

TD1 : RECHERCHES LOCALES

Exo1

Calculer une solution initiale aléatoire pour chacun des problèmes suivants :

1. Max $f(x) = x^3 - \sin x$ pour $x \in [-5, 5]$
2. Min $f(x_1, x_2, x_3) = x_1^2 - x_2^2$ pour $(x_1, x_2) \in [-5, 5]^2$
3. Min $f(x_1, x_2, x_3) = x_1^2 - x_2^2 + x_3^2$ pour $(x_1, x_2, x_3) \in [-5, 5]^3$
4. KSP simple
5. TSP symétrique
6. Coloration d'un graphe de n sommets
7. 8-queens

Exo2

Le tableau suivant récapitule les solutions optimales et approchées des moyennes de 5 types d'instances en utilisant une méthode exacte et 3 méthodes approchées :

Instance	Méthode exacte	Méthode approchée 1	Méthode approchée 1	Méthode approchée 1
1	15	15	16	16
2	26	27	27	26
3	33	35	34	35
4	47	49	50	52
5	60	62	61	62

- 1) S'agit-il d'un problème de minimisation ou de maximisation ?
- 2) Marquer la meilleure solution approchée pour chaque instance.
- 3) Calculer pour chaque instance, la déviation et le taux d'approximation pour chaque approche.
- 4) Déduire le facteur d'approximation de chaque approche.

Exo3

1. Considérons le KSP simple 0/1 (n , $v[]$, $w[]$, C).
2. Soit la fonction de voisinage qui consiste à inverser 1 bit.

a) Soit l'instance :

n=5, C= 10,					
v_i	10	12	8	15	6
w_i	2	5	1	3	4

Et soit la solution courante $x_0 = (10011)$.

Calculer :

- Le nombre de voisins de x_0 .
 - Tous les voisins de x_0 .
 - Le premier meilleur voisin de x_0 .
 - Le meilleur voisin de x_0 .
- b) Ecrire l'algorithme qui retourne :
- Un voisin aléatoire.
 - Tous les voisins.
 - Le premier meilleur voisin.
 - Le meilleur voisin.

c) Evaluer la complexité de chacun de ces algos.

3. Soit la fonction de voisinage qui consiste à inverser 2 bits différents.

a) Soit l'instance :

n=5, C= 10,					
vi	10	12	8	15	6
wi	2	5	1	3	4

Et soit la solution courante $x_0 = (10011)$.

Calculer :

- Le nombre de voisins de x_0 .
- Tous les voisins de x_0 .
- Le premier meilleur voisin de x_0 .
- Le meilleur voisin dex_0 .

b) Ecrire l'algorithme qui retourne :

- Un voisin aléatoire.
- Tous les voisins.
- Le premier meilleur voisin.
- Le meilleur voisin.

Exo4

On se propose de résoudre le PVC ($n, d[][]$) par une méthode de recherche locale.

1. Soit la fonction de voisinage qui consiste à permuter 2 villes.

Soit l'instance : $n=5, d[][] = \begin{bmatrix} 0 & 3 & 5 & 6 & 2 \\ 3 & 0 & 2 & 8 & 2 \\ 5 & 2 & 0 & 7 & 6 \\ 6 & 8 & 7 & 0 & 1 \\ 2 & 2 & 6 & 1 & 0 \end{bmatrix}$

Et soit la solution courante $x_0 = (23154)$.

a) Calculer :

- $f(x_0)$.
- Le nombre de voisins de x_0 .
- Tous les voisins de x_0 .
- Le premier meilleur voisin de x_0 .
- Le meilleur voisin dex_0 .

b) Ecrire l'algorithme qui retourne :

- Un voisin aléatoire.
- Tous les voisins.
- Le premier meilleur voisin.
- Le meilleur voisin.

c) Evaluer la complexité de chacun de ces algos.

2. Soit la fonction de voisinage qui consiste à insérer une ville aléatoire a un endroit aléatoire.

a) Calculer :

- Le nombre de voisins de x_0 .
- Tous les voisins de x_0 .
- Le premier meilleur voisin de x_0 .
- Le meilleur voisin dex_0 .

b) Ecrire l'algorithme qui retourne :

- Un voisin aléatoire.

- Tous les voisins.
 - Le premier meilleur voisin.
 - Le meilleur voisin.
- c) Evaluer la complexité de chacun de ces algos.

Exo5

Afin de résoudre le problème SAT-3 (n,m) (n est le nombre de variables et m est le nombre de clauses) par une méthode de recherche locale, nous définissons la fonction fitness d'une solution x comme étant le nombre de clauses satisfaites (=1) dans x. La fonction de satisfiabilité f est représentée par une matrice m*3 telle que :

$f_{ij} \in \mathbb{Z}, i = 1, m; j = 1, 3$ et :

$$\begin{cases} \text{si } f_{ij} = 0 \text{ la clause } i \text{ ne contient pas un littéral} \\ \text{si } f_{ij} > 0 \text{ la clause } i \text{ contient le littéral } x_{f_{ij}} \\ \text{si } f_{ij} < 0 \text{ la clause } i \text{ contient le littéral } \overline{x_{f_{ij}}} \end{cases}$$

Exemple : n=5 variables, m=4 clauses, $f = (x_2 \vee x_3 \vee \overline{x_5}) \wedge (x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2})$

Devient $f = \begin{bmatrix} 2 & 3 & -5 \\ 1 & -2 & 0 \\ 1 & 2 & -3 \\ -2 & 0 & 0 \end{bmatrix}$

Soit les solution $x_0 = 10001$, $x_1 = 11101$, $x_2 = 10111$. Calculer $f(x_0)$, $f(x_1)$, $f(x_2)$.
Reprendre les questions de l'exo1.

Exo6

On se propose de calculer les entiers a,b,c,d compris entre 1 et 20 qui satisfont: $a+2b+3c+4d=50$.

- 1) Calculer la taille de l'espace de recherche.
- 2) Proposer une fonction fitness pour évaluer les solutions candidates de ce problème.
- 3) Ecrire un algorithme générant une solution aléatoire à ce problème.
- 4) Considérons la fonction de voisinage qui consiste à modifier un et un seul des entiers a,b,c,d d'une solution. Ecrire l'algorithme qui retourne un voisin aléatoire de la solution (a,b,c,d).

Exo7

Dans l'algorithme ci-contre, la fonction Rand() retourne un nombre réel r tel que $0 \leq r < 1$.

- 1) Que fait cet algorithme ?
- 2) Qu'appelle-t-on cette méthode ? Citer un avantage et un inconvénient de cette méthode.
- 3) Evaluer en fonction de x0 et y0 le nombre d'itérations de cet algorithme.
- 4) Modifier cet algorithme pour calculer le maximum de la fonction :

$$g(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i^2 \text{ pour } x_i \in [-2, +2]$$

en utilisant la même méthode et ses paramètres.

```
Float f(float x, float y)
{ Return x * x + y * y - x * y - 2 ; } ;

Main void ()
{ Float x0 = 5 * Rand() ;
  Float y0 = 5 * Rand() ;
  Float xopt , yopt ;
  Do
  { fopt = f(x0,y0) ;
    xopt = x0 ; yopt = y0 ;
    x0 = x0 - 0.1 + 0.2 * Rand() ;
    y0 = y0 - 0.1 + 0.2 * Rand() ; }
  While (f(x0,y0) <= fopt) ;
  Printf (xopt , yopt , fopt) ; }
```

Exo8

En appliquant la méthode du recuit simulé pour un certain problème d'optimisation d'une fonction f , on est arrivé à une température T et à la solution courante x_0 tel que $f(x_0) = 50$. Cette solution a 2 voisins x_1 et x_2 tels que $f(x_1) = 52$ et $f(x_2) = 49$. Pour chacune de ces solutions, on se propose de déterminer la probabilité pour qu'elle soit choisie pour être la prochaine solution courante ($P(x_0 \leftarrow x_1)$ et $P(x_0 \leftarrow x_2)$ pour remplacer x_0).

- 1) Supposons que $T = 1000$:
 - a) Calculer ces probabilités pour un problème de maximisation.
 - b) Même question pour un problème de minimisation.
- 2) Supposons que $T = 5$:
 - c) Calculer ces probabilités pour un problème de maximisation.
 - d) Même question pour un problème de minimisation.
- 3) Que peut-on déduire ?
- 4) Sachant que le pallier est k et que le taux de réduction de la température est α .
Exprimer en fonction de k et α le nombre d'appels de la fonction de voisinage entre $T = 1000$ et $T = 5$.

Exo9

L'heuristique du plus proche voisin pour le PVC s'énonce comme suit :

On choisit un sommet (ville) arbitraire.

On part au sommet voisin le plus proche, puis de celui-là, à son plus proche voisin non visité, etc...;

Jusqu'à ce que tous les sommets aient été parcourus, où l'on revient au départ.

- 1) Donner un modèle puis une formulation mathématique pour le PVC.
- 2) Rappeler la taille de l'espace de recherche.
- 3) Ecrire l'algorithme de recherche exhaustive.
- 4) Ecrire l'algorithme de l'heuristique ci-dessus.
- 5) Exprimer sa complexité.

Exo 10

On veut colorer un graphe avec le nombre minimum k de couleurs.

1. Ecrire la formulation mathématique de ce problème.
2. Proposer une fonction fitness.
3. Proposer une fonction de voisinage.

Exo 11

Soit à résoudre le problème de 8 reines par une recherche locale..

4. Ecrire la formulation mathématique de ce problème.
5. Proposer une fonction fitness.
6. Proposer une fonction de voisinage.

Exo 12

Un sudoku 4x4 se compose d'une grille de 4x4 cases ; divisée en 4 régions de 2x2 cases; la grille est déjà partiellement remplie ; il faut la compléter avec des nombres entre 1 et 4 ; de telle sorte qu'un chiffre n'apparaisse jamais 2 fois dans chaque ligne, chaque colonne et chaque région. Caractériser puis formuler ce problème d'optimisation.