IN104 - Projet informatique

Introduction aux algorithmes génétiques

1 Présentation du projet

1.1 Introduction

La sélection naturelle est un concept ayant prouvé son efficacité dans de nombreux domaines et bien au delà de la théorie de l'évolution en biologie. Les concepts de l'évolution et de l'hérédité peuvent être utilisés dans le cadre de nombreux problèmes d'optimisation et ont permis l'émergence d'une approche connaissant de plus en plus de succès dans le domaine de l'optimisation «ămulticritère », les algorithmes génétiques. Dans ce projet, nous nous proposons d'implanter une version générique des algorithmes génétiques et d'en dériver deux version permettant de résoudre des problèmes classiques d'optimisation.

1.2 Présentation du problème d'optimisation

Un problème typique d'optimisation s'exprime ainsi :

Maximise

$$f(x_1, x_2, ..x_k)$$

En respectant l'ensemble des contraintes suivantes :

$$\begin{array}{lcl} h_1(x_1, x_2, \ldots, x_k) & \leq & 0 \\ h_1(x_1, x_2, \ldots, x_k) & \leq & 0 \\ \vdots & & \\ h_m(x_1, x_2, \ldots, x_k) & \leq & 0 \\ h_{m+1}(x_1, x_2, \ldots, x_k) & = & 0 \\ \vdots & & \\ h_{m+n}(x_1, x_2, ..x_k) & = & 0 \end{array}$$

L'expression $f(x_1, \ldots, x_k)$ est appelée la fonction objectif. La fonction objectif peut-être n'importe quelle fonction qui calcule quelques dépendant de x. Il peut aussi bien s'agir d'une fonction calculant un taux de rendement interne ou alors le nombre de connexions entres les différents composants d'un circuit. L'ensemble des contraintes peut inclure des inégalités aussi bien que des égalités. De plus, les variables de ces égalités ou inégalités peuvent être des variables discrètes ou continues. Pour résoudre ce type de problèmes, il est possible d'utiliser différentes techniques pouvant être classé grosso modo en trois catégoriesă :

Les techniques de programmation linéaire ou non-linéaire Ces algorithmes sont basés sur des propriétés mathématiques des fonctions optimisées. Si ces algorithmes sont relativement efficaces, leur domaine d'utilisation est assez étroit.

Les heuristiques de recherche Ces algorithmes sont désormais facilement exploitables étant donné l'explosion des performances des ordinateurs et de la mémoire disponible. Ces approches sont très efficaces pour explorer les optima locaux de la fonction objectif et peuvent être facilement applicable à des grandes classes de problèmes.

Les algorithmes génétiques Ces algorithmes reprennent deux concepts de la théorie de l'évolution : la définition d'un processus de sélection ainsi qu'un recours à des pseudo-mutations aléatoires permettant d'obtenir un nouvel ensemble de solutions à partir de solutions précédemment envisagés. Par rapport aux heuristiques de recherche, cette approche permet d'une part des recherches sur des grands domaines de définition, incluant de facto l'étude de plusieurs optima locaux, d'autre part cette approche ne nécessite que des adaptations mineures à chacune des classes de problèmes.

1.2.1 Fonctionnement d'un algorithme génétique

Avant de pouvoir utiliser un algorithme générique, il est nécessaire de pouvoir définir pour le problème donnéã :

- Une méthode permettant de représenter la solution sous une forme manipulable par la machine, la plupart du temps, il s'agit d'un vecteur de valeurs booléennes.
- Une fonction permettant de calculer la qualité d'une solution.

Initialisation Un ensemble de plusieurs solutions est engendré de manière aléatoire. Cet ensemble constitue ce qui est appelé la population initiale. La taille de la population initiale dépend de la nature du problème et surtout de l'existence de nombreux optima locaux. Typiquement, les populations peuvent contenir plusieurs milliers de solution possible. Habituellement, la population initiale est engendrée de manière aléatoire afin de couvrir complètement le domaine des solutions possibles.

Sélection A une époque donnée, un échantillon de la population existante est sélectionné afin d'engendre la nouvelle génération. La sélection s'effectue selon un processus de sélection des solutions ayant la meilleure adéquation. Plusieurs fonction de sélection sont possibles, la plus simple consisté à sélectionner les solutions ayant la meilleur adéquation, d'autres fonctions stochastiques de sélections sont moins sélectives et ont pour but de préserver des solutions ayant une « médiocre adéquation » afin d'éviter une convergence vers un optimum local.

Reproduction L'étape suivante consiste à construire un nouvel ensemble de solutions à partie de l'ensemble des solutions venant d'être sélectionné. Des nouvelles solutions peuvent être engendrées à partir de solutions courantes par deux techniques, soit par le croisement d'une paire de solutions, soit par la mutation de deux solutions.

Croisement Une paire de solutions parmi les solutions précédemment est sélectionnée. Chacune des solutions est en fait un vecteur de bits $({}^1b_1, \ldots, {}^1b_n)$ et $({}^2b_1, \ldots, {}^2b_n)$. Nous choisissons au hasard un point de coupure k tel que $k \geq 1$ et $k < n$. Un premier descendant des solutions $({}^1b_1, \ldots, {}^1b_n)$ et $({}^2b_1, \ldots, {}^2b_n)$ est construit en concaténant aux k premiers bits de la première solutions les $k-1$ derniers bits de la seconde solution. De même un deuxième descendant en concaténant aux k premiers bits de la première solutions les $k-1$ derniers bits de la seconde solution. Au final la paire de solutions s'écrit comme suit :

$$({}^1b_1, \ldots, {}^1b_k, {}^2b_{k+1}, \ldots, {}^2b_n) \text{ et } ({}^2b_1, \ldots, {}^2b_k, {}^1b_{k+1}, \ldots, {}^1b_n)$$

Mutation La mutation standard dans le cadre d'un algorithme génétique consiste en la probabilité qu'un bit appartenant à une séquence de bits représentant une solution possible puisse changer d'état, c'est-à-dire passer de zéro à un ou passer de un à zéro. La méthode standard consiste à générer une variable aléatoire pour chacun des bits dans une séquence. Cette variable aléatoire va permettre de déterminer si le bit doit être modifier ou non.

Terminaison Le processus générationnel continue tant qu'une condition de terminaison n'est pas atteinte. Les conditions de terminaison sont plus diverses et parmi les conditions de terminaisons les plus courantes nous pouvons citerǎ :

- Une solution vérifie un critère d'adéquation
- Le nombre maximal d'itérations est atteint
- Le temps d'exécution maximal est atteint
- La qualité des meilleures solutions a atteint un plateau et les itérations successives ne semblent plus produire de meilleurs résultats.

Bien entendu, ces conditions de terminaisons peuvent être combinées entre elles.

Description de l'algorithme en pseudo-code

```
Sélection la population initiale \\
Repeat \\
  Sélectionne une partie de la population en fonction de leur adéquation
  Sélectionne les paires des meilleures solutions pour construire de \\
  nouvelles solutions. \\
  Produit une nouvelle génération en combinant les paires des meilleures \\
  solutions et en introduisant des mutations. \\
Until la condition de terminaison est vérifiée
```

1.3 Un petit problème NP-complet

Un problème intéressant en théorie de la complexité mais aussi en cryptographie est le suivant : considérons un ensemble d'entiers relatifs, est-ce que la somme des éléments d'un sous-ensemble de cet ensemble a pour valeur 0. Par exemple, pour l'ensemble $7, 3, 2, 5, 8$, la réponse au problème est «oui» puisqu'il existe la somme des éléments du sous-ensemble $3, 2, 5$ est égale à 0. De plus, il est souhaitable de trouver le ou les sous-ensembles ayant le plus grand nombre d'éléments. Les algorithmes génétiques permettent de trouver de manière relativement simple les solutions à ce type de problèmes.

1.4 Objet du projet

Le projet a pour but d'implanter un algorithme génétique et de l'utiliser pour déterminer les plus grands sous-ensemble d'un ensemble d'entiers relatifs tel que la somme des éléments de ce sous-ensemble est 0.