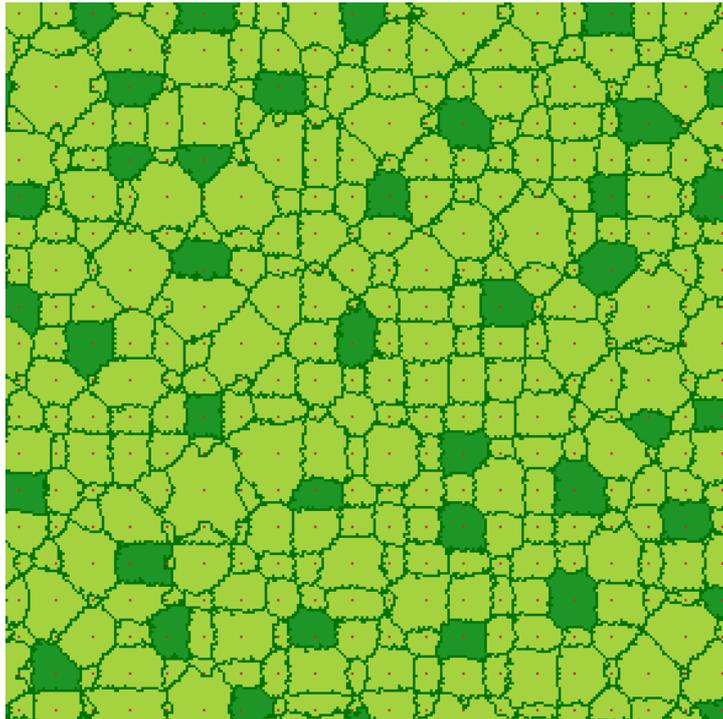


Sélection numérique guidée et sélection numérique autonome d'une éclaircie dans le simulateur SimCop



Rapport de stage présenté en vue de l'obtention du

**Diplôme d'Etudes Approfondies
mention "Biologie Forestière"
de l'Université Henri Poincaré, Nancy 1**

Présenté le 30 Août 2005 par

Patrick Millan

Jury composé de : **J.F. Dhôte, J.C. Hervé, F. Lebourgeois,
J.M. Ottorini, J.C. Pierrat**

Rapporteurs : **J.C. Hervé, J.C. Pierrat**

Responsable de stage : **Jean-Marc Ottorini** (UMR LERFOB)

UMR Laboratoire d'Etude des Ressources Forêt-Bois,
Unité Mixte de Recherches INRA-ENGREF 1092, Equipe Croissance et Production, France

Résumé

L'automatisation des éclaircies facilite la simulation des différents traitements et scénarii sylvicoles. L'évaluation d'une éclaircie utilise comme critères le nombre d'arbres sélectionnés et le coefficient d'éclaircie k , auxquels est ajoutée une notion de répartition spatiale. Trois méthodes d'évaluation de cette répartition ont été comparées, les méthodes du plus proche voisin, de l'inertie et des Quadrats. Deux modes de sélection d'arbres ont été retenus : la sélection numérique guidée et la sélection numérique autonome. La sélection numérique guidée permet des échantillonnages représentatifs stratifiés avec ou sans biais. La sélection numérique autonome fait appel aux algorithmes génétiques. Ces algorithmes nécessitent des ajustements lors de l'implémentation (pour le codage du problème, les techniques de croisement et de mutation), et lors de l'optimisation (le scaling, le sharing, ainsi que le choix des paramètres de dimensionnement du problème).

Mots clés: éclaircie, sélection numérique guidée, autonome, méthodes du plus proche voisin, inertie, Quadrats, algorithmes génétiques, scaling, sharing, dimensionnement.

Abstract

The thinnings automation facilitates the simulation of various forestry treatments and scenarii. The evaluation of a thinning uses as criteria the number of selected trees and the thinning factor k , to which is added a notion of spatial distribution. Three methods of evaluation of this distribution were compared: the methods of the Nearest Neighbour (or Knn) the inertia, and the Quadrats. Two selection modes of trees were retained: the guided digital selection and the off-line digital selection. The guided digital selection allows stratified representative samplings with or without bias. The off-line digital selection is based on genetic algorithms. These algorithms require adaptations for their implementation (the problem coding, the crossing and alteration techniques), and during the optimization process (the scaling and the sharing, the dimensionnement parameters choice of the problem).

Keywords: thinning, guided digital selection, off-line digital selection, Nearest Neighbour, inertia, Quadrats, genetic algorithms, scaling, sharing, dimensionnement.

Sélection numérique guidée et sélection numérique autonome d'une éclaircie dans le simulateur SimCop

Patrick Millan

Centre de Recherches de Nancy, I.N.R.A.,
Champenoux, 54280 Seichamps

Résumé

L'automatisation des éclaircies facilite la simulation des différents traitements et scénarii sylvicoles. L'évaluation d'une éclaircie utilise comme critères le nombre d'arbres sélectionnés et le coefficient d'éclaircie k , auxquels est ajoutée une notion de répartition spatiale. Trois méthodes d'évaluation de cette répartition ont été comparées, les méthodes du plus proche voisin, de l'inertie et des Quadrats. Deux modes de sélection d'arbres ont été retenus : la sélection numérique guidée et la sélection numérique autonome. La sélection numérique guidée permet des échantillonnages représentatifs stratifiés avec ou sans biais. La sélection numérique autonome fait appel aux algorithmes génétiques. Ces algorithmes nécessitent des ajustements lors de l'implémentation (pour le codage du problème, les techniques de croisement et de mutation), et lors de l'optimisation (le scaling, le sharing, ainsi que le choix des paramètres de dimensionnement du problème).

1. Introduction

Le système de simulation SimCoP (« Simulateur de Croissance et de Développement de Conifères en Peuplement ») a été développé pour permettre d'évaluer les effets de divers traitements et scénarii sylvicoles sur des peuplements de Douglas [*Pseudotsuga menziesii* (Mirb.) Franco]. Ce simulateur exploite de nombreuses informations détaillées sur la croissance individuelle des arbres, établies au moyen d'analyses de tiges et de branches, provenant d'arbres abattus (Ottorini, 1991), suivant des principes très proches de ceux présentés antérieurement par Mitchell (1975).

Le but de ce travail sera de faciliter et d'accélérer le choix des arbres à éclaircir au cours du processus de simulation. Les différentes méthodes de sélection testées devront fournir à l'utilisateur une première approximation à la solution attendue, qu'il pourra éventuellement remanier.

2. Matériel et méthodes

2.1. Les caractéristiques d'une éclaircie

Les différentes caractéristiques de l'éclaircie retenues sont:

- Le nombre d'arbres à éclaircir N_e .

- Le coefficient d'éclaircie k correspondant à la formule: $k = \frac{\bar{v}_e}{\bar{v}}$, avec \bar{v}_e volume moyen des arbres éclaircis et \bar{v} volume moyen du peuplement.

- La répartition spatiale des arbres à éclaircir qui doit être la plus uniforme possible sur la placette.

2.2. Données fournies par le simulateur

À chaque interruption, le simulateur crée un fichier de données représentant un peuplement à un moment donné. Ce fichier contient des informations propres à chacun des arbres composant le peuplement, et à la ou les espèces présentes.

Chaque Arbre est décrit par son numéro identifiant, sa hauteur, son volume, son diamètre à 1,30 mètre, sa surface de projection du houppier, sa position spatiale

sur la placette convertie en coordonnées graphiques et le profil de sa tige (couples hauteur-diamètre).

Est également présente une carte codée de la projection plane des houppiers sous la forme d'un tore.

La forme du houppier étant propre à chaque espèce, dans le cas du Douglas, elle suit la formule suivante

$$r = b * d * \log\left(\frac{l}{c} + 1\right)$$

avec b, c, d paramètres de forme des dimensions du houppier, r rayon du houppier, l distance par rapport à l'apex.

2.3. L'interface graphique

Afin de faciliter le développement d'algorithmes de sélection et la

visualisation de leurs résultats, une nouvelle interface graphique a été développée. Pour qu'elle soit accessible au plus grand nombre de machines, elle a été codée dans le langage de programmation objet Java, et adjointe au simulateur.

Cette interface est composée de cinq éléments répartis dans trois fenêtres graphiques.

La fenêtre principale comporte trois panneaux permettant la visualisation de la carte, du profil de tige, du houppier et de sa surface de projection, ainsi que la répartition par classes de volume des arbres du peuplement (Figure 1).

Les deux autres fenêtres permettent le paramétrage des sélections numériques guidées et autonomes des arbres à couper (Figure 2 et 3).

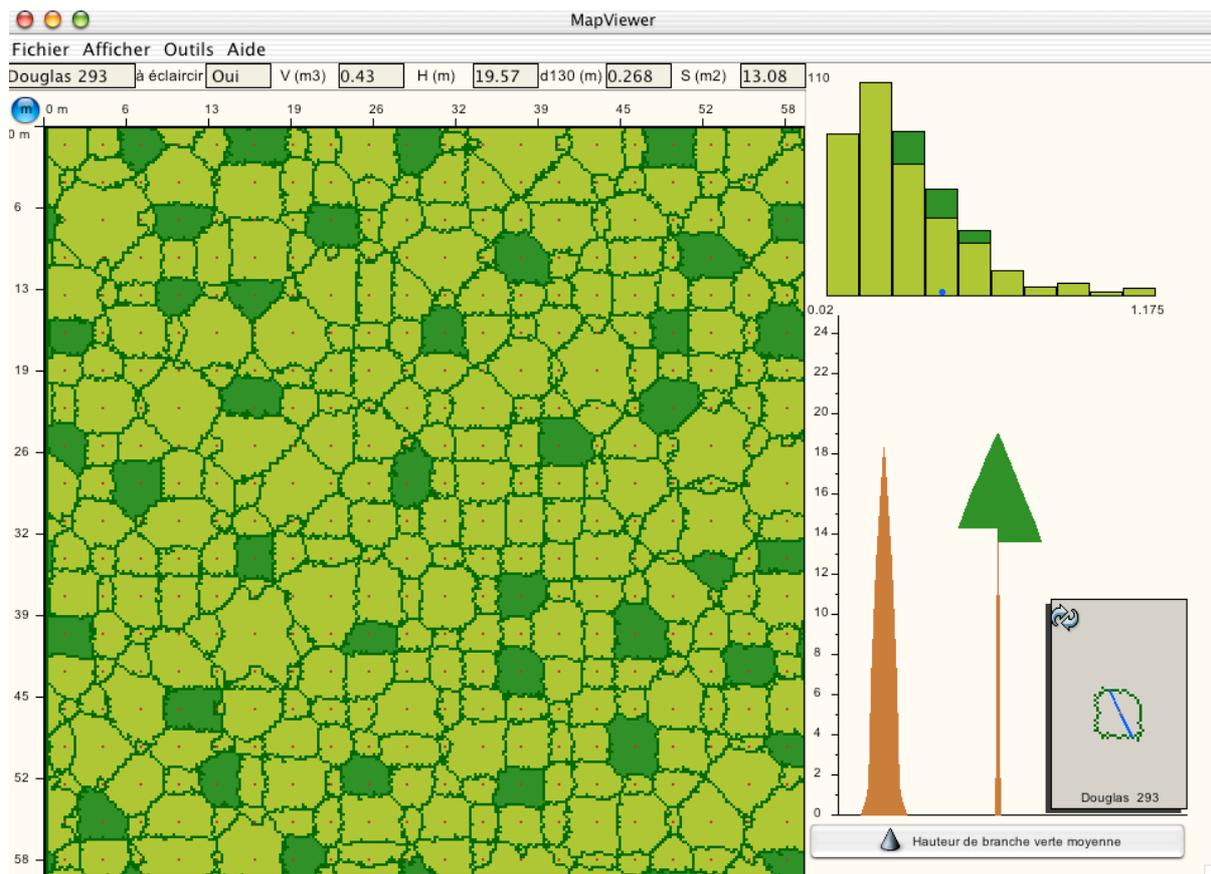


Figure 1 Fenêtre principale de l'interface graphique du visualisateur de cartes de placettes forestières dans le simulateur SimCoP.

2.4. La sélection numérique guidée

2.4.1. Répartition des arbres en classes

Figure 2 Interface de sélection numérique guidée.

Les arbres sont regroupés par défaut en classes de volume. Une option, dans le code, permet de choisir entre des classes de volume ou de diamètres à 1,30m. La création des classes et leur nombre se font selon la méthode de calcul décrite par Wand (1997), soit

(1) $R = V_{\max} - V_{\min}$
 V_{\max} Volume Maximum, V_{\min} Volume Minimum, et R la largeur de l'intervalle de variation

$$(2) h_{\square} = \frac{R}{1 + \log_2(N)}$$

h_{\square} la largeur d'une classe, et N le nombre d'arbres du peuplement

$$(3) nbV = (\text{int})\left(\frac{R}{h_{\square}} + 1\right)$$

nbV le nombre de classes de volume

2.4.2. Prélèvement des arbres dans les classes

2.4.2.1. Généralités

Soit k le coefficient d'éclaircie
 $k = \frac{\bar{v}_e}{v}$.

Par définition, le volume éclairci \bar{v}_e est tel

$$\text{que } \bar{v}_e = \frac{1}{n_e} \sum_{i=1}^{n_e} v_{ei}.$$

Avec la répartition des arbres en classes,

$$\text{on a } \bar{v}_e = \frac{1}{n_e} \sum_{j=1}^{n_c} n_{ecj} \bar{v}_{ecj}, \text{ où } \bar{v}_{ecj} \text{ est le}$$

volume moyen éclairci pour une classe j .

Par approximation, on a $\bar{v}_{ecj} \approx \bar{v}_{cj}$, d'où

$$\bar{v}_e = \sum_{j=1}^{n_c} \frac{n_{ecj}}{n_e} \bar{v}_{ej}$$

On note alors $\frac{n_{ecj}}{n_e} = p_j$ le taux de

prélèvement dans une classe, d'où

$$\bar{v}_e = \sum_{j=1}^{n_c} p_j \bar{v}_{cj}.$$

Chaque taux doit alors respecter les contraintes suivantes:

$$p_j \leq 1, \text{ et } \sum_{j=1}^{n_c} p_j = 1$$

2.4.2.2. Échantillonnage représentatif stratifié

Dans le cas particulier d'un échantillonnage représentatif stratifié, le taux de prélèvement d'une classe est égal à sa proportion dans le peuplement

$$P_{0j} = \frac{n_j}{n},$$

d'où

$$\bar{v}_e = \sum_{j=1}^{n_c} \frac{n_j}{n} \bar{v}_{cj} = \frac{1}{n} \sum_{j=1}^{n_c} n_j \bar{v}_{cj} = \frac{1}{n} \sum_{j=1}^{n_c} v_{cj} = \bar{v}$$

$$\text{Dès lors } k = \frac{\bar{v}_e}{v} = 1.$$

Ce calcul est celui réalisé par défaut au moment du choix du nombre total d'arbres à éclaircir dans le peuplement.

2.4.2.3. Échantillonnage biaisé

Pour permettre un sur-échantillonnage des petites ou des grandes classes, on attribue un coefficient à chaque taux de prélèvement tel que:

$p_{ij} = q_1 \square p_{0j}$ avec j appartenant aux petites classes.

$p_{ij} = q_2 \square p_{0j}$ avec j appartenant aux classes intermédiaires.

$p_{ij} = q_3 \square p_{0j}$ avec j appartenant aux grandes classes.

Avec comme contraintes

$$\square P_{1j} + \square P_{2j} + \square P_{3j} = 1$$

$$q_1 \square \square P_{1j} + q_2 \square \square P_{2j} + q_3 \square \square P_{3j} = 1$$

et $q_i \square p_{0j} \square 1$

Dès lors, lorsque l'on sur-échantillonne les petites classes et/ou les classes intermédiaires, on aura $k < 1$. Inversement, lorsque l'on sur-échantillonne les grandes classes et/ou les classes intermédiaires, on aura $k > 1$.

Dans les faits, ces calculs sont rapportés à des groupes de classes, dans lesquels l'échantillonnage est à nouveau représentatif stratifié.

2.4.3. Procédure de sélection des arbres à éclaircir

Le tirage des numéros identifiants les arbres, au moment de la sélection, peut s'avérer être contraignant et nécessiter un temps de calcul important.

Trois algorithmes ont été testés afin de permettre cette sélection.

Les variables utilisées communes à ces trois algorithmes sont

Arbre Arbre sélectionné pour l'éclaircie.

Ne Nombre d'arbres à éclaircir.

Peuplement Liste d'arbres du peuplement.

Liste Liste d'arbres à éclaircir.

2.4.3.1. Algorithme 1

Ce premier algorithme est un tirage avec remplacement. Il est intuitif, mais

aussi, lourd à gérer pour un ordinateur. Si le nombre d'arbres à éclaircir est proche du nombre d'arbres du peuplement, alors le nombre d'itération nécessaire à l'obtention du résultat souhaité peut devenir prohibitif. Il peut en résulter des problèmes de stabilité de calcul.

tirageAléatoire(Peuplement) Fonction retournant un arbre pris aléatoirement dans la liste d'arbres du peuplement.

Début

Tant que **Ne** \neq 0

Arbre = **tirageAléatoire(Peuplement)**

Tant que **Liste** contient **Arbre**

Arbre = **tirageAléatoire(Peuplement)**

Fin tant que

Ajouter **Arbre** à **Liste**

Décrémenter **Ne** de 1

Fin tant que

Fin

2.4.3.2. Algorithme 2

Une variante sans remplacement de l'algorithme 1 utilise une liste d'arbres intermédiaire (copie, ou clone, de la liste d'arbres principale). C'est sur cette liste que les tirages sont réalisés. Après chacun d'eux, l'arbre sélectionné y est effacé.

Le nombre d'itération de cet algorithme est fixe, mais la répétition des parcours de tableaux en fait un algorithme de complexité de type quadratique $O(n^2)$.

ListeClone Liste clonée d'arbres du peuplement.

tirageAléatoire(ListeClone) Fonction retournant un arbre pris aléatoirement dans la liste clonée d'arbres.

Début

ListeClone = clone de **Peuplement**

Tant que **Ne** \neq 0

Arbre = **tirageAléatoire(ListeClone)**

Effacer **Arbre** de **ListeClone**

Ajouter **Arbre** à **Liste**

Décrémenter **Ne** de 1

Fin tant que

Fin

2.4.3.3. Algorithme 3

Cet algorithme est de complexité n , ce qui lui garantit une rapidité d'exécution, et une fiabilité de résultats.

ListeCouples : Liste des couples ($Arbre_i, Valeur_i$)

valeurAléatoire() : Fonction retournant un nombre réel tiré aléatoirement dans une loi Uniforme entre $[0, 1]$.

Début

Pour chaque Arbre $Arbre_i$ de Peuplement

$Valeur_i = \text{valeurAléatoire}()$

Ajouter couple($Arbre_i, Valeur_i$) à

ListeCouples

Fin pour chaque

Trier ListeCouples selon $Valeur_i$ dans l'ordre croissant

Liste = Ne premiers Arbres de ListeCouples

Fin

Il est également possible de faire dépendre la probabilité de sélection d'un arbre de sa probabilité relative à celle des autres arbres. On cherche alors à échantillonner dans les différentes classes selon les proportions définies précédemment. Pour que l'espérance du nombre d'arbres tiré par classes soit conforme à $\frac{N_{ClasseArbre}}{N_{Peuplement}}$, les valeurs

associées à chaque arbre sont tirées dans un intervalle $[0, A_i]$, où A_i est déterminé de façon à respecter ces proportions, soit

$$A_i = 1 \times \frac{N_{ClasseArbre}}{N_{Peuplement}} \quad (\text{Dhôte \& Le Moguedec 2005}).$$

2005).

2.5. La sélection numérique autonome

2.5.1. Les algorithmes génétiques

Pour résoudre le problème de l'optimisation de la sélection des arbres à éclaircir, nous avons fait appel à des techniques dites d'intelligence artificielle. Nous avons développé et testé des algorithmes génétiques. Ces algorithmes sont classés parmi les algorithmes évolutionnaires affiliés aux algorithmes de vie artificielle.

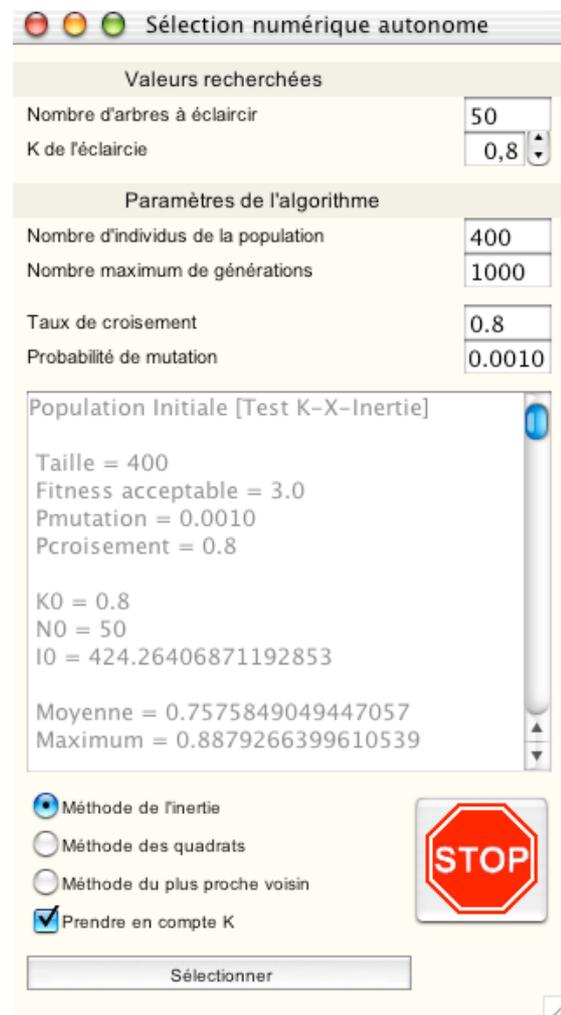


Figure 3 Interface des algorithmes génétiques.

2.5.1.1. Principe général

Inspiré de la théorie darwinienne de l'évolution (Darwin, 1858), l'algorithme génétique recherche le ou les maxima

d'une fonction définie sur un espace de données. Pour l'utiliser, il est nécessaire de définir les cinq éléments suivants

- Le codage de chaque situation faisant partie d'un ensemble de situations possibles.

- Le mécanisme de génération d'une population initiale aléatoire de situations. Chacune de ces situations est alors assimilée à un individu.

- La fonction à optimiser, définie pour chaque individu qui retourne une valeur appelée fitness ou fonction d'évaluation de l'individu

- Les opérateurs permettant de diversifier la population de façon préférentielle au cours des générations et d'explorer en théorie la quasi-intégralité de l'espace d'état. Ces opérateurs sont les suivants

- L'opérateur de croisement recompose les gènes, ou éléments codant les individus existant dans la population par simple ou double crossover (uniforme ou non).

- L'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états, en faisant apparaître des individus que les croisements ne peuvent pas produire.

- Les paramètres de dimensionnement du problème, comme la taille de la population, le nombre total de générations ou critère d'arrêt de l'algorithme, les probabilités d'application des opérateurs de croisement et de mutation ... Ces paramètres seront un compromis entre le temps de calcul et la qualité de la solution.

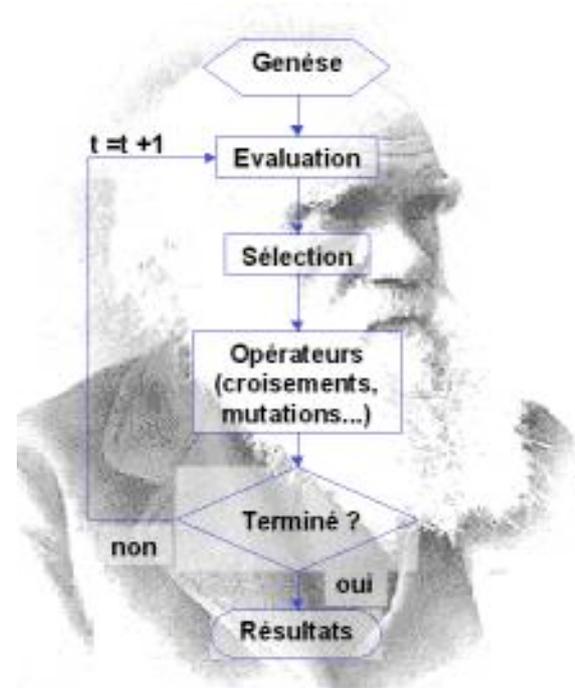


Figure 4 Principe général d'un algorithme génétique (Magnin 2002).

2.5.1.2. Le fitness ou adaptabilité

L'algorithme génétique est un algorithme d'optimisation. Il est donc nécessaire de définir une fonction d'évaluation permettant de quantifier l'adaptabilité (ou fitness) en vue de réaliser une sélection.

Dans notre cas, cette fonction prendra en compte les caractéristiques propres à l'éclaircie. Des paramètres de dimensionnement du problème seront testés pour plusieurs valeurs.

2.5.1.3. Les différentes approches du codage de l'algorithme génétique

Dans le cas présent, on considère comme individu une liste d'arbres à éclaircir. Cet individu peut être codé de deux façons différentes.

2.5.1.3.1. Première approche du codage

On considère la liste comme une suite N de valeurs binaires égales à 0 ou 1, i.e. des bits. Chaque bit est associé à un arbre déterminé du peuplement codé sur

une liste référence. La valeur «□□» signifie que l'arbre est choisi pour faire partie de l'éclaircie. Au moment de la génération de la population initiale, chaque bit de chaque individu a 50% de chance d'être initialisé à 0 ou 1. Il en résulte une population possédant des individus avec tous environ $N/2$ bits initialisés à 1 et $N/2$ bits initialisés à 0.

L'opérateur de croisement recompose les gènes par simple crossover.

L'avantage de cette méthode de codage est de nous permettre un certain degré de liberté quant au nombre exact d'arbres que l'on souhaite éclaircir. On s'autorise à prendre quelques arbres en plus ou en moins si les autres critères d'évaluation s'en trouvent considérablement améliorés. Ce type de codage est le plus employé, ces propriétés sont les mieux connues, et il bénéficie d'un champ d'application plus général.

Cette méthode correspond à l'algorithme de base simple de Goldberg (Goldberg Daniel E. 1989).

2.5.1.3.2. Seconde approche du codage

On considère la liste comme une suite de numéros identifiants de chaque arbre à éclaircir.

Au moment de la génération de la population initiale, les individus sont hétérogènes en ce qui concerne la nature des bases codant la séquence génétique (On a autant de type de bases que de numéros identifiant les arbres du peuplement étudié).

L'opérateur de croisement recompose les gènes par double crossover uniforme, car cette méthode est particulièrement appropriée à cette forme de codage par entier, et à la nécessité d'éviter leurs répétitions. Cette méthode permet d'obtenir toujours le nombre d'arbres souhaités, sans la souplesse de la première méthode.

2.5.1.4. Les Crossovers

2.5.1.4.1. Le simple crossover, ou croisement en un point

Le simple crossover est utilisé dans le cas du codage sous forme de suite de bits. On détermine au hasard un point de croisement et on aboute en intervertissant les listes des deux parents pour obtenir les deux enfants (Figure 5).

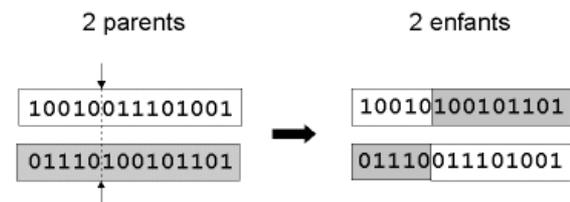


Figure 5 Le simple crossover (Magnin 2002).

2.5.1.4.2. Le double crossover uniforme

2.5.1.4.2.1. Généralités

Le double crossover uniforme (Jourdan 2003) se fait dans le cas d'un codage sous forme de liste de numéros d'arbres et nécessite la mise en place d'un algorithme plus complexe. Il reprend le principe général décrit par la Figure 6, en ajoutant des contraintes empêchant la création de doublon de numéros d'arbres dans les listes enfants.

Ce type de croisement est considéré comme plus efficace que le précédent dans le cas d'un codage des chromosomes sous une autre forme qu'une suite de bits (Beasley 1993, Jourdan 2003).

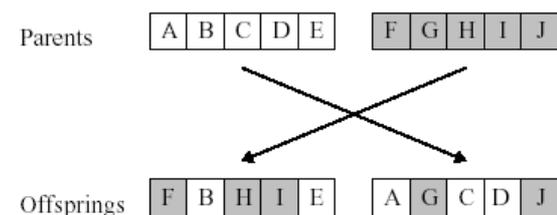


Figure 6 Le double crossover uniforme (Jourdan 2003).

2.5.1.4.2.2. Algorithme du double crossover uniforme

P1, P2 parents 1 et 2.

E1 et **E2** enfants 1 et 2.

size taille d'une liste d'arbres

Début.

Création de **E1** et **E2** listes vides de taille **size**.

Clonage de **P1** et **P2** en **L1** et **L2**.

Détermination «aléatoire» de 2 points de croisement **c1** et **c2** tel que $1 \leq c1 \leq c2 \leq size$.

Pour chaque valeur **i** comprise entre **c1** et **c2**, soit

Ni numéro à l'indice **i** de la liste **P1**.

Recopier **Ni** aux indices **i** d'**E1** et **E2**.

Supprimer **Ni** de **L1** et **L2**.

Fin pour chaque.

Pour chaque valeur **j** comprise entre 0 et **c1**, et entre **c2** et **size**.

Nj numéro à l'indice **j** de la liste **P2**.

Si **Nj** non présent dans **E1** (et donc non présent dans **E2**)

Recopier **Nj** aux indices **j** d'**E1** et **E2**.

Supprimer **Nj** de **L1** et **L2**.

Fin si.

Fin pour chaque.

Pour chaque enfant.

Pour chaque indice **k** de la liste.

Si absence de numéro à l'indice **k**.

Tirer un numéro **N** aléatoirement dans la liste clone correspondant.

(**L1** pour **E1**, et **L2** pour **E2**).

Recopier **N** à l'indice **k**.

Supprimer **N** de la liste clone correspondant.

Fin si.

Fin pour chaque.

Fin pour chaque.

Fin.

2.5.1.5. Les Mutations

Dans le cas d'une liste de bits, la mutation consiste à changer un 0 en 1 et inversement.

Dans le cas d'une liste de numéros, il faudra s'assurer que les numéros ne soient pas répétés suite à la mutation. Le tirage du numéro se fait aléatoirement dans la liste référence d'arbres. Pour éviter de retirer à chaque fois un arbre déjà présent dans la liste en cours de mutation, on inverse les numéros d'indice, soit

Début.

Pour chaque indice **i** d'un individu.

Ni numéro d'indice **i** de l'individu.

Si mutation.

Tirer numéro **N** d'arbre dans liste référence.

Pour chaque indice **j** d'un individu.

Nj numéro d'indice **j** de cet individu.

Si **N = Nj**.

Permuter **Ni** et **Nj**.

Sinon.

Remplacer **Ni** par **N**.

Fin si.

Fin pour chaque.

Fin si.

Fin pour chaque.

Fin.

Routine «Permuter **Ni** et **Nj**»

Remplacer **N** par **Ni**.

Remplacer **Ni** par **Nj**.

Remplacer **Nj** par **N**.

2.5.1.6. Les Mécanismes de sélection roulette

2.5.1.6.1. Généralités

La sélection roulette (ou Wheel selection) est une manière de choisir des membres d'une population d'individus de façon proportionnelle à leurs aptitudes. Elle ne garantit pas que le membre le plus apte survivra à la prochaine génération, mais il a de grandes chances d'y parvenir.

On peut représenter le mécanisme de sélection par le modèle suivant. Les aptitudes de toute la population sont représentées par les secteurs d'une roulette. Chaque secteur correspond à l'aptitude d'un membre de la population. La taille d'un secteur est proportionnelle à l'aptitude (fitness score) de l'individu représenté.

Le membre le plus adapté se voit attribuer le secteur le plus important. Pour choisir un individu, il ne nous reste plus qu'à faire «tourner une boule sur la roulette» (comme une roulette de casino [NTTDR]) et sélectionner l'individu sur lequel celle-ci s'arrête

2.5.1.6.2. Algorithme de la sélection par roulette

On représente les aptitudes des individus par des segments contigus de longueur proportionnelle à celle-ci, et on sélectionne l'individu correspondant au segment qui contient la valeur d'un tirage uniforme aléatoire.

Random = Nombre réel tiré aléatoirement selon une loi Uniforme entre $[0, 1]$.

SommeFitness = Somme des fitness de tous les individus de la population.

Début.

Somme = 0.

j = 0.

Stop = **Random** * **SommeFitness**

Tant que **Somme** ≤ **Stop** ou **j** ≤

Taille de la population, faire

Incrémenter **j** de 1.

Ajouter le fitness de

l'individu **j** de la population

à **Somme**.

Fin tant que.

Retourner individu **j** de la population.

Fin.

2.5.1.7. L'optimisation de l'évolution du fitness

Les processus de sélection présentés sont très sensibles aux écarts de fitness et dans certains cas, un très bon individu risque d'être reproduit trop souvent et peut provoquer l'élimination complète de ses congénères. On obtient alors une population trop homogène contenant une diversité génétique réduite, et dans le pire des cas un seul type d'individu.

Pour éviter ces comportements, il existe des principes comme le scaling et le sharing qui empêchent les individus «forts» d'être trop systématiquement sélectionnés par rapport aux plus «faibles». Il est également possible de modifier le processus de sélection en introduisant des tournois entre parents et enfants, basés sur une technique proche du recuit simulé (basée sur les principes de la thermodynamique).

2.5.1.7.1. Le scaling

Le scaling permet une évaluation graduelle du fitness entre chaque individu. Ce changement d'échelle permet de diminuer ou d'accroître artificiellement l'écart relatif entre les fitness des individus.

Le processus de sélection s'opère sur une fonction de fitness modifiée.

Parmi les fonctions de scaling envisagées, on a le scaling linéaire, le scaling exponentiel, le scaling parabolique, le scaling exponentiel par palier, et le scaling mixte puissance-exponentiel.

2.5.1.7.2. Le sharing

Le sharing, ou partage des modes, influe sur les fitness afin d'éviter le regroupement des individus autour d'un sommet (ou mode) dominant. Il permet la «division» de la population sur plusieurs optimums locaux (Figure 6).

On peut donc utiliser des courbes de

fitness à plusieurs sommets et/ou pénaliser le fitness en fonction du taux d'agrégation de la population dans le voisinage d'un individu : plus les individus sont regroupés, plus leur fitness est faible, et des individus proches les uns des autres doivent partager leur fitness. Dans la pratique, pour estimer ce taux d'agrégation, on ouvre un domaine autour d'un individu, puis on calcule les distances entre les individus contenus dans ce voisinage et ce dernier. Il faut donc pouvoir définir une distance représentative entre deux individus. Cette technique a toutefois l'inconvénient d'avoir une complexité de type quadratique $O(n^2)$.

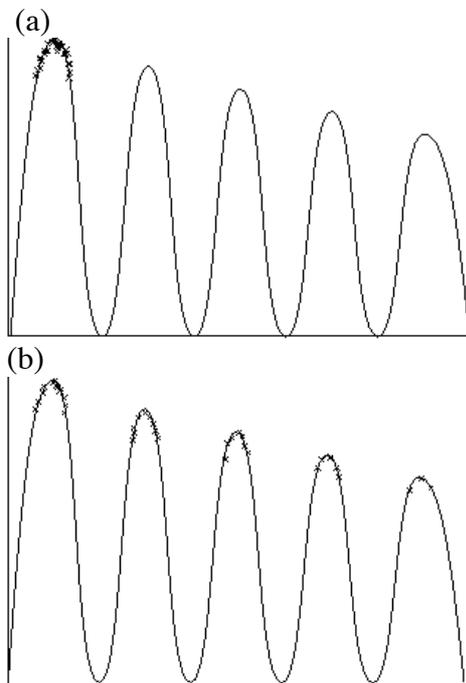


Figure 6 La méthode du sharing. (a. sans sharing : concentration des individus sur un seul mode □ b. avec sharing : répartition des individus sur l'ensemble des modes).

2.5.1.7.3. Exemple □ la sélection du nombre d'arbres à éclaircir

On impose à la courbe de fitness un maximum qui favorisera les individus présentant un nombre donné d'arbres à éclaircir.

Les courbes suivantes sont des courbes de fitness scaling prises dans le cas

de la résolution du problème simple suivant □

Pour un peuplement de N arbres ($N=400$), on cherche à sélectionner les individus ayant 50 arbres éclaircis.

Le maximum de fitness est fixé à « □ □ » pour un nombre de tiges à éclaircir de 50.

On favorise, dans une population de chaînes de 400 bits, les chaînes dont le nombre de « □ □ » tend vers 50. Nous tirerons parti de cette étude pour incorporer l'optimisation du k et la répartition des arbres (comme nous le verrons ultérieurement).

Pour chaque courbe, on a □

$X_{max} = 400$

$x_0 = 60$, pour les graphiques 50 pour les tests

$x_m = X_{max}/2$

$c = 30.0$

Nous avons testé chacune des courbes en notant le nombre de générations nécessaires G_n pour atteindre l'objectif souhaité. Ce nombre est indépendant de la machine sur laquelle le test est réalisé.

Au cours des tests de chacune des courbes, les valeurs fixées par défaut sont, dans le cas où elles ne sont pas précisées :

- La probabilité de mutation $P_{mutation} = 10^{-4}$
- La probabilité de croisement effectif de 0,8
- La population de 400 individus.

Chaque test est répété 10 fois avec une population initiale différente mais contrôlée à chaque fois.

Pour cela, le germe permettant la génération de suites de nombres pseudo-aléatoires par le système est imposé. Pour les dix tests, ces germes sont les nombres 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Ainsi, les différences observées en terme de nombre de générations nécessaires pour atteindre une des solutions seront uniquement dues à la façon dont les données sont traitées par l'algorithme.

Le nombre de génération maximale G_{\max} est fixé à 1000; au delà, l'algorithme s'interrompt.

2.5.1.7.3.1. La courbe linéaire dissymétrique par morceaux (Chapeau chinois)

$$q1(\mathbf{X}) = 1/(x0 - xm) * \mathbf{X} - xm / (x0 - xm)$$

$$q2(\mathbf{X}) = 1/x0 * \mathbf{X}$$

$$q3(\mathbf{X}) = q1(\mathbf{X}) \text{ si } \mathbf{X} < xm, \text{ sinon } 0$$

$$q4(\mathbf{X}) = q2(\mathbf{X}) \text{ si } \mathbf{X} > 0, \text{ sinon } 0$$

$$q(\mathbf{X}) = q3(\mathbf{X}) \text{ si } \mathbf{X} > x0, \text{ sinon } q4(\mathbf{X})$$

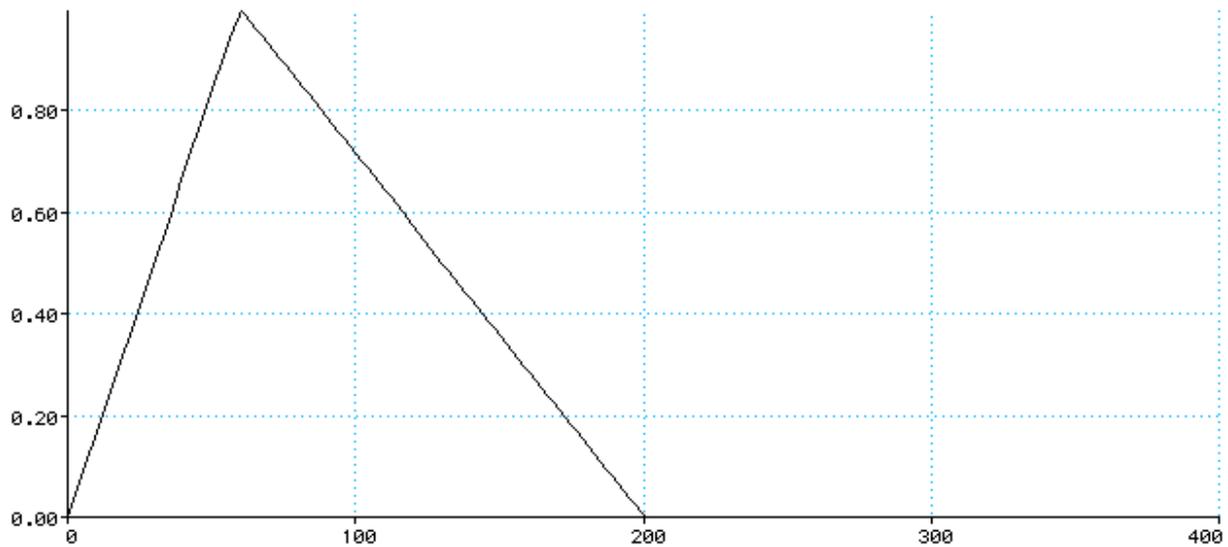


Figure 7 Courbe linéaire dissymétrique par morceaux (Chapeau chinois).

2.5.1.7.3.2. La Courbe exponentielle

$$a = (1 / x0) * c1$$

$$c1 = c / x0$$

$$g1(\mathbf{X}) = \mathbf{exp}(-a * (x0 - \mathbf{X})^2)$$

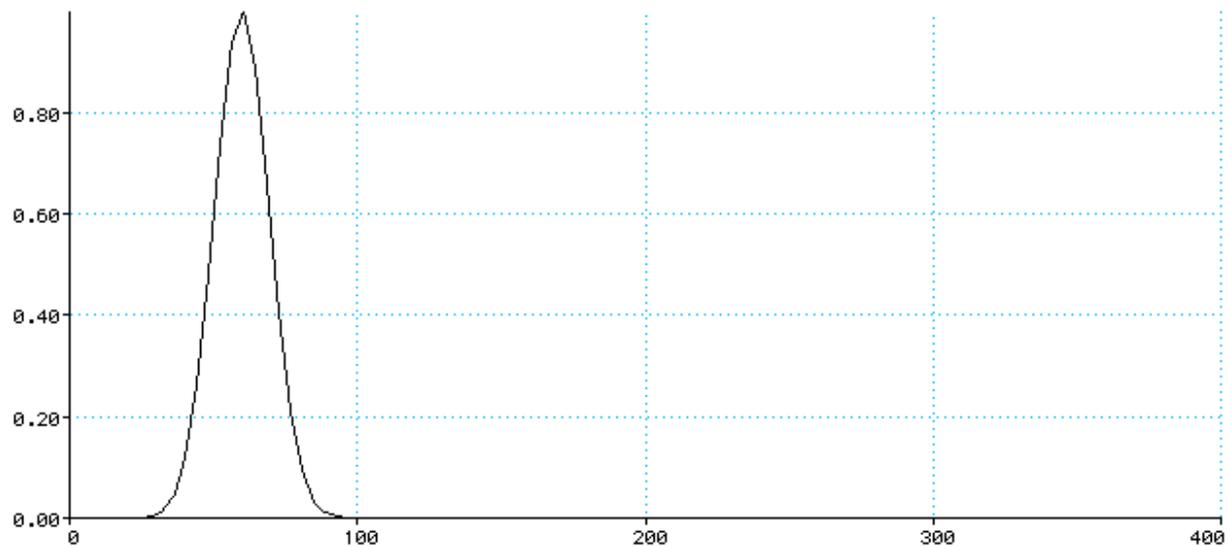


Figure 8 Courbe exponentielle.

2.5.1.7.3.3. La courbe exponentielle dissymétrique par morceaux

La courbe $g(\mathbf{X})$ est la concaténation de deux courbes $g1(\mathbf{X})$ et $g2(\mathbf{X})$ avec dissymétrie en $x0$.

$$c2 = c / (X_{max} - x_0)$$

$$g2(X) = \exp(-b * (X - x_0)^2)$$

$$b = (1 / (X_{max} - x_0)) * c2$$

$$g(X) = g1(X) \text{ si } X < x_0, \text{ sinon } g2(X)$$

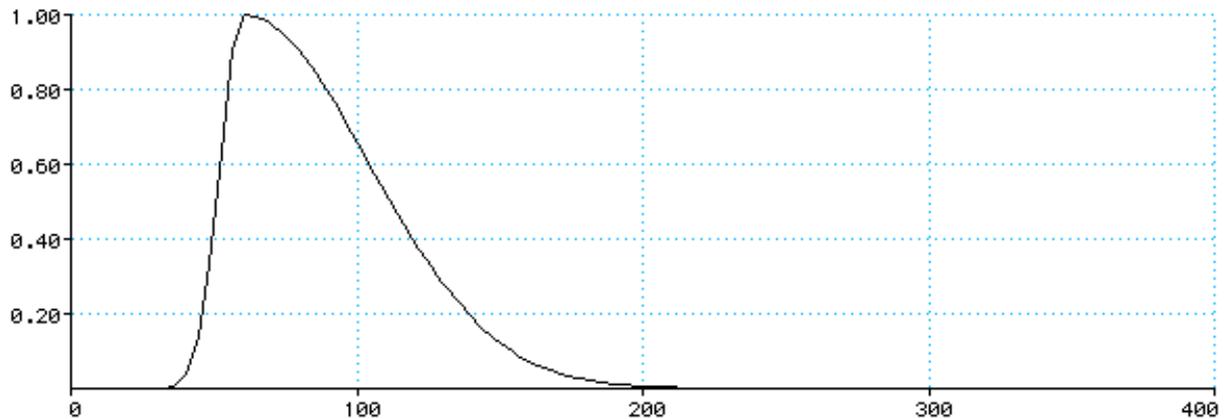


Figure 9 Courbe exponentielle dissymétrique par morceaux

2.5.1.7.3.4. La courbe exponentielle dissymétrique par morceaux à palier

La courbe $h(X)$ est la concaténation de deux courbes $f(X)$ et $g(X)$ avec palier. On a :

$$\begin{aligned} d1 &= 1.2 \\ d2 &= 0.01 \\ d3 &= 1 \end{aligned}$$

$$\begin{aligned} d4 &= 0 \\ \text{palier} &= 0.8 \end{aligned}$$

$$f(X) = d1 * g(X) + d2 \text{ si } g(X) < \text{palier}, \text{ sinon } d3 * g(X) + d4$$

$$h(X) = f(X) \text{ si } f(X) < \text{palier}, \text{ sinon } (g(X) \text{ si } g(X) > \text{palier}, \text{ sinon } \text{palier})$$

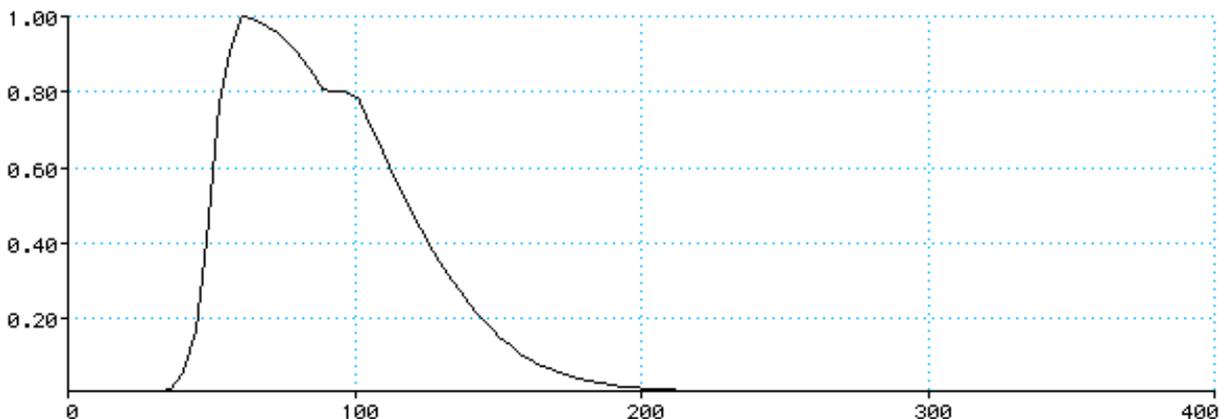


Figure 10 Courbe exponentielle dissymétrique par morceaux à palier.

2.5.1.7.3.5. La courbe parabolique dissymétrique par morceaux

La courbe $p(X)$ est la concaténation de deux courbes $p3(X)$ et $p4(X)$ avec dissymétrie en x_0 . On a :

$$\begin{aligned} xm1 &= xm \\ x1 &= X_{max} - x_0 \end{aligned}$$

$$\begin{aligned} \text{num1} &= 1 - (x1 - xm1) / (x0 - xm1) \\ \text{den1} &= (x1^2 - xm1^2) - (x0^2 - xm1^2) * (x1 - xm1) / (x0 - xm1) \\ t1 &= \text{num1} / \text{den1} \\ t2 &= (1 - t1 * (x0^2 - xm1^2)) / (x0 - xm1) \\ t3 &= -xm1^2 * t1 - xm1 * t2 \end{aligned}$$

$$\begin{aligned} xm2 &= 0 \\ x2 &= -x_0 \end{aligned}$$

$$\begin{aligned} \text{num2} &= 1-(x_2-xm_2)/(x_0-xm_2) \\ \text{den2} &= (x_2^2-xm_2^2)-(x_0^2-xm_2^2)*(x_2-xm_2)/(x_0-xm_2) \\ t_4 &= \text{num2}/\text{den2} \\ t_5 &= (1-t_4*(x_0^2-xm_2^2))/(x_0-xm_2) \\ t_6 &= -xm_2^2*t_4-xm_2*t_5 \end{aligned}$$

$$p_1(X) = t_1*X^2+t_2*X+t_3$$

$$p_2(X) = t_4*X^2+t_5*X+t_6$$

$$p_3(X) = p_1(X) \text{ si } X < xm_1, \text{ sinon } 0$$

$$p_4(X) = p_2(X) \text{ si } X > 0, \text{ sinon } 0$$

$$p(X) = p_3(X) \text{ si } X > x_0, \text{ sinon } p_4(X)$$

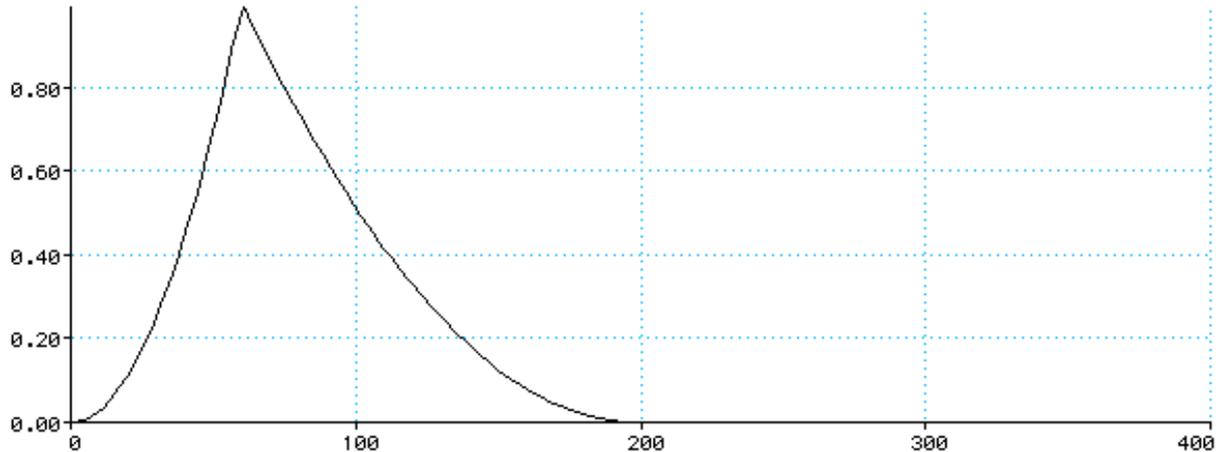


Figure 11 Courbe parabolique dissymétrique par morceaux.

2.5.1.7.3.6. La Courbe mixte puissance/exponentielle dissymétrique par morceaux (Courbe en aileron de requin)

$$n_1(X) = p_4(X) \text{ si } g_1(X) > p_4(X), \text{ sinon } g_1(X)$$

$$n_2(X) = p_3(X) \text{ si } g_2(X) > p_3(X), \text{ sinon } g_2(X)$$

$$k(X) = n_1(X) \text{ si } X < x_0, \text{ sinon } n_2(X)$$

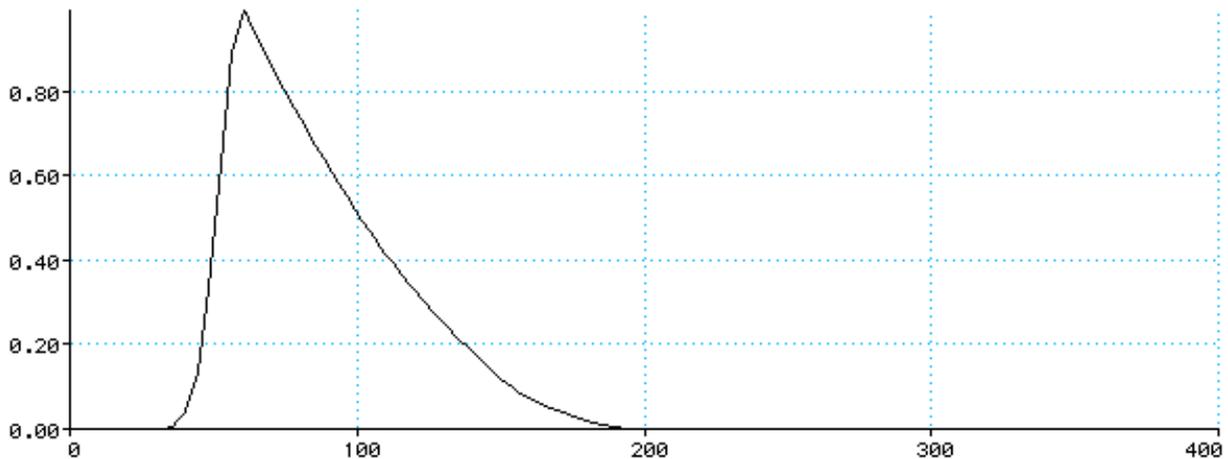


Figure 12 Courbe en aileron de requin.

2.5.1.7.3.7. La courbe en aileron de requin avec effet de sharing initial

$$n_3(X) = p_1(X) \text{ si } g_1(X) > p_1(X), \text{ sinon } g_1(X)$$

$$n_4(X) = p_1(X) \text{ si } g_2(X) > p_1(X), \text{ sinon } g_2(X)$$

$$l(X) = n_3(X) \text{ si } X < x_0, \text{ sinon } n_4(X)$$

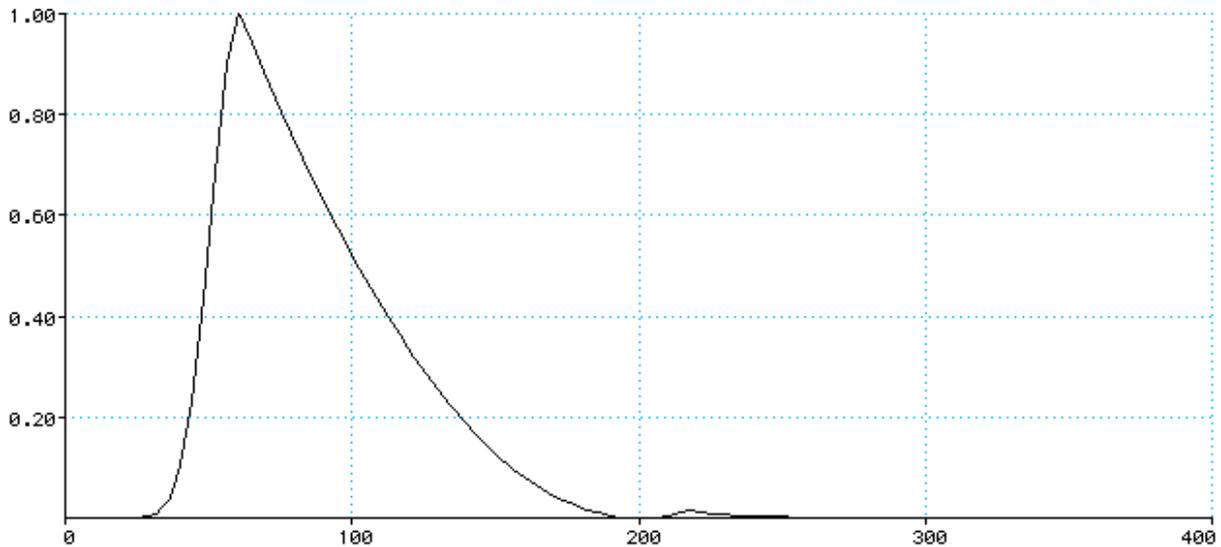


Figure 13 Courbe en aileron de requin avec effet de sharing initial.

2.5.2. La répartition spatiale des arbres

2.5.2.1. La méthode du plus proche voisin, principe général

Soit un ensemble de N points distribués sur une surface S , on note d la densité moyenne des points par unité de surface à l'intérieur de l'espace considéré $d = \frac{N}{S}$. Pour chaque point i , on calcule la distance $D_{\min(i)}$ qui le sépare de son voisin le plus proche. On calcule ensuite la moyenne D_0 des distances au plus proche

voisin. La distance théorique moyenne au plus proche voisin D_T dans le cas d'une distribution aléatoire uniforme est donnée par :

$$D_T = 0.5/\sqrt{d}$$

L'indice de dispersion est donné par :

$$R = D_0 \square D_T$$

La valeur de cet indice permet de se faire une idée de la forme de la distributions, sachant qu'elle varie entre 0 (concentration totale) et 2.149 (dispersion maximale selon une grille hexagonale).

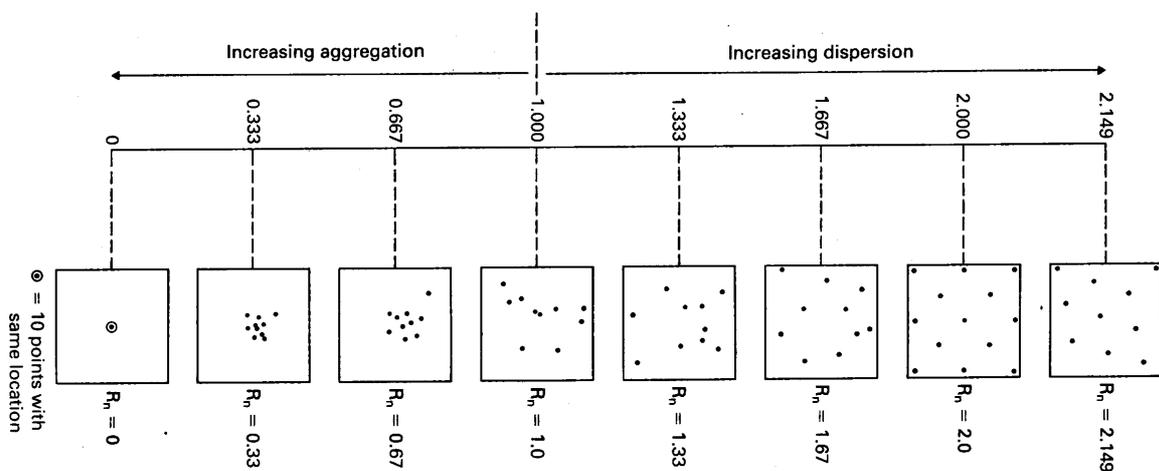


Figure 14 Dispersion des points en fonction de l'indice de dispersion de la méthode du plus proche voisin.

Le principe de cette méthode est simple mais difficile à mettre en oeuvre (plus proches voisins d'ordre n). La complexité algorithmique de cette méthode est de type quadratique $O(n^2)$.

La complexité de cet algorithme est beaucoup trop grande pour obtenir un résultat en un temps raisonnable, même si on peut utiliser une puissance de calcul phénoménale.

On sera donc conduit à rechercher une solution la plus proche possible d'une solution optimale en procédant par essais successifs. Puisque toutes les combinaisons ne peuvent être essayées, certains choix stratégiques doivent être faits. Ces choix sont très dépendants du problème traité, et constituent ce qu'on appelle une heuristique. Le but d'une heuristique est donc de ne pas essayer toutes les combinaisons possibles avant de trouver celle qui répond au problème, afin de trouver une solution approchée convenable (qui peut être exacte dans certains cas) dans un temps raisonnable.

2.5.2.2. L'inertie, calcul de dispersion autour d'un point central, Distance-type / Point moyen

Tout comme une valeur centrale est précisée par un paramètre de dispersion statistique, un point central peut être accompagné d'un paramètre de dispersion spatiale qui indique l'éloignement des points ou de la population par rapport à celui-ci. On utilisera des paramètres différents selon que l'on a retenu comme point central le point moyen ou le point médian.

Ayant déterminé le point moyen, on peut chercher à mesurer la dispersion des lieux autour de ce point. Par analogie avec la statistique univariée, on peut mesurer la dispersion des lieux autour du point moyen. Pour cela, on calcule la dispersion, ou l'inertie, qui est définie par la formule de calcul suivante dans le cas de calcul non pondéré.

Le paramètre de dispersion associé

est la distance-type sD qui est la racine carrée de la moyenne du carré des distances, c'est-à-dire la racine carrée de la somme des variances de X et de Y.

$$\begin{aligned} \sigma_D &= \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 + (Y_i - \bar{Y})^2} \\ &= \sqrt{\sigma_X^2 + \sigma_Y^2} \end{aligned}$$

2.5.2.3. Méthode des Quadrats

2.5.2.3.1. Principe général

La méthode des Quadrats consiste à diviser l'espace d'étude d'un maillage carré. Le principe général de constitution des Quadrats est que chaque carreau doit avoir a priori la même probabilité d'accueillir un point.

2.5.2.3.2. Calcul de l'indice de concentration

La seconde étape de l'analyse consiste à examiner la distribution des fréquences de points par carreaux et à déterminer si elle se rapproche plutôt d'une distribution concentrée ou régulière. Pour cela, on dispose d'un indice qui est le rapport entre la densité moyenne de points par carreaux (D) et la variance de cette densité $V(D)$. Le rapport variance/densité $V(D)/D$ tend vers 1 dans le cas d'une distribution régulière.

La densité moyenne est calculée selon la formule $D = \frac{N}{\sum K_n}$, avec N le nombre

total de points, K_n le nombre de Quadrats de n points, et la variance selon

$$V(D) = \frac{\sum_{n=0}^{n_{\max}} K_n (n - D)^2}{\left(\sum K_n\right) - 1}$$

Par exemple, un rapport de 1.5 signale une tendance assez marquée à la concentration de points dans certains secteurs du territoire. La variance est élevée, ce qui signifie que quelques carreaux sont pleins et beaucoup d'autres vides.

2.5.3. Application de la théorie des algorithmes génétiques à l'éclaircie

Le calcul du fitness d'un individu i lors de l'éclaircie se fera sous la forme d'une combinaison de fonctions

$$fitness(i) = a \cdot f(i) + b \cdot g(i) + c \cdot h(i)$$

Avec $f(i)$ pour le nombre d'arbres à éclaircir, $g(i)$ pour le coefficient d'éclaircie k , et $h(i)$ pour la répartition spatiale.

a , b , c sont des facteurs à ajuster.

3. Résultats et discussion

3.1. Sélection numérique guidée

Les résultats obtenus lors de la sélection automatique guidée sont conformes aux prévisions. Un suréchantillonnage des petites classes entraîne une diminution de k . Un suréchantillonnage des grandes classes entraîne une augmentation de k . Les contraintes ne permettent de déterminer au plus que deux coefficients sur trois.

3.2. Comparaison des propriétés générales des courbes

Le nombre de générations nécessaires à l'algorithme génétique utilisant la courbe en chapeau chinois pour trouver une solution satisfaisante au problème posé (c'est-à-dire sélectionner un individu de 50 arbres éclaircis dans une population de 400 individus) est supérieur au nombre maximum fixé. La graduation du fitness entre chaque individu n'est pas assez prononcée pour que la population converge rapidement vers l'optimum imposé.

Pour une courbe exponentielle par morceau, dissymétrique, à palier, on note que la population a de grandes difficultés à franchir le palier. Lorsque les individus se trouvent au niveau du palier, une variation dynamique des taux de mutation et/ou de croisement n'arrive pas à résoudre ce problème. Toutefois, une augmentation de la diversité de la population a été observée.

On note la présence d'individus de même fitness, mais de séquence chromosomique couvrant une large gamme de possibilité. Cette solution pourrait être utile pour empêcher que l'évolution de la population soit limitée par les optima locaux.

Pour la courbe exponentielle, le nombre de génération nécessaire varie de façon aléatoire en fonction des caractéristiques de la population d'origine. D'autre part, si l'on donne trop peu de valeur aux individus qui ont des caractéristiques éloignées de celles que l'on veut valoriser, alors il n'y a pas démarrage de l'évolution

Les autres courbes donnent des résultats comparables (Tableau 1). Les meilleurs résultats sont obtenus avec la courbe en aileron de requin. Elle bénéficie d'une meilleure convergence vers l'optimum hérité de la courbe parabolique, et d'une meilleure pente initiale héritée de la courbe dissymétrique exponentielle.

Pour cette courbe, les meilleurs paramètres de dimensionnement du problème sont une population de 1000 individus et une probabilité de mutation de 10^{-3} (Tableau 2).

3.3. La sélection pour le nombre d'arbres à éclaircir

Pour la sélection du nombre de tiges à éclaircir, la courbe de fitness en aileron de requin, avec comme valeur à évaluer le nombre d'arbres éclaircis, a donnée les meilleurs résultats.

3.4. La sélection pour le coefficient d'éclaircie

Le sélection du coefficient d'éclaircie a été effectuée à l'aide de la courbe parabolique. Le choix de cette courbe se justifie par le fait qu'elle couvre l'ensemble des valeurs de k possible. Contrairement au nombre d'arbres à éclaircir, on ignore à quelle valeur le k moyen probable de la population initiale va se situer.

3.5. La sélection pour les critères de la répartition spatiale

La sélection pour la répartition spatiale a été effectuée à l'aide de la courbe parabolique, avec comme valeur à évaluer l'indice de dispersion correspondant à la méthode choisie. Ce choix se justifie pour les mêmes raisons

que le coefficient d'éclaircie.

Les sommets de chaque courbe sont ajustés respectivement sur l'indice R de 2.149 pour la méthode du plus proche voisin, et le rapport $V(D)/D$ de 1 pour la méthode des Quadrats. La diagonale de la placette donne une valeur maximum de l'inertie

Tableau 1 Comparaison des performances des courbes de fitness en terme de nombre de générations nécessaires à la sélection d'un individu présentant 50 tiges à éclaircir. La taille de la population est de 400 et la probabilité de mutation de 10^{-4} .

Population initiale contrôlée	Nombre de générations par courbe de fitness			
	Exponentielle Dissymétrique	Parabolique	Aileron de requin Sharing	Aileron de requin
1	505	246	340	313
2	517	421	291	241
3	687	471	372	350
4	857	479	556	445
5	288	400	359	299
6	452	251	396	284
7	702	265	339	288
8	477	337	278	388
9	340	392	249	342
10	535	319	314	349
Moyenne	536	358,1	349,4	329,9
Ecart-Type	171,7	87,5	85,3	58,3

Tableau 2 Comparaison des performances induites en terme de nombre de générations nécessaires à la sélection d'un individu présentant 50 tiges à éclaircir par les paramètres de dimensionnement. La courbe de fitness est la courbe en aileron de requin. Ces paramètres sont la taille de la population (T_p) et la probabilité de mutation (P_m).

Population initiale contrôlée	Nombre de générations en fonction de la taille de la population initiale T_p et de la probabilité de mutation P_m			
	T_p 1000	T_p 1000	T_p 400	T_p 400
	P_m 0,001	P_m 0,0001	P_m 0,001	P_m 0,0001
1	156	146	200	313
2	148	153	184	241
3	166	159	293	350
4	151	161	218	445
5	114	160	212	299
6	139	154	190	284
7	115	164	205	288
8	157	173	195	388
9	116	147	223	342
10	144	154	231	349
Moyenne	140,6	157,1	215,1	329,9
Ecart-Type	19,1	8,1	31,1	58,3

3.6. Analyse des méthodes d'évaluation de la répartition des arbres éclaircis

3.6.1. Difficultés rencontrées avec la méthode du plus proche voisin

La méthode du plus proche voisin possède un certain nombre de défauts et de limites, qui peuvent être partiellement résolus, comme

- Les effets de bordure : Lorsque l'espace d'étude est délimité de façon arbitraire, on risque de surestimer la distance au plus proche voisin des points situés en périphérie, car leur plus proche voisin peut se situer en dehors de la zone d'étude. Ce problème peut être résolu en ajoutant une zone "buffer", un couloir autour de la zone d'étude où les points sont relevés mais pas utilisés dans l'analyse.

- Les effets d'échelle : Il peut arriver que des agrégations se produisent à un niveau, par exemple, lorsque les points vont 2 par 2, ce qui va engendrer une impression de concentration qui serait démentie à une autre échelle. On peut résoudre ce problème en complétant l'analyse du plus proche voisin par l'analyse de la distance aux 2 e, 3 e, 4 e... voisins (Figures 15 et 16).

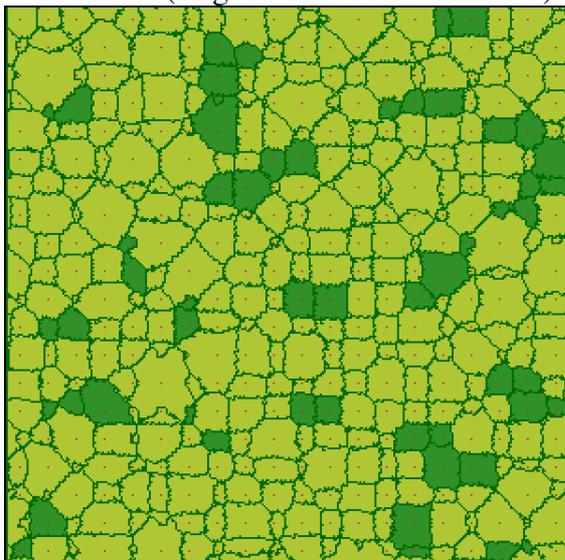


Figure 15 Effet d'agrégation (Exemple après 1000 générations: $N_e=56$ et $R=2.1$).

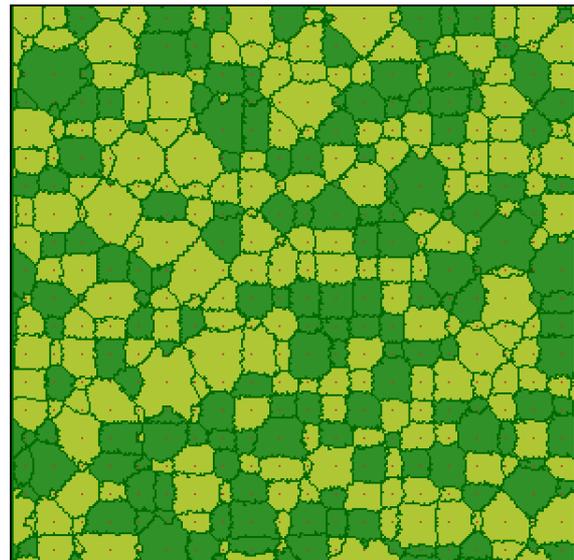


Figure 16 Répartition spatiale de l'éclaircie par la méthode du plus proche voisin (Exemple après 9 générations \square fitness=1, $N_e=175$, $R=2.149$, $\square D = 121.65$, $V(D)/D = 0.43$).

3.6.2. Difficultés rencontrées avec la méthode de calcul de la dispersion autour d'un point central

Cette méthode a pour conséquence de repousser les arbres sélectionnés vers les limites de la carte du peuplement (Figure 17).

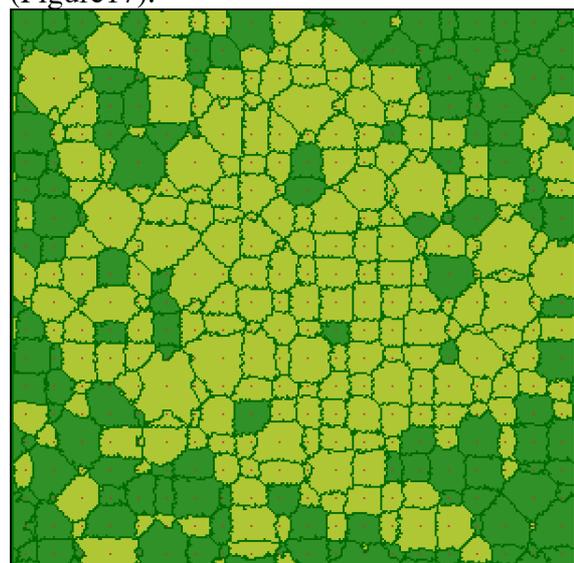


Figure 17 Répartition spatiale de l'éclaircie par la méthode de calcul de l'Inertie (Exemple après 1000 générations, : $N_e=150$, $R=2,21$ $\square D=145.36$, $V(D)/D=0.75$).

3.6.3. Difficultés rencontrées avec la méthode des Quadrats

La méthode des Quadrats n'est pas exempte de faiblesses. Le choix de la taille et de la forme des carreaux peut en effet entraîner des conclusions différentes pour une même distribution. Ceci peut dans certains cas révéler des problèmes d'échelle, une distribution pouvant être concentrée à l'échelle locale et dispersée aux échelles supérieures, et inversement (Figure 18).

De plus, cette méthode, lorsqu'elle est utilisée comme seul critère d'évaluation du fitness, a une certaine tendance à diminuer spontanément et fortement le nombre d'arbres à éclaircir (de 40 à 50 arbres environ pour une population initiale de 200 arbres éclaircis en moyenne à partir d'un peuplement de 400 arbres). Ceci pose problème pour son intégration dans un polynôme de calcul du fitness comprenant le nombre espéré d'arbres à éclaircir. En effet, l'intégration de deux caractères évolutifs dans un polynôme n'est en principe possible qu'à la seule condition que ces deux caractères soient peu ou pas liés génétiquement. Il est donc nécessaire que la sélection d'un caractère soit relativement indépendante de la sélection d'un second caractère.

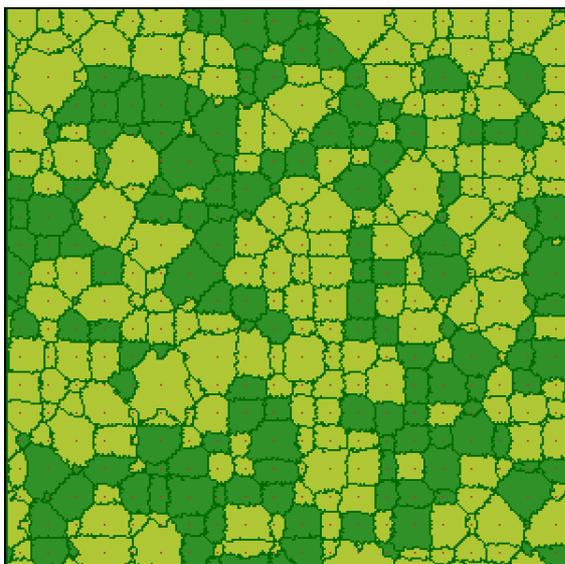


Figure 18 Répartition spatiale de l'éclaircie par la méthode des Quadrats. (Exemple après 47 générations \square fitness=1.0, $N_e=160$, $V(D)/D = 1.0$)

3.7. Combinaison des valeurs d'adaptation

3.7.1. Combinaison des contraintes portant sur le nombre de tige et le facteur d'éclaircie dans une fonction de fitness

Dans un premier temps, on cherche à optimiser à la fois le k souhaité et le nombre de tiges à éclaircir.

Le calcul du fitness d'un individu, se fera sous la forme d'une combinaison: $fitness(i) = a \square f(i) + b \square g(i)$.

On se donne comme valeur maximale de fitness 2. Ce maximum est atteint pour les valeurs maximales de k et n_e , soit 1 et 1. Pour le nombre d'arbres à éclaircir et le k souhaités, on a:

$$fitness(i \max) = a \square f(i \max) + b \square g(i \max) \\ = a \square 1 + b \square 1 = a + b = 2$$

Le choix de a et b peut être arbitraire tant que leur somme est égale à 2, que tous deux sont positifs, et qu'ils ne sont pas égaux à 0 ou 2 ($0 < a < 2$, $0 < b < 2$, et $a + b = 2$).

Toutefois, l'algorithme privilégie certaines valeurs qui permettront à la population de tendre plus rapidement vers l'optimum imposé. On accorde la priorité, donc le plus grand facteur, à la fonction dont on suppose que cet optimum est le plus éloigné du point moyen d'origine de la population.

Pour le cas d'un nombre d'arbres à éclaircir de 50 sur un peuplement de 400 arbres, et un k souhaité de 0,8, les valeurs $a=1.8$ et $b=0.2$ ont donné des résultats acceptables en ce qui concerne le temps de calcul nécessaire pour trouver une valeur finale acceptable. L'optimum de chacune de ces deux valeurs n'a pas été recherché car il dépend du nombre d'arbres à éclaircir, de la taille du peuplement, de sa répartition en classes de volume de ces arbres et du k souhaité.

3.7.2. Combinaison des contraintes portant sur le nombre de tiges, le facteur d'éclaircie et la répartition spatiale dans une fonction de fitness

Sur le même principe, on cherche maintenant à combiner la répartition spatiale au k et au nombre de tiges à éclaircir.

Le calcul du fitness se fera sous la forme:

$$fitness(i) = a \cdot f(i) + b \cdot g(i) + c \cdot h(i)$$

L'optimum imposé sera 3, d'où:

$$fitness(i_{max})$$

$$= a \cdot f(i_{max}) + b \cdot g(i_{max}) + c \cdot h(i_{max})$$

$$= a \cdot 1 + b \cdot 1 + c \cdot 1$$

$$= a + b + c = 3$$

A priori, le choix de a , b et c peut être également arbitraire tant que leur somme est égale à 3, que tous trois sont positifs, et qu'ils ne sont pas égaux à 0 ou 3 ($0 < a < 3$, $0 < b < 3$, $0 < c < 3$ et $a + b + c = 3$).

Pour le cas d'un nombre d'arbres à éclaircir de 50 sur un peuplement de 400 arbres, et un k souhaité de 0,8, les valeurs $a=2,0$, $b=0,7$, et $c=0,3$ ont donné des résultats acceptables en ce qui concerne le temps de calcul nécessaire pour trouver une valeur finale acceptable.

Remerciements

Je remercie J.M. Ottorini pour la grande liberté et la grande confiance accordées à mon travail, pour son soutien indispensable à la rédaction ainsi qu'à l'évaluation de ce manuscrit, G. Le Moguedec, pour ses supports dans les analyses mathématiques et N. Le Goff pour ses aides à la mise en page.

Bibliographie

Beasley, D., Bull, D.R., et Martin, R.R., «An Overview of Genetic Algorithms: Part 2, Research topics », University Computing, vol. 15, n°4, p. 170-181, 1993.

Darwin Charles, «The Origin of Species ». 1859.

Dhôte, J.F., Le Moguedec, G., «Présentation du modèle Fagacées» 2 Février 2005, p 16.

Goldberg David E., «Genetic Algorithms in Search, Optimization and Machine Learning». The University of Alabama. Addison-Wesley Publishing Company, INC.1989.

Jourdan Leatitia, «Métaheuristique pour l'extraction de connaissances» application à la génomique», thèse Université de sciences et technologies de Lille, U.F.R. d'I.E.E.A., Numéro d'ordre»368. Année 2003.

Magnin Vincent, «Algorithmes Evolutionnaires et Algorithmes Génétiques».

http://www.eudil.fr/~vmagnin/coursag/alg_o_evo.html , Mai 2002.

Mitchell, K.J., «Dynamics and Simulated Yield of Douglas-fir». Forest Science Monograph, vol. 17, 1975, p 39.

Ottorini, J.M., «Growth and development of individual Douglas-fir in stands for applications to simulation in silviculture.» Annales des Sciences forestières, vol. 48, 1991, pp. 651-666.

Wand, M.P., "Statistical Computing and Graphics, Data-Based Choice of Histogram Bin Width" The American Statistician, February 1997, vol 51, No. 1.

Sommaire

Pages

1	1. Introduction
1	2. Matériel et méthodes
1	2.1. Les caractéristiques d'une éclaircie
1	2.2. Données fournies par le simulateur
2	2.3. L'interface graphique
3	2.4. La sélection numérique guidée
3	2.4.1. Répartition des arbres en classes
3	2.4.2. Prélèvement des arbres dans les classes
3	2.4.2.1. Généralités
3	2.4.2.2. Échantillonnage représentatif stratifié
4	2.4.2.3. Échantillonnage biaisé
4	2.4.3. Procédure de sélection des arbres à éclaircir
4	2.4.3.1. Algorithme 1
4	2.4.3.2. Algorithme 2
5	2.4.3.3. Algorithme 3
5	2.5. La sélection numérique autonome
5	2.5.1. Les algorithmes génétiques
5	2.5.1.1. Principe général
6	2.5.1.2. Le fitness ou adaptabilité
6	2.5.1.3. Les différentes approches du codage de l'algorithme génétique
6	2.5.1.3.1. Première approche du codage
7	2.5.1.3.2. Seconde approche du codage
7	2.5.1.4. Les Crossovers
7	2.5.1.4.1. Le simple crossover, ou croisement en un point
7	2.5.1.4.2. Le double crossover uniforme
7	2.5.1.4.2.1. Généralités
8	2.5.1.4.2.2. Algorithme du double crossover uniforme
8	2.5.1.5. Les Mutations
8	2.5.1.6. Les Mécanismes de sélection roulette
8	2.5.1.6.1. Généralités
9	2.5.1.6.2. Algorithme de la sélection par roulette.
9	2.5.1.7. L'optimisation de l'évolution du fitness
9	2.5.1.7.1. Le scaling
9	2.5.1.7.2. Le sharing
10	2.5.1.7.3. Exemple de la sélection du nombre d'arbres à éclaircir
11	2.5.1.7.3.1. La courbe linéaire dissymétrique par morceaux (Chapeau chinois)
11	2.5.1.7.3.2. La Courbe exponentielle
11	2.5.1.7.3.3. La courbe exponentielle dissymétrique par morceaux

12	2.5.1.7.3.4. La courbe exponentielle dissymétrique par morceaux à palier
12	2.5.1.7.3.5. La courbe parabolique dissymétrique par morceaux
13	2.5.1.7.3.6. La Courbe mixte puissance/exponentielle dissymétrique par morceaux (Courbe en aileron de requin)
13	2.5.1.7.3.7. La courbe en aileron de requin avec effet de sharing initial
14	2.5.2. La répartition spatiale des arbres
14	2.5.2.1. La méthode du plus proche voisin, principe général
15	2.5.2.2. L'inertie, calcul de la dispersion autour d'un point central, Distance-type / point moyen
15	2.5.2.3. Méthode des Quadrats
15	2.5.2.3.1. Principe général
15	2.5.2.3.2. Calcul de l'indice de concentration
16	2.5.3. Application de la théorie des algorithmes génétiques à l'éclaircie
16	3. Résultats et Discussion
16	3.1. Sélection numérique guidée
16	3.2. Comparaison des propriétés générales des courbes
16	3.3. La sélection pour le nombre d'arbres à éclaircir
16	3.4. La sélection pour le coefficient d'éclaircie
17	3.5. La sélection pour les critères de la répartition spatiale
18	3.6. Analyse des méthodes d'évaluation de la répartition des arbres éclaircis
18	3.6.1. Difficultés rencontrées avec la méthode du plus proche voisin
18	3.6.2. Difficultés rencontrées avec la méthode de calcul de la dispersion autour d'un point central
19	3.6.3. Difficultés rencontrées avec la méthode des Quadrats
19	3.7. Combinaison des valeurs d'adaptation
19	3.7.1. Combinaison des contraintes portant sur le nombre de tige et le facteur d'éclaircie dans une fonction de fitness
20	3.7.2. Combinaison des contraintes portant sur le nombre de tiges, le facteur d'éclaircie et la répartition spatiale dans une fonction de fitness
20	Remerciements
20	Bibliographie