1 et mourir dans la classe l entre deux instants t et t+1.

$Usher(N_1^t, ..., N_L^t | \theta_1, ..., \theta_L) = (N_1^{t+1}, ..., N_L^{t+1})$ avec $N_l^{t+1} = N_{l,p}^t - N_{l,d}^t - N_{l,q}^t + N_{l-1,q}^t$ et $(N_{l,p}^{t+1}, N_{l,q}^{t+1}, N_{l,d}^{t+1}) \sim \mathcal{M}(N_l^t, \theta_l)$. Recru$^t \sim \mathcal{P}(\sum_{l=1}^{L} \lambda N_l^t)$ et $N_{L,q}^t = 0$.

5.3 Processus utilisés pour la simulation

Nous définissons dans cette section les différents processus utilisés lors de la conception de ce simulateur. Notons que les equations et méthodes utilisées ici pour prédire l'évolution des différents processus ont été étudiés attentivement car, dans des modèles qui utilisent des processus que l'on répète fréquemment, une erreur sur un processus causera alors une accumulation d'erreur et du coût un résultat statistiquement incorrect. [12]

5.3.1 Processus de croissance

Dans ce processus de croissance, nous partons d'un vecteur d'effectif qui à l'initial contient les effectifs des espèces par classes de diamètre ou si l'on veut pour un groupe donné ; ensuite nous générons un nouveau vecteur qui représente les effectifs des espèces ayant grandis à l'année suivante pour les différentes classes de diamètre, et donc les valeurs suivent une loi binomiale ; ce nouveau vecteur est généré en fonction de la probabilité de croissance dans ce groupe. Ceci dis à l'année suivante, nous pouvons simuler le processus et avoir le nombre d'espèce ayant grandis par classe de diamètre.

5.3.2 Processus de mortalité

Dans ce processus de mortalité, nous partons d'un vecteur d'effectif qui à l'initial contient les effectifs des espèces par classes de diamètre ; ensuite nous générons un nouveau vecteur qui représente les effectifs des espèces ayant trouvés la mort à l'année suivante pour ces différentes classes de diamètre, et donc les valeurs suivent une loi binomiale ; ce nouveau vecteur est généré en fonction de la probabilité de décès dans ce groupe. Ceci dis à l'année suivante, nous pouvons simuler le processus et avoir le nombre d'espèce qui meurt par classe de diamètre.

5.3.3 Processus de recrutement

Dans notre modèle, n'oublions pas que chaque classe possède une probabilité de décès, donc pour pouvoir garder une stabilité du modèle nous avons besoins de nouvelle recru qui vienne remplacer ceux qui trouvent la mort à un moment donné t. Pour ce processus, nous partirons au départ avec une valeur qui est le nombre de recru pour chaque groupe ; ensuite ces valeurs serons mises à jour après chaque simulation. Recru^t sera modélisé en fonction d'une loi qui est celle de poisson et dépendra linéairement du nombre d'individu présent dans nos différentes classes.

5.4 Remarque

Vu le fait que nous disposons pas assez d'information pour concevoir un modèle par espèce, le modèle des espèces ici sera groupé par processus et par classe de diamètre (12 dans le cas de notre simulateur) ;

6 Proposition d'algorithme par processus

Les algorithmes proposés ci-dessous serons exécutés après avoir fait au préalable les opérations suivantes :

- Générer à l'initial les différents vecteurs d'effectifs des espèces pour toutes les classes de diamètre.
- Générer à l'initial les vecteurs de probabilités de croissance de chaque groupe.

Algorithm 1 Algorithme de croissance par groupe

Etape1 : Utiliser le vecteur N_l^t d'effectif de chaque classe $C_{l,l=1..12}$ de diamètre ; donc un total de douze classes et chaque effectif est en fait l'effectif dans une classe.

Etape2 : Utiliser la valeur de probabilité de croissance $V_{i,i=1..n}$

Etape 3 : Utiliser ensuite le vecteur N^t et la valeur de probabilité de croissance pour générer le vecteur N_q^t qui est en fait le vecteur représentant le nombre de grandis pour toutes les classes de diamètre correspondantes. Ce vecteur est en fait fonction de l'effectif de l'espèce dans les différentes classes concernées. .

Résultat obtenu : En réalité le résultat obtenu ici est en fait l'effectif $N_{l,q}^t$ de notre modèle. Notons que les valeurs de cet effectif $N_{l,q}^t$ suivent une loi binomiale.

Algorithm 2 Algorithme de croissance par groupe

```
{Principe : Le principe est celui de l'algorithme 1 ci dessus.}

Début :

Initialisation : l := 1;

for all i = 0      Nbregroupe do
   for all elt de N^t do
      Elt := élément de N^t ;
      Vi :=ValeurProbabilitéGrandis (Groupe_i) ; {// valeur de Probabilité de croissance}
      N^t_{l,q} :=GenerateurBinomial(Elt,Vi) ;
      l++ ; {// l ici en fait représente la classe et varie de 1 a 12}

   end for

end for
Retourner N^t_{l,q} ;
```

Algorithm 3 Algorithme de mortalité par groupe

Etape1 : Utiliser le vecteur N_l^t d'effectif de chaque classe $C_{l,l=1..12}$ de diamètre ; ou ici on a un total de douze classes et chaque effectif est en fait l'effectif dans une classe.
Etape2 : Utiliser la valeur de probabilité de mortalité $V_{i,i=1..n}$

Etape 3 : Utiliser ensuite le vecteur N^t et la valeur de probabilité de mortalité par classe pour générer le vecteur N_d^t qui est en fait le vecteur représentant le nombre de grandis pour toutes les classes de diamètre correspondantes. Ce vecteur est en fait fonction de l'effectif de l'espèce dans les différentes classes concernées.

Résultat obtenu : En réalité le résultat obtenu ici est en fait l'effectif $N_{l,d}^t$ de notre modèle. Notons que les valeurs de cet effectif $N_{l,d}^t$ suivent une loi binomiale.

Algorithm 4 Algorithme de mortalité par groupe

{**Principe :** Le principe est celui de l'algorithme 3 ci dessus.}

Début :

Initialisation : $l := 1$;

for all $i = 0$ $Nbregroupe$ **do**
  for all elt de N^t **do**
    $Elt :=$ élément de N^t ;
    Vi :=ValeurProbabilitéMort ($Groupe_i$) ; {// valeur de Probabilité de croissance}
    $N_{l,d}^t$:=GenerateurBinomial(Elt,Vi) ;
    l++ ; {// l ici en fait représente la classe et varie de 1 a 12}

  end for

end for
Retourner $N_{l,d}^t$;

Algorithm 5 Algorithme de recrutement par groupe

Etape1 : Utiliser le vecteur N_l^t d'effectif de chaque classe $C_{l,l=1..12}$
Etape2 : Faire une somme d'effectif de ce vecteur ;

Etape 3 : Générer une valeur de recru qui sera fonction du paramètre lambda et de l'effectif total du vecteur (N_l^t) ;

Résultat obtenu : En réalité le résultat obtenu ici est en fait les recrus du modèle pour l'ensemble des classes. $\text{Recru}^t \sim \mathcal{P}(\sum_{l=1}^{L} \lambda N_l^t)$ et $N_{L,q}^t$ =0. Avec lambda le poids du processus de régénération par classe.

Algorithm 6 Algorithme de recrutement par groupe

{**Principe :** Le principe est celui de l'algorithme 5 ci dessus.}

Début :

Initialisation : $i = 1$, $som = 0$; lambda (fixe a une valeur) ;

for all $i = 0$ $Nbregroupe$ **do**
  for all Elt du groupe i **do**
    som := som+Elt ; // une somme entière des effectifs
    recrut i :=Generateurpoisson (lambda *som) ;

  end for
  Retourner $(recrut_i)$;

end for

Algorithm 7 Algorithme reconstitution du vecteur d'effectif

1. **Etape1 :** Générer un vecteur d'effectif (N^t) pour chaque classe de chaque groupe. Par la suite, nous notons ce vecteur effectif (N_l^t), avec l représentant la classe.
2. **Etape2 :** Générer l'effectif des grandis à partir de l'algorithme 1 ce qui nous donnera $N_{l,q}^t$.
3. **Etape 3 :** Générer l'effectif des Morts à partir de l'algorithme 3 ce qui nous donnera $N_{l,d}^t$.
4. **Etape 4 :** Générer l'effectif des recrus à partir de l'algorithme 5 ce qui nous donnera Recru^t.
5. **Etape 5 :** Générer l'effectif N_l^{t+1} à partir de ces valeurs obtenus plus haut c'est-à-dire : $N_l^{t+1} = N_{l,p}^t - N_{l,d}^t - N_{l,q}^t + N_{l-1,q}^t$.

Algorithm 8 Algorithme de reconstitution du vecteur d'effectif

{**Principe :** Le principe est celui de l'algorithme 7 ci dessus.}

Début :

Initialisation : $i = 1$, $som = 0$; lambda (fixe a une valeur) ;

for all $i = 1 \qquad NbreAnne$ **do**
  for all $j = 0 \qquad Nbregroupe$ **do**
    N_l^t := vecteur d'effectif ;
    $N_{l,d}^t$:= vecteur d'effectif des morts ;
    $N_{l,q}^t$:= vecteur d'effectif des grandis ;
    Recru^t :=valeur d'effectif des recrus ;
    $N_l^{t+1} = N_{l,p}^t - N_{l,d}^t - N_{l,q}^t + N_{l-1,q}^t$;
    N_l^t :=N_l^{t+1} ;
    Update ($N_{l,d}^t$) ;
    Update ($N_{l,q}^t$) ;
    Update (Recru^t) ;
    Update (N_l^t) ;
  end for
end for
Retourner N_l^{t+1} ;

Troisième partie

Présentation du simulateur et Résultats

Les modèles d'évolution de peuplement dans la pratique consistent à faire une ou plusieurs simulations ; ceci en partant à l'initial avec un ensemble de donnée, notre simulateur calcule alors successivement une estimation des effectifs des processus de croissance,mortalité, recrutement du peuplement par classe de diamètre et ceci pour plusieurs périodes. Ensuite le simulateur évalue les effectifs de croissance,mortalité, recrutement pour estimer les états successifs du peuplement jusqu'à la fin de la simulation.

7 Outils utilisés et environnement de développement

Pour le travail d'implémentation, nous avons utilisé principalement la librairie UncommonsMaths vu que cette librairie renferme beaucoup de fonctions mathématiques ;

- Un générateur binomiale BinomialGenerator de la librairie UncommonsMaths [5] qui nous aide à créer un générateur de valeurs distribuées sous forme de loi binomiale avec un certains nombre de paramètres spécifiques. BinomialGenerator (int n, double p, Random rng) Paramètres : n – Le nombre d'essai.
- Générateur de poisson [6] : qui nous servira ici à générer nos valeurs de nouveau recru dans le modèle qui suivent une loi de poisson ceci en fonction du peuplement.

Le simulateur qui en ressort de ce stage est en fait conçu sous l'environnement de développement éclipse Helios , en JAVA SE. Cet environnement tournant sous une machine Toshiba triple core ayant Ubuntu 13.04 comme système d'exploitation.

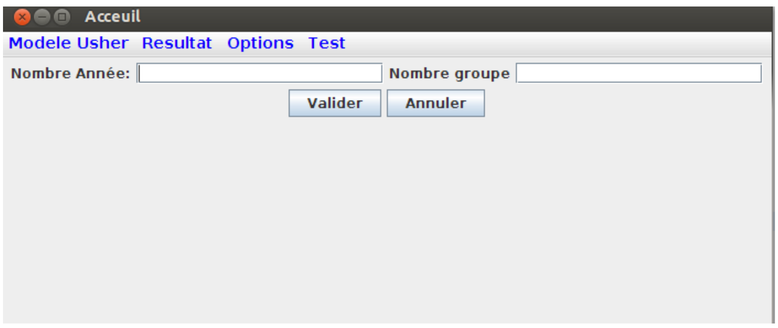


fig 2 : Interface d'accueil du simulateur.

5. org.apache.commons.math
6. org.apache.commons.math.distribution.PoissonDistribution

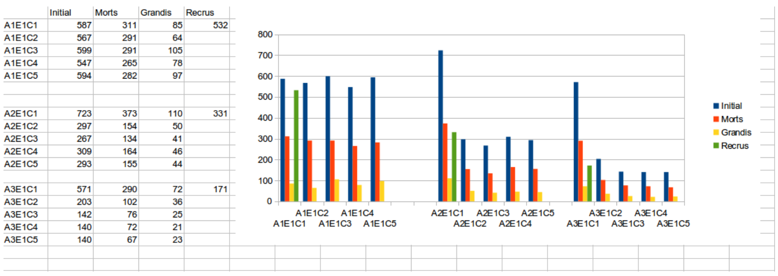
8 Présentation du simulateur

Notre simulateur pour le moment génère des résultats qui sont stockés dans des fichiers et donc l'interprétation est faite à partir des différentes cases du fichier CSV donc les valeurs seront ensuite injectées dans le Système de Gestion de Bases de Données Postgres.

Au niveau du cas statique qui est en effet une simulation sur 3 ans avec 3 groupes, on n'a pas besoin de remplir les champs nombres d'années et nombres de groupe. Il suffit juste de cliquer sur le bouton test en bleu pour lancer cette simulation statique.

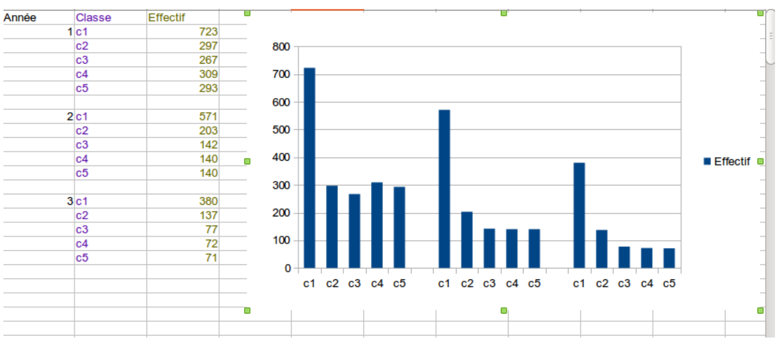
Au niveau du cas dynamique qui est en effet une simulation sur un nombre d'année voulu et un nombre de groupe voulu, on a juste besoin de remplir les champs nombres d'années et nombres de groupes. Il suffit juste de cliquer sur le bouton Valider pour lancer cette simulation dynamique.

En sortie nous avons plusieurs fichiers, le principal ici est le fichier csv qui ressort juste les effectifs globaux après une simulation complète et un autre fichier qui garde juste une trace de l'évolution des différents effectifs dans le temps. Soit les deux images suivantes :



	Initial	Morts	Grandis	Recrus
A1E1C1	587	311	85	532
A1E1C2	567	291	64	
A1E1C3	599	291	105	
A1E1C4	547	265	78	
A1E1C5	594	282	97	
A2E1C1	723	373	110	331
A2E1C2	297	154	50	
A2E1C3	267	134	41	
A2E1C4	309	164	46	
A2E1C5	293	155	44	
A3E1C1	571	290	72	171
A3E1C2	203	102	36	
A3E1C3	142	76	25	
A3E1C4	140	72	21	
A3E1C5	140	67	23	

fig 3 : Fichier récapitulatif des effectifs des différents processus.



Année	Classe	Effectif
1	c1	723
	c2	297
	c3	267
	c4	309
	c5	293
2	c1	571
	c2	203
	c3	142
	c4	140
	c5	140
3	c1	380
	c2	137
	c3	77
	c4	72
	c5	71

fig 4 : Fichier récapitulatif des effectifs des espèces.

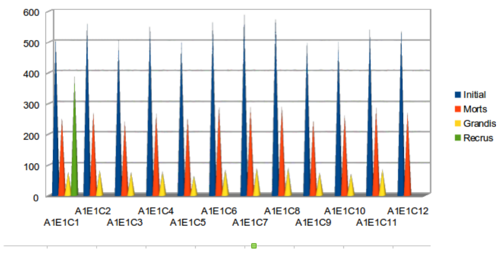
Les deux images représentent en fait une simulation sur 3 ans avec 5 classes d'arbres. La première image représente l'évolution de l'effectif de l'espèce 1. De façon générale, c'est la forme du fichier CSV qui est généré par notre simulateur.Ce fichier CSV contient la somme d'effectif après chaque année de simulation ceci pour toutes les espèces et pour toutes les classes. La deuxième image nous donne un aperçu de l'évolution des différents processus du modèle (effectif initiale, effectif des morts, des grandis, des recrus). C'est aussi la forme du fichier XLS[7] libre qui est généré par notre simulateur.

9 Quelques Résultats

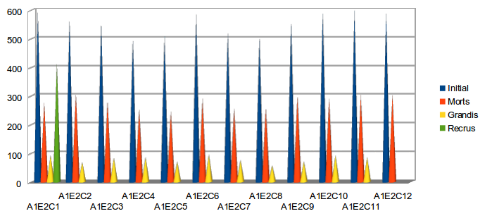
Nous présenterons ici quelques résultats obtenus pour une simulation sur 3 ans avec 3 espèces et 12 classes de diametre ; ceci en partant a l'initial avec un ensemble de donnée. le simulateur calculera alors successivement une estimation de la croissance du peuplement pour plusieurs périodes ceci en intégrant les processus parallèles (mortalité,croissance).Ensuite le simulateur évaluera les conditions de croissance, et estimera les états successifs du peuplement jusqu'à la fin de la simulation. mètre ; Ce jeu de données est obtenus avec la matrice de probabilités de transition suivante :

$$\begin{array}{ccc} 0{,}5 & 0{,}3 & 0{,}2 \\ 0{,}5 & 0{,}3 & 0{,}2 \\ 0{,}5 & 0{,}3 & 0{,}2 \end{array}$$

	Initial	Morts	Grandis	Recrus
A1E1C1	519	254	76	389
A1E1C2	567	274	83	
A1E1C3	507	242	76	
A1E1C4	560	268	79	
A1E1C5	515	256	64	
A1E1C6	569	292	85	
A1E1C7	592	288	89	
A1E1C8	587	292	90	
A1E1C9	509	249	75	
A1E1C10	504	264	71	
A1E1C11	545	285	85	
A1E1C12	549	273	0	



	Initial	Morts	Grandis	Recrus
A1E2C1	594	278	93	415
A1E2C2	570	309	70	
A1E2C3	562	285	84	
A1E2C4	502	256	87	
A1E2C5	513	247	71	
A1E2C6	585	292	96	
A1E2C7	525	259	77	
A1E2C8	512	261	58	
A1E2C9	565	303	72	
A1E2C10	596	294	93	
A1E2C11	599	304	87	
A1E2C12	593	306	0	



	Initial	Morts	Grandis	Recru
A1E3C1	512	261	71	
A1E3C2	566	284	78	
A1E3C3	555	291	81	
A1E3C4	599	279	88	
A1E3C5	600	291	97	
A1E3C6	509	242	73	
A1E3C7	573	298	91	
A1E3C8	572	287	89	
A1E3C9	537	266	76	
A1E3C10	563	279	81	
A1E3C11	549	290	83	
A1E3C12	584	283	0	

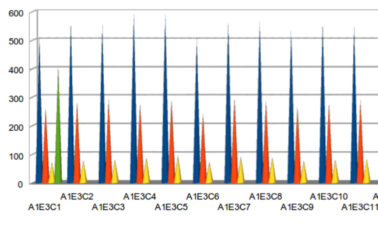


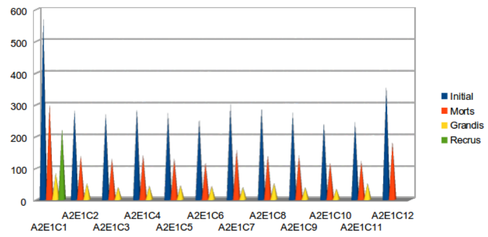
Table 1 – Données à l'initial et évolution effectifs des processus

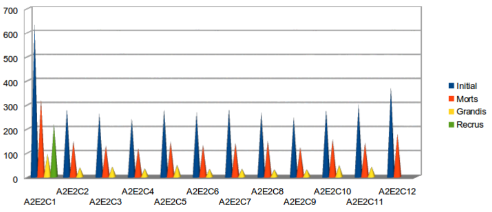
7. C'est le fichier de calcul de libre office qui est un logiciel open source

	Initial	Morts	Grandis	Recrus
A2E1C1	578	303	85	226
A2E1C2	286	139	52	
A2E1C3	272	130	40	
A2E1C4	289	142	44	
A2E1C5	274	130	45	
A2E1C6	256	118	43	
A2E1C7	300	157	40	
A2E1C8	294	141	53	
A2E1C9	275	140	39	
A2E1C10	244	118	34	
A2E1C11	246	122	52	
A2E1C12	361	184	0	

	Initial	Morts	Grandis	Recrus
A2E2C1	638	324	96	218
A2E2C2	284	148	40	
A2E2C3	263	130	43	
A2E2C4	243	120	35	
A2E2C5	282	147	51	
A2E2C6	268	133	33	
A2E2C7	285	142	34	
A2E2C8	270	149	32	
A2E2C9	248	125	32	
A2E2C10	281	157	50	
A2E2C11	301	143	43	
A2E2C12	374	181	0	

	Initial	Morts	Grandis	Recru
A2E3C1	589	297	86	
A2E3C2	275	136	54	
A2E3C3	261	128	40	
A2E3C4	313	153	51	
A2E3C5	300	149	58	
A2E3C6	291	141	37	
A2E3C7	257	123	39	
A2E3C8	287	144	45	
A2E3C9	284	138	46	
A2E3C10	279	146	34	
A2E3C11	257	131	27	
A2E3C12	384	178	0	





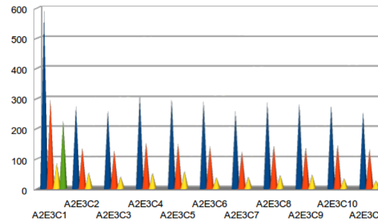
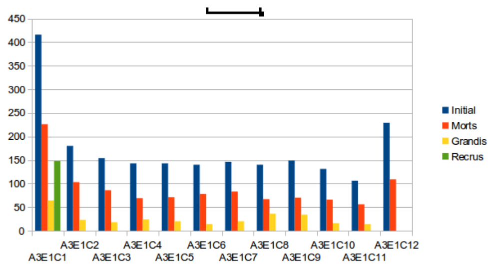


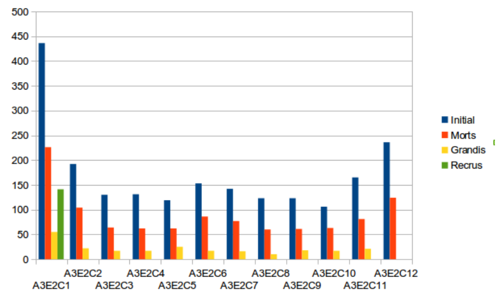
TABLE 2 – Données à l'initial et évolution effectifs des processus la deuxième année

	Initial	Morts	Grandis	Recrus
A3E1C1	416	226	64	148
A3E1C2	180	103	23	
A3E1C3	154	86	18	
A3E1C4	143	69	24	
A3E1C5	143	71	20	
A3E1C6	140	78	14	
A3E1C7	146	83	20	
A3E1C8	140	67	36	
A3E1C9	149	70	34	
A3E1C10	131	66	16	
A3E1C11	106	56	14	
A3E1C12	229	109	0	

	Initial	Morts	Grandis	Recrus
A3E2C1	436	226	55	141
A3E2C2	192	104	22	
A3E2C3	130	64	17	
A3E2C4	131	62	17	
A3E2C5	119	62	25	
A3E2C6	153	86	17	
A3E2C7	142	77	16	
A3E2C8	123	60	10	
A3E2C9	123	61	18	
A3E2C10	106	63	17	
A3E2C11	165	81	21	
A3E2C12	236	124	0	

	Initial	Morts	Grandis	Recru
A3E3C1	433	216	72	
A3E3C2	171	80	31	
A3E3C3	147	76	19	
A3E3C4	149	68	30	
A3E3C5	144	78	19	
A3E3C6	171	93	20	
A3E3C7	132	58	21	
A3E3C8	137	70	28	
A3E3C9	145	72	23	
A3E3C10	145	78	16	
A3E3C11	133	75	23	
A3E3C12	233	109	0	





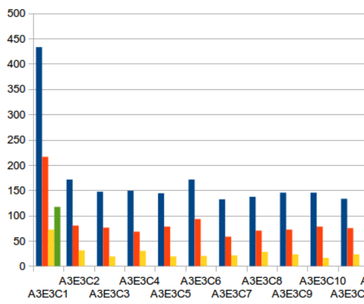


TABLE 3 – Données à l'initial et évolution effectifs des processus la troisième année

10 Discussion et perspectives

Nous dirons que le simulateur conçu lors de notre stage produit bien les résultats attendus en intégrant bien la dimension stochastique dans ses prédictions ce qui est l'innovation pour un modèle d'Usher densité dépendant ; toutefois, il est important de noter que dans le cadre de la mise au point de notre simulateur, certaines données par espèces peuvent être bruitées et dans ce cadre que s'inscrit les travaux connexes faits parallèlement pour pouvoir déterminer en cas de bruitage les différents paramètres des espèces et aussi la détermination des différents groupes des différents espèces.Ces résultats pourront être injectés au simulateur pour lui fournir des informations en plus pour les simulations. Il est important de rappeler ici que la documentation n'a pas été très aisé pour la réalisation de ce simulateur en JAVA car il n'y a pas d'outils équivalent pour le moment aidant dans la gestion de ressources de forêt tropicales. De plus, notons que le fait que notre simulateur peut être étendu de façon à faire des prédictions sur des peuplements différents.

Conclusion

L'objectif pour nous dans ce stage était le développement d'un simulateur en JAVA pour pouvoir faire la gestion de ressources naturelles ou plus précisément la prédiction forestière. Nous avons commencé par étudier le modèle de gestion de ressources forestières qui sied le mieux avec notre problème ; Donc nous avons commencé par une étude du modèle d'Usher stochastique et les différents processus y intervenant.Notre travail a été divisé en trois parties donc un état de l'art, une présentation du simulateur et les modèles et processus y intervenant et enfin une présentation de résultat. Le simulateur a été mis sur pied en se basant sur le modèle non seulement d'Usher mais aussi le modèle de classe et un certains nombre de processus tels que les processus de mortalité, croissance, recrutement. Notons que nos principaux résultats ici sont pour des ressources particulières qui sont les arbres, mais toutefois, une extension de d'autres ressources peuvent être simulé par notre outil. Les principaux résultats pour le moment restent bien ceux attendus par le projet de ce stage car nous arrivons à pouvoir faire des prédictions en se basant sur le modèle étudié pour le faire. A la fin de la simulation nous générons un certains nombres de fichier dont le principal est le fichier au format libre CSV qui présente les principaux résultats. Le choix du CSV nous est motivé par le fait qu'il peut facilement être utilisable dans un outil comme R pour pouvoir faire des interprétations graphiques plus simplifiées, et aussi par le fait que son intégration dans un SGBD[8] est aisée. Nous devons tout de même dire que notre simulateur n'est pas encore fini car il y a encore un certains nombre de travaux connexes à lui associer et de modules aussi à l'instar de celui des interprétations graphiques.

Remerciements

Je remercie vivement tous ceux qui de près ou de loin ont contribué à mener à terme cette phase d'initiation à la recherche.Je remercie le Docteur MOTO MPONG Serge et le Professeur Vivien Rossi qui m'ont accepté dans ce projet ; merci encore pour le suivi, la patience et le contrôle des travaux de ce mémoire.

J'adresse mes remerciements sincères à mes professeurs du département d'informatique pour leurs enseignements ; également au Docteur DADA Jean pierre et Romain Atangana pour leur soutien.

Tous les membres de ma famille : mes parents Takam Isaac et Ngameni Rose, Arleo Etame et Ngueng Monique ; mes frères Takam Durel, Takam Guidel et mes soeurs Takam Leslie, Takam Jornelle, Etame Arlette ; mes tantes Aicha Issa, Florence Vegnie qui ont toujours su m'encourager.

8. Système de Gestion de Bases de Données

La famille Stofack, qui m'a acceptée comme leur propre enfant. Mes remerciements à mes amis et camarades qui m'ont aidé à améliorer mes idées par des discussions et propositions. Je remercie enfin mes proches (Arnaud Stofack, Heudjeu Olivier,Moumi jerome, Fouejio Nongni, Amelie Habit, Koagne Julio,Tangueu Valerie, Domou, Ngouana Paulin, Mbonji Fabrice, Vanessa Elemva, Dr Ngouana Brice, Mich Maf, Mbianda, Harmonie Noujip, Nlend Ariane, Darius Tetsa, Nyobe, Fotsing Armand, Bitoulefock, ...) qui ont su me motiver ; sans oublier un merci particulier à Estelle Bakakeu et ma fille Mambé Gabriella ;

Références

[1] Pierrette Chagneau. *Modélisation bayésienne hiérarchique pour la prédiction multivariée de processus spatiaux non gaussiens et processus ponctuels hétérogènes d'intensité liée à une variable prédite. Application à la prédiction de le régénération en forêt tropical humide.* PhD thesis, Université Montpellier II, décembre 2009.

[2] Landry Clara. *Modélisation bayésienne de la dynamique d'un peuplement forestier avec prise en compte de la densité-dépendance.* Université Montpellier II, juin 2010.

[3] Patricia Le Crenn. Capsis : modéliser et simuler la croissance des peuplements forestiers. 2012.

[4] SAMBAKHE Diariétou. Analyse des sources de biais dans les modèles de croissance forestière dus aux changements d'échelle d'application. pages 26–30, Septembre 2012.

[5] E.S. Allman et J.A. Rhodes. Dynamiques aleatoires : chaines de markov. juin 2004.

[6] Vivien Rossi Fabien Campillo, Frederic Mortiery. Chaines de markov modeles matriciels. mars 2010.

[7] http ://ur-bsef.cirad.fr/principaux-projets/modele-matriciel-stochastique stomat. Consulté le : 15 septembre 2014.

[8] Sebastien Jesel. Regeneration ecology and population dynamics of dicorynia guianensis(caesalpiniaceae)in a french guiana rainforest. pages 8–12, oct 2007.

[9] Usher M.B. *A matrix model for forest management.* Université Montpellier II, juin 1969.

[10] Dakis-Yaoba Ouedraogo. *Prediction de la dynamique forestiere à l'aide d'un modele matriciel qui incorpore la variabilite de la reponse des espèces à l'environnement.* PhD thesis, Universite Montpellier II, 2011.

[11] Nicolas Picard Sylvie Gourlet-Fleury, Alain Franc. Une introduction à la modélisation des forêts hétérogènes. page 54–57.

[12] Jerome K Vanclay. Growth modelling and yield prediction for sustainable forest management. pages 4–6, juin 2003.