Groupe 2-4SIM4: Anas HOUISSA

Problème d'ordonnancement

Algorithme de recherche locale

Exercice 2 :

Question 1 : Donner le pseudo-code de l'algorithme de la recherche locale.

```
Début
  Choisir une solution initiale S
  Tant que [critère = false] faire
    Pour chaque solution voisine S' de S faire
      Evaluer la qualité de S' en utilisant la fonction d'évaluation
    Fin Pour
    Sélectionner la meilleure solution S' parmi les voisins évalués
    Si S' est meilleure que S alors
      S = S'
    Fin Si
  Fin Tant que
Retourner la meilleure solution S trouvée
Fin
```

Question 2 : Expliquer l'utilité des différents paramètres de l'algorithme.

Paramètre 1 :

La solution initiale <choisie au hasard> est importante car elle peut avoir un impact sur la qualité de la solution finale et la rapidité de la résolution. Si la solution initiale est proche de l'optimum, l'algorithme peut converger plus rapidement vers une solution optimale.

Paramètre 2 :

Les opérateurs locaux <transformations appliquées aux solutions> permettent de trouver des solutions voisines à partir de la solution courante. Ils sont utilisés pour explorer l'espace des solutions et trouver des solutions de meilleure qualité.

Paramètre 3 :

La fonction d'évaluation permet de mesurer la qualité des solutions en fonction du critère d'optimisation. Il est utilisé pour évaluer chaque solution générée par l'algorithme et déterminer si elle est meilleure que la solution courante.

Paramètre 4 :

Le critère d'arrêt est important pour éviter une boucle infinie et pour arrêter l'algorithme après n itérations ou lorsque la solution courante ne peut plus être améliorée

Question 2 : Comment peut-on appliquer cet algorithme pour résoudre votre problème d'optimisation?

Pour résoudre un problème d'ordonnancement en utilisant la recherche locale, on peut considérer les différentes tâches à ordonnancer comme des variables et les différentes séquences d'ordonnancement comme des solutions. On peut ensuite utiliser les opérateurs locaux pour intervertir l'ordre de certaines tâches dans une séquence et évaluer la qualité des solutions en utilisant un critère d'optimisation tel que le temps d'exécution total. Le processus de recherche locale peut être répété jusqu'à ce qu'une solution satisfaisante soit trouvée.

On considère trois tâches A, B, C avec des durées d'exécution sur deux machines M1 et M2. Les durées d'exécution de chaque tâche sur chaque machine sont données dans le tableau suivant :

Tâches	Machine M1	Machine M2
A	5 heures	4 heures
B	3 heures	5 heures
C	2 heures	1 heure

On cherche à déterminer la meilleure solution d'ordonnancement pour ces tâches, c'est-à-dire l'ordre dans lequel les tâches doivent être exécutées et sur quelle machine.

Solution initiale :

Machine M1 : [A, B, C] **Machine M2 : vide**

Cela signifie que toutes les tâches sont exécutées sur la machine M1, dans l'ordre A, B, C. Nous allons maintenant utiliser la recherche locale pour améliorer cette solution.

Nous allons déplacer une tâche de la machine M1 vers la machine M2 pour créer une nouvelle solution voisine. Nous allons évaluer la qualité de la nouvelle solution obtenue.
Cette étape sera répétée jusqu'à ce qu'aucune amélioration ne puisse être apportée à la solution courante.

Itération 1 :

M1 [A,C] ➔ 7 heures
M2 [B] ➔ 5 heures

} Durées d'exécution : 12 heures

Itération 2 :

M1 [C] ➔ 2 heures
M2 [A,B] ➔ 9 heures

} Durées d'exécution : 11 heures

Itération n :

M1 [B] ➔ 3 heures
M2 [A,C] ➔ 5 heures

} Durées d'exécution : 8 heures

<u>Solution optimale :</u>

En utilisant la recherche locale, nous avons trouvé la solution optimale suivante :

- Machine M1 : B
- Machine M2 : A, C

Dans cette solution, les tâches A et C sont exécutées sur la machine M2 et la tâche B est exécutée sur la machine M1. La qualité de cette solution est optimale car elle minimise le temps total d'exécution de toutes les tâches.