I. Benkalai, P. Baptiste, D. Rebaine

Ordonnancement d'ateliers de type flow shop avec opérateurs en mode d'affectation libre

Présenté par : Imène Benkalai

17ème conférence ROADEF
Société Française de Recherche Opérationnelle et d'Aide à la Décision

Compiègne, le 10-02-2016

Sommaire

Ordonnancement

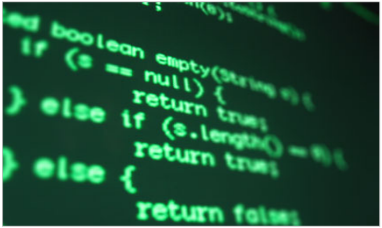
- n tâches.
- m machines.
- p_{ij} : Le temps de traitement de la tâche j sur la machine i.
- C_j : Le temps auquel la tâche j quitte le système.
- d_j : La date d'échéance de la tâche j.

Applications

- Gestion d'un chantier.
- Exécution de programmes informatiques.
- Organisation de départs et d'arrivées dans un aéroport.

Applications

- Gestion d'un chantier.
- Exécution de programmes informatiques.
- Organisation de départs et d'arrivées dans un aéroport.



Applications

- Gestion d'un chantier.
- Exécution de programmes informatiques.
- Organisation de départs et d'arrivées dans un aéroport.

RM et RH

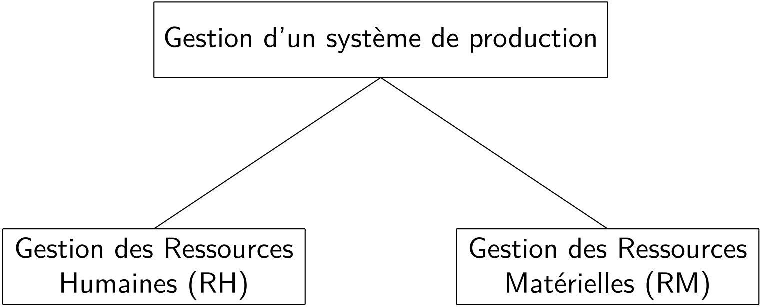


FIGURE – 1. Les deux axes de la gestion de production.

Intégration d'opérateurs

- Problèmes complexes.
- Difficilement modélisables.
- Multiples contraintes.
- Hypothèses simplificatrices.
- Les décisions "*séparées*" sont moins susceptibles d'être valides.
- Variation des durées de traitement des tâches.

Influence des opérateurs

- Nombre.
- Compétences.
- Performances.
- Etc.

Points à prendre en considération

- La configuration (où sont les opérateurs ?).
- L'affectation (qui fait quoi ?).
- La séquence des différentes tâches sur les machines (dans quel ordre ?).
- Les dates de début et de fin de traitement des tâches ainsi que les intervalles de temps d'utilisation des différentes affectations (quand ?).

Probléme étudié

- Problème de flow shop standard.
- Prise en compte des opérateurs avec ordre connu au d épart.
- Nombre d'opérateurs inférieur au nombre de machines.
- Mode d'affectation libre.
- l types d'opérateurs avec différents niveaux de performance.
- Étude de complexité et méthodes.
- Objectif : minimiser $L_{\max} = \max\limits_{j=1,\dots,n} L_j$ où $L_j = C_j - d_j$.

Probléme étudié

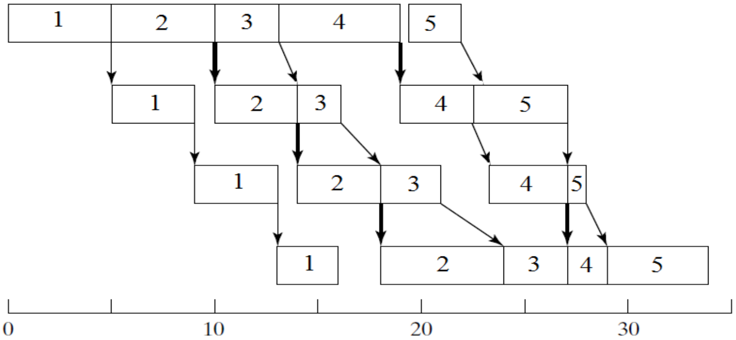


FIGURE – 2. Problème du flow shop avec opérateurs.

Revue de la littérature

- Foisonnement de travaux portant sur le flow shop.
- 1200 articles sur le flow shop de permutation en 2009.
- Très peu d'articles sur le flow shop standard.

Flow shop standard

- Études comparatives du flow shop standard avec celui de permutation.
- Heuristiques constructives.
- Métaheuristiques.

Flow shop avec contraintes de ressources

- Ordonnancement cyclique avec robots.
- Transformation du problème en TSP.
- Ordonnancement avec un opérateur et plusieurs machines.
- Ordonnancement dans des environnements à machines parallèles.
- Ordonnancement dans des environnements Open shop.

Cas à $k \geq 3$ opérateurs

Théorème

Le problème $F_m|res\ \ell k_h 1, S, v_h|L_{\max}$ est NP-difficile au sens fort pour $k \geq 3$ et ce, même pour le cas où les v_h sont tous égaux.

Démonstration.

Transformation en problème à machines parallèles uniformes avec préemption et contraintes de précédence.

□

Cas à $k = 2$ opérateurs

Théorème

Le problème $F_m|res\ 2k_h1, S, v_h|L_{\max}$ est résoluble en temps polynomial.

Démonstration.

Transformation en problème à deux machines parallèles uniformes avec préemptions et contraintes de précédence.

$\square$

Cas à $k = 1$ opérateur

Théorème

Le problème $F_m | res\ 111, S | L_{\max}$ est résoluble en temps polynomial.

Démonstration.

Transformation en problème à machine unique avec préemption et contraintes de précédence.

□

Cas à $k = 2$ opérateurs

La solution est construite en trois étapes :

- Modification des dates d'échéance.
- Construction d'intervalles d'ordonnancement.
- Application d'un algorithme d'ordonnancement par priorité.

Cas à $k = 2$ opérateurs

Pour (toute opération (i, j) sans successeur)
$\{d'_{ij} \leftarrow d_{ij}\,;\}$
Créer deux listes ordonnées par date d'échéance modifiées d'_{ij} et $b'_{ij} = d'_{ij} - p_{ij}$;
Tant que ($\exists\ (i, j)$ / d_{ij} non modifiée et $d_{i'j'}$ modifiée $\forall (i', j') \in S(i, j)$)
{ Choisir (i, j) ;

Chercher les successeurs de cette opération dans les deux listes ordonnées ;

Calculer $\sum_{(i',j') \in S(i,j)} p_{i'j'}(d'_{i''j''})\ \forall (i'', j'') \in S(i, j)$;

$$d'_{ij} \leftarrow min\left\{ d_{ij}, \min_{d'_{(i'',j'')}/(i'',j'') \in S(i,j)} \left\{ d'_{(i'',j'')} - \frac{\sum_{(i',j') \in S(i,j)} p_{i'j'}(d'_{i''j''})}{v_1 + v_2} \right\} \right\};$$

Pour (chaque $(i', j') \in S(i, j)$)
$\{d'_{ij} \leftarrow min\{d'_{ij}, b'_{i'j'} = d'_{i'j'} - p_{i'j'}\}\,;\}$
Insérer d'_{ij} et $b'_{ij} = d'_{ij} - p_{ij}$ dans les listes ordonnées ;

}

FIGURE – 3. Algorithme pour la modification des dates d'échéance.

Cas à $k = 2$ opérateurs

$\Delta \leftarrow 0$; $u \leftarrow m$; $v \leftarrow m$; $T \leftarrow b_m$; $P \leftarrow 0$;
Tant que $(\Delta < t_{r+1} - t_r)$
{
Tant que $(T = b_{u-1} + \Delta)$
{ $u \leftarrow u - 1$ (Si $u = 1$, $b_0 = -\infty$); $P \leftarrow P + \Delta$;}
Tant que $(T = b_{v+1})$
{ $v \leftarrow v + 1$ (Si $v = z$, $b_{z+1} = \infty$);}
$T_1 \leftarrow (\frac{(b_{u-1}+\Delta)(m-u+s)-(v-u+1)T)}{((m-u+s)-(v-u+1))}$; $T_2 \leftarrow b_{v+1}$; $T_3 \leftarrow T + \frac{(t_{r+1}-t_r-\Delta)(m-u+s)}{(v-u+1)}$;
$T' \leftarrow min\{T_1, T_2, T_3\}$; $P \leftarrow P + (v - u + 1)(T' - T)$; $\Delta \leftarrow \frac{P}{(m-u+s)}$; $T \leftarrow T'$;
}

FIGURE – 4. Algorithme pour la construction d'intervalles fixes.

Cas à $k = 2$ opérateurs

Soit $x_{ij}^{(r)}$ la quantité de l'opération (i, j) exécutée durant l'intervalle $[t_r, t_{r+1}]$.

$$x_{ij}^{(r)} = \begin{cases} \Delta, & q < u, \\ max\{0, T - b_q^{(r)}\}, & q \geq u, \end{cases}$$

avec T, u et v déterminés par l'algorithme en Figure 3, v étant l'indice tel que $b_v^{(r)} \leq T < b_{v+1}^{(r)}$ et q l'indice correspondant à l'opération (i, j) après ré-indexation.

Cas à $k = 2$ opérateurs

Calculer le degré intérieur de chaque opération (i, j) ;
Créer une file Q contenant les opérations de degré intérieur égal à 0 ;
Créer une liste A contenant les opérations disponibles pour traitement* ;
$r \leftarrow 0$;
Tant que $(Q \neq \varnothing)$
{ $r \leftarrow r + 1$; $t_r \leftarrow t_{r-1}$; $A \leftarrow A \cup Q$; $Q \leftarrow \varnothing$;
  Tant que $(A \neq \varnothing)$
  { $t_{r+1} \leftarrow t_r$;
    Exécuter l'algorithme en Figure 3 sur les opérations dans A ;
    Si (les valeurs trouvées (Δ, u, v, T) violent $b_j + \Delta \leq d_j$, $j = 1, \ldots, u - 1$)
    { Ré-exécuter l'algorithme en Figure 2 avec $t_{r+1} = t_r + \min_y \{p_j^y / j = 1, \ldots, u - 1\}$;
      $A \leftarrow A \backslash \{$opérations complétées dans$[t_r, t_{r+1}]\}$;
      Mettre à jour les degrés intérieurs des opérations ;
      Mettre à jour Q ; $A \leftarrow A \cup Q$; $r \leftarrow r + 1$;
    }
  }
}

N.B : Pour la construction d'intervalles variables, on procède de manière analogue à la différence des points suivants :

— $T \leq d_j$, $j = u, u + 1, \ldots, v$. Si $T = d_j$, $j \geq u$, le calcul s'arrête.
— Si u est décrémenté, on vérifie si $b_{u-1} + \Delta > d_{u-1}$, si c'est le cas, le calcul s'arrête.

* Une opération est dite disponible à la date t si le traitement de tous ses prédécesseurs est complété et s'il reste un temps de traitment non nul à effectuer sur cette opération.

FIGURE – 5. Algorithme pour la résolution de $F_m | res\ 2k_h1, S, v_h | L_{\max}$.

Cas à $k = 1$ opérateur

Tant que $(B \neq \varnothing)$
$\{$ Choisir (i,j) / $S(i,j) = \varnothing$ et $L_{ij} = \min\limits_{(i,j)\in B} \{P(B) - d_{ij}\}$;
$\quad B \leftarrow B \backslash \{(i,j)\}$;
$\}$

FIGURE – 6. Algorithme pour la résolution de $F_m|res\ 111, S|L_{\max}$.

Conclusion

- Problèmes très complexes.
- Nécessité de l'intégration de la composante humaine ainsi que ses caractéristiques.
- Étude de la complexité du problème.
- Méthodes polynomiales pour les cas à un et deux opérateurs.

Perspectives

- Domaine d'études largement ouvert.
- Nécessité de concevoir des modèles plus réalistes.
- Tester les heuristiques pour les cas avec $k \geq 3$.
- Étudier d'autres objectifs.

Références

- K. R. Baker, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints. *Operations Research*, 31(2) :381–386, 1983.
- P. Baptiste, V. Giard, A. Hait, and F. Soumis. *Gestion de production et ressources humaines*. Presses Internationales Polytechnique, 2005.
- J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Handbook on scheduling : From theory to applications*. Springer, 2007.

Références

- E. L. Lawler. Preemptive scheduling of precedence-constrained jobs on parellel machines. In *Deterministic and stochastic scheduling, Proceedings of the NATO advanced study and reasearch institute on theoretical approaches to scheduling problems*, pages 101–123. D. Reidel publishing Co., 1982.
- J. D. Ullman. Complexity of sequencing problems. In *Computer and Job/shop scheduling theory*. Wiley & Sons, Inc. New York, 1976.

Je vous remercie de votre aimable attention

Questions ?