Centre Universitaire de Mila

Institut des sciences et de la technologie

1ère Master STIC

Année : 2018-2019

Module :

Contrôle d’optimisation combinatoire et

métaheuristiques

**Exercice 1: 6 points**

1. Citer les caractéristiques des problèmes de la classe NP.

* Les problèmes de la classe NP sont des problèmes de décision qui peuvent être décidés sur une machine non déterministe en temps polynomial.
* Les problèmes NP admettent un algorithme polynomial capable de tester la validité d’une solution du problème.

1. Quelle est la différence entre un problème NP et un problème NP-complet.

La différence est que un problème NP-Complet est un problème NP mais possède en particulier les caractéristiques suivantes :

* les problèmes NP-complets sont les plus difficiles de la classe NP.
* Tout problème dans NP peut être réduit en un problème NP-complet celui-ci en temps polynomial.
* Si on trouve un algorithme polynomial pour un problème NP-complet, on trouve alors automatiquement une résolution polynomiale de tous les problèmes de la classe NP.

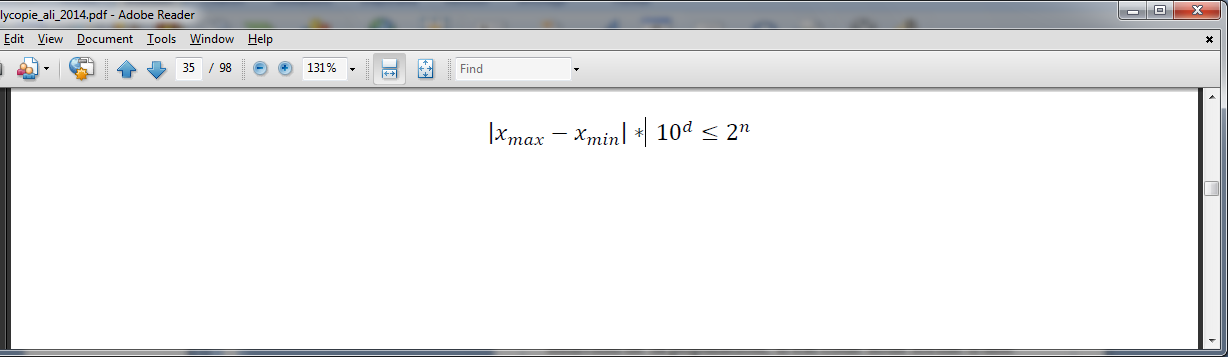
1. Dans quel cas on doit s’orienter vers les métaheuristiques pour résoudre un problème donné.

Dans le cas où la taille du problème est très grande qui signifie que les méthodes exactes ne prouvent pas donnée une solution dans un temps raisonnable.

1. Quels problèmes nous essayons d’éviter par l’utilisation des recuits simulés et de la recherche taboue.

Par l’utilisation des recuits simulés et de la recherche taboue on essaye d’éviter de tomber dans un minimum local et aussi, par l’utilisation de la recherche taboue, d’éviter le problème de cycle.

1. Nous cherchons à trouver le maximum d’une fonction *f* (x) sur l’intervalle [0, 2.56] en utilisant un algorithme génétique. Calculer la taille du chromosome (représentation binaire) dans le cas où *x* est un entier et dans le cas où *x* est un réel avec deux nombre après la virgule.



1. |2.56-0| \* 102 ≤2n  🡺 265≤2n 🡺 28≤2n 🡺 n=8

**Exercice 2: 3.5 points**

On cherche à trouver le plus court chemin entre le nœud n0 et n6.

1. Résoudre ce problème par l’algorithme A\*.

|  |  |  |
| --- | --- | --- |
|  | **Open** | **Closed** |
| **1** | (n0, 50, void) | Vide |
| **2** | (n1, 50, n0), (n3, 60, n0), (n2, 70, n0) | (n0, 50, void) |
| **3** | (n3, 60, n0), (n2, 70, n0), (n5, 12, n1) | (n0, 50, void), (n1, 50, n0), |
| **4** | (n2, 70, n0), (n4, 90, n2) ,(n5, 120, n1) | (n0, 50, void), (n1, 50, n0), (n3, 60, n0) |
| **5** | (n3, 50, n2), (n4, 60, n2) ,(n5, 120, n1) | (n0, 50, void), (n1, 50, n0), (n2, 70, n0) |
| **6** | (n4, 60, n2) ,(n5, 120, n1) | (n0, 50, void), (n1, 50, n0), (n2, 70, n0), (n3, 50, n2), |
| **7** | (n6, 70, n4) ,(n5, 120, n1) | (n0, 50, void), (n1, 50, n0), (n2, 70, n0), (n3, 50, n2), (n4, 60, n2) |
| **8** | (n5, 120, n1) | (n0, 50, void), (n1, 50, n0), (n2, 70, n0), (n3, 50, n2), (n4, 60, n2) , (n6, 70, n4) |

**Solution:** n0, n2, n4, n6

**Résolution par l’algorithme glouton.**

**A🡪C ; L=20**

**C🡪F ; L=30**

**F🡪G ; L=70**

Est-ce qu’on peut confirme que les solutions trouvées sont optimales ou non

Pour l’algorithme glouton on ne peut pas confirmer mais pour l’algorithme A\* on peut confirmer parce *h*(n) est admissible.

**Exercice 4: 6**

1. Résolution par l’algorithme glouton
2. Résolution par branch and bound



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Taches  Personnes | T1 | T2 | T3 | T4 |  |
| P1 | 10 | 3 | 8 | 9 | 3 |
| P2 | 7 | 5 | 4 | 8 | 4 |
| P3 | 6 | 9 | 2 | 9 | 2 |
| P4 | 8 | 7 | 10 | 5 | 5 |
|  |  | | | | 14 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Taches  Personnes | T1 | T2 | T3 | T4 |  |
| P1 | 10 | 3 | 8 | 9 | 10 |
| P2 | 7 | 5 | 4 | 8 | 4 |
| P3 | 6 | 9 | 2 | 9 | 2 |
| P4 | 8 | 7 | 10 | 5 | 5 |
|  |  | | | | 21 |

La solution est T1🡨P2, T2🡨P1, T3🡨P3, T4🡨P4 avec b=17

1. Si on utilise la méthode de recuits simulés avec T=10. Calculer pour chacun des voisins de la solution s la probabilité d’être choisi comme solution courante.

S= (T1🡨P1, T2🡨P4, T3🡨P3, T4🡨P2) f(s)= 10+7+2+8 =27

|  |  |
| --- | --- |
| N(s) | Probabilité |
| (T1🡨P4, T2🡨P1, T3🡨P3, T4🡨P2) | f(s0)= 22 <f(s) 🡺 P=1 |
| (T1🡨P1, T2🡨P3, T3🡨P4, T4🡨P2) | f(s0)= 37 > f(s) 🡺 P=0,36 |
| (T1🡨P1, T2🡨P4, T3🡨P2, T4🡨P3) | f(s0)=30 <f(s) 🡺 P=0,74 |
| (T1🡨P2, T2🡨P4, T3🡨P3, T4🡨P1) | f(s0)=25 <f(s) 🡺 P=1 |

1. La méthode de recherche taboue :

Initialisation :

S🡨 (T1🡨P1, T2🡨P4, T3🡨P3, T4🡨P2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T | (T1🡨P1, T2🡨P4, T3🡨P3, T4🡨P2) |  |  |  |

Best🡨(T1🡨P1, T2🡨P4, T3🡨P3, T4🡨P2)

Itération 1 :

N(s) ={ ((T1🡨P4, T2🡨P1, T3🡨P3, T4🡨P2), (T1🡨P1, T2🡨P3, T3🡨P4, T4🡨P2), (T1🡨P1, T2🡨P4, T3🡨P2, T4🡨P3), (T1🡨P2, T2🡨P4, T3🡨P3, T4🡨P1)}

S0= (T1🡨P4, T2🡨P1, T3🡨P3, T4🡨P2)

f(s0)=22

best= T1🡨P4, T2🡨P1, T3🡨P3, T4🡨P2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T | (T1🡨P1, T2🡨P4, T3🡨P3, T4🡨P2) | (T1🡨P4, T2🡨P1, T3🡨P3, T4🡨P2) |  |  |

Itération 2 :

N(s) ={ (T1🡨P1 , T2🡨P4, T3🡨P3, T4🡨P2), (T1🡨P4, T2🡨P3, T3🡨P1, T4🡨P2), (T1🡨P4, T2🡨P1, T3🡨P2, T4🡨P3), (T1🡨P2, T2🡨P1, T3🡨P3, T4🡨P4) }

S0= (T1🡨P2, T2🡨P1, T3🡨P3, T4🡨P4)

f(s0)=17

best=(T1🡨P2, T2🡨P1, T3🡨P3, T4🡨P4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T | (T1🡨P1, T2🡨P3, T3🡨P4, T4🡨P2) | (T1🡨P1, T2🡨P4, T3🡨P3, T4🡨P2) | (T1🡨P2, T2🡨P1, T3🡨P3, T4🡨P4) |  |

|  |
| --- |
| binf=60 |
| b=0 |
|  |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |

|  |
| --- |
| bsup=60 |
| b=20 |
| P=2 |