Ordonnancement sur machines parallèles avec prise en compte de l'état de santé : modélisation mathématique

Margaux Nattaf[1] et Stéphane Dauzère-Pérès[2]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP[†], G-SCOP, Grenoble, France
{margaux.nattaf}@grenoble-inp.fr

[2] Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, CMP, Department of Manufacturing Sciences and Logistics, Gardanne, France
{dauzere-peres}@emse.fr

Mots-clés : *Ordonnancement à machines parallèles, fabrication de semi-conducteurs, santé des équipements*

Introduction De nombreuses études se sont intéressées à l'ordonnancement en fabrication de semi-conducteurs. Or, dans ces études, le statut d'une machine est souvent binaire [3], i.e. 1 signifie que la machine est disponible pour exécuter une tâche et 0 que la machine n'est pas disponible (panne, maintenance, non-conformité des paramètres, ...). Cependant, même si la machine est indiquée comme étant disponible, la qualité du processus peut ne pas être garantie en raison de la détérioration de la machine [1]. Le problème traité dans ce résumé, inspiré du problème étudié dans [2], vise à prendre en compte de la détérioration des machines dans l'ordonnancement en intégrant la notion de santé des machines.

Le problème traité vise à ordonnancer un ensemble de tâches $\mathcal{N}$, dont chacune est associée à une famille et $\mathcal{F}$ est l'ensemble de ces familles, sur un ensemble de machines $\mathcal{M}$. Á chaque famille $f \in \mathcal{F}$ est associée une durée p_f et un temps de transition (*setup*) s_f.

Á l'instant $t = 0$, chaque machine $m \in \mathcal{M}$ possède un niveau de santé donné $H_m \in [0, 100]$. Ce niveau de santé évolue au cours de l'ordonnancement. En effet, lors de l'exécution d'une tâche j, le niveau de santé de la machine décroît en fonction de la famille de la tâche. Une "pénalité" $\alpha^m_{f(j)} > 0$ est appliquée à H_m , i.e. si j appartient à f et est exécutée au temps t, alors on a $H_m(t+p_f) = H_m(t) - \alpha^m_f$. De même, un temps de transition peut détériorer l'état de santé de la machine, et une pénalité β^m_f est appliquée à l'état de santé de m quand un temps de transition est nécessaire. Dans ce cas, on a $H_m(t + s_f) = H_m(t) - \beta^m_f$. Notons que, dans de rares cas, un temps de transition peut améliorer l'état de santé de la machine mais ce cas n'est pas considéré dans cette étude.

De plus, on associe à chaque famille deux valeurs LH_f et UH_f. Ces valeurs catégorisent le niveau de santé de la machine au moment de l'exécution d'une tâches en trois catégories : favorable (supérieur à UH_f), acceptable (entre LH_f et UH_f) et défavorable (inférieur à LH_f). Alors, on dit qu'un risque existe lorsqu'une tâche est exécutée sur une machine présentant un niveau de santé acceptable pour la tâche. L'objectif est de minimiser deux critères : le risque total et la somme des dates de fin des tâches. Ce problème sera dénoté par PEHI (*scheduling Problem with Equipement Health Index*).

Enfin, des opérations de maintenances peuvent être réalisées afin d'améliorer la santé de la machine. En utilisant une transformation simple, ces opérations peuvent être traitées comme des tâches appartenant à une nouvelle famille.

Programmation Linéaire en Nombre Entiers Dans le cas général, un Programme Linéaire en Nombre Entiers (PLNE) indexé par le temps peut être utilisé pour résoudre le

[†]Institute of Engineering Univ. Grenoble Alpes

problème. Le PLNE utilise les variables suivantes : $x^m_{f,t} \in \{0,1\}$ représente les dates de début des tâches ; $z^m_{f,t} \in \{0,1\}$ modélise la présence d'un temps de transition ; $H^m_t \in \mathbb{R}$ représente l'état de santé de la machine m et $Q^m_{f,t} \in \mathbb{R}$ est utilisé pour représenter le risque associé à l'exécution d'une tâche de la famille f au temps t sur la machine m.

Sans rentrer dans le détail des contraintes, il est clair que le PLNE a un nombre important de variables. De ce fait, son utilisation est limitée à de "petites" instances, i.e. peu de tâches et/ou familles et/ou machines. Le but de cette étude est de connaître les limites exactes du modèle PLNE et de savoir quelles sont les tailles d'instances que l'on peut espérer résoudre avec un solveur standard. Des résultats expérimentaux permettant de répondre à cette question seront présentés lors de la conférence.

D'autres modélisations de problèmes d'ordonnancement sous la forme d'une PLNE existent. Parmi les plus connues, nous citerons les modèles avec des variables de séquencement ou de positionnement. Ces modélisations sont cependant difficiles à appliquer dans le cas du PEHI. En effet, l'état de santé des machines et l'évaluation du risque associé à l'ordonnancement d'une tâche sur une machine nécessite de connaître le temps exact à laquelle la tâche s'exécute.

Génération d'instances Afin de générer des instances les plus réalistes possibles, nous avons utilisé le schéma suivant. Tout d'abord, nos instances sont générées à partir de celles proposées dans [3]. De ces instances, nous récupérons le nombres de tâches et de machines ainsi que le nombre de familles et les caractéristiques suivantes : la durée p_f, le temps de transition s_f et le nombre de tâches dans chaque famille. Les opérations de maintenances sont considérées comme 4 fois plus longues que la tâche de plus grande durée.

Au début de l'ordonnancement (au temps 0), l'état de santé d'une machine est générée de manière aléatoire dans l'intervalle $[50, 100]$. Les états de santé de chaque famille sont ensuite générées de la façon suivante :

1. $UH_f = 80, \ \forall f \in \mathcal{F}$,
2. LH_f est tiré dans l'ensemble $\{80; 70; 60; 50\}$ avec les probabilités suivantes : $\{20\%; 20\%; 30\%; 30\%\}$.
3. Nous avons généré des pénalités indépendantes des machines utilisées. Pour générer les pénalités associée à l'exécution d'une tâche, $\alpha^m_f = \alpha_f$, nous avons choisi aléatoirement une valeur dans l'ensemble $\{2 \cdot p_f; 1,5 \cdot p_f; p_f; 0,5 \cdot p_f\}$ avec les probabilités suivantes : $\{20\%; 20\%; 30\%; 30\%\}$.
4. Pour générer les pénalités associées au setup, $\beta^m_f = \beta_f$, nous avons choisi aléatoirement une valeur dans l'ensemble $\{1,5 \cdot s_f; s_f; 0,5 \cdot s_f\}$ avec les probabilités suivantes : $\{30\%; 40\%; 30\%\}$.

La présentation à la conférence inclura des analyses détaillées des résultats expérimentaux sur ces instances.

Références

[1] Argon Chen and GS Wu. Real-time health prognosis and dynamic preventive maintenance policy for equipment under aging markovian deterioration. *International Journal of Production Research*, 45(15) :3351–3379, 2007.

[2] Yu-Ting Kao, Stéphane Dauzère-Pérès, Jakey Blue, and Shi-Chung Chang. Impact of integrating equipment health in production scheduling for semiconductor fabrication. *Computers & Industrial Engineering*, 120 :450 – 459, 2018.

[3] Margaux Nattaf, Stéphane Dauzère-Pérès, Claude Yugma, and Cheng-Hung Wu. Parallel machine scheduling with time constraints on machine qualifications. *Computers & Operations Research*, 107 :61 – 76, 2019.