

Exercice 1 : Questions de cours (05 points)

- 1) $f(x_0) \leq f(x^*)$ dans le cas de maximisation. $f(x_0) \geq f(x^*)$ dans le cas de minimisation.....1
- 2) Oui lorsque f satisfait : $\exists x_1, x_2 / x_1 \neq x_2$ et $f(x_1) = f(x_2)$ (f est surjective).....1
- 3) Une position d'une particule est une solution du POC.
La vitesse d'une particule sert à modifier la position donc à explorer une nouvelle solution.....1
- 4) Représenter la solution par un vecteur et définir une distance entre ses composantes.....1
- 5) Full, grow, ramped half and half.....1

Exercice 2 (07 points)

- 1) Une matrice x ($2 \times n/2$) des entiers 1 à n , où chaque colonne j est un binôme si $\{x[0][j], x[1][j]\} \notin U$.
Une matrice carrée binaire $x(n \times n)$ dont $n \geq 1$, t.q. si $x[i][j] = 1$ et $\{i, j\} \notin U$ alors $\{i, j\}$ est un binôme.....1
- 2) Taille de l'espace de recherche = $n(n-1)/2 - |U|$0.5
- 3)
$$\begin{cases} \min(\max_{i=1, n, j=1, n} x_{ij} * (c_i + c_j)) \\ x_{ij} \in \{0, 1\} \\ u_{ij} \neq 1 \end{cases} \dots\dots\dots 1.5$$
- 4) a) `Void Initialisation(){`
`for (k=0 ; k<m ; k++)`
`for (i=0 ; i<n ; i++)`
`j=int(n*rand(0,1))`
`pop[k][i][j]=1;}`.....1.5
Complexité = $O(m \times n)$0.5
b) `Void Evaluation(){`
`for (k=0 ; k<m ; k++)`
`{s=0;`
`for (i=0 ; i<n ; i++)`
`for (j=0 ; j<n ; j++)`
`if (c[i]+c[j]>s) s = c[i]+c[j];`
`fitness[k] = s; }`.....1.5
Complexité = $O(m \times n^2)$0.5

Exercice 3 (08 points)

- 1) $\text{Prob}(D, C \rightarrow A) = \frac{(\tau_{CA})^{\alpha} * (\mu_{CA})^{\beta}}{(\tau_{CA})^{\alpha} * (\mu_{CA})^{\beta} + (\tau_{CB})^{\alpha} * (\mu_{CB})^{\beta}} = 0.4$1
 $\text{Prob}(D, C \rightarrow B) = \frac{(\tau_{CB})^{\alpha} * (\mu_{CB})^{\beta}}{(\tau_{CA})^{\alpha} * (\mu_{CA})^{\beta} + (\tau_{CB})^{\alpha} * (\mu_{CB})^{\beta}} = 0.6$ ou bien $1-0.4=0.6$ (les villes restantes sont A et B).....1
- 2) $\text{Prob}(D, C, A \rightarrow B) = 1$ (B est la seule ville restante).....1
- 3) $\text{length}(DACB) = DA + AC + CB + BD = 5 + 3 + 2 + 5 = 15$1
`Int len(int[] x){ s=0;`
`for (i=0 ; i<n-1 ; i++) s+=d[x[i]][x[i+1]] ;`
`return s+ = d[x[n-1]][x[0]] ;}`.....1.5

4) Delta =

	A	B	C	D
A	0	0	0.66	0
B	0	0	0	0.66
C	0	0.66	0	0
D	0.66	0	0	0

.....1

```
Void MAJDELTA (int[] x) {
    L= Length(x);
    for (i=0 ; i<n-1 ; i++) Delta[x[i]][x[i+1]] += Q/L ;
    Delta[x[n-1]][x[0]] += Q/L ;}.....1.5
```