

# APNOLAB

Projet industriel 2020 FCEA Innovation / ENSMSE

## Rapport préliminaire

### Document confidentiel

Etude de l'intérêt de raffinement de la technique de « force brute » pour identifier les solutions optimales à l'aide de techniques d'algorithmes stochastiques de type Métropolis et/ou recuit simulé.

### Table des matières

1	Etat du projet .....	2
1.1	Rappels sur les FCM .....	2
1.2	Fonctionnement du programme .....	2
1.3	Remarques générales .....	3
1.4	Utilisation du programme .....	3
2	Optimisation de la recherche .....	4
2.1	Visualisation des résultats .....	4
2.2	Critique du modèle .....	4

Statut	Document de travail
Version	1.0
Date	01/12/2020
Auteurs	Léo THEODON

Évolutions

--	--	--

# 1 ETAT DU PROJET

## 1.1 RAPPELS SUR LES FCM

Une FCM (Fuzzy Cognitive map) peut être représentée sous forme d'un graphe. L'état d'une FCM composée de  $n$  concepts et à l'instant  $k$  peut quant à lui être représenté à l'aide d'un vecteur d'état  $A(k) = [A_1(k), \dots, A_n(k)]$ .

Le système est initialisé avec un vecteur  $A(0)$  et converge (ou non) vers un état d'équilibre.

La valeur de chaque concept à l'instant  $k + 1$  est calculé à l'aide de la valeur de chaque concept à l'instant précédent. En particulier, on a (inférence de Kosko) :

$$A_i(k + 1) = f \left( \sum_{j \neq i} A_j(k) \times w_{ij} \right)$$

Parfois, on préférera l'expression suivante (inférence de Kosko modifiée) :

$$A_i(k + 1) = f \left( A_i(k) + \sum_{j \neq i} A_j(k) \times w_{ij} \right)$$

Dans les expressions précédentes, la fonction  $f$  est la fonction d'activation ou fonction de seuil, à valeur dans  $[0; 1]$  ou  $[-1; 1]$ . Il peut s'agir d'une fonction de type signum, sigmoïde ou encore tangente hyperbolique par exemple.

Les poids  $w_{ij}$  quant à eux représentent les valeurs qui étiquettent les arrêtes du graphe. Ce sont des valeurs comprises entre 0 et 1 (ou entre  $-1$  et 1). Ainsi,  $w_{ij}$  indique d'influence du concept  $C_i$  de valeur  $A_i$  sur le concept  $C_j$  de valeur  $A_j$ .

**Remarque** — En général, les valeurs  $w_{ij}$  des poids sont normalisées dans l'intervalle  $[0; 1]$  ou  $[-1; 1]$  et représentent la force de la relation entre les concepts  $C_i$  et  $C_j$ .

De plus, par nature même, les valeurs  $A_i$  associées aux concepts  $C_i$  sont normalisées, et forment une représentation fuzzifiée des valeurs réelles. Il convient donc de les défuzzifier une fois la convergence établie.

## 1.2 FONCTIONNEMENT DU PROGRAMME

A l'heure actuelle, le programme consiste en un package Java qui lit deux fichiers de configuration en entrée, recherche une solution optimale en « force brute », et renvoie les paramètres ayant permis d'atteindre cet optimum.

Le premier fichier à être lu est [WildApolab.xml](#) qui contient en particulier toutes les informations nécessaire à la formalisation de la FCM : nœuds, connecteurs, etc... La lecture de ce fichier permet de stocker la FCM en mémoire sous forme d'un objet map.

Le second fichier est [Wild\\_try.xml](#) dont la structure est la suivante :

Les nœuds « concept » indiquent quels sont les acteurs pour lesquels on modifiera les paramètres afin de chercher un optimum. Ils permettent d'indiquer les informations suivantes :

- **rmin** : borne inférieure de l'intervalle de recherche.
- **rmax** : borne supérieure de l'intervalle de recherche.
- **step** : le pas utilisé afin de discrétiser cet intervalle.

```
1 <?xml version="1.0"?>
2 <patient>
3   <concept name="Rain">
4     <rmin>0.0</rmin>
5     <rmax>1.0</rmax>
6     <step>0.025</step>
7   </concept>
8   <concept name="Rangers">
9     <rmin>0.0</rmin>
10    <rmax>1.0</rmax>
11    <step>0.025</step>
12  </concept>
13  <cost name="Herbivores">
14    <weight>-0.25</weight>
15  </cost>
16  <cost name="Predators">
17    <weight>0.1</weight>
18  </cost>
19  <cost name="Grassland">
20    <weight>0.05</weight>
21  </cost>
22 </patient>
```

Les nœuds « cost » sont utilisés afin d'indiquer comment sera calculée la fonction de « coût » que l'on cherche à optimiser. Un nœud « cost » possède un attribut « name » qui correspond au nom d'un nœud de la FCM, et « weight » permet d'indiquer le poids qui sera attribué à l'output de ce même nœud dans le calcul de la fonction de coût.

### **1.3 REMARQUES GENERALES**

La FCM que l'on utilise actuellement est basée sur un exemple fourni par le développeur de la librairie JFCM. Il s'agit d'une simulation intitulée WildlifePark qui, comme son nom l'indique, représente l'évolution de la population d'herbivores et carnivores dans un parc naturel. D'autres facteurs entrent en compte, comme le nombre de gardes forestiers, la quantité de pluie, le nombre de braconniers attirés par les prix des animaux au marché noir, ou encore la quantité de prairie disponible.

### **1.4 UTILISATION DU PROGRAMME**

Après avoir configuré correctement les deux fichiers XML (celui contenant les informations de la FCM n'a, a priori, pas besoin d'être modifié), l'utilisateur n'a plus qu'à lancer le programme afin d'obtenir la liste des valeurs de paramètres permettant d'obtenir la solution optimale.

L'utilisateur peut sélectionner autant de nœuds qu'il souhaite afin d'effectuer la recherche d'optimum, ainsi que pour le calcul de la fonction de coût.

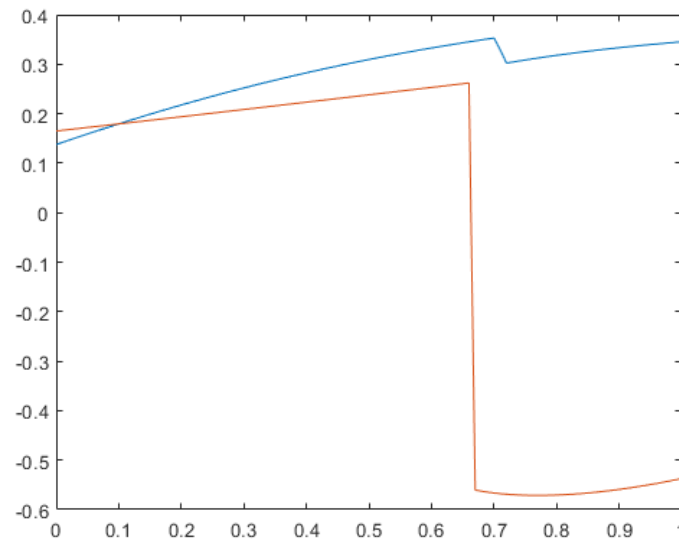
Les résultats sont affichés directement sur la sortie standard.

## 2 OPTIMISATION DE LA RECHERCHE

### 2.1 VISUALISATION DES RESULTATS

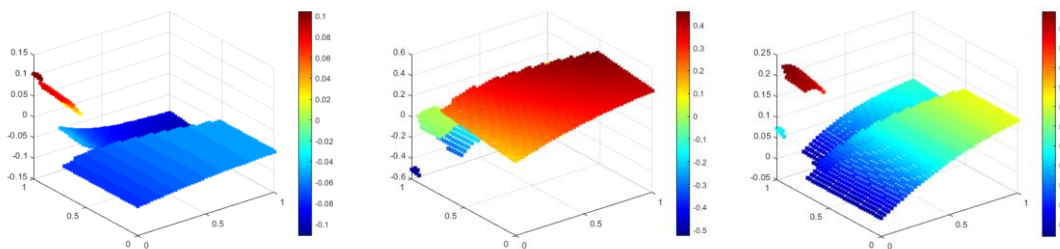
Avant de penser à optimiser la façon dont la recherche de l'optimum est effectué, on peut songer à se donner une idée de l'allure de la fonction de coût.

Pour ce faire, on réalise plusieurs simulations, avec un seul paramètre à optimiser, puis deux.



Le graphique ci-dessus illustre l'allure de la fonction de coût en faisant varier le taux de précipitation (en bleu) et le nombre de gardes forestiers (en rouge).

On peut déjà constater que chaque courbe laisse apparaître de manière évidente des cassures, voir des discontinuités, ce qui semble naturel, étant donné la nature très interdépendante du modèle.



Les trois figures ci-dessus permettent de représenter l'allure de la surface obtenue lors de la représentation de la fonction de coût, en faisant varier le nombre de gardes forestiers et le taux de précipitation. Les trois courbes sont obtenues en choisissant différents paramètres en compte dans le calcul de la fonction de coût, avec des poids différents. Par exemple, la seconde image est obtenue en cherchant à équilibrer le nombre d'herbivores et l'étendue de prairie, alors que la troisième image est obtenue en cherchant à équilibrer le nombre d'herbivores et de carnivores.

### 2.2 CRITIQUE DU MODELE

On peut constater que les courbes obtenues peut être segmentée en plusieurs zones bien distinctes. Les cassures et discontinuités sont évidentes. Néanmoins, au sein de chacune de ces zones, les courbes et surfaces sont extrêmement lisses et régulières.

Le fait est que l'on peut raisonnablement supposer que la FCM qui sera utilisée afin de modéliser notre problème d'apnée du sommeil, et la fonction de coût qui lui sera associée, offriront des profils très différents.

En effet, il semble peu probable que le fait de maximiser ou minimiser l'un des paramètres d'entrée (par exemple les pression maximales ou minimales) puissent offrir des solutions optimales, en particulier dès lors que l'on prend en compte le « bien être » du patient.

Dans le cas présent, les différentes zones et discontinuités peuvent être expliquées simplement par le fait qu'il semble exister des valeurs seuils au-delà desquels les populations d'herbivores ou de carnivores s'effondrent par exemple.

Par conséquent, il serait intéressant de voir si dans notre cas particulier, ces discontinuités existeraient toujours. En effet, si d'aventure la surface à optimiser est extrêmement régulière, un algorithme de descente de gradient pourrait se révéler des plus efficaces. En revanche, si l'on observe de grande discontinuités et variations, il semble évident qu'un algorithme d'optimisation globale ou mixte serait plus adapté.

C'est dans cette optique que l'on peut recommander de se pencher sur l'implémentation d'un algorithme stochastique de type recuit simulé, dont on sait qu'il permet en moyenne de converger vers un optimum global, s'il existe.