USTHB, FEI,
Département d'Informatique
Master SII, S2

Bab-Ezzouar le 13 Juin 2012

Epreuve sur les méta-heuristiques
et les algorithmes évolutionnaires

Le problème de satisfiabilité noté SAT en abrégé, occupe une place centrale en intelligence artificielle et en théorie de la complexité. C'est le premier problème qui a été démontré NP-complet et est donc considéré comme l'ancêtre de tous les problèmes NP-complets. Il est par conséquent intraitable lorsque sa taille est importante. Il peut être défini par la paire (instance, question) comme suit :
Instance : une formule booléenne sous la forme normale conjonctive CNF
Question : existe-t-il une instanciation des variables pour que la formule soit vraie ?
L'exemple suivant permet d'illustrer une instance de SAT :
Soient X = {x1, x2, x3} un ensemble de 3 variables booléennes (n dans le cas général) et C = {c1, c2, c3, c4} un ensemble de 4 clauses (k dans le cas général), une clause étant une disjonction de littéraux et un littéral une variable booléenne avec ou sans négation.
c1 = x1
c2 = -x1 + x2
c3 = -x1 + x3
c4 = -x2 + -x3
L'opérateur + représente l'opération booléenne de disjonction et l'opérateur - celle de la négation.

1) Comment représenter une solution du problème ? (3pts)

La solution est un vecteur de n valeurs booléennes, chacune affectée à une variable. Elle peut être représentée par une chaine binaire.

2) Donner une borne supérieure du nombre maximum de ces solutions ? Que peut-on en conclure ? (3pts)

chaque variable peut prendre la valeur 0 ou 1, donc 2 valeurs possibles. Au total le nombre de solutions possibles est égal à 2^n. Nous concluons qu'en pratique les algorithmes exacts et notamment la recherche exhaustive des solutions, ne peuvent pas résoudre efficacement le problème.

3) Nous nous intéressons au développement d'un algorithme génétique pour résoudre le problème SAT.
   - Proposer une fonction d'évaluation d'un individu ou solution (2pts)

Une solution qui résout une instance de SAT, satisfait les clauses de cette dernière. Par conséquent une fonction d'évaluation peut être :
- maximiser le nombre des clauses satisfaites
- minimiser le nombre de clauses insatisfaites

   - Fournir les détails de l'initialisation de l'algorithme. Expliquer clairement la création de la population initiale. Donner les instructions correspondantes. (4pts)

Initialisation :
pop-size = 100 ;

i = 1 ;
/* créer la population initiale
Pour j = 1 à pop-size faire

- s[j] = générer aléatoirement un tableau de n valeurs binaires ;
- f(s[j]) = nombre de clauses satisfaites par la solution s[j] ;

TC = taux de croisement ;
TM = taux de mutation ;

- Comment appliquer les opérateurs génétiques sur les individus du problème. (4pts)

croisement :
- on choisit deux solutions au hasard
- puis on choisit un point de coupure pour les solutions c ϵ [1, n]
- puis on applique le changement des parties coupées. Il n'y a pas de problème d'admissibilité, toutes les solutions générées sont réalisables.

Mutation : une méthode serait de choisir aléatoirement une position du vecteur solution est d'inversion le bit correspondant

- Proposer une stratégie de remplacement des individus pour le renouvellement de la population. (2pts)

une stratégie serait de ranger tous les individus de la population et ceux nouvellement créés selon leur fonction d'évaluation et d'éliminer ceux qui ont les plus faibles valeurs de cette fonction.

- Ecrire l'algorithme génétique adapté au problème SAT. Quels sont les paramètres empiriques ? (2pts)

Algorithme génétique :
- initialisation
- tant que i <= max-iter faire
  - pour j = 1 à TC faire appliquer le croisement
  - pour j = 1 à TM faire appliquer la mutation
  - appliquer le remplacement

Les paramètres empiriques sont :
max-iter : le maximum d'itérations
TC : Taux de croisement
TM : Taux de mutation

BON COURAGE !