INTELLIGENCE ARTIFICIELLE

Artificial Bee Colony Algorithm



Julien ARNOLD - Adrien COUCHOT – Daniel WETTEL

Table des matières

[Introduction 3](#_Toc470871949)

[Organisation 3](#_Toc470871950)

[Description des classes 4](#_Toc470871951)

[MyAlgorithm 4](#_Toc470871952)

[Problem 4](#_Toc470871953)

[SetUpParams 4](#_Toc470871954)

[Solution 4](#_Toc470871955)

[Les fonctions 5](#_Toc470871956)

[Problèmes rencontrés 5](#_Toc470871957)

[Démonstration 6](#_Toc470871958)

[Conclusions 7](#_Toc470871959)

# Introduction

L’algorithme ABC (Artificial Bee Colony) est une technique d’optimisation qui simule le comportement d’une colonie d’abeille pour résoudre différents problèmes.

Il y a 3 types d’abeilles : les employées, les spectatrices et les éclaireuses.

Les employées cherchent de la nourriture dans les alentours de la source de nourriture qu’elles ont en mémoire et partagent ces informations avec les spectatrices.

Les spectatrices essayent ensuite de choisir les meilleures sources parmi celles que les employées leur ont transmises.

Les éclaireuses sont d’anciennes employées dont la source a été abandonné et qui en cherche donc une nouvelle.

## Organisation

|  |  |  |  |
| --- | --- | --- | --- |
| Début | Fin | Intitulé | Participants |
| 18/10 | 13/11 | Prise d'informations | Tous |
| 14/11 | 14/11 | Mise en place planning | Tous |
| 14/11 | 08/12 | Développement des classes |  |
|  | 08/12 | MyAlgorithm | Adrien + Julien |
|  | 24/11 | Solution | Daniel |
|  | 24/11 | SetParams, Problem | Julien |
| 24/11 | 24/11 | Interface graphique | Daniel |
| 08/12 | 15/12 | Nettoyage de classe et Optimisation 1 |  |
|  | 15/12 | MyAlgorithm | Adrien |
| 15/12 | 28/12 | Nettoyage de classe et Optimisation 2 |  |
|  | 28/12 | MyAlgorithm | Adrien + Julien |
| 22/12 | 30/12 | Edition du rapport | Tous |
| 25/12 | 30/12 | Interface graphique 2 + Sortie Texte | Daniel |
| 30/12 | 30/12 | Optimisation 3 | Julien + Adrien |

# Description des classes

Le projet est constitué des 4 classes principales imposées dans le template fournit, d’un fichier contenant l’implémentation des différents benchmarks et d’une interface graphique réalisé sous Visual Studio.

## MyAlgorithm

C’est la classe principale dans laquelle s’exécute l’algorithme avec les paramètres fixés par les classes Problem et SetUpParams en utilisant la classe Solution.

- evolution() : C’est la boucle principale, NbRun indépendants sont exécutés où on effectue NbEvolution fois : sendEmployedBees(), SendOnLookerBees() et sendScoutBees().

- initialize() : Fait que les NbRun soient indépendant.

- sendEmployedBees() : utilise BeesWork() avec un paramètre aléatoire.

- SendOnLookerBees() : utilise BeesWork() avec un paramètre choisi par CalculateProbabilities().

- CalculateProbabilities() : choisit un paramètre en fonction de la valeur de la source et sa proximité.

- BeesWork() : effectue une mutation en fonction du paramètre reçu.

- sendScoutBees() : permet d’éviter de rester bloquer dans des minimums locaux.

## Problem

La classe Problem contient les limites, la dimension, et la fonction de benchmark qui définissent le problème à traiter.

Des getters, setters et surcharges d’opérateurs sont présents dans cette classe.

## SetUpParams

La classe SetUpParam contient les paramètres relatifs à l’exécution de l’algorithme (nombre d’exécution, nombre d’évolution par exécution, taille de la population, taille de la solution, nombre d’essaies max) ainsi que des getters pour accéder à ces données.

## Solution

La classe Solution contient une instance de la classe Problem et un vecteur de double représentant les solutions. Elle s’occupe notamment d’initialiser les données et de calculer le fitness.

- initialize() : initialise les source de nourriture.

- CalculateFitness() : calcule et renvoie le fitness

- Divers getters et setters pour chaque pour chaque propriété ainsi que des surcharges d’opérateurs pour faciliter la programmation.

# Les fonctions

|  |  |  |
| --- | --- | --- |
| Nom | Fonction | Meilleur résultat obtenu |
| Ackley |  | 1.03\*e-013 |
| Rastrigin |  | 1.18\*e-011 |
| Rosenbrock |  | 0.0466 |
| Schaffer |  | 1.92\*e-010 |
| Schwefel |  | -29.81607472 |
| Weierstrass |  | -55.376 |

Les meilleurs résultats ont été obtenus avec 1 exécution et 100 000 évolutions.

# Problèmes rencontrés

La plus grosse difficulté a été d’adapté l’algorithme en C à notre structure, de plus, l’algorithme en C obtient des résultats bien meilleur en un temps bien plus court.

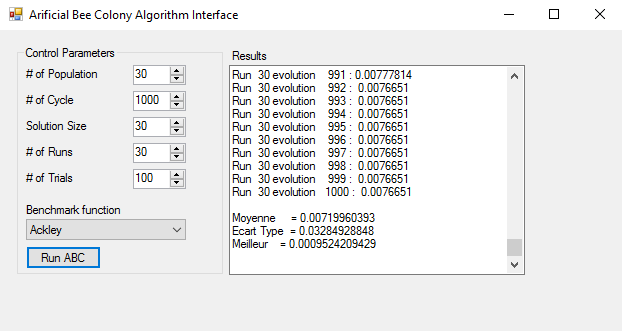
Nous avions essayé de trier le tableau des fitness afin de réduire le temps des calculs mais cet essai c’est avéré infructueux car il influait de façon non négligeable les probabilités qu’une solution soit choisie par rapport à une autre.

Nous avons réussi à beaucoup améliorer notre algorithme en optimisant nos fonctions, modifiant certaines mais les résultats ne sont pas encore ceux attendus.

# Démonstration

L’interface graphique nous invite à rentrer les paramètres du problème, à choisir la fonction de benchmark utilisé (ici Ackley) et à cliquer sur le bouton « Run ABC ».

Les résultats de chaque exécution sont ensuite affichés dans la textebox adjacente après avoir été calculés avec à la fin la moyenne, l’écart type et le meilleur fitness.



# Conclusions

## Adrien

Ce projet a été une toute nouvelle expérience pour moi, ça a été une manière intéressante et instructive de découvrir l’IA. Malgré les difficultés rencontrées, je trouve que notre groupe s’en est très bien sorti, nous avons su progresser et résoudre les problèmes. Nous avons fait beaucoup de recherches avant d’attaquer le problème et elles nous ont été d’une grande aide.

La plus grosse difficulté rencontrée, selon moi, a été d’implémenter le code dans une structure prédéfinie qui est très différente de celle de l’algorithme en C que nous avons récupéré.

## Daniel

Pour moi, le projet aura été intéressant bien que compliqué à comprendre à cause des nombreuses formules mathématiques nécessaires au bon fonctionnement de l’algorithme. Le travail en groupe c’est très bien passé et je pense que nous avons globalement réussi à accomplir les tâches demandées.

Cependant, si ce projet était à refaire, je pense que l’on préparerait d’abord tous les outils nécessaires au développement du projet (je pense notamment à l’interface) car changer d’IDE et de structure en fin de projet constitue une perte de temps évitable.

## Julien

L’intelligence artificielle est une technologie qui prend de plus en plus d’ampleur de nos jours, j’ai toujours trouvé cela très intéressant mais je ne m’étais jamais intéressé à son fonctionnement. Ce projet m’a permis de le comprendre et de l’implémenter à l’aide de fonctions existantes et de l’entraide du groupe. Cette expérience a été très enrichissante et m’a donné envie de progresser dans ce domaine en implémentant d’autres fonctions, par exemple les réseaux de neurones artificiels.

Malgré la difficulté de ce projet, les outils mis à notre disposition nous ont permis de réaliser un ensemble correct avec des résultats satisfaisants.