

Sébastien Blin
Victor Drouin Viallard



Université du Québec à Chicoutimi

8INF844- TP1

Simulation multi-agents avec NetLogo

Sous l'encadrement de Abdenour Bouzouane

10 février 2017

1 Synthèse de l'article

1.1 Introduction

L'article *Flocks, Herds, and Schools : A Distributed Behavioral Model* (Craig W. Reynolds) décrit comment simuler des flocks en considérant chaque individu à part, sans avoir à scripter le mouvement de chaque individu, mais plutôt en fonction de l'environnement perçu par ceux-ci. Le flock sera ainsi le résultat du mouvement de chaque individu. À première vue, la simulation d'un flock peut paraître difficile, car il y a beaucoup d'individus à simuler et très dur à maintenir (éviter les collisions entre toutes les particules à chaque frame par exemple), mais ce papier propose une méthode pour réussir à simuler ce mouvement sans avoir à réaliser la gestion des collisions. Ainsi, Le flock sera le résultat du mouvement d'un seul individu appliqué à tous les oiseaux.

On considère ici le flock comme un ensemble de particules améliorées (*boid*). Chaque particule représente un individu (ici un oiseau) et possède des variables (couleur, localisation, vitesse, etc) auxquelles on donne une forme (qui est plus significative visuellement et ajoute la notion d'orientation) ainsi qu'une vitesse maximale.

1.2 Fonctionnement de l'algorithme

Les observations montrent que certains animaux se regroupent en flock afin de se protéger des prédateurs, chercher de la nourriture, etc. Pour participer à un flock, un oiseau doit pouvoir se coordonner avec les oiseaux qui l'entourent. Vu qu'un flock est possiblement infini en taille (certains bancs de poissons ont des millions d'individus), on suppose que l'animal ne considère que lui-même, quelques uns de ses voisins et l'ensemble du flock. On peut alors considérer trois forces :

- *Collision Avoidance*, pour éviter de rentrer en collision avec les autres oiseaux.
- *Velocity Matching* (heading + speed => vecteur direction), pour rester à la vitesse des oiseaux voisins.
- *Flock Centering*, pour rester proche des oiseaux voisins et aller vers le centre du flock car y a la même densité partout autour de lui.

On observe donc deux forces opposées : l'une qui pousse l'oiseau à rester dans le flock, l'autre qui le pousse à éviter les autres oiseaux ; il faut y prêter attention car cela peut amener l'oiseau à ne pas savoir où se déplacer. Les trois forces agissent indépendamment et propose une solution différente au choix de la direction à prendre. On peut alors décider d'atténuer plus ou moins telle ou telle force et les combiner pour obtenir le mouvement résultant des 3 forces, etc. Si moyenner ces trois forces est la solution la plus simple, Il semble plus intéressant de les prioriser puis de les accumuler dans l'ordre jusqu'à atteindre une magnitude totale maximale, pour éviter les problèmes d'opposition évoqués précédemment. Par exemple, si on regarde en priorité la force de *Collision Avoidance* et qu'elle est de magnitude plus forte que celle maximale alors on

suivra directement cette force pour éviter en priorité la collision et les autres forces seront temporairement écartées.

1.3 Améliorations possibles

On peut enfin ajouter des fonctionnalités comme l'initialisation de la position des obstacles, des oiseaux ou encore leur donner un but de migration au fur et à mesure. La plus intéressante des fonctionnalités est l'évitement d'objets. Pour la réaliser, deux solutions sont envisageables. La première, *force field concept*, consiste à avoir une force de répulsion autour de l'objet à éviter, mais si l'oiseau arrive pile en face d'une ligne du champ, il ne sera pas repoussé mais ralenti jusqu'à se bloquer. De plus, si l'oiseau vole à côté d'un mur il n'a pas besoin de tourner même s'il est dans le champ, ce qui nous amène à la seconde, *steer-to-avoid*, où l'oiseau ne considère que les obstacles face à lui. S'il en trouve un, il calcule alors un vecteur afin d'éviter l'objet.

1.4 Conclusion

On peut trouver plusieurs applications concrètes à la simulation d'un flocking, par exemple dans la simulation de trafics routiers ou encore pour l'observation d'animaux. Enfin, il est possible d'améliorer l'algorithme en le parallélisant ou en étant sûr que N soit assez petit (en considérant seulement une partie du flock par exemple).

2 Simulation avec NetLogo