

PRÉSENTATION DU LABORATOIRE

Ce laboratoire a deux objectifs :

- Se familiariser et maîtriser un logiciel d'ordonnancement «LEKIN».
- Appliquer les techniques et les algorithmes d'ordonnancement vus en classe et s'initier à d'autres techniques.

I. DÉROULEMENT DU LABORATOIRE

1ÈRE ÉTAPE : SE FAMILIARISER AVEC LE LOGICIEL «LEKIN»

- Voir DEMO_LEKIN.pdf dans MOODLE.
- Lancez le logiciel et essayez de faire des petits essais pour voir si vous avez compris le fonctionnement de ce dernier. Si vous éprouvez des difficultés, appelez le démonstrateur qui vous réexpliquera le fonctionnement de ce logiciel.

2ÈME ÉTAPE : PROBLÈME TYPE ET ALGORITHME JACKSON

- Le démonstrateur présentera au tableau l'algorithme JACKSON. Prenez des notes ! (L'annexe 2 fournit un résumé de cet algorithme).
- Résoudre le problème ci-dessous, intitulé « **Problème type :JobShop** » en utilisant cet algorithme. Le chargé du laboratoire vous accompagnera durant cette étape et vous aidera à trouver la solution et l'évaluer sous «LEKIN».

PROBLÈME TYPE : JOBSHOP

Dans un atelier d'usinage, on reçoit dix commandes (P1, P2 ... P10) à usiner sur les deux machines de l'atelier : la machine 'A' et la machine 'B'.

Le tableau ci-dessous résume les gammes opératoires de ces produits.

Gamme opératoire				
	1ère Opération		2ème Opération	
	Machine	Durée	Machine	Durée
P1	A	3		
P2	A	5	B	4

P3	A	3	B	7
P4	B	5		
P5	B	2	A	4
P6	B	4	A	4
P7	A	4		
P8	A	5	B	5
P9	B	2		
P10	B	6	A	3

- Déterminer l'ordonnancement qui minimise le temps moyen dans le système ?

3ÈME ÉTAPE : LES PROBLÈMES À REMETTRE

- Le travail à remettre portera sur deux problèmes : un du type FLOWSHOP et l'autre du type JOBSHOP. ***Vous devez remettre un seul word ou pdf avec toutes les réponses*** dans deux sections séparées qui s'intitulent respectivement « **Problème à remettre 1 : FlowShop** » et « **Problème à remettre 2 : JobShop** ». Le document word ou pdf remis doit s'intituler : NomPrenom1_NomPrenom2_Lab5. Vous devez inscrire à la page de garde vos noms, prénoms ainsi que vos matricules.
- Pour chacun de ces problèmes vous devez développer au **moins trois solutions** différentes dont l'une en utilisant l'algorithme approprié (expliquer votre choix et votre démarche) et les autres en utilisant les règles de priorité et les heuristiques proposées par «LEKIN» (SPT, FCFS, LPT). ***Pour chacune des solutions, vous devez fournir :***
 1. La règle ou l'heuristique utilisée
 2. La séquence obtenue
 3. Le diagramme de Gantt
 4. Et les mesures de performances sous forme de graphique
- Présenter votre choix de solution pour chaque problème en le justifiant.

II. PROBLÈMES À REMETTRE :

Problème 1 : FlowShop

On considère le problème d'ordonnancement décrit dans le tableau ci-dessous :

Jobs	Trimming (M1)	Drilling (M2)	Polishing (M3)	Testing (M4)
J1	3	6	6	5
J2	9	2	9	8
J3	8	8	5	4
J4	6	3	5	4
J5	9	2	9	8
J6	2	7	1	5

Toutes les tâches (J1 à J6) doivent passer par les quatre opérations (les 4 machines) et dans le même ordre.

On vous demande de trouver :

1. Un ordonnancement qui minimise le temps moyen dans le système (utiliser l'algorithme du *Campell, Dudek & Smith* (CDS), voir l'annexe 3).
2. Un ordonnancement qui minimise le makespan.
3. Le meilleur ordonnancement qui permet à la tâche de perçage (Drilling) de finir le plus tôt.

Problème 2 : JobSHOP

Dans ce problème, on dispose de deux machines '**A**' et '**B**' et on s'intéresse à l'usinage de 14 produits (P1 à P14) dont les gammes opératoires sont fournies dans le tableau ci-dessous. Certains doivent passer sur une seule machine, d'autres sur les deux et dans des ordres différents.

	Gamme opératoire			
	1ère Opération		2ème Opération	
	Machine	Durée	Machine	Durée
P1	A	3		
P2	A	5	B	4

P3	A	2	B	7
P4	B	5		
P5	B	2	A	4
P6	B	4	A	4
P7	A	4		
P8	A	5	B	5
P9	A	3	B	4
P10	B	2	A	3
P11	B	2		
P12	B	6	A	3
P13	A	5		
P14	B	7	A	8

On vous demande de trouver :

4. Un ordonnancement qui minimise le temps moyen dans le système.
5. Un ordonnancement qui minimise le makespan.
6. Un ordonnancement qui permette à la machine 'A' de finir le plus tôt possible.

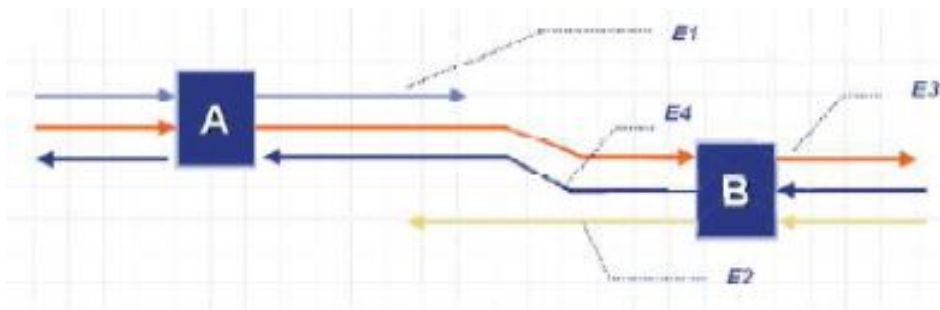
ANNEXES

ANNEXE 1 : DÉFINITIONS

- Problème d'ordonnancement de type **Flow-shop** : C'est le cas où tous les produits suivent la même séquence de production (atelier à cheminement unique).
- Problème d'ordonnancement de type **Job-shop** : C'est le cas où les produits suivent des séquences de production différentes (atelier à cheminement multiple).
- **Makespan** : c'est la date de fin de l'ordonnancement. Soit C_i la date de fin d'exécution du job i . Alors, le **makespan** (C_{max}) est calculé de la manière suivante : $C_{max} = \max\{C_i\}$.

ANNEXE 2 : ALGORITHME DE JACKSON

Pour ordonnancer n tâches dans une organisation **jobshop** à deux machines A et B avec l'algorithme de Jackson, suivre la démarche suivante :



Étape 0. Définir les ensembles suivants :

- E1 : ensemble de tâches comportant une seule opération sur A (gamme: A)
- E2 : ensemble de tâches comportant une seule opération sur B (gamme: B)
- E3 : ensemble de tâches comportant une première opération sur A et une deuxième opération sur B (gamme: A-B)
- E4 : ensemble de tâches comportant une première opération sur B et une deuxième opération sur A (gamme: B-A)

Étape 1. Ordonnancer les tâches de E3 avec Johnson

Étape 2. Ordonnancer les tâches de E4 avec Johnson

Étape 3. Exécuter les tâches sur A dans l'ordre suivant : E3, E1, E4

Étape 4. Exécuter les tâches sur B dans l'ordre suivant : E4, E2, E3

ANNEXE 3 : ALGORITHME DE CDS^{1,2}

Cette heuristique consiste simplement à générer $m-1$ sous problèmes de type **FlowShop à 2 machines**, à les résoudre et à sélectionner la meilleure solution.

Le sous problème k est défini par :

- *processing time* sur la machine virtuelle 1 : $p'_{i,1} = \sum_{j=1}^k p_{i,j}$
- *processing time* sur la machine virtuelle 2 : $p'_{i,2} = \sum_{j=m+1-k}^m p_{i,j}$
- Pour chacun de ces problèmes, on calcule l'ordre optimal avec l'algorithme de Johnson et on applique alors cet ordre sur le problème de base pour obtenir la(es) performance(s) $P(k)$.
- Ensuite, il suffit de choisir le meilleur sur l'ensemble des $P(k)$.

EXEMPLE D'UTILISATION DE L'ALGORITHME DE CDS

Soit le problème *flowShop* suivant :

Job	Machine (M1)	Machine (M2)	Machine (M3)
J1	8	7	6
J2	7	3	10
J3	4	6	8
J4	5	2	9

- L'objectif est de trouver la(es) séquence(s) d'ordonnancement qui minimise(nt) le *makespan* et le temps moyen passé dans le système.
- **Itération 1 (k=1)**. On applique l'algorithme de Johnson sur **M1** et **M3**. On trouve la séquence d'ordonnancement suivante : {J3;J4;J2;J1}. Cette séquence doit s'appliquer sur le problème *flowShop* initial de trois machines M1, M2 et M3. En utilisant Lakin, on trouve $C_{max} = 43$ et $\bar{F} = 31.25$.
- **Itération 2 (k=2)**. On construit deux machines virtuelles M1' et M2' de telle façon que le temps opératoire de chaque job sur M1' est égal à la somme des temps opératoires sur M1 et M2, alors que le temps opératoire sur M2' est égal à la somme des temps opératoires sur M2 et M3:

¹ Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management science*, 16(10).

² <http://drupal.mgi.polymtl.ca/?q=book/export/html/57>

Job	Machine (M1')	Machine (M2')
J1	15	13
J2	10	13
J3	10	14
J4	7	11

- En appliquant l'algorithme de Johnson sur **M1'** et **M2'**, on trouve la séquence d'ordonnement suivante : {J4;J2;J3;J1}. Cette séquence doit s'appliquer sur le problème *flowShop* initial de trois machines. En utilisant Lakin, on trouve $C_{max} = 40$ et $\bar{F} = 29$.
- Il est clair donc que la deuxième itération permet d'avoir des meilleurs résultats pour minimiser le makespan et le temps moyen passé dans le système. Donc la séquence d'ordonnement qui doit s'appliquer sur les trois machines M1, M2 et M3, selon l'algorithme de *Campell-Dudek-Smith* est: {J4;J2;J3;J1}.