

IN104 - Projet informatique

Algorithmes génétiques

Vladimir Paun

U2IS - ENSTA ParisTech

Détails livrable

12 mai 2015

Table des matières

1	Format livrable	3
1.1	Exemple numérique du fichier d'entrée	3
1.1.1	Remarque	3
1.2	La fourniture de l'exécutable	3
1.3	La fourniture du code source	5
1.4	Le rapport	5
1.4.1	Remarque qualitative	5
2	Evaluation	6

1 Format livrable

Le livrable du Projet IN104 - Algorithmes génétiques attendu est constitué de deux parties :

1. le projet informatique même sous la forme d'un exécutable ;
2. le code source ;
3. le rapport.

Chaque livrable devrait pouvoir accepter un fichier d'entrée qui comporte l'ensemble initial qui correspond à l'ensemble d'entiers pour lequel il faut trouver le plus grand sous-ensemble de somme nulle. Pour faciliter l'évaluation des projets et les tests, ce format est standardisé :

- le cardinal de l'ensemble E , sur la première ligne ;
- les éléments $e_i \in E$ sur la deuxième ligne, séparés par des virgules (*comma separated values*).

1.1 Exemple numérique du fichier d'entrée

```
1001
313,13579,-3597,...,-13
```

1.1.1 Remarque

Dans le traitement de ce fichier d'entrée vous ne devriez pas supposer que l'utilisateur fournit une entrée correcte (voir la partie suivante).

1.2 La fourniture de l'exécutable

Le binaire du Projet doit compiler, il faut donc privilégier une fourniture *propre* suffisamment testé pour compiler dans tous les cas sans erreur d'exécution. Si certaines parties/fonctions ne sont pas abouties, il faut plutôt les mettre en commentaire et spécifier leur fonctionnement ainsi que les difficultés rencontrées dans le rapport.

L'exécutable devrait respecter les conditions suivantes :

- compiler sans erreur ;
- contrôler le format du fichier d'entrée pour éviter les erreurs (*programming defensive*, ne jamais supposer que l'utilisateur de votre programme donnera les entrées correctes en ce qui concerne la taille, le type, etc.) ;
- proposer un menu visuel (console ou graphique) à l'utilisateur ;
- affichage pertinent qui comporte :
 - temps d'exécution ;
 - nombre d'itérations ;
 - *vitesse de convergence* ;
 - meilleur individu à tout moment ou à la demande (penser à un fichier de configuration pour certains éléments) ;
 - affichage de l'individu (pas en binaire) ;
 - affichage de la population pour voir l'évolution (un graphique dynamique serait idéal).

Le menu devrait proposer un maximum des options suivantes :

- changer le temps de l'exécution de l'algorithme ;
- relancer l'algorithme sans quitter le programme ;
- génération de ensemble initial :
 - aléatoire qui change à chaque exécution :
 - spécifier la taille de l'ensemble ;
 - spécifier les limites de l'intervalle d'entiers relatifs de l'ensemble ;
 - paramétrer l'algorithme (taille population, taille individu, méthode de sélection privilégiée, etc.) → penser à utiliser un fichier de configuration ;
 - aléatoire qui ne change pas à chaque exécution ;
 - lecture d'un fichier standardisé ;
 - spécifier le nom du fichier d'entrée standardisé ;
 - spécifier le chemin du fichier standardisé ;

1.3 La fourniture du code source

Dans l'évaluation de la qualité du code source nous allons tenir compte de facteurs suivants :

- efficacité / optimisation (justifications avec le calcul de complexité temporelle et spatiale) ;
- lisibilité du code (choix de noms de variables) ;
- organisation du code (partage en bibliothèques, séparation fonctionnelle) ;
- tests de chaque fonction - la pertinence des tests est cruciale (éviter les tests de type `print("Résultat f(x) = " + f(x))`).

1.4 Le rapport

Le rapport doit au moins être constitué des éléments suivants :

- présentation d'une vue globale du projet qui devrait relever la compréhension du projet, l'acquisition d'une démarche projet, etc. ;
- partage détaillé des tâches au sein du binôme ;
- explication de l'implémentation :
 - justification du choix des structures de données ;
 - calcul de complexité (temporelle et mémoire) ;
 - analyse critique (points forts et faibles par exemple ;
- explications concernant la manière de compiler les codes sources et exécuter le programme (paramètres, fichiers d'entrée et de sortie, commandes, etc.) ;
- captures d'écran de l'utilisation et des résultats ;
- présentation et interprétation des résultats (graphiques, etc.).

1.4.1 Remarque qualitative

Le rapport devrait être écrit avec la supposition que le code binaire ne sera jamais lu (sachant qu'il serait lu quand même, mais il ferait tout simplement l'objet d'une évaluation, et donc d'une note, distincte qui va contribuer à la note finale). Il faudrait donc vous démarquer des autres projets.

2 Evaluation

Une version minimale du projet, accompagné par son rapport, seront nécessaire pour valider le cours projet. Cette version comporte un algorithme principal qui fait intervenir tous les étapes d'un algorithme génétique formalisés dans le document *Sujet détaillé*.

La version améliorée doit comporter plusieurs version de sélection, croisement, fonction de fitness, etc. qui pourront éventuellement changer pendant une même exécution.