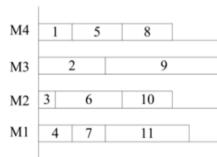


Ordonnancement de Machines Parallèles

Avec Opérations Pondérées

Abderazak HSAINI

1 Nov 2024





1 Introduction

- ▶ Introduction
- ► Modèle Mathématique
- ► Heuristique
- ► Métaheuristique
- ► Résultats
- ► Conclusion

- Le problème d'ordonnancement de machines parallèles consiste à affecter des opérations à plusieurs machines identiques.
- Chaque opération a un temps de traitement et un poids représentant sa priorité.
- Objectif : Minimiser la somme pondérée des temps de fin et le makespan.
- Ce problème est NP-difficile, ce qui rend les méthodes exactes inefficaces pour les grandes instances.



2 Modèle Mathématique

- ▶ Introduction
- ► Modèle Mathématique
- ► Heuristique
- ► Métaheuristique
- ► Résultats
- ► Conclusion



Modèle Mathématique : Paramètres et Variables

2 Modèle Mathématique

• Paramètres :

- m: Nombre de machines.
- n: Nombre d'opérations.
- t_j : Temps de traitement de chaque opération j.
- w_j : Poids de chaque opération j.

• Variables de Décision :

- x_{iy} : Variable binaire (1 si l'opération i est affectée à la machine y, sinon 0).
- F_i : Temps de fin de l'opération j.
- F: Makespan, soit le temps total maximal pour compléter toutes les opérations.

Fonction Objectif et Contraintes

2 Modèle Mathématique

Fonction Objectif:

$$\min \Gamma = \sum_{j=1}^{n} w_j F_j + F$$

où w_j pondère chaque temps de fin et F est le makespan total.

Contraintes:

- Affectation unique : chaque opération est affectée à une seule machine.
- Non-préemption : une machine traite une opération à la fois.
- Succession : les temps de fin tiennent compte des dépendances.
- Makespan: le temps de fin ne dépasse pas F.



3 Heuristique

- Introduction
- ▶ Modèle Mathématique
- ▶ Heuristique
- ► Métaheuristique
- ▶ Résultats
- ▶ Conclusion

Heuristique : List Scheduling Pondéré (LS)

3 Heuristique

- Solution initiale : utilisation de l'heuristique List Scheduling Pondéré.
- Étapes :
 - 1. Trier les opérations par ratio $\frac{w_j}{t_j}$.
 - 2. Affecter chaque opération à la machine, minimisant son temps de fin.
 - 3. Calculer le makespan et la somme pondérée des temps de fin.
- Cette heuristique fournit une solution rapide comme point de départ pour la métaheuristique.



4 Métaheuristique

- ▶ Introduction
- ▶ Modèle Mathématique
- ► Heuristique
- \blacktriangleright Métaheuristique
- ► Résultats
- ► Conclusion



Métaheuristique : Recherche Tabou (Tabu Search)

4 Métaheuristique

- La recherche tabou améliore une solution initiale en explorant son voisinage.
- Utilisation d'une liste tabou pour éviter les retours en arrière indésirables.
- La solution initiale provient de l'heuristique List Scheduling Pondéré.



Étapes de la Recherche Tabou

4 Métaheuristique

- 1. Solution Initiale : obtenue via List Scheduling Pondéré.
- 2. Génération de Voisins : permutation des tâches entre les machines.
- 3. Liste Tabou : stockage des solutions récemment explorées.
- 4. Critère d'Amélioration : acceptation de solutions améliorant le makespan ou la somme pondérée.
- 5. Critère d'Arrêt : arrêt après un nombre fixé d'itérations sans amélioration.



Table des matières ⁵ Résultats

- ▶ Introduction
- ► Modèle Mathématique
- ► Heuristique
- ► Métaheuristique
- ► Résultats
- ► Conclusion



Résultats : Évaluation des Performances 5 Résultats

- Tests sur deux types d'instances :
 - Petite instance : 5 opérations et 3 machines.
 - Grande instance : 50 opérations et 10 machines.
- Petite Instance : convergence rapide vers une solution optimale.
- Grande Instance: amélioration significative avec la recherche tabou.



Résultats : Paramètres de la Métaheuristique ⁵ Résultats

• Taille de la Liste Tabou :

- Influence sur l'évitement des cycles et la diversité d'exploration.
- Réglage : généralement entre $0.1 \times n$ et $0.2 \times n$.

• Critère d'Aspiration :

- Permet de revenir sur une solution taboue si elle est significativement meilleure.
- Réglage : une amélioration de δ en makespan (par exemple, 5-10%).

• Critère d'Arrêt :

- Définit la convergence par nombre d'itérations ou non-amélioration.
- Réglage : $100 \times n$ ou 10-20 itérations sans amélioration.



6 Conclusion

- ▶ Introduction
- ► Modèle Mathématique
- ► Heuristique
- ► Métaheuristique
- ► Résultats
- ► Conclusion

Conclusion et Perspectives

6 Conclusion

- Le problème d'ordonnancement de machines parallèles est un défi complexe.
- La combinaison d'une heuristique initiale et de la recherche tabou permet d'approcher une solution de haute qualité.
- Résultats encourageants sur des instances de grande taille où les méthodes exactes échouent.
- Perspectives : exploration d'approches hybrides et d'optimisation des paramètres pour des gains de performance supplémentaires.