**Rapport projet LO21**

Léo Angonnet – Hugo Allainé

Sommaire

[Description des choix de conception et d’implémentation 2](#_Toc123309511)

[Algorithmes des sous-programmes 3](#_Toc123309512)

[Sous-programme « individu.c » 3](#_Toc123309513)

[Sous-programme « population.c » 4](#_Toc123309526)

[Jeux d’essais 5](#_Toc123309527)

[Commentaires sur les résultats 6](#_Toc123309528)

# Description des choix de conception et d’implémentation

# Algorithmes des sous-programmes

## Sous-programme « individu.c »

Importation du fichier « individu.h »

Définition de la fonction initRandomIter :

Création du premier élément de la liste

Donne une valeur aléatoire entre 0 et 1 au premier élément

Créé un pointeur temporaire sur le premier élément

Boucle longIndiv de fois :

Début

Créé un nouvel élément

Donne une valeur aléatoire entre 0 et 1 au nouvel élément

Fixe le prochain élément à NULL

Fixe l'élément suivant de l'élément précédent au nouvel élément

Place le pointeur temporaire sur le nouvel élément

Fin boucle

Renvoie le premier élément de la liste

Définition de la fonction initRandomRecIn, arguments (type individu, entier i (compteur)):

Si le compteur est égal à longIndiv

Renvoie la liste

Sinon

Création d'un nouvel élément

Donne une valeur aléatoire entre 0 et 1 au nouvel élément

Fixe le prochain élément à NULL

Fixe l'élément suivant de l'élément précédent au nouvel élément

Renvoie le résultat de la fonction avec le nouvel élément et le compteur incrémenté de 1 en arguments

Définition de la fonction initRandomRec :

Création du premier élément de la liste

Donne une valeur aléatoire entre 0 et 1 au premier élément

Création d'un compteur initialisé à 1

Renvoie le résultat de la fonction initRandomRecIn avec le premier élément et le compteur en arguments

Définition de la fonction convertIndivToInt, argument(type individu):

Création d'une variable pour stocker la valeur

Création d'un compteur égal à longIndiv-1

Boucle tant que l'élément de la liste n'est pas NULL

Décrémentation du compteur

Passe le bit de la base 2 à la base 10

Passage à l'élément suivant de la liste

Renvoie la valeur

# Définition de la fonction crossTwoLists, arguments(type individu,type individu) :

# Création d'un pointeur temp1 temporaire sur la première liste

# Création d'un pointeur temp2 temporaire sur la deuxième liste

# Initialisation de deux entier, rand1 et rand2

# Boucle tant que temp1 et temp2 ne sont pas nul

# Si un nombre aléatoire entre 0 et 100 est inférieur à pCross fois 100

# Un entier temp est égal à la valeur de temp1

# la valeur de temp1 est égale à la valeur de temp2

# la valeur de temp2 est égale à temp

# temp1 est égal à temp1 suivant

# temp2 est égal à temp2 suivant

# Fin

Définition de la fonction quality, argument(entier value) :

Un flottant X égal à

Un flottant quality égal à

Renvoie quality

## Sous-programme « population.c »

Définition de la fonction initPopulation, argument(entier taillePop) :

Création du premier élément de la liste

Création d'un Individu aléatoire pour le premier élément

Fixation de la qualité du premier élément en utilisant la fonction

de qualité et la conversion de l'Individu en entier

Fixation de l'élément précédent du premier élément à nul

Création d'un pointeur temp temporaire sur le premier élément

Boucle taillePop de fois :

Création d'un nouvel élément

Création d'un Individu aléatoire pour le nouvel élément

Fixation de la qualité du nouvel élément en utilisant la fonction de qualité et la conversion de l'Individu en entier

Fixation du prochain élément du nouvel élément à NULL

Fixation de l'élément précédent à temp

Fixation du l'élément suivant de temp au nouvel élément

Fixation de temp à son suivant

renvoie élément

# Jeux d’essais

# Commentaires sur les résultats