II : Ordonnancement « Gestion » en Atelier spécialisés

Pour un ordonnancement « Gestion » au sein des Atelier spécialisés, on a deux Modèles à prévoires :

* **Modèles statique** Pour lesquelles on cherches l'ordonnancement (Gestion) optimal d'un ensemble donné de tache sur une Période donnée

* **Modèles dynamiques** : se caractérisent par des arrivées successives de taches, le plus souvent dans un ~~...~~ univers aléatoires.

-I) Ordonnancement suivant la Règle du Temps opératoire minimum (T.O.M)

* Ordonnancement sur une Machine « A » (Centre de Production)

Ordonnancement sur la machine (centre de Production) A

On a : 5 tâches sur la machine

tableau (1) T. opératoire (en Centième d'heures)

tâche (i)	1	2	3	4	5
t_i temps Opératoire	50	150	80	200	30

Il est claire que quel-que soit l'ordre choisi, le temps opératoire total est le même, alors il faudra donc définir un autre Critère de choix entre tous les ordonnancement Possible.

Tableau (2) un ordonnancement Possible

ordre (J)	1	2	3	4	5
tache Programm (i)	3	4	1	5	2
T_J Temp opérateur	80	200	50	30	150

* **Digramme de GANTT** : Illustant tout d'abord une Technique de Visualisation d'un ordonnancement. (Gestion) le diagramme de Gantt

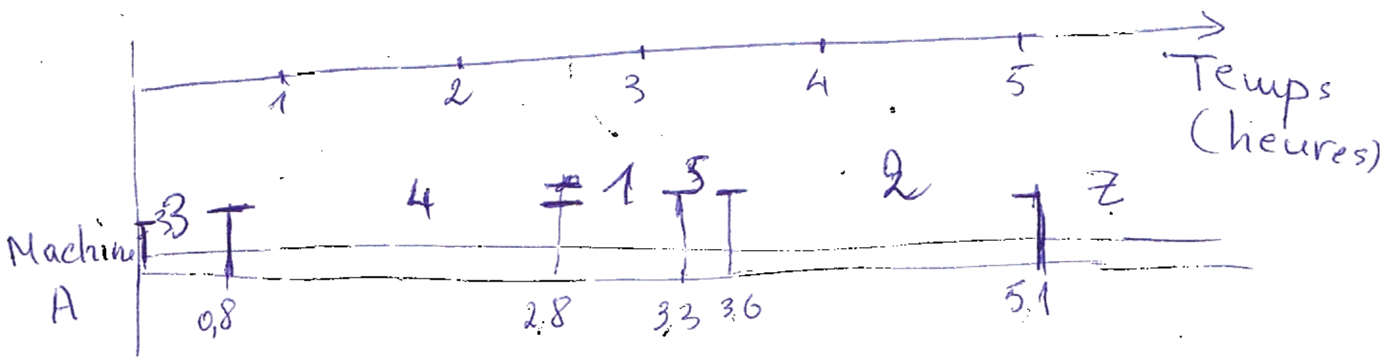


Diagramme de Gantt

il permet de visualiser à la fois :

- l'utilisation des Moyens productifs
- l'avancement de l'execution de tâches

On a alors donc, - 5 taches sur la machine A. avec 510 Centième d'heure

* Notons A_j le temps d'achevement de la tache programée en Position "J", alors $A_4 = T_1 + T_2 + T_3 + T_4$

C.a.d

ordre (J)	1	2	3	4	5
T_j	80	200	50	30	150
A_j	80	280	330	360	510

Tableau 3a Temps d'achevement des Taches

Le temps d'achevement Moyen vaut Alors

$$\bar{A} = \frac{80 + 280 + 330 + 360 + 510}{5} = 312$$

En general :

$$\bar{A} = \frac{1}{5}\sum_{j=1}^{5} A_j = \frac{1}{5}\Big[T_1 + (T_1+T_2) + (T_1+T_2+T_3) + (T_1+T_2+T_3+T_4) + (T_1+T_2+T_3+T_4+T_5)\Big] = \frac{1}{5}\Big[5T_1 + 4T_2 + 3T_3 + 2T_4 + T_5\Big]$$

il s'agit donc d'une somme pondérée des Temps operatoire, on remarque que chaque temps operatoire etant pendérée par un facteur d'autant plus grand qu'il se trouve exécuté plus tôt dans l'ordonnancement alors pour minimiser le temps d'achevement Moyen est cel du temps operatoire minimum ~~alors~~ il s'agit d'executer les taches per ordre croissant de durée :

$$T_1 \leq T_2 \leq \cdots \leq T_j \leq \cdots \leq T_n$$

l'application de cette règle donne $\bar{A} = 218$ (voir Tableau B)

ordre (j)	1	2	3	4	5
taches (i)	5	1	3	2	4
T_j	30	50	80	150	200
A_j	30	80	160	310	510

Tableau B

on peut Montrer que la regle T O M revient à minimiser le retard Moyen, le retard d'une tache étant la différence entre le Moment où la tache est terminée et celui où elle aurait été terminée si l'on l'avait commencé en premier lieu.

* Ordonnancement avec deux centre de Production

* Chaque tache nécessite pour son exécution le passage sur deux Machine, les Machines "A" et "B"
Soient t_{iA} et t_{iB}. le temps d'exécution de la tache "i" sur A et B. on va utilisé comme Critère d'ordonnancement. La minimisation du Temps total d'execution des Taches sur les deux Machines. On va distinguer deux Cas

- le cas où toutes les taches sont à exécuter sur A puis B;
- le cas où toutes les taches n'ont pas le même ordre de passage sur les deux Machines.

* Cas où toutes les tâches sont exicutes sur "A" puis "B"

tache(i)	1	2	3	4	5
t_{iA}	50	150	80	200	30
t_{iB}	60	50	150	70	200

Tableau : ord. sur deux Machines

~~Pour~~ · la méthode optimal pour minimiser le temps total.

Remarque : On a interêt a mettre en tête la tâche de Temps t_{iA} le plus faible, de façon similaire, lors de l'execution de la dernière tâche sur la machine B la machine A doit. On a donc interêt a mettre en fin la tâche de durée d'execution t_{iB} minim.

- Pour cela On doit se basee sur (l'algorithme de Johnson) Qui calcule l'ordonnancement minimisant le temps total d'execution des tâches :

1- Rechercher la tache "i" de temps d'execution t_{im} minimum

2- Si m=A placer cette tâche à la premiere place disponible
 Si m=B " —— " —— " ~~à la~~ dernière —"—"

3- supprimer la tâche "i" des tâches encore à programmer
 Retour en "1"

alors On a : [5 ($t_{5A}=30$) première position, puis 1 ($t_{1A}=50$)
, 2 ($t_{2B}=50$) est mise en derniere position, puis la tâche 4 ($t_{4B}=70$) en avant dernière, Enfin la tâche 3 est mise à la dernière place disponible.]

alors on à l'ordre : $\boxed{5 \mid 1 \mid 3 \mid 4 \mid 2}$

On obtient le graphique de Gantt à la figure 2-2 où le passage d'une tâche d'une machine à l'autre est réalisé à l'aide d'une flèche verticale.

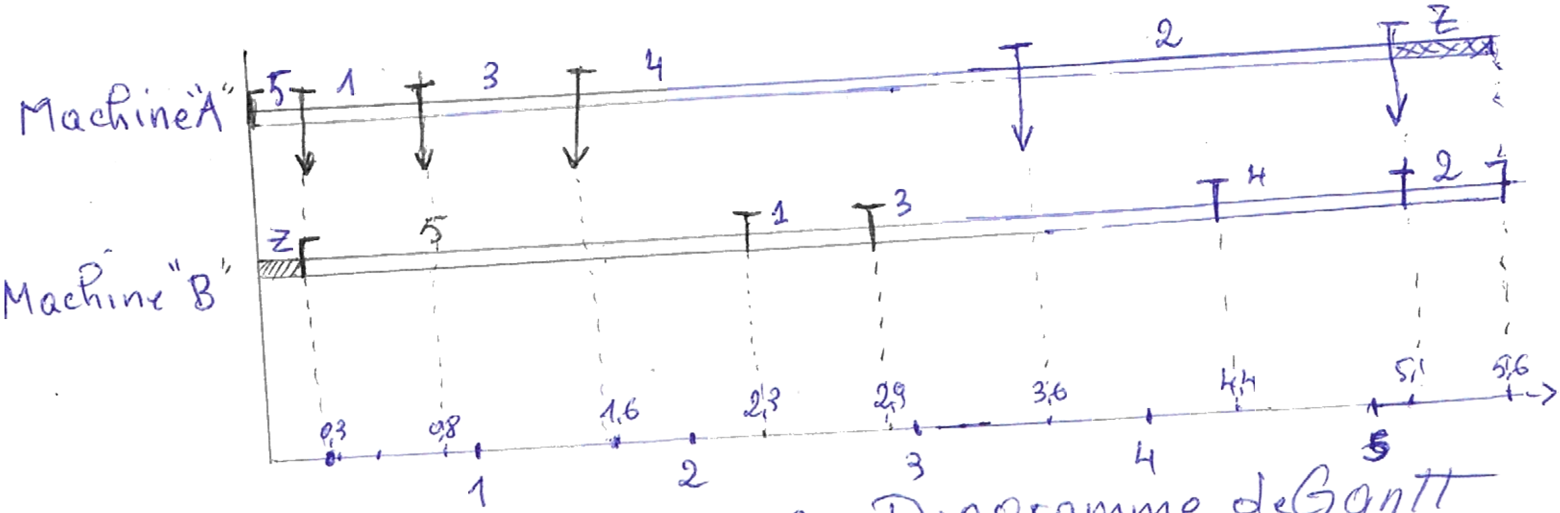


Fig 2-2 Diagramme de Gantt

N.B: Z : la machine dort, $T_{total} = 5{,}6$, $t_{tA} = 5{,}1$.

Cas de tâches ne s'éffectuant pas dans le meme ordre

dans le cas plus general plus général, certaines taches ne necessitent que le passage sur une machine, d'autre sur les deux dans un ordre ou d'autre, les données numeriques sont reprises sur le tableau 2.3

taches à effectuer sur A puis B

taches(i)	1	2	3	4	5	6
t_{iA}	50	80	10	50	30	70
t_{iB}	30	60	30	0	0	0

taches à effectuer sur B puis A

taches(i)	7	8	9	10	11
t_{iB}	90	20	10	40	10
t_{iA}	70	30	100	0	0

Tableau 2.3 illustration de l'algorithme de Jackson

* l'ordonnancement qui minimise le temps total d'execution des taches sur les deux Machines est obtenu par l'algorithme de Jackson qui est une généralisation de Johnson.

il consiste tout simplement a:

1) faire une partition de l'ensemble de "n" tache en:

- l'ensemble A de taches ne passant que sur A: $\mathcal{A}=\{4,5,6\}$
- 〃 — 〃 $\mathcal{B}$ 〃 — 〃 〃 — B: $\mathcal{B}=\{10,11\}$
- 〃 — $\mathcal{AB}$ 〃 de taches 〃 passant 〃 sur A puis B: $\mathcal{AB}=\{1,2,3\}$
- 〃 — $\mathcal{BA}$ = — 〃 — 〃 — sur B puis A: $\mathcal{BA}=\{7,8,9\}$

2) calculer un ordonnancement pour chaque sous-ensemble

- l'ordonnancement optimal pour $\mathcal{AB}$ par Johnson: 3,2,1
- 〃 — 〃 — 〃 $\mathcal{BA}$ 〃 — 〃: 9,8,7
- un ordonnancement arbitraire pour $\mathcal{A}$ (par exemple TOM): 5,4,6
- 〃 — 〃 — 〃 — $\mathcal{B}$ (— 〃 — 〃): 11,10

3) Remarquons que l'on a intérêt à obtenir le plus vite possible sur "A" les taches qui doivent ensuite aller sur B et à mettre en dernière place sur A celle qui doivent d'abord aller sur B, ceci conduit à combiner les ordonnancement de la manière suivante:

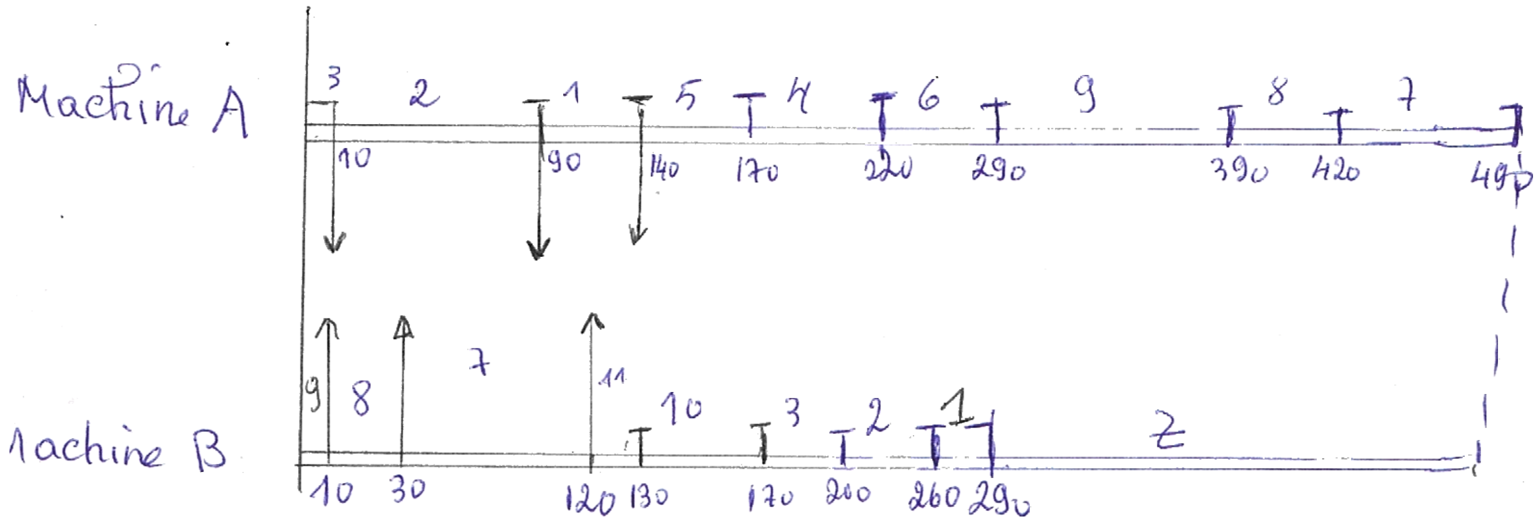
- Pour la machine A: la sequence optimal pour le sous ensemble $\mathcal{AB}$ puis les taches de $\mathcal{A}$ puis la sequence optimale du sous-ensemble $\mathcal{BA}$

3, 2, 1, 5, 4, 6, 9, 8, 7

• Pour la Machine B : la sequence optimale pour le sous ensemble BA, puis ~~la sequence~~ les taches de B, puis la séquence optimale du sous-ensemble AB

9.8.7.11.10,3,2,1

On obtient le diagramme de Gantt (fig.2.4)



(Fig.2.4) Algorithme de Jackson

* Ordonnancement sur Trois Machines

l'algorithme de Johnson ne s'applique qu'en présence de deux machines. Cependant, le cas de Trois machines peut se ramener au cas de deux machines si la machine B est complètement dominée par la machine A ou par la machine C c-a-d. si l'on se trouve dans le cas où

minimum $t_{iA} \geq$ maximum t_{iB}

soit dans le cas où

Illustrons ceci sur l'exemple du Tableau 2.4, où l'on constate que minimum $t_{iA} = 12 =$ maximum t_{iB}

On est donc bien dans les conditions d'applications énoncées ci-dessus.
Remarquez qu'il suffit qu'une des deux conditions soit vérifiée. Ainsi dans l'exemple, la seconde condition n'est pas vérifiée et l'algorithme s'applique bien.

tache	1	2	3	4	5	6	7
Assemblage	20	12	19	16	14	12	17
Inspection	4	1	9	12	5	7	8
Expédition	7	11	4	18	8	3	6

Tableau 2-4 temps opératoire avec 3 Machines

lorsque l'on se trouve dans un des deux cas on reformule alors le problème en un problème à deux machines, la première groupant les M A et B ($t_{iAB} = t_{iA} + t_{iB}$), et la seconde groupant les M B et C ($t_{iBC} = t_{iB} + t_{iC}$)

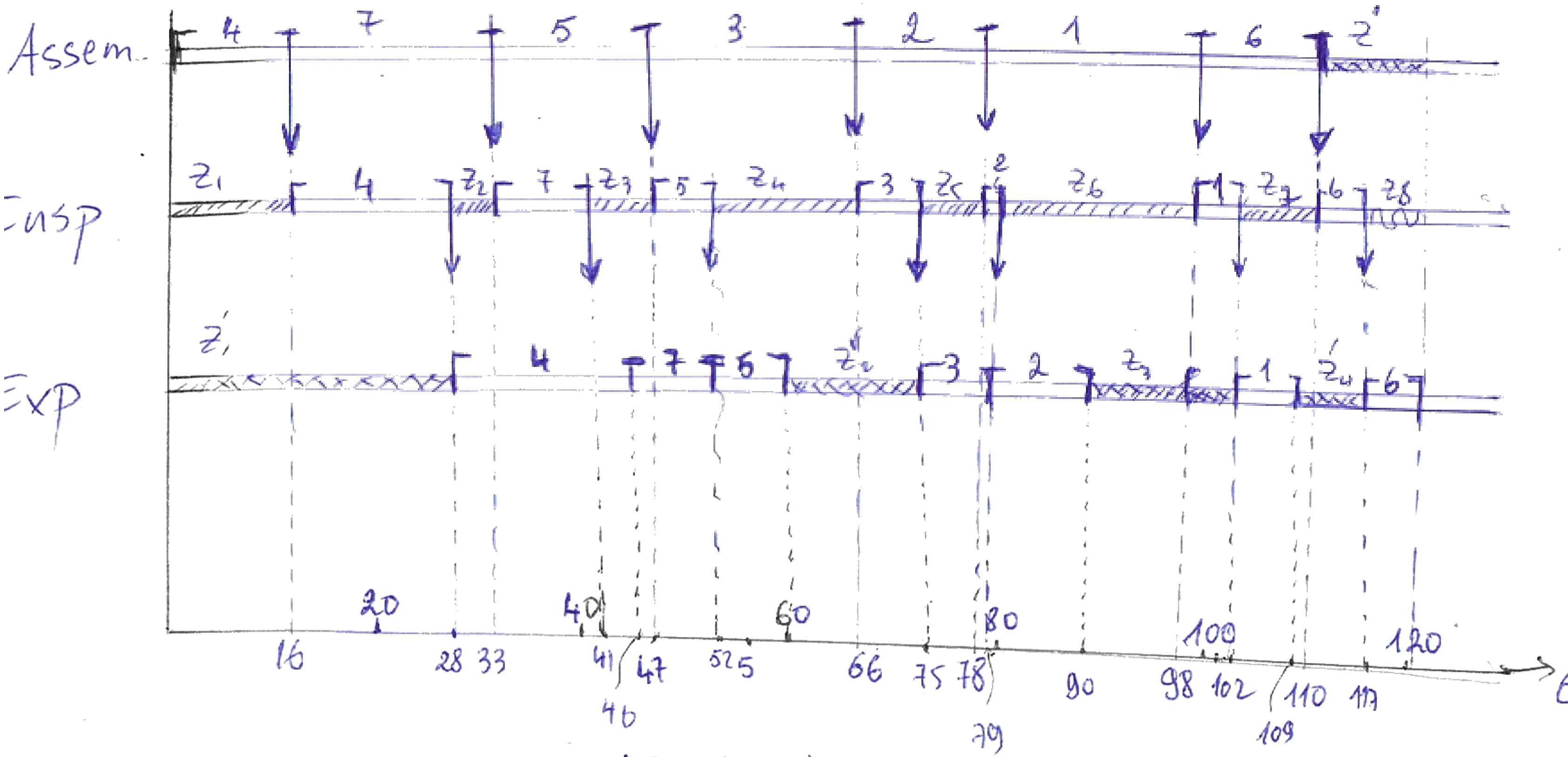
tache	1	2	3	4	5	6	7
Assemblage + inspection	24	13	28	28	19	19	25
Inspection + Expedition	11	12	13	30	13	10	14

On applique alors l'algorithme de Johnson à ce Problème à deux machine, pour déterminer l'ordonnancement optimal.

Place	1	2	3	4	5	6	7
tache	4	7	5	3	2	1	6

On peut alors tracer le diagramme de Gantt Correspondant au problème original, C'est-à-dire Celui avec Trois machines (voir fig. 2.5)

dans le cas où la machine Central n'est pas dominée par la première où la Troixième, le problème Peut être modélisé Comme un problème en nombres entiers et resolu par une technique de Progra-mmation en nbre entiers telle que la méthode dit "Branch and bound"



Temp Total de l'ordon. egale a 120.

Fig (2-5) Ordonnancement avec 3. machines

Remarque: 1- $\sum z_{insp} = z_1 + z_2 + z_3 + z_4 + z_5 + z_{6} + z_{7} + (z_8) = 16 + 5 + 6 + 14 + 3 + 19 + 8 + 3 = 74$

Pour l'inspection on a 61,66% du Temp mort, reste = 38,33% de travail

2- pour Expedition : la même chose

$\sum z_{Exp} = 63$, $\{ \sum z_{Exp}\% = 52,5\%$, T de travaile %, 47,5% $\}$

3- temp d'achevement pour la tâche 2 par exemple - insp = 78, Exp = 90,

* # Ordonnancement de "n" taches sur "n" Centres de Production

le problème en générale ~~qui est form~~ a été formalisé en termes de programmation dynamique et en linéaire en nbre entiers, On doit etudier le probleme des contraintes supplementaires comme la date de livraison une capacité de production ... etc.

lorsque l'ordre de passage des taches est identiques et que le nbre de Centre de production ne dépasse pas quelque dizaines une solution souvent proche de l'application de l'algorithme de Johnson sur des ~~reg~~ groupements de Centre de productions successifs, exactement à le manière précedente pour le cas de Trois machines C.a.d. 3 Centre de production dont celui du Milieu est dominé. En arrive enfin a une methode heuristique donnant une solution approchée

Exemple : prenons le cas de 5 Centres de production Notés A.B.C.D et E il faut resoudre les 4 Problèmes suivants par l'algorithme de Johnson

- avec la premiere et la dermère machine
  $\{A\} \rightarrow \{E\}$
- $\{A,B\}$ — $\{D,E\}$
- $\{A,B,C\}$ — $\{C,D,E\}$
- $\{A,B,C,D\}$ — $\{B,C,D,E\}$

- On prend alors le meilleur des temps totaux d'execution des taches ainsi Trouvés,

Prenons (Illustrons) ceci sur un exemple à 4 Centre de production. le tableau (2.5) represente les données du Probleme.

tâches	t_{iA}	t_{iB}	t_{iC}	t_{iD}
1	50	43	15	4
2	89	99	95	77
3	7	47	20	98
4	8	64	12	94
5	61	19	65	14
6	1	80	66	78

~~Temps optimo~~

Tableau (2.5) temps opératoires avec quatre Machines

- le premier problème fictif consiste à ne considérer que les machines A et D, il conduit à l'ordonnancement suivant la Méthode de Johnson;

1. 2. 3. 4. 5. 6

6	3	4	2	5	1

Qui conduit à un temps de 51,2 heures

le deuxième problème fictif consiste à considérer les machines A+B et C+D, comme illustre au tableau 2.6).

tâches	$t_{iA} + t_{iB}$	$t_{iC} + T_{iD}$
1	50+43=93	15+4=19
2	89+99=188	95+77=172
3	7+47=54	20+98=118
4	8+64=72	12+94=106
5	61+19=80	65+14=79
6	1+80=81	66+78=144

tableau 2.6. deuxième problème fictif

Methode de Johnson

1	2	3	4	5	6
3	4	6	2	5	1

qui conduit à un temps de 48,7 heurs

Troixieme et dernier probleme fictif.

A+B+C et B+C+D, il donne la meme solution avec 48,7 heurs, et avec tous les ordonnancement possible : a 48,5 heures