INTELLIGENCE ARTIFICIELLE

Artificial Bee Colony Algorithm



Julien ARNOLD - Adrien COUCHOT – Daniel WETTEL

Table des matières

[Introduction 3](#_Toc470871949)

[Organisation 3](#_Toc470871950)

[Description des classes 3](#_Toc470871951)

[MyAlgorithm 3](#_Toc470871952)

[Problem 3](#_Toc470871953)

[SetUpParams 3](#_Toc470871954)

[Solution 3](#_Toc470871955)

[Les fonctions 3](#_Toc470871956)

[Problèmes rencontrés 3](#_Toc470871957)

[Démonstration 3](#_Toc470871958)

[Conclusions 3](#_Toc470871959)

# Introduction

Artificial Bee Colony Algorithm : L’optimisation de recherche de source pour une colonie d’abeille.

## 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 |

# Description des classes

## 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

…

## SetUpParams

…

## Solution

…

# Les fonctions

|  |  |  |
| --- | --- | --- |
| Nom | Fonction | Meilleur résultat obtenu |
| Ackley |  |  |
| Rastrigin |  |  |
| Rosenbrock |  |  |
| Schaffer |  |  |
| Schwefel |  |  |
| Weierstrass |  |  |

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

# Démonstration

Voir annexes.

# 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 recherche 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