

TD série N°3: projet de montage d'un respirateur multi-modules

Christoph Samuel

I-Position du problème :

Le sujet porte sur la gestion de projet pour la **fabrication d'un respirateur** multi-modules dans une **usine du groupe** Industriel Air Liquide. Le projet doit être mené **en urgence** et la fabrication doit être intégralement automatisée. Le processus de fabrication implique **17 tâches** primitives codées A, B, C ... P, Q qui doivent être exécutées dans un **intervalle de temps très court**. En examinant le processus **d'exécution du cycle**, il m'est fourni le tableau suivant:

Code tâches	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Durée	7	7	10	3	5	1	5	4	8	12	7	1	6	1	5	7	2
Tâches antérieures	G M	H	E N Q	-	H	B P	-	G M	«D+3»	A L	«*+2»	K O	I	«J*+3»	G M	E N Q	«H*+2»

« * + n » signifie que la tâche ne peut commencer que n unités de temps après le lancement du cycle

« T* + n » signifie que la tâche ne peut commencer que n unités de temps après le début de T

« T + n » signifie que la tâche ne peut commencer que n unités de temps après la fin de T

L'**objectif de ce troisième TD** est de résoudre, dans la pratique de certaines techniques de construction de **graphes** introduites en cours, un **problème concret d'ordonnancement**. Ce troisième problème me permet d'aborder la **notion de chemin critique** d'un graphe.

Un chemin critique désigne la **liste ordonnée des opérations** nécessaires pour obtenir le résultat voulu, dont la **durée totale** donne la durée du projet. C'est donc aussi, parmi les **différents chemins** constitués par les tâches, **le plus long chemin obtenu**.

L'un des **problèmes de l'ingénieur** consiste à s'assurer que le **montage respecte** bien la **cohérence d'ordonnancement** du modèle. Dans le cadre de ce TD, j'ai décidé de m'aider de JGraphT, une bibliothèque Java implémentant l'**algorithme PERT** et qui propose une **interface de création et traitement de graphes**. Elle permet entre autres de **calculer le chemin critique du modèle MPM**. Je me suis également servi de **yEd Graph Editor**, une application me permettant de créer des graphes, pour une représentation graphique plus adéquate **du modèle MPM**.

II-Réalisation :

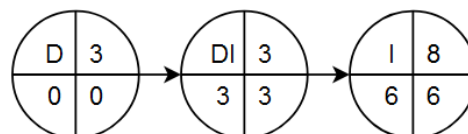
Lors de la **fabrication en urgence** de respirateur multi modules, il est **primordial** pour l'usine de s'assurer que le temps de **production soit minimum**. Le problème à résoudre est **un problème typique d'ordonnancement**. Ainsi, je privilégierais un **modèle de graphe** permettant d'établir une stratégie de **chemin critique**. Pour résoudre ce problème, je vais le visualiser à l'aide d'un **modèle de réseau MPM**. Pour cela, je vais réaliser les étapes suivantes:

- 1- Déterminer les différents niveaux de tâches.
- 2- Modéliser le réseau MPM pour le cycle de fabrication.
- 3- Calculer à base de ce modèle la date la plus précoce de démarrage de chaque tâche.
- 4- En déduire de ce qui précède la durée optimale du cycle de fabrication.
- 5- Calculer la date la plus tardive de démarrage de chaque tâche.
- 6- Indiquer les marges totales qu'on peut observer pour chaque tâche.

1-Détermination des différents niveaux de tâches

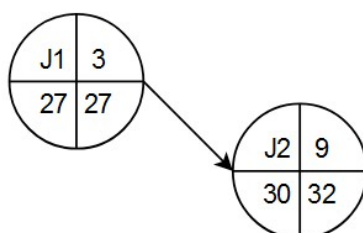
Afin de déterminer les différents **niveaux de tâches** du **modèle de réseau MPM**, il faut résoudre les problèmes de **tâches antérieures** des tâches I, K, N et Q. Il en découle la création des **nouvelles tâches** suivantes ainsi que la création du nouveau **tableau de précédence des tâches**:

Pour résoudre le problème « ***+2** », il faut initialiser la date de début au plus tôt de K à la valeur 2.
Pour le problème « **D+3** », il faut introduire une tâche fictive DI de durée 3 postérieure à D et antérieure à I

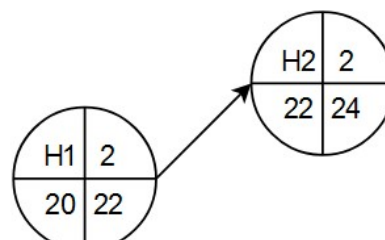


nouvelle tâche via « **D+3** »

Pour résoudre le problème « **J*+3** », il faut subdiviser la tâche J en 2 sous-tâches, J1 de durée 3 et J2 qui dure le reste, soit $12-3 = 9$. La tâche J1 sera antérieure à J2, ce qui était postérieure à J sera maintenant postérieure à J2 et N sera postérieure à J1.



nouvelle tâche via « **J*+3** »



nouvelle tâche via « **H*+2** »

Les différents niveaux sont définis successivement:

Niveau initial

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 1: D, G, K

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 2: DI

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 3: I

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 4: M

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 5: A, H1, O

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 6: H2, L, Q

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 7: B, E, J1

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 8: J2, N

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 9: C, P

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Niveau 10: F

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

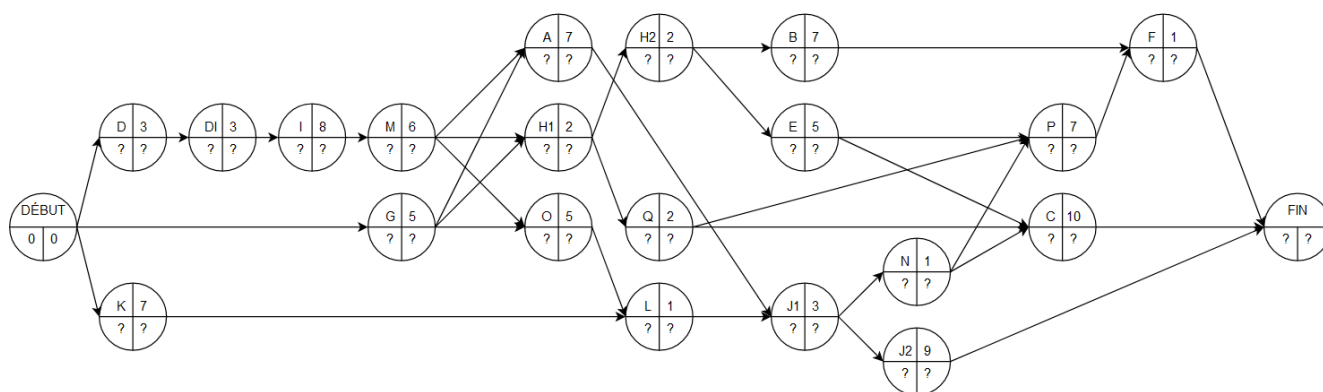
Niveau final

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
-------------	---	---	---	---	----	---	---	---	----	----	---	----	----	---	---	---	---	---	---	---

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
Tâches antérieures	G M	H2	E Q N	-	D	H2	B P	-	G M	H1	DI	A L	J1	-	K O	I	J1	G M	E N Q	H1
Durée	7	7	10	3	3	5	1	5	2	2	8	3	9	7	1	6	1	5	7	2

2- Modélisation du réseau MPM pour le cycle de fabrication

Le modèle proposé est un **graphe orienté** composé de **20 nœuds** et de **33 arcs**. Il est présenté sous la forme d'un **réseau MPM provisoirement incomplet**, tel que:



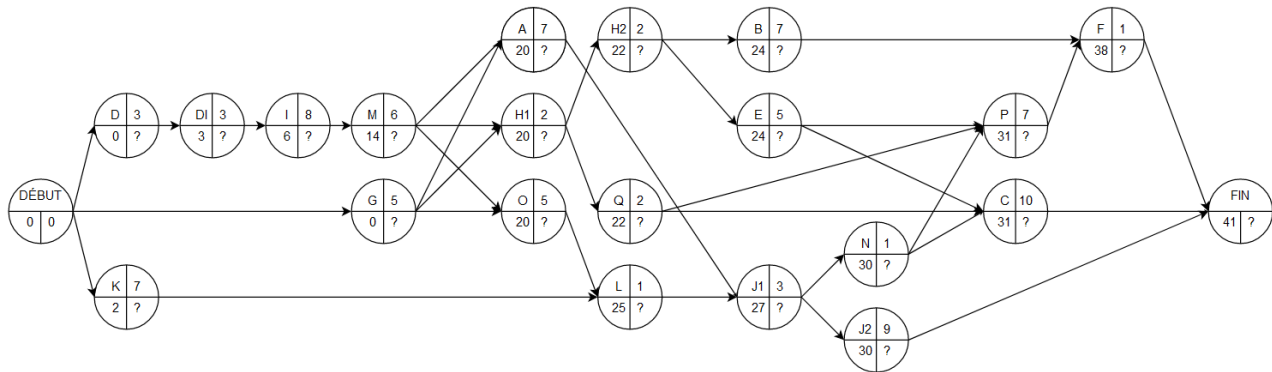
modèle MPM provisoirement incomplet

3- Calcul de la date la plus précoce de démarrage de chaque tâche

Dans le réseau MPM, la **date de démarrage au plus tôt** d'une tâche T est obtenue en calculant la **longueur du plus long chemin entre le sommet** représentant la **tâche DEBUT** et le sommet correspondant à une **tâche T**. Les résultats de **l'application de cette méthode** sont résumés dans le tableau suivant:

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
Tâches antérieures	G M	H2	E Q N	-	D	H2	B P	-	G M	H1	DI	A L	J1	-	K O	I	J1	G M	E N Q	H1
Durée	7	7	10	3	3	5	1	5	2	2	8	3	9	7	1	6	1	5	7	2
Date au plus tôt	20	24	31	0	3	24	38	0	20	22	6	27	30	2	25	14	30	20	31	22

Après **ajout des valeurs obtenue**, j'actualise mon **modèle MPM**, en mettant à jour les dates au plus tôt pour chaque tâche, j'obtiens le **nouveau modèle MPM** définit ci-dessous.



modèle MPM date au plus tôt

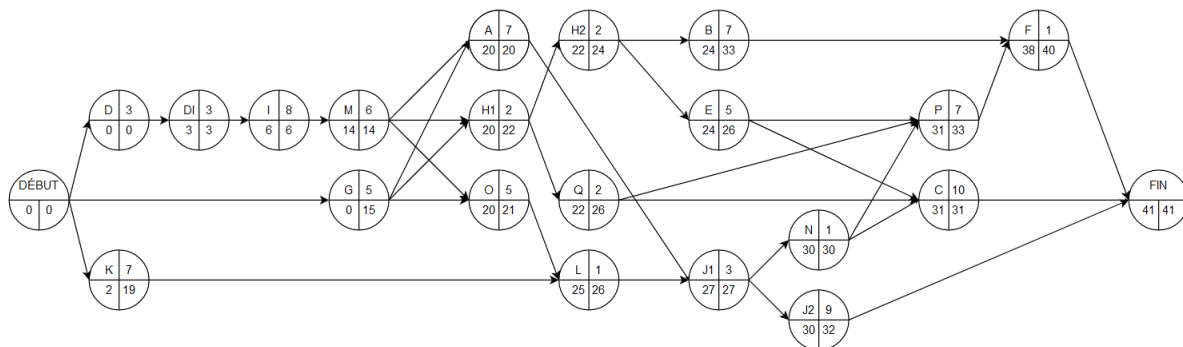
4- Détermination de la durée optimale du cycle de fabrication

De ce calcul, on peut **déduire la durée optimale** du cycle d'assemblage en calculant **la date au plus tôt de la tâche FIN**: $\text{date} + \text{tôt}(\text{FIN}) = 41$

5- Calcul de la date la plus tardive de démarrage de chaque tâche

Dans le réseau MPM, la **date de démarrage au plus tard** d'une tâche T est obtenue en calculant la **différence entre la durée optimale** du cycle et la **longueur du plus long chemin** entre le sommet correspondant à **une tâche T et le sommet FIN**. Les résultats de l'application de cette **méthode** sont résumés dans le tableau qui suit:

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
Tâches antérieures	G M	H2	E Q N	-	D	H2	B P	-	G M	H1	DI	A L	J1	-	K O	I	J1	G M	E N Q	H1
Durée	7	7	10	3	3	5	1	5	2	2	8	3	9	7	1	6	1	5	7	2
Date au plus tard	20	33	31	0	3	26	40	15	22	24	6	27	32	19	26	14	30	21	33	26



modèle MPM complet

En effet, après le lancement de **mon programme java** (Montage_respirateur_170423.java), le terminal via ApacheNetBeans m’affiche le **chemin critique du montage** ainsi que la **valeur de la date au plus tard** du cycle de fabrication:

```
-----< com.mycompany:MontageRespirateur >-----
Building MontageRespirateur 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ MontageRespirateur ---
Le chemin critique est :
DEBUT -> D -> DI -> I -> M -> A -> J1 -> N -> C -> FIN

Date au plus tard de la tâche FIN :
41

BUILD SUCCESS

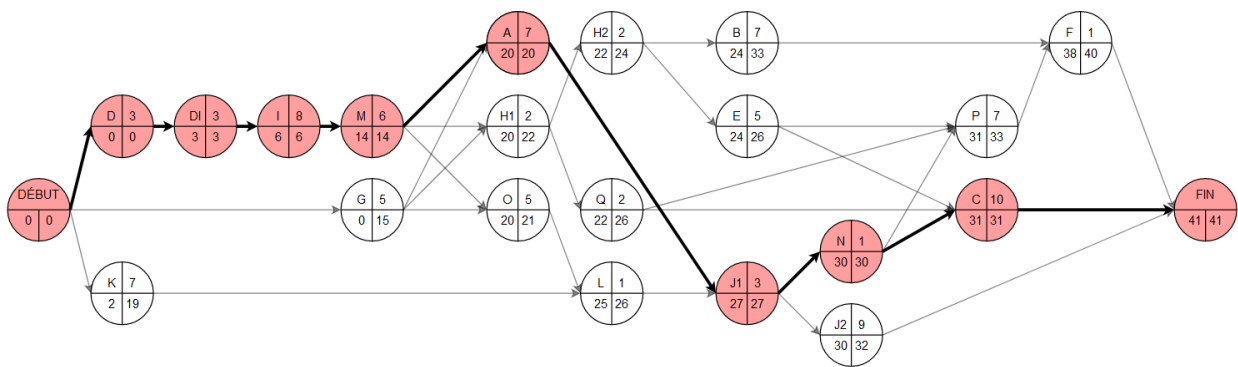
Total time: 1.140 s
Finished at: 2023-04-17T19:39:57+02:00
-----
```

terminal ApacheNetBeans

Les **tâches critiques** sont toutes les tâches T telles que : **date+tôt(T) = date+tard(T)**. D’après les **résultats obtenus** via les questions précédentes, il en ressort le tableau suivants, parmi lequel, les **tâches critiques sont colorées en rouge**.

Code tâches	DEBUT	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q	FIN
Durée	0	7	7	10	3	3	5	1	5	2	2	8	3	9	7	1	6	1	5	7	2	0
Date au plus tôt	0	20	24	31	0	3	24	38	0	20	22	6	27	30	2	25	14	30	20	31	22	41
Date au plus tard	0	20	33	31	0	3	26	40	15	22	24	6	27	32	19	26	14	30	21	33	26	41

Une fois la certitude des valeurs fournies par mon programme, codant le **modèle MPM** et le faisant tourner dans **l’algorithme PERT** ainsi que des valeurs de **dates au plus tôt et au plus tard** obtenues, je réalise le graphe avec le **chemin critique** afin de visualiser aux mieux les résultats.



modèle MPM chemin critique

Le **chemin critique** est le chemin traversant les sommets «**tâches critiques**». Dans notre cas le chemin est le suivant: DEBUT → D → DI → I → M → A → J1 → N → C → FIN. Sa longueur détermine la **durée optimale du cycle d'assemblage**.

6- Présentation des marges totales qu'on peut observer pour chaque tâche

On appelle **marge d'une tâche** le **retard** qu'il est **possible de tolérer** dans sa réalisation **sans affecter la durée optimale** prévue du projet. Les marges des tâches composant le **chemin critique** sont **nécessairement nulles**, il existe **deux types de marges**:

- «**Marge Totale**» est de mesurer le degré de liberté disponible pour **retarder le démarrage** d'une tâche, ou allonger sa durée sans affecter la durée optimale du projet.

$$Mt = \text{Date+tard}(T) - \text{Date+tôt}(T)$$

- «**Marge Libre**» d'une tâche T et on note ML(T) le **retard admissible** dans sa réalisation **sans modifier les dates au plus tôt** des tâches suivantes **sans allonger la durée optimale du projet**.

$$ML(T) = \min [\text{date+tôt}(Si) - \text{date+tôt}(T)] - \text{durée}(T), \text{ avec } Si = \text{tâches succédant à } T$$

A l'aide des **définitions et des résultats précédents**, j'obtiens le tableau suivant qui me fournit les **marges totales et les marges libre** de toute les **tâches**.

Code tâches	A	B	C	D	DI	E	F	G	H1	H2	I	J1	J2	K	L	M	N	O	P	Q
Tâches antérieures	G M	H2	E Q N	-	D	H2	B P	-	G M	H1	DI	A L	J1	-	K O	I	J1	G M	E N Q	H1
Durée	7	7	10	3	3	5	1	5	2	2	8	3	9	7	1	6	1	5	7	2
Date au plus tôt	20	24	31	0	3	24	38	0	20	22	6	27	30	2	25	14	30	20	31	22
Date au plus tard	20	33	31	0	3	26	40	15	22	24	6	27	32	19	26	14	30	21	33	26
Marge totale	0	9	0	0	0	2	2	15	2	2	0	0	1	17	1	0	0	1	2	4
Marge libre	0	7	0	0	0	2	2	15	0	0	0	0	2	16	1	0	0	0	0	4

III-Bilan/Conclusion :

Sur le plan de l'**application des modèles** et algorithmes de recherche opérationnelle sur les graphes, ce TD démontre qu'il est aisé de **ramener de nombreux problèmes réels à un problème de recherche opérationnelle**, ici celui de l'**étude d'un chemin critique** au sein d'un **modèle MPM** et la **puissance de ses solutions algorithmiques**, utilisant l'**algorithme PERT** pour résoudre les-dits problèmes. J'ai appris à appliquer mes connaissances **via un problème classique d'ordonnancement**.

Sur le plan de **résolution des problèmes réels**, cet exemple me démontre l'importance du **chemin critique** au sein d'un **problème classique d'ordonnancement** et cela me donne un aperçu d'un futur travail qui m'attend.