

Maîtrise Sciences Co : 4ème TD

1 Construction et initialisation d'un objet `int** population`

- Définition d'un objet `int** population` dans le `main.c`.
- Allocation `int* initPopulation()` et désallocation `void endPopulation(int** pop)` dynamique (`malloc` et `free`) de la population dans `population.c`.
- Initialisation des valeurs des gènes de chaque chromosome de la population grâce à la méthode `int* initChromosome()` dans `chromosome.c`

2 Opérateur d'évaluation de la population

- En utilisant la méthode `int evalChromosome(int*)`, définir une méthode `int* evalPopulation(int** population)`, qui retourne un tableau d'entiers correspondant au fitness de chaque chromosome dans la population.
- Définir `int* tabFitness` dans le `main`, qui récupère les valeurs calculées dans `evalPopulation`.

3 Opérateur de sélection et de création de la génération suivante

- Définition d'une méthode `int** selectPopulation(int** population, int* tabFitness)` dans `population.c`.
- L'allocation pour la nouvelle population doit se faire à ce niveau.
- Sélection par tournoi : on tire aléatoirement `NB_MAX_OPPONENTS` entiers (chaque entier compris entre 0 et `NB_MAX_CHROMOSOMES`), correspondant aux chromosomes sélectionnés pour participer au tournoi.
- On ne garde qu'un chromosome, celui ayant le fitness le plus grand, le vainqueur du tournoi. On doit recopier explicitement le chromosome vainqueur dans la nouvelle population.
- On recommence cette procédure `NB_MAX_CHROMOSOMES` fois afin de constituer la nouvelle population.