

Simulateur d'Incendie

Vidal ATTIAS - 2017

• *Simulation numérique* • *Modèle probabiliste* • *Complexité*

TABLE DES MATIÈRES

I.	MODÉLISATION DE LA GÉOGRAPHIE	2
II.	MODÉLISATION NAÏVE DE LA PROPAGATION	5
III.	COMMENTAIRES SUR LA SIMULATION	7
IV.	LA COMPLEXITÉ DU PROGRAMME	8
V.	ALGORITHME GÉNÉRIQUE	10

INTRODUCTION

Le projet présenté ici est de simuler de façon numérique la propagation d'un feu en forêt. Pour cela, nous allons modéliser notre forêt par une matrice carrée de 50×50 . Nous travaillerons avec plusieurs paramètres différents nous aurons donc autant de matrices dont les coordonnées correspondront à un même point de l'espace. Nous allons par la suite construire un modèle d'automate probabiliste et le programmer en langage Python. Finalement, après nous être assurés que le modèle n'est pas trop aberrant d'un point de vue physique, même si ce n'est pas le but du projet, nous allons essayer de faire de l'optimisation d'un point de vue informatique, pour gagner en complexité.

I. MODÉLISATION DE LA GÉOGRAPHIE

Pour avoir assez de paramètres pour construire un modèle un peu complexe, nous allons introduire les notions d'hygrométrie et de terrain combustible.

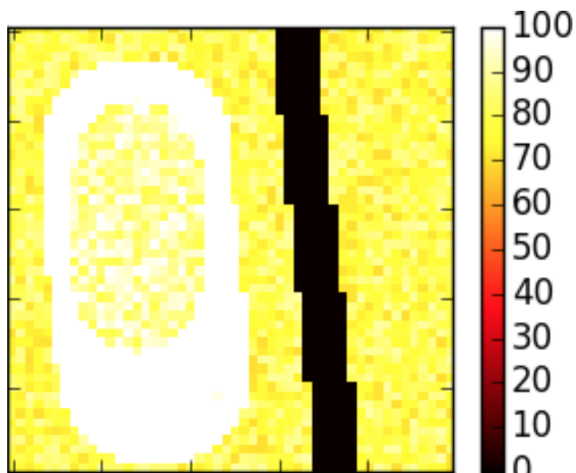


FIGURE 1 – Combustible

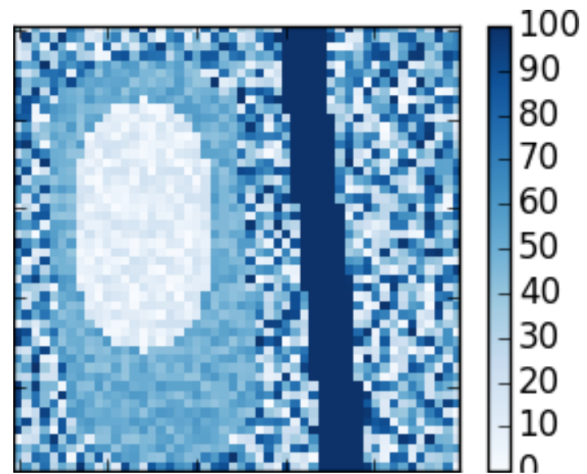


FIGURE 2 – Hygrométrie

L'hygrométrie correspond à l'humidité du terrain en pourcentage en chaque case et le combustible à une certaine réserve de matière à consommer en pourcentage, ce qui permet de s'assurer que le feu ne sera pas éternel, car une réserve faible de combustible impliquera une mort du feu. Ces deux paramètres joueront aussi un rôle dans l'efficacité de la propagation du feu.

De plus, nous supposons l'existence de quatre types de terrain :

- **L'HERBE** : C'est celle qui prédomine, de couleur verte.
 - Hygrométrie : 0-100
 - Combustible : 70-90
- **L'EAU** : C'est un terrain très simple qui va servir à modéliser une rivière et aura le simple but de vérifier que le feu ne traverse pas l'eau.
 - Hygrométrie : 100
 - Combustible : 0
- **LA FORÊT SÈCHE** : C'est un terrain représenté en gris.
 - Hygrométrie : 0-20
 - Combustible : 80-100
- **LA FORÊT HUMIDE** : C'est un terrain représenté en marron.
 - Hygrométrie : 40-60
 - Combustible : 100

De plus, nous allons mettre en place une matrice dite « matrice de feu », qui donne l'avancement du feu pour chaque case. Le feu est en intensité de 0 à 3, 0 étant l'absence de feu et 3 l'intensité maximale du feu.

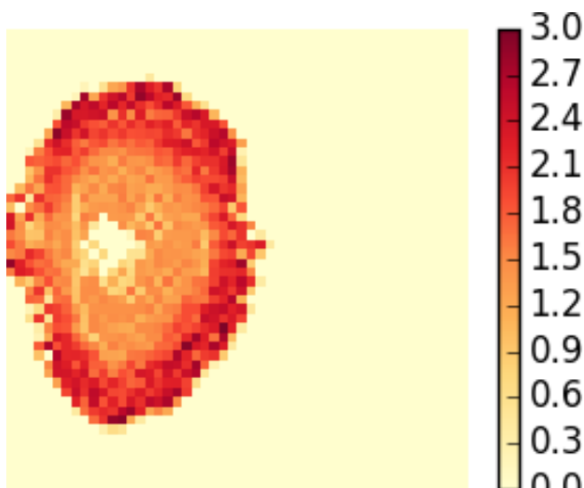


FIGURE 3 – Matrice feu

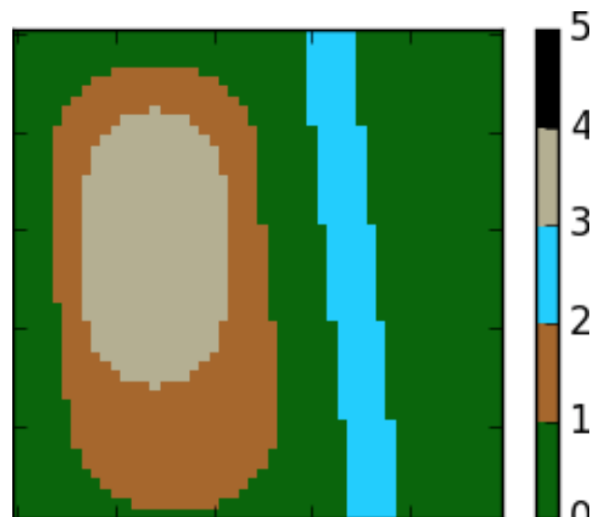


FIGURE 4 – Visuel du terrain

Vue d'ensemble des matrices

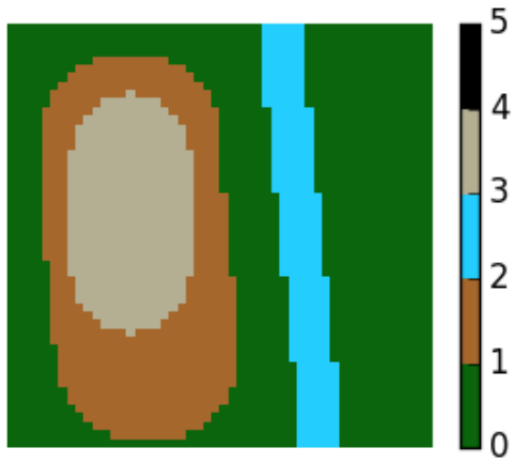


FIGURE 5 – Nature du terrain

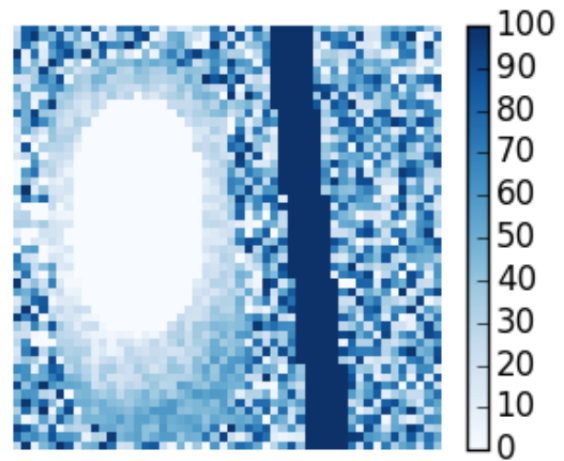


FIGURE 6 – Hygrométrie

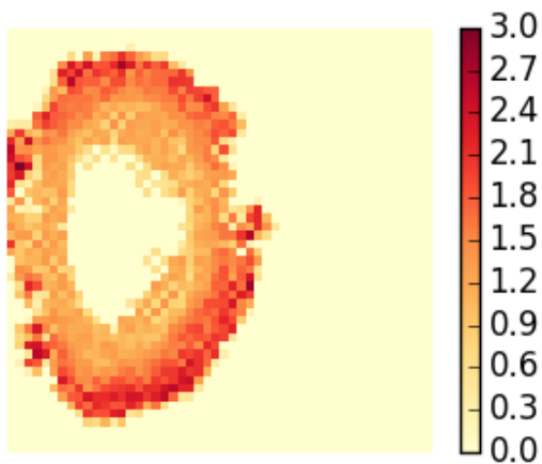


FIGURE 7 – État du feu

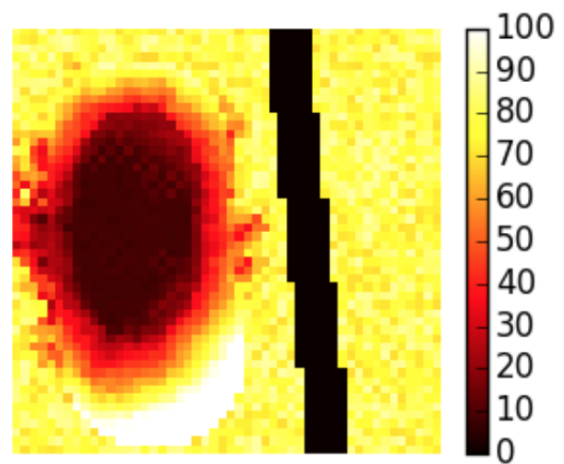


FIGURE 8 – Combustible terrain

II. MODÉLISATION NAÏVE DE LA PROPAGATION

La propagation du feu se fera ici en plusieurs étapes. Nous allons principalement raisonner en termes d'objets. J'ai créé deux objets, `Interface` qui s'occupera de tout ce qui est interface, c'est à dire l'affichage, la détermination des points à calculer etc., et un objet `Feu` qui s'occupera de calculer la propagation du feu en un point donné. On initialise le feu en allumant un foyer à 3 en un certain point (j'ai ici systématiquement pris le point [23, 10] pour des raisons que j'expliciterais plus loin). Le déroulement d'une génération suit le plan suivant :

1. On crée une matrice `matFeuTemp` qui servira de matrice tampon pour pouvoir mettre à jour le feu de la matrice au rang `n+1` sans modifier la matrice du rang `n`, dont on aura toujours besoin pour les autres points.
2. On parcourt la matrice de feu avec une double boucle `for` pour les `x` et les `y`. On définit que si la case a un intensité de feu d'au moins 1, elle est potentiellement capable d'enflammer ses voisines. On lance donc la méthode `Feu.propagerFeu(case, Interface)` qui permet cette propagation. Cette méthode commence par calculer une probabilité associée à chaque case adjacente probabilité qui se formule :

$$p = \frac{\text{combustible} \cdot (100 - \text{hygrometrie})}{100^2}$$

Avec `hygrometrie` et `combustible` les valeurs correspondantes à la case adjacente en question. On fait tirage à l'aide de la fonction `random`, et si la condition de probabilité est satisfaite, alors on définit une intensité maximale possible via la formule

$$\text{feuMax} = \text{combustibleAdj} \cdot (100 - \text{hygroCase}) \cdot \frac{3}{100^2}$$

L'intensité de la case adjacente sera alors le maximum du feu maximal et de la somme du feu à l'état `n` et d'une quantité additive, donnée par la formule

$$\text{add} = 0.1 + 0.4 \cdot (\text{feuAdj} - 1)$$

On va enfin inscrire un booléen sur la case adjacente qui vient d'être traitée signifiant qu'elle a déjà été visitée et ne peut plus être évaluée pour cette génération. On fait cette opération pour les quatre cases autour de la case étudiée.

Une fois la méthode **Feu.propagerFeu(case, Interface)** terminée, on évalue pour la case en question son combustible et l'humidité. Pour cela, on va juste retrancher au combustible l'intensité du feu. Pour l'hygrométrie, on va retrancher la moitié de l'intensité du feu. On fait toujours bien attention à s'arrêter à 0 pour éviter les valeurs négatives.

Exemple : si notre case [i] [j] a à la génération **n** une hygrométrie de 80 et un combustible de 50 et une intensité de 2, alors à la génération **n+1**, son hygrométrie sera de 79 et son combustible 48.

3. On va ici remplacer la matrice de feu de la génération **n** par la matrice de feu tampon, afin d'avoir notre génération **n+1**. De plus, si le combustible est inférieur à 10, on considère que le terrain ne peut plus supporter le feu et va progressivement en fait baisser l'intensité. On va donc systématiquement retrancher 0.3 d'intensité de feu à toutes les cases dont le combustible est inférieur à 10.
4. Cette étape est la plus simple, il s'agit ici de tracer les matrices afin d'avoir un aperçu de la propagation visuel. Cette étape est soumise à un booléen passé en paramètre car elle est assez lourde en termes de calculs, donc pour des calculs à la chaîne il vaut mieux l'enlever. On va par ailleurs systématiquement enregistrer les images générées afin d'avoir une vidéo exploitable.

III. COMMENTAIRES SUR LA SIMULATION

Une simulation sur 400 générations a été lancée, dont on peut voir les résultats dans le dossier `/simulations`. De plus, il a été créé une vidéo sous le nom `/videos/video.mov`.

Nous pouvons faire quelques observations intéressantes. Premièrement, on remarque que le feu se propage en anneau, avec un front de propagation du feu et un front d'ignition du feu. Au final, on observe que le feu s'éteint effectivement, après avoir consumé l'écrasante majorité de la forêt. C'est là que l'on peut faire l'observation la plus importante de ce projet, c'est que l'on voit bien que la forêt sèche est très vite consumée, se propageant surtout dans ce milieu, alors que la progression est bien plus lente dans la forêt humide. C'est là le but du projet, c'est de modéliser une forêt d'arbres secs entouré d'arbres humides et d'essayer de trouver un optimum dans les rayons de ces zones de sorte à ce que le feu se propage le moins possible. L'objectif pour cela est d'utiliser de l'algorithmique génétique. Nous allons le schématiser de la façon suivante :

- La population considérée est un ensemble de simulations d'environ 200 générations, ce qui est suffisant pour la durée de vie du feu.
- L'initialisation se fera par génération aléatoire d'une population de taille m .
- L'évaluation pourra se faire sur deux critères, soit la surface consumée à la fin de la simulation, soit le nombre de générations nécessaires avant d'avoir une extinction du feu.
- La sélection se fera par conservation de la moitié des simulations les mieux évaluées.
- Le croisement se fera en prenant la moyenne des rayons des cercles considérés pour les meilleures simulations deux à deux.
- La mutation se fera en faisant légèrement varier les rayons d'un quart des simulations choisies aléatoirement.

IV. LA COMPLEXITÉ DU PROGRAMME

On l'a vu, on a commencé par un traitement naïf de notre matrice. Cependant, parcourir de façon aussi naïve peut amener à une surcharge de calculs très importante, sachant qu'avec l'extinction du feu, il est peu probable qu'une très grande partie de la forêt soit à calculer à chaque génération. On a donc mis une condition de calcul, à savoir que le feu soit plus intense que 1. Il a néanmoins été cherché d'avoir une preuve expérimentale de l'efficacité de cette condition. C'est pourquoi l'on mis en place un compteur de « calculs » qui compte le nombre de fois que l'on effectue un calcul avec condition. J'obtenais au début la figure suivante

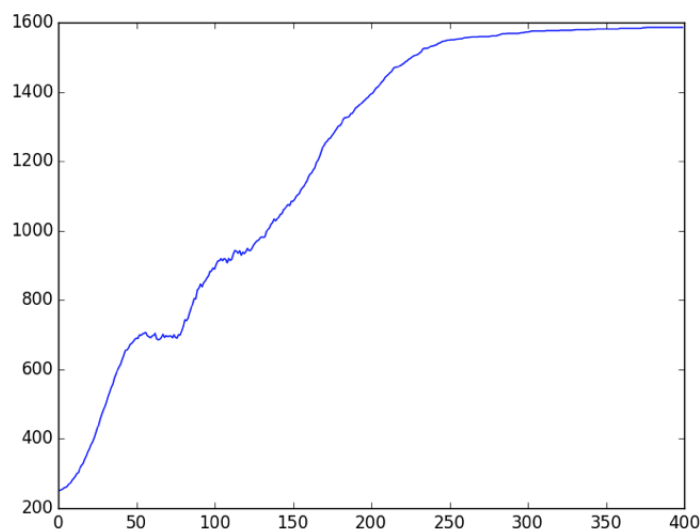


FIGURE 9 – Nombre de calculs en fonction des générations (1)

Qui donne le nombre de calcul en fonction des génération de la simulation. On observe bien une progression du nombre de calculs du à la propagation du feu et donc du nombre de cases à évaluer. Seulement, on observe une asymptote croissante en fin de simulation, alors qu'il n'y a presque plus de feu.

Après recherche, il s'avère que cette croissance était due à l'étape 3, on cherchait à éteindre le feu à la condition que le combustible soit en dessous de 10, ce qui faisait que l'on évaluait toutes les cases consommées, mêmes si elles sont éteintes. Après correction de cette condition, on trouve la courbe suivante :

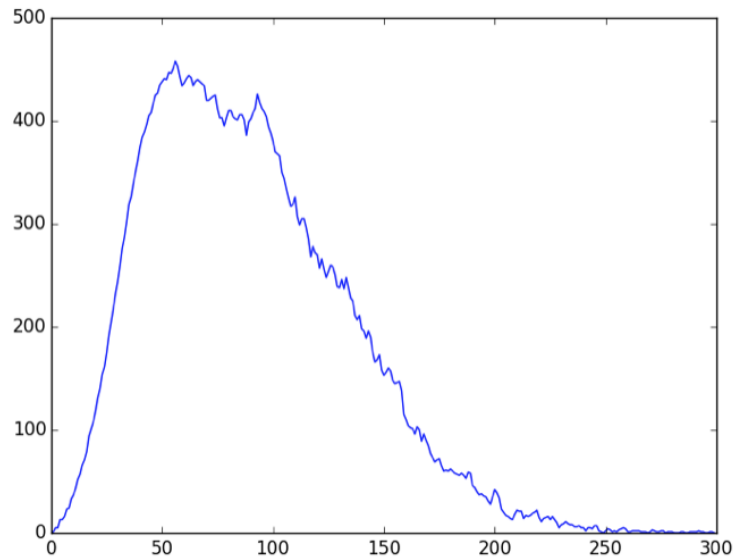


FIGURE 10 – Nombre de calculs en fonction des générations (2)

En faisant un grand nombre de simulations et en faisant la moyenne de ces courbes, on obtient la figure suivante :

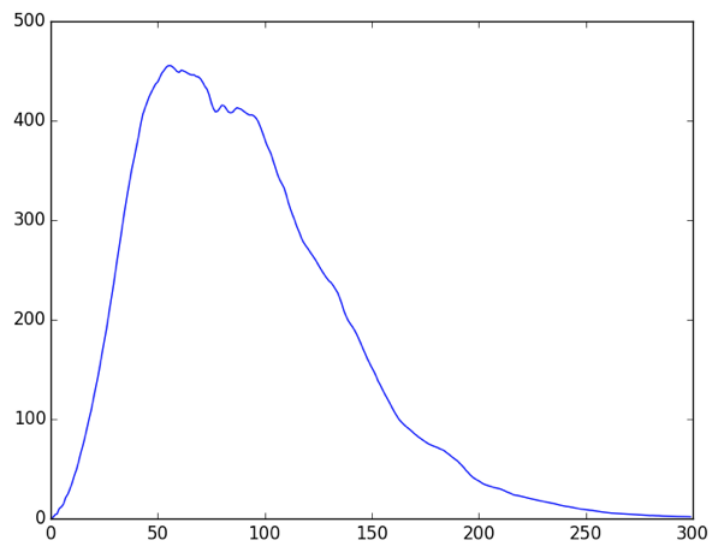


FIGURE 11 – Nombre de calculs en fonction des générations (3)

On a ici quelque chose de grandement satisfaisant. On observe clairement une chute du nombre de calculs moyen, ce qui nous permet d'accélérer grandement la phase « algorithme génétique ». D'ailleurs, en termes de temps, voici ce que l'on obtient en moyenne :

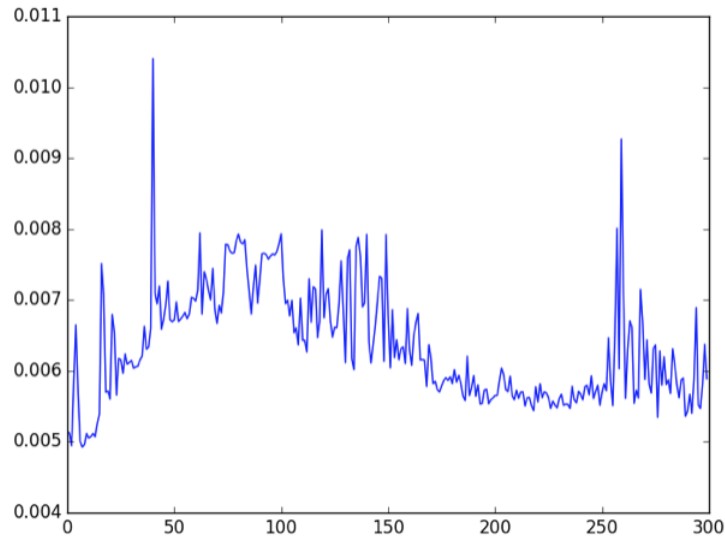


FIGURE 12 – Temps d'exécution en fonction des générations

On observe une chute du temps de génération lorsque le feu décroît, ce qui est encore une piste encourageante.

V. ALGORITHME GÉNÉRIQUE

Malheureusement, notre implémentation d'un algorithme génétique n'est pas bien concluante. En effet, on se rend compte au bout de quelques simulations qu'un optimum ne peut être trouvé, car les hypothèses de travail sont beaucoup trop simplistes. Nous ne disposons pas d'assez de temps pour matériellement pouvoir conclure ce projet, néanmoins je compte le continuer dans le futur, en essayant de considérer une maison que l'on souhaiterait sauver du feu, comment dès lors disposer la forêt autour de cette maison pour atteindre ce but ?