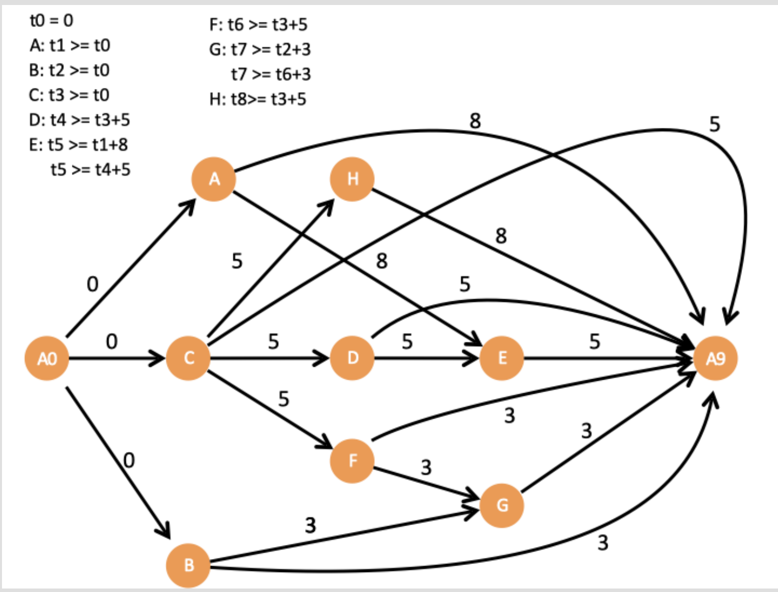
TD 7 : Problème d'ordonnancement

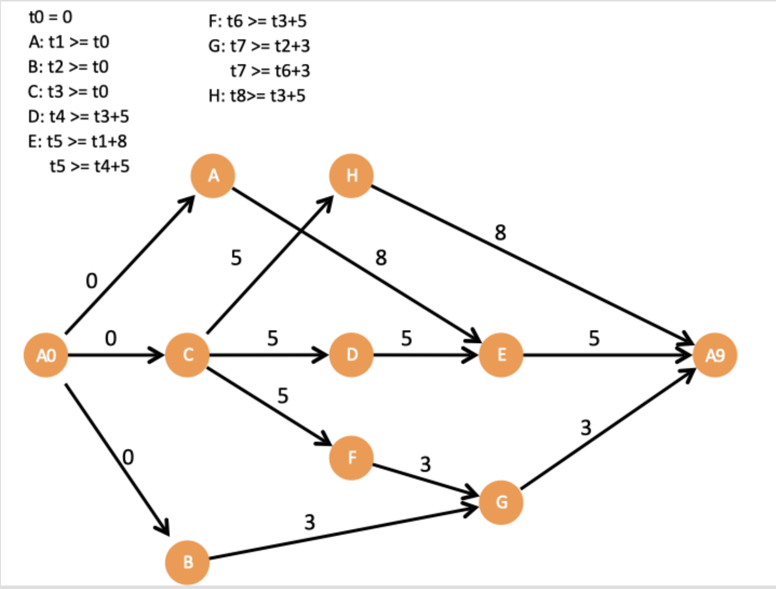
1 Application du cours - Mise en place d'un atelier d'embouteillage de vin

Tâche	Durée	Description	Contraintes
A	8	Commander la cuverie	-
B	3	Commander le matériel d'embouteillage	-
C	5	Terrassement et fondations	-
D	5	Installer infrastructure cuverie	C terminé
E	5	Installer cuverie	A et D terminés
F	3	Installer infrastructure matériel d'embouteillage	C terminé
G	3	Installer matériel d'embouteillage	B et F terminés
H	8	Installer le matériel de manutention	C terminé

QUESTION 1 – Donner le graphe en représentation potentiel-tâche.



Nous pouvons simplifier le graphe en retirant les arêtes redondantes.



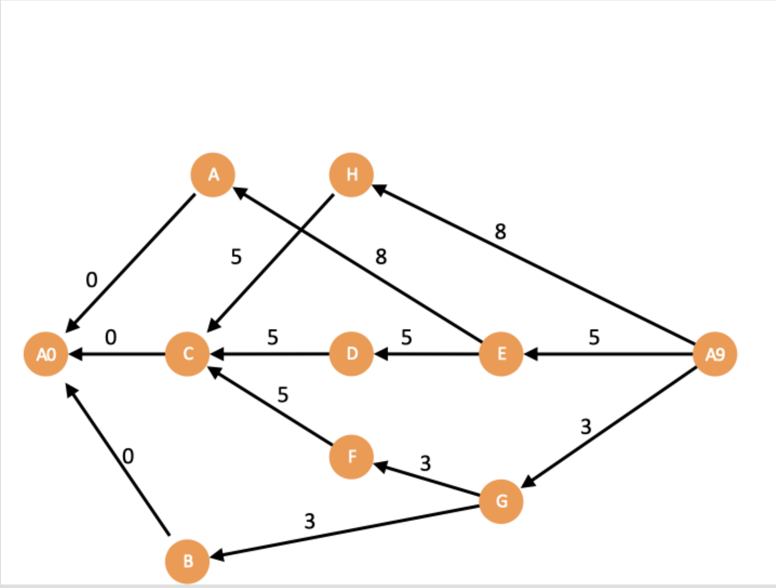
Question 2 – Donner les ordonnancements au plus tôt.

Ordonnancement au plus tot en utilisant Bellman-Ford (avec max).

	A_0	A	B	C	D	E	F	G	H	A_9
$k=0$	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
$k=1$	0	0	0	0	5	8	5	3	5	8
$k=2$	0	0	0	0	5	10	5	8	5	13
$k=3$	0	0	0	0	5	10	5	8	5	15

Question 3 – Donner l'ordonnancement au plus tard, pour la durée minimum des travaux.

Ordonnancement au plus tard avec $F = 15$. Graphe inversé puis Bellman-Ford.



	A_9	H	G	F	E	D	C	B	A	A_0
$k=0$	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
$k=1$	0	8	3	3	5	5	5	3	8	0
$k=2$	0	8	3	6	5	10	13	6	13	8
$k=3$	0	8	3	6	5	10	15	6	13	13
$k=4$	0	8	3	6	5	10	15	6	13	15
$t_i = F - \Delta^{-1}(9,i)$	15	7	12	9	10	5	0	9	2	0

QUESTION 4 – Indiquer la marge totale pour chaque sommet ainsi que leur mage libre (pour les deux ordonnancements).

Calculons les marges. Rappel :
- marge totale : $m_i = \lambda'_i - \lambda_i$
- marge d'un chemin u : $m(u) = max_{i \in u} m_i$
- marge libre pour une tâche A_i d'un ordonnancement (t_i) donné :

$$\mu_i = min_{i \xrightarrow{a_{ij}} j} (t_j - t_i - a_{ij})$$

	A_0	A	B	C	D	E	F	G	H	A_9
λ_i	0	0	0	0	5	10	5	8	5	15
λ'_i	0	2	9	0	5	10	9	12	7	15
marge totale	0	2	9	0	0	0	4	4	2	0
marge libre au plus tôt	0	2	5	0	0	0	0	4	2	–
marge libre au plus tard	0	0	0	0	0	0	0	0	0	–

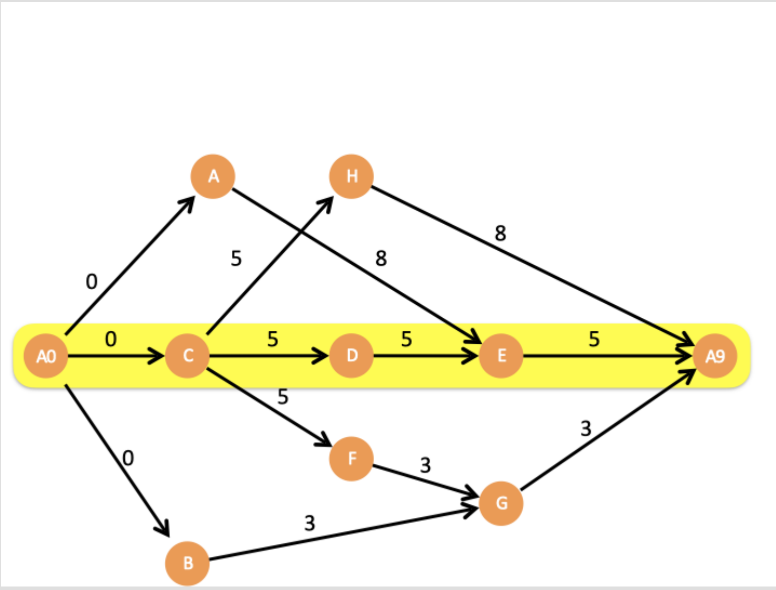
La dernière ligne est logique car les tâche sont déjà effectuées au plus tard ; elles ne peuvent donc pas prendre du retard supplémentaire.

QUESTION 5 – Indiquer la marge du chemin passant par les tâches B et G. De plus, indiquer un chemin critique.

Marge du chemin $u = A0 \to B \to G \to A9$:

$$m(u) = max(0, 9, 4, 0) = 9.$$

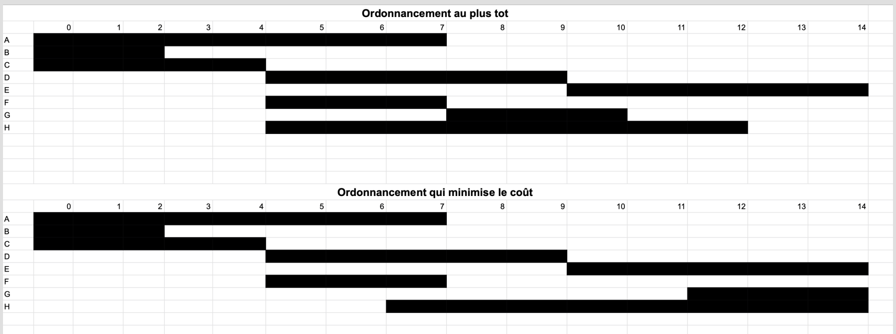
Chemin critique :



Nous supposons maintenant que chaque l'usine a un coût de fonctionnement horaire non négligeable que son propriétaire souhaite minimiser. Le coût horaire dépend du nombre de tache simultanées :

— 1 tâche : 1
— 2 tâches : 2
— 3 tâches : 4
— 4 tâches : 8
— 5 tâches : 16

QUESTION 6 – Proposer un ordonnancement qui permet de minimiser le coût.



Coût ordonnancement au plus tôt : $3 \times 4 + 2 \times 2 + 3 \times 8 + 2 \times 4 + 4 + 2 \times 2 + 2 \times 1 = 56$
Coût second ordonnancement : $3 \times 4 + 2 \times 2 + 2 \times 4 + 8 + 2 \times 2 + 2 \times 2 + 3 \times 4 = 52$

2 Difference bound matrix

Les tâches d'ordonnancement sont présentées par des contraintes de la forme $t_j - t_i \leq a_{ij}$. Nous allons voir ici un méthode permettant de décider si un tel ensemble de contraintes admet une solution.

Définition

Étant donné un ensemble $X = \{x_1, \dots, x_n\}$ de variables, une DBM est une matrice $M = (a_{ij})_{1 \leq i,j \leq n}$. Elle représente l'ensemble de contraintes

$$\mathcal{C}(M) = \{x_j - x_i \leq a_{ij} \mid 1 \leq i, j \leq n\}.$$

Nous noterons également $G(M)$ le graphe pondéré ayant M comme matrice d'adjacence, i.e. $G(M) = (S, V)$ avec $S = \{x_1, \dots x_n\}$ et $V = \{x_i \xrightarrow{a_{ij}} x_j\}_{1 \leq i,j \leq n}$.

Question 7 – Montrer que $\mathcal{C}(M)$ n'admet pas de solution si $G(M)$ admet un cycle de poids strictement négatif.

Supposons que $G(M)$ admet un cycle de poids strictement négatif et que $\mathcal{C}(M)$ admet une solution f, i.e. $f(x_j) - f(x_i) \leq a_{ij}$ pour tout $1 \leq i, j \leq n$.

Par hypothèse, il existe $u = x_{i_1}.x_{i_2}.\dots.x_{i_k}.x_{i_1}$ tel que $a_{i_1 i_2} + \dots a_{i_{n-1} i_n} + a_{i_n i_1} < 0$. Par conséquent, $(f(x_{i_2}) - f(x_{i_1})) + \dots + (f(x_{i_n}) - f(x_{i_{n_1}})) + (f(x_{i_1}) - f(x_{i_n})) < 0$. Or, ceci est une somme télescopique ; nous obtenons donc $f(x_{i_1}) - f(x_{i_1}) < 0$. Contradiction. Donc $\mathcal{C}(M)$ n'admet pas de solution. $\square$

Question 8 – Montrer que si $\mathcal{C}(M)$ n'admet pas de solution alors $G(M)$ admet un cycle de poids négatif.

Nous supposerons que tous les sommets sont accessibles depuis un noeud racine que nous supposerons être x_1.

Indice : considérer le graphe G' des plus courts chemins, i.e. $x_i \xrightarrow{a'_{ij}} x_j \in G$ ssi le plus court chemin de x_i à x_j ans G a pour poids a'_{ij}.

Supposons que G n'admet pas de cycle de poids strictement négatif. Nous savons donc que G' est bien défini. Définissons la valuation f telle que $f(x_1) = 0$ et $f(x_i) = a'_{1i}$. Cela définit entièrement f car tous les sommets sont accessibles depuis x_1 par hypothèse.

Montrons que f satisfait $\mathcal{C}(M)$: Par l'absurde, supposons qu'il existe une contrainte $x_j - x_i \leq a_{ij}$ qui n'est pas satisfaite, i.e. $f(x_j) - f(x_i) > a_{ij}$. Considérons dans G le chemin $u = x_1 ... x_i x_j$ (ce chemin existe car x_i est accessible depuis x_1). W.l.o.g. nous supposons que $x_1 ... x_i$ est le plus petit chemin de x_1 à x_i. Le poids de u, $\omega(u) = a'_{1i} + a_{ij}$. Nous avons donc :

$$\omega(u) = a'_{1i} + a_{ij} < a'_{1i} + f(x_j) - f(x_i) = a'_{1i} + a'_{1j} - a'_{1i} = a'_{1j}.$$

Ceci est une contradiction car a'_{1j} est supposé être le poids du plus petit chemin de x_1 à x_j dans G.

Question 9 – En admettant que la question 5 est vraie pour tout graphe, proposer un algorithme pour déterminer si un ensemble de contraintes admet une solution.

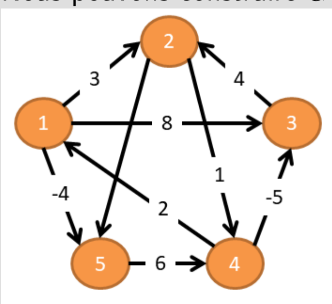
Algorithme de Floyd-Wharshall qui permet de détecter les cycles de poids négatif.

Soit S l'ensemble composé des contraintes suivantes :

$$\begin{array}{lll} x_2 - x_1 \leq 3 & x_2 - x_3 \leq 4 & x_4 - x_4 \leq 6 \\ x_3 - x_1 \leq 8 & x_4 - x_3 \geq 5 & x_4 - x_2 \leq 1 \\ x_1 - x_5 \geq 4 & x_1 - x_4 \leq 2 & x_5 - x_2 \leq 7. \end{array}$$

Question 10 – En utilisant les questions précédentes, l'ensemble S admet-il une solution ?

Nous pouvons construire G qui représente cet ensemble. L'arête $2 \to 5$ a pour poids 7.



En appliquant l'algorithme nous ne détectons pas de cycle de poids strictement négatif. L'ensemble S admet donc une solution.

0	3	8	∞	-4
∞	0	∞	1	7
∞	4	0	∞	∞
2	∞	-5	0	∞
∞	∞	∞	6	0

0	3	8	∞	-4
∞	0	∞	1	7
∞	4	0	∞	∞
2	5	-5	0	-2
∞	∞	∞	6	0

0	3	8	4	-4
∞	0	∞	1	7
∞	4	0	5	11
2	5	-5	0	-2
∞	∞	∞	6	0

0	3	8	4	-4
∞	0	∞	1	7
∞	4	0	5	11
2	-1	-5	0	-2
∞	∞	∞	6	0

0	3	-1	4	-4
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0

0	1	-3	2	-4
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0