

Questions de cours (04 pts)

- 1) Citer trois propriétés communes à toutes les métaheuristiques.
- 2) Dans quels cas la méthode de la descente fournirait-elle une solution optimale ?
- 3) Quel est l'inconvénient majeur de l'algorithme de recuit simulé ?
- 4) Comment se comporte la recherche taboue quand la liste taboue est trop courte ?
- 5) Que veut dire un algorithme ε -approché ?

Exercice 1 (06 pts)

En appliquant la méthode du recuit simulé à un problème d'optimisation, on est arrivé à une itération où la température $T = 8$ et la valeur de la fonction objectif f de la solution courante x_0 est 1000. Cette solution a 2 voisins x_1, x_2 tels que $f(x_1) = 960$ et $f(x_2) = 1024$ respectivement. Pour chacune de ces 2 solutions, on se propose de déterminer la probabilité pour qu'elle remplace la solution courante x_0 .

- 1) Calculer ces probabilités pour un problème de maximisation.
- 2) Même question pour un problème de minimisation.
- 3) Que peut-on en déduire ?
- 4) Supposons que $T_{\max} = 500$ et $T_{\min} = 4$, le temps de recuit est Annealing = 10 et que le taux de réduction de la température est $\rho = 0.8$.
 - a) Calculer le nombre des itérations restantes avant que l'algorithme se termine.
 - b) Interpréter le résultat.

Exercice 2 (10 pts)

- 1) Donner le résultat retourné par l'algorithme ci-contre pour

$n = 5$ et $x_0 = \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & 2 & 1 \\ \hline \end{array}$

- 2) Que fait cet algorithme en général ?
Evaluer sa complexité temporelle.

```
int f (int n , int[] x) {  
    int s=0;  
    for (i = 0 ; i < n ; i++)  
        for (j = 0 ; j < i ; j++)  
            if (x[j] == x[i]) break ;  
            if (j==i) s+=1 ;  
    return s ; }  
}
```

- 3) Soit un problème d'optimisation combinatoire dont la solution est un vecteur d'entiers inférieurs à un entier donné n et soit la fonction de voisinage qui consiste à ajouter 1 à un élément aléatoire de ce vecteur (sans dépasser l'intervalle $[0, n-1]$).
 - a) Donner tous les voisins de la solution x_0 de la question (1).
 - b) Quel est le nombre de voisins d'une solution x de taille n ?
 - c) Ecrire l'algorithme qui retourne un voisin aléatoire d'une solution x .
 - d) Ecrire un algorithme qui calcule une solution initiale aléatoire x_0 de taille n .
 - e) Ecrire l'algorithme qui retourne le meilleur voisin d'une solution x minimisant la fonction f de l'algorithme de la question (1).
 - f) Quel problème vu au cours qui pourrait avoir une telle solution ?