

**Gnebehi Bagre Jean-Philippe**

**L3 INFORMATIQUE**

**2017-2018**

**OPTIMISATION DU TEMPS D'EXECUTION  
D'UNE APPLICATION CRITIQUE**

**Khadir Ouriachi**

**Enseignant chercheur à Université de Pau**



## I- POSITION DU PROBLEME

A l'heure actuelle le niveau de production des applications dans le monde est un grand succès . Cependant pour être compétitif , il est fortement capital que ces applications aient un temps d'exécution minimal.

Dans le cadre de notre formation à l'Université de Pau dirigé par l'enseignant chercheur Khadir Ouriachi , nous nous confrontons à un problème d'**optimisation du temps d'exécution d'une application critique**.

Le problème soumis à notre sagacité consiste donc à optimiser la durée du cycle . Cela consiste donc à rechercher un chemin optimal dans le cycle.

## II- REALISATION

### II-1 MODELE DE GRAPHE

Pour formaliser le problème nous allons modéliser un graphe qui met en évidence le cycle d'exécution des applications à l'aide d'un graphe orienté  $G=(S,A)$ , défini comme suit :

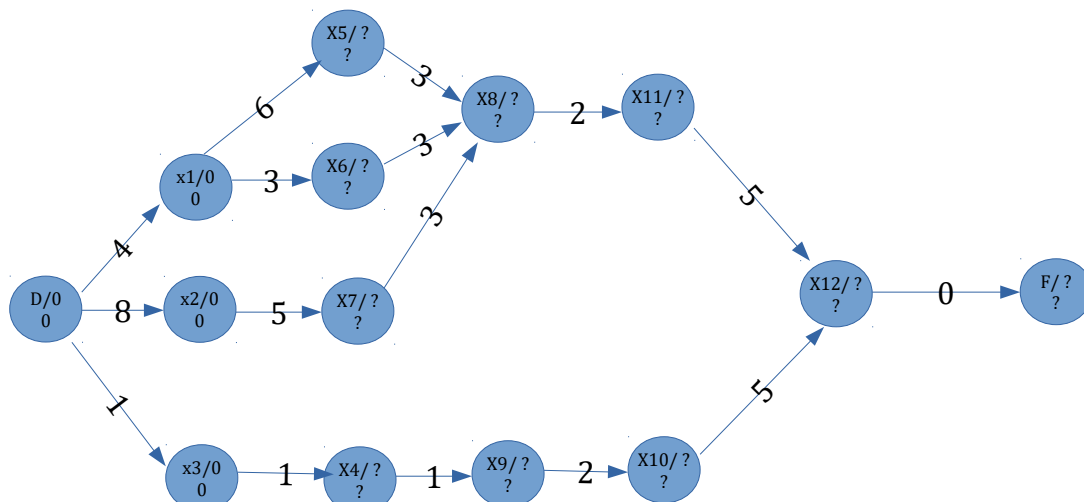
$S= \{ D, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, F \}$

$A= \{ (D, X1), (D, X2), (D, X3), (X1, X5), (X1, X6), (X2, X7), (X3, X4), (X5, X8), (X6, X8), (X7, X8), (X4, X9), (X8, X11), (X9, X10), (X10, X12), (X11, X12), (X12, F) \}$

Comme vu en cours nous allons utiliser la convention suivante pour représenter les sommets du modèle de graphes :

Code thread / Date plus tôt  
Date au plus tard

Le modèle de graphe est donc le suivant :



Le modèle sera complet quand on aura calculé pour chaque tâche sa date de départ au plus tôt et sa date de départ au plus tard.

## II-2 PROBLÈME DE RECHERCHE DU CHEMIN OPTIMAL PARTANT DU SOMMET DÉBUT ET ATTEIGNANT LE SOMMET REPRÉSENTANT LE THREAD T

Pour préserver la relation de précédence on doit imposer qu'un thread T :

- doit être exécutée qu'une seule fois
- seulement si tous les threads qui le précèdent ont déjà été exécutés

Pour déterminer la date au plus tôt d'un thread il est primordial de faire le calcul :

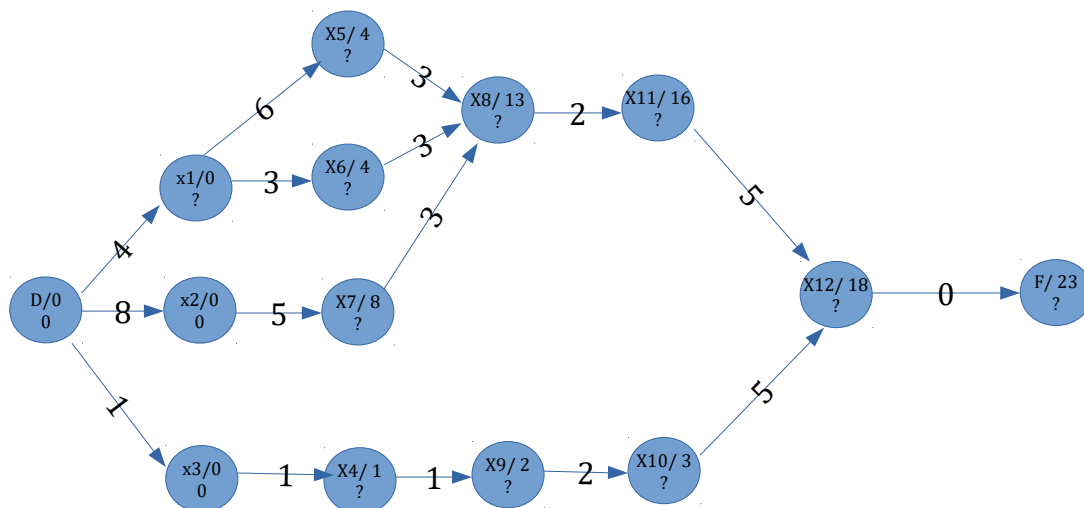
- de la longueur  $L_T$  du chemin le plus long
- entre les sommets DEBUT et T

Vu qu'on calcule la date de départ du thread T il faut faire une soustraction entre la longueur  $L_T$  du chemin le plus long et la durée  $t_T$  du thread

On obtient la formule suivante du calcul de la date au plus tôt d'un thread T:

$$Dtôt(T) = L_T - t_T$$

On obtient le modèle suivant plus complet :



## II-3 RECHERCHE DE CHEMIN OPTIMAL ENTRE LES SOMMETS DÉBUT ET FIN

Si on fixe à 0 la date de départ au plus tôt de la tâche **DEBUT**, la date de départ au plus tôt de la tâche **FIN** représente la durée optimale du cycle.

La date au plus tôt de la tâche FIN passe donc par la recherche du chemin le plus long partant du sommet **DEBUT** et joignant le sommet **FIN**.

On obtient :

$$Dt\hat{o}t(FIN) = L_{FIN} - t_{FIN}$$

Avec  $L_{FIN} = 23$  et  $t_{FIN} = 0$

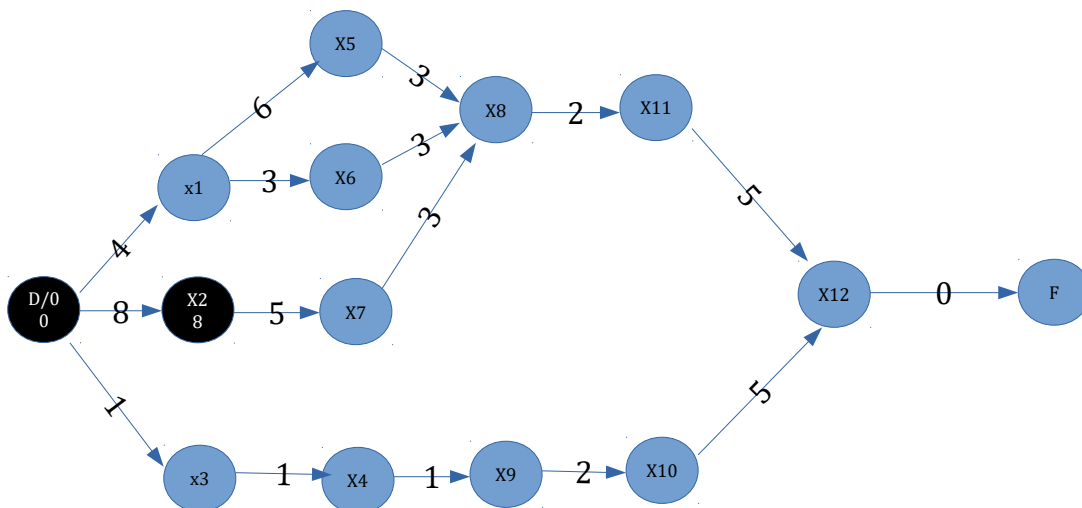
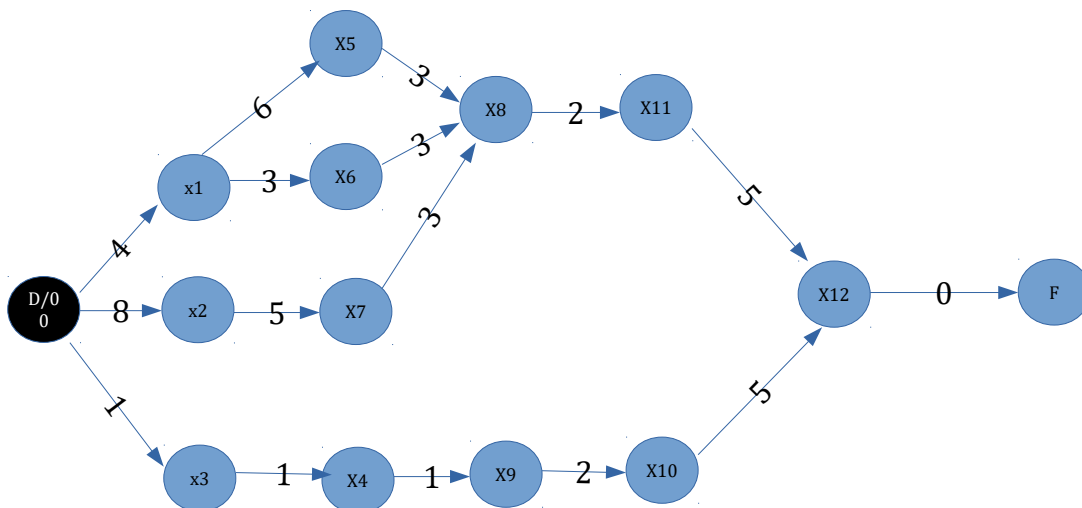
Ainsi

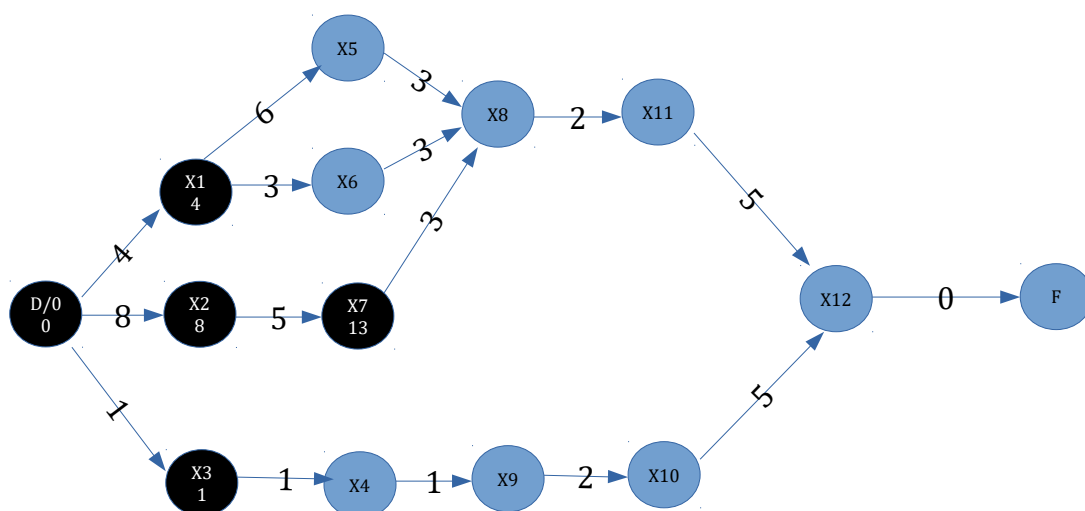
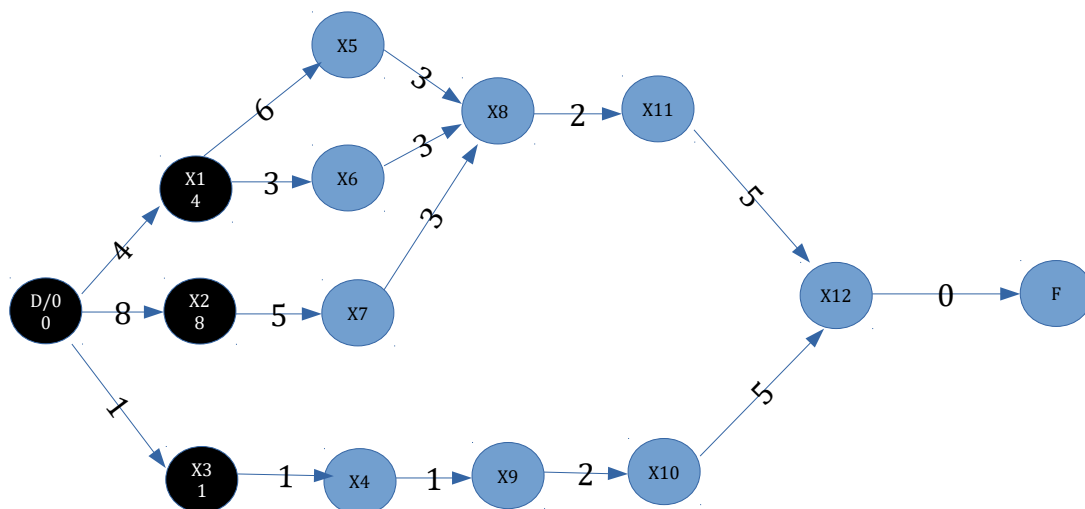
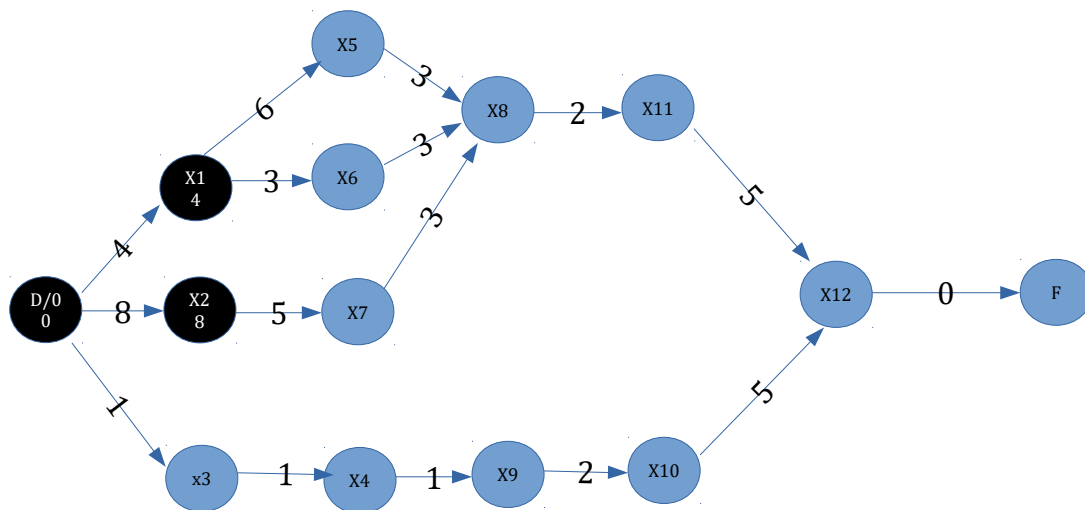
$$Dt\hat{o}t(FIN) = 23$$

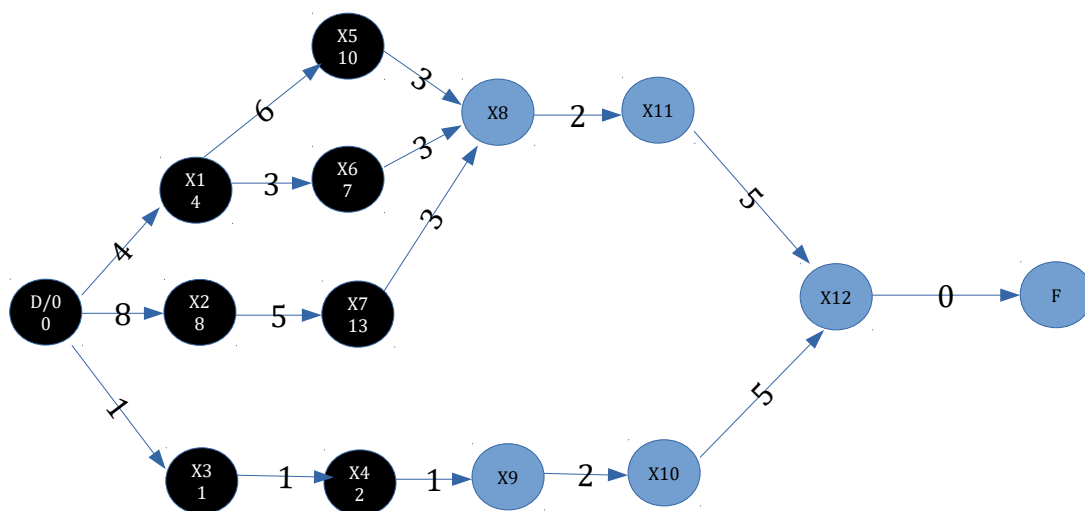
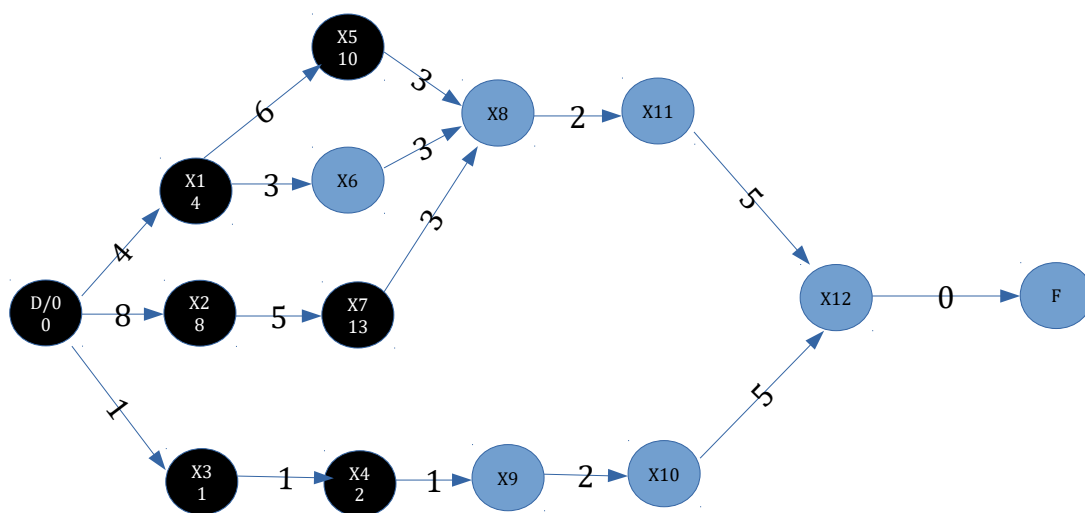
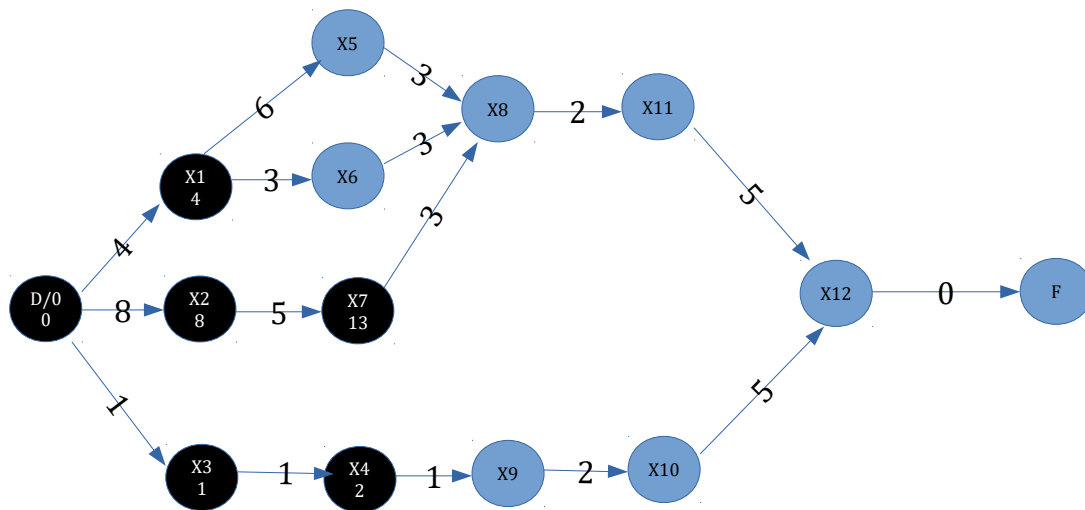
## II-4 SOLUTION AU PROBLEME

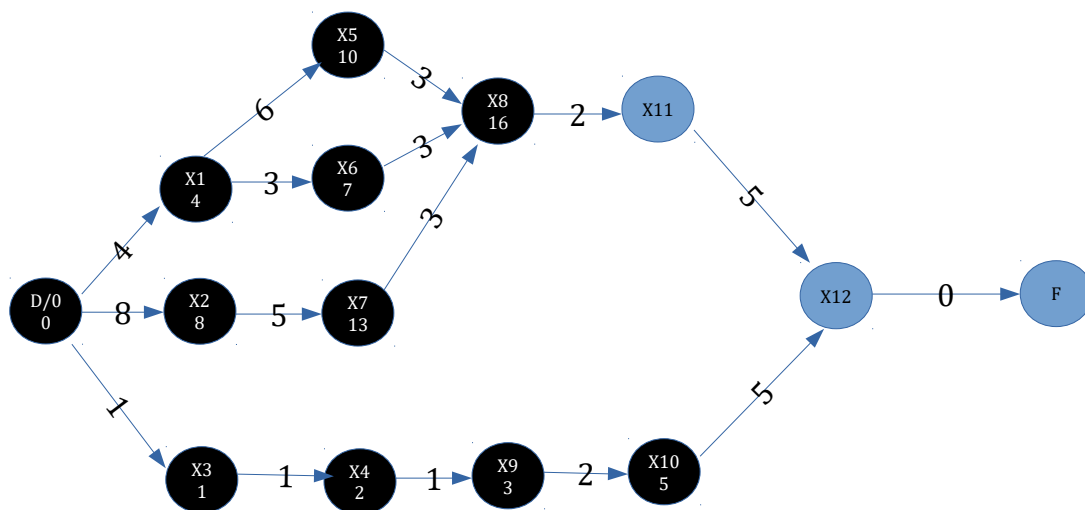
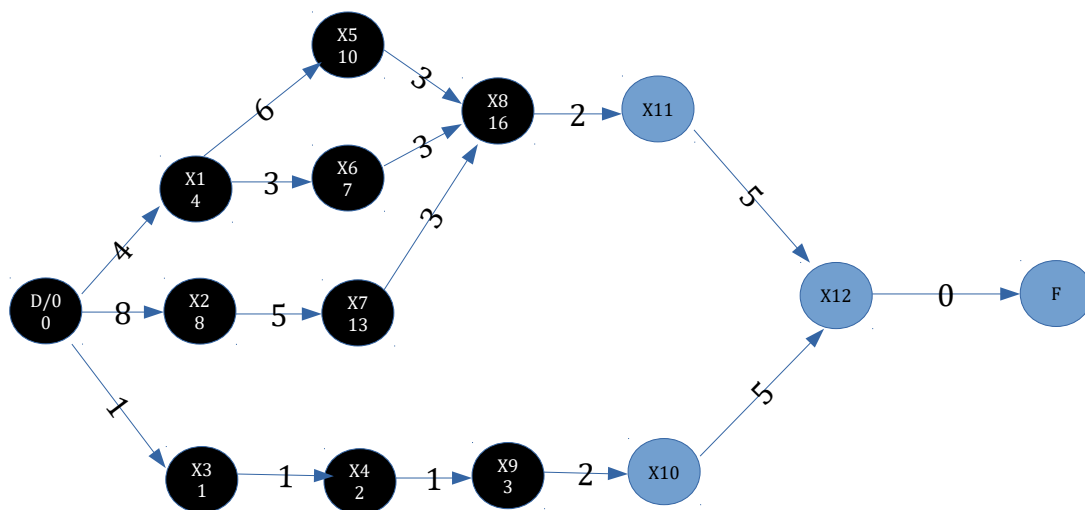
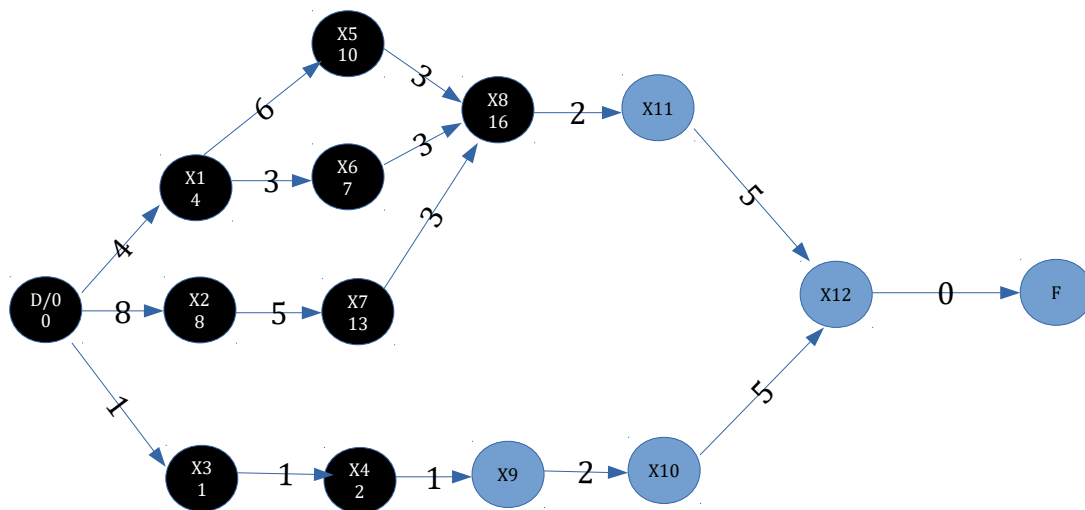
### II-4.1 ALGORITHME DE BELLMAN

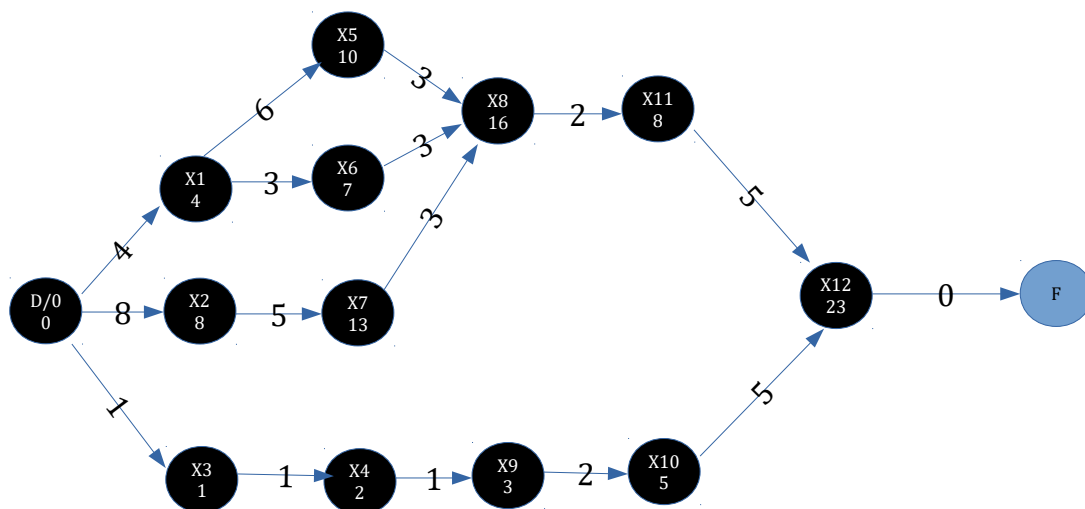
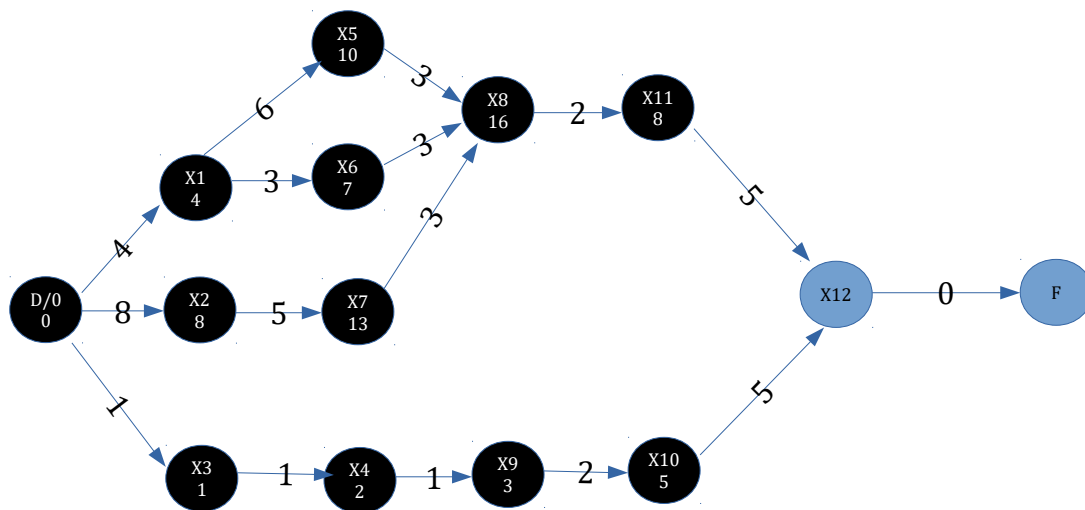
La procédure consiste à rechercher dans le modèle de graphe le chemin le plus long entre les sommets.



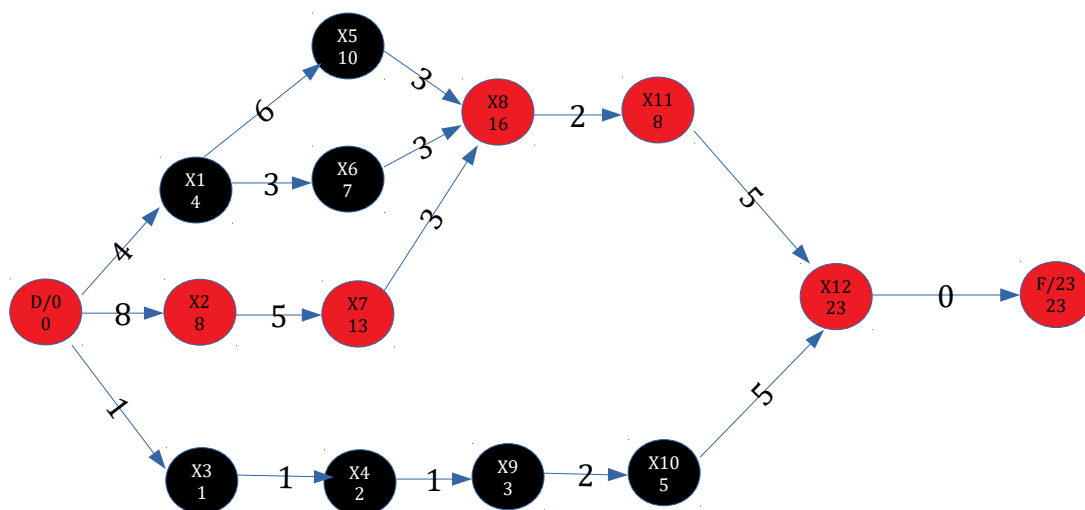






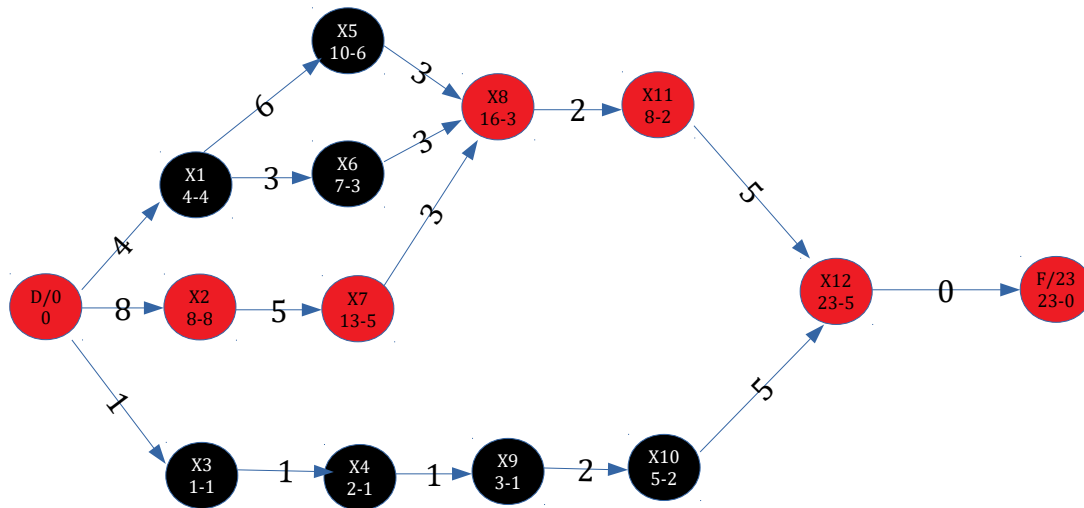


**Le résultat est le suivant**



## II-4.2 CALCUL DES DATES AU PLUS TÔT





## II-5 RECHERCHE DE CHEMIN OPTIMAL ENTRE CHAQUE SOMMETS DÉBUT ET LE SOMMET FIN

Le but est d'estimer la durée maximale en retardant le démarrage d'une tâche augmenter la durée optimale du cycle.

On se réfère à la durée minimale qui peut séparer la date de démarrage au plus tard de la tâche et la date du fin de cycle.

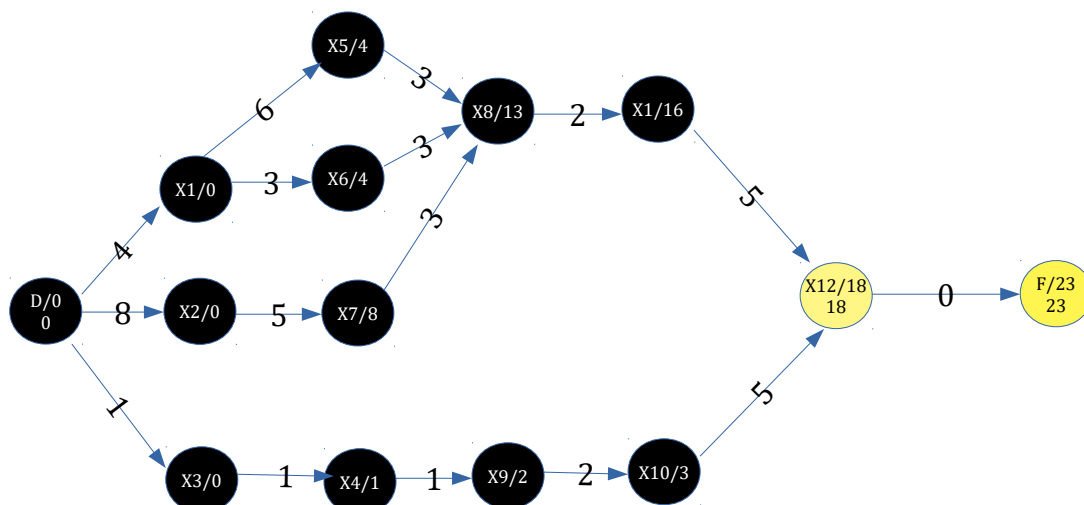
D'où le calcul pour chaque tâche T du chemin le plus court entre le sommet Fin et le sommet représentant T.

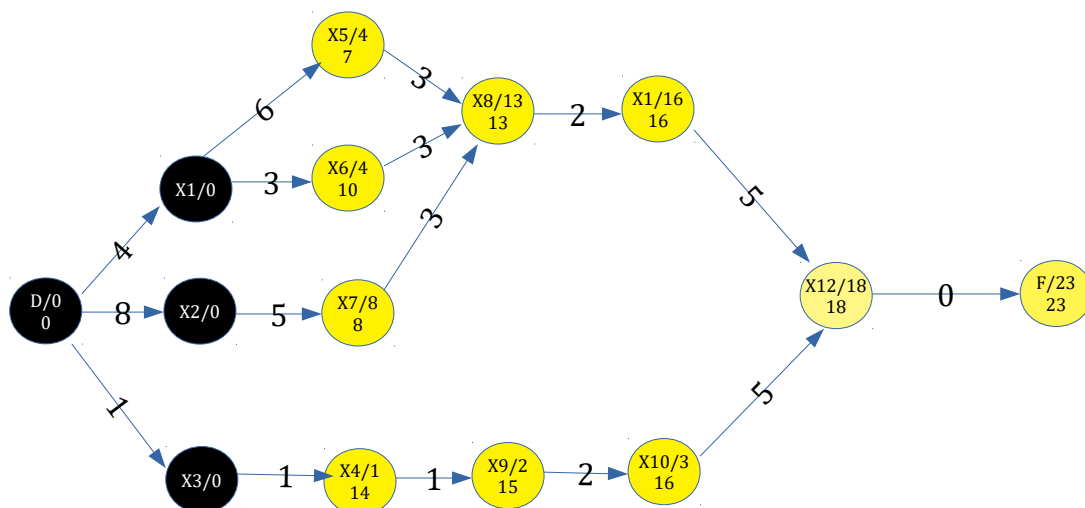
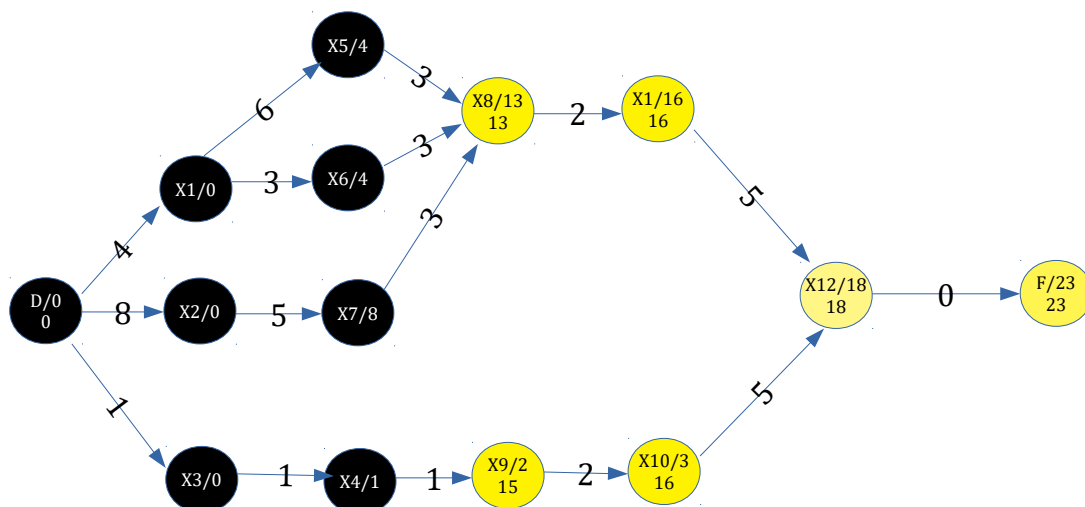
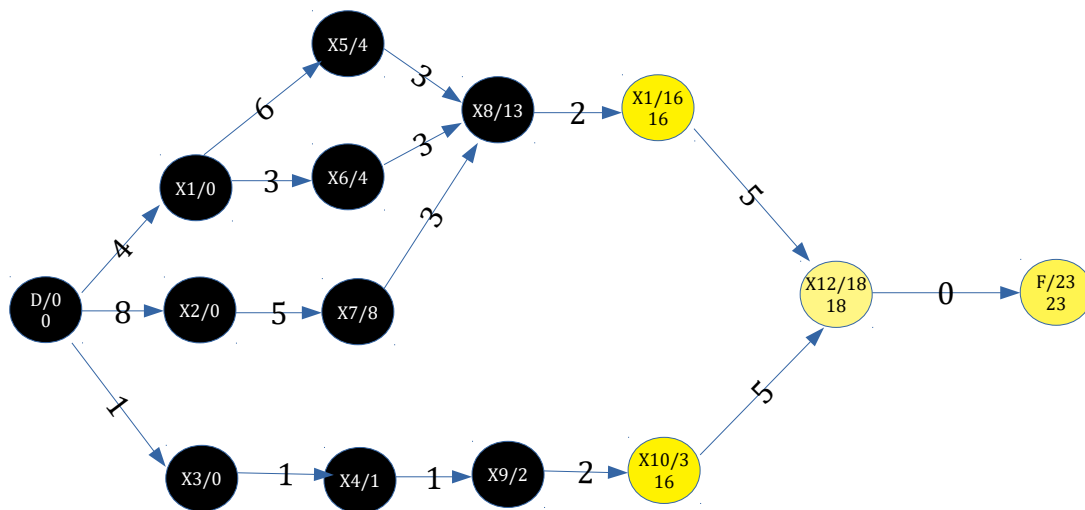
## II-6 SOLUTION POUR CALCULER LA DATE AU PLUS TARD

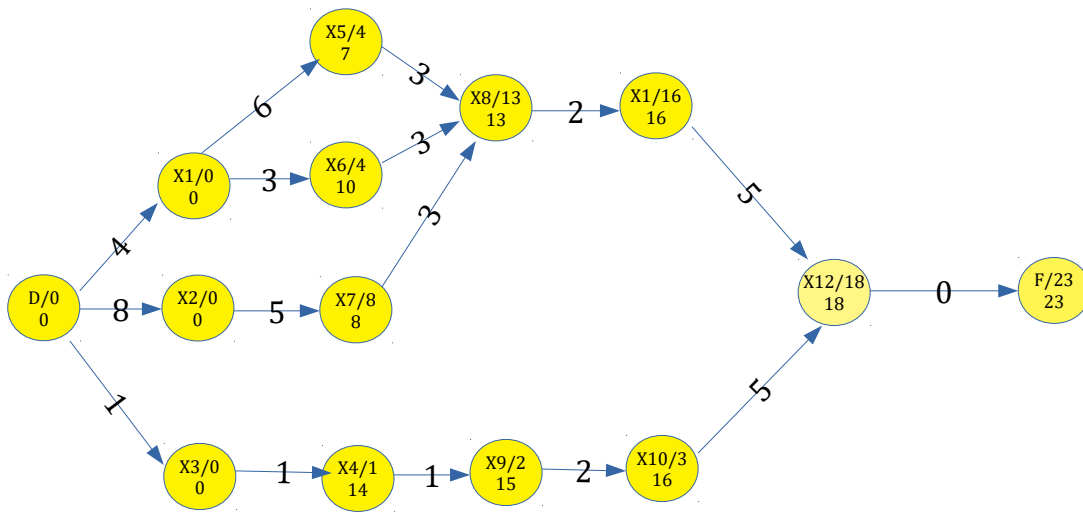
Soit CT, la longueur du plus court chemin entre le thread T et la tâche FIN et  $t_T$  la durée du thread T

La date au plus tard est calculée en utilisant la formule suivante :

$$Dtard(T) = 23 - CT - t_T$$





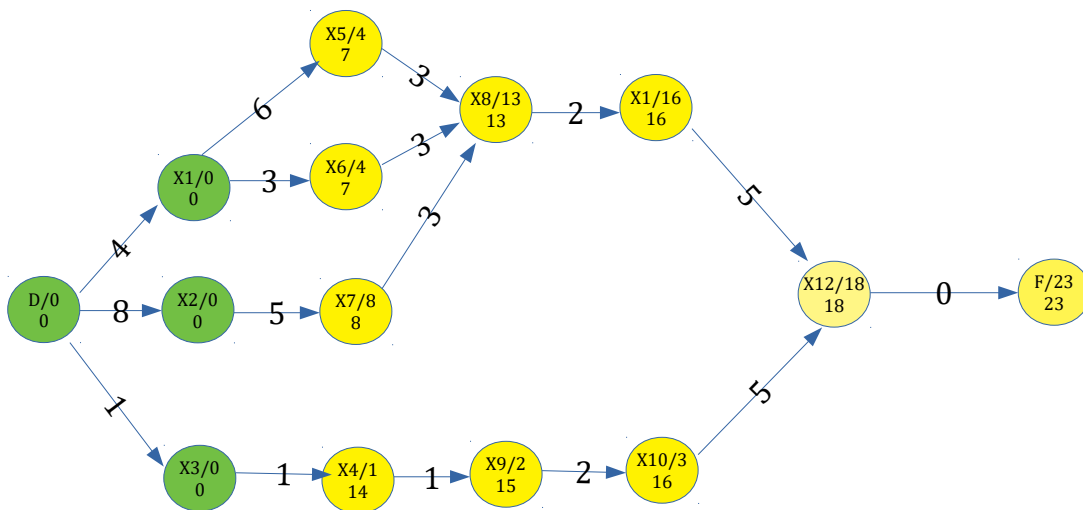


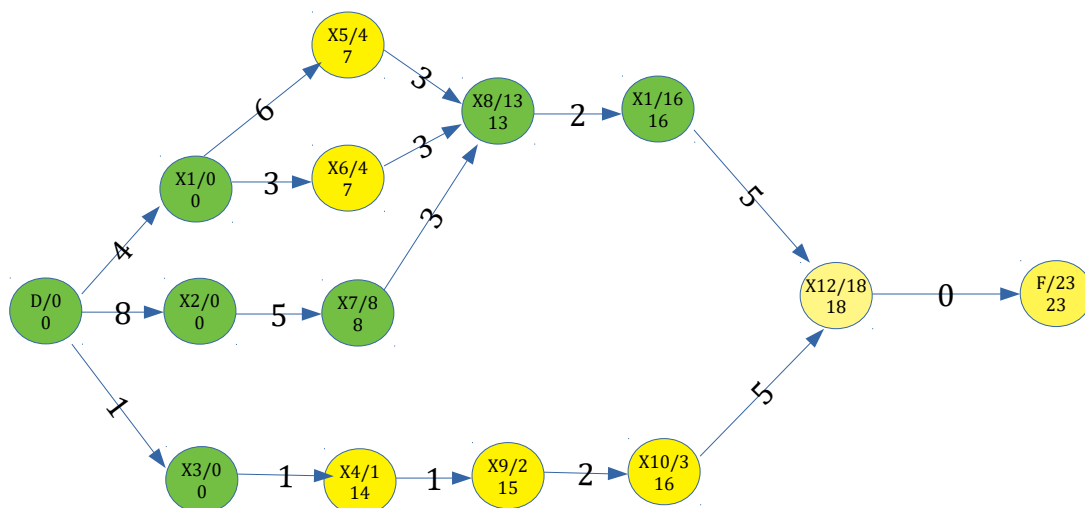
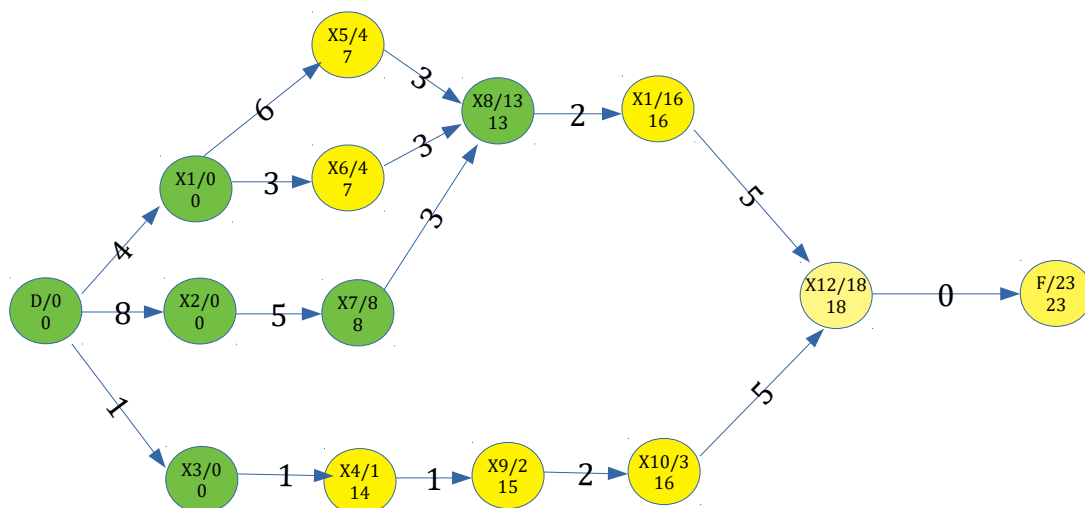
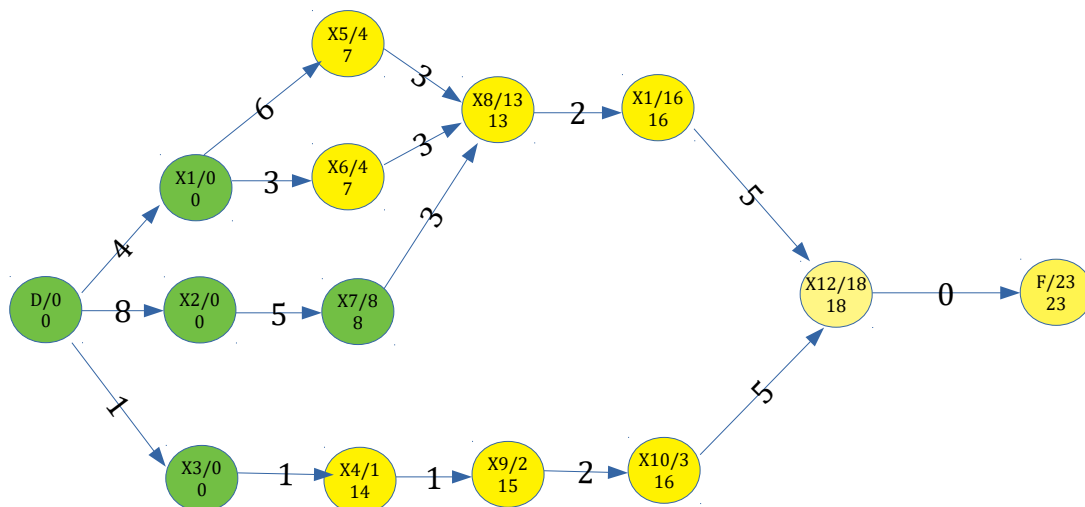
## II-7 PROCÉDURE POUR DÉTERMINER LE CHEMIN CRITIQUE

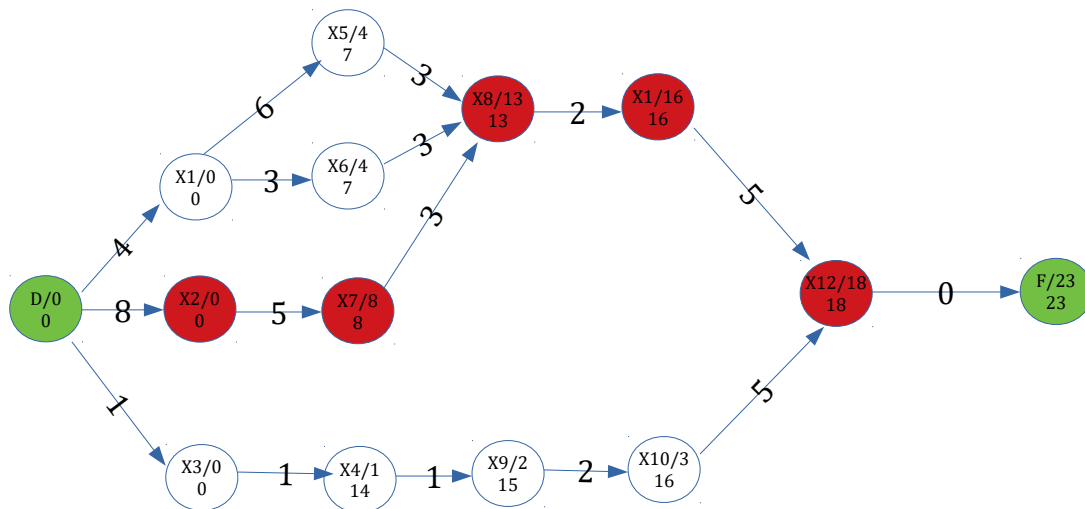
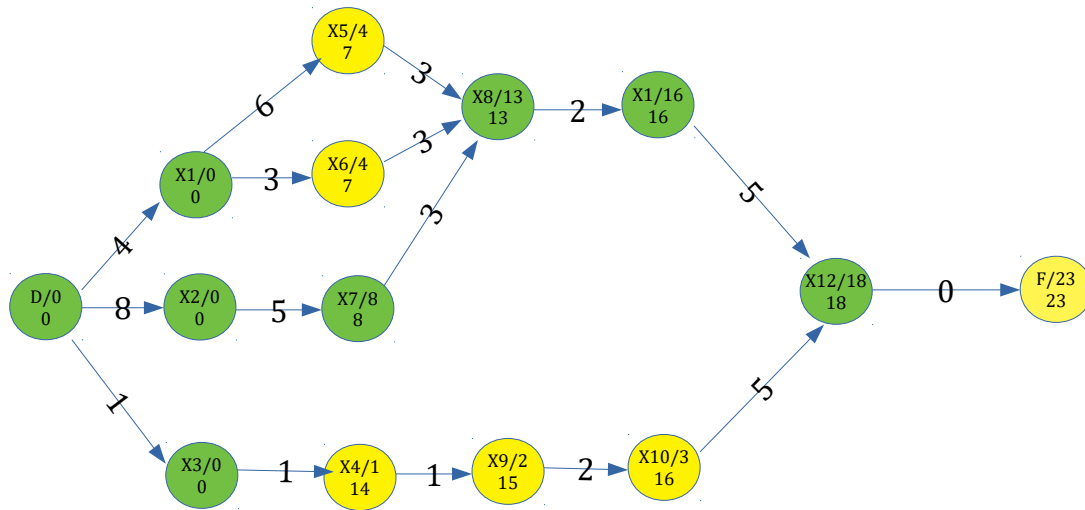
La procédure la plus simple consiste à marquer les sommets tels que la date au plus tôt est égale à la date au plus tard partant du sommet DÉBUT

$$Dtôt(s) = Dtard(s)$$

La procédure s'arrête lorsque le sommet FIN est marqué







## II-7 COMPARAISON

Le chemin critique est le chemin le plus long entre les sommets DEBUT et FIN.

Cela signifie que la tâche FIN est atteinte à partir de la tâche DEBUT au plus tôt après une durée 23 qui garantit que n'importe laquelle des tâches est exécutée.

## III – CONCLUSION

Au terme de notre étude, nous retenons que l'algorithme de Bellman nous permet de procéder à une optimisation du temps d'exécution d'une application critique.

Aussi nous retenons d'abord que le chemin critique fixe une durée minimale pour le cycle d'exécution.

D'autre part, il signifie également que la tâche FIN peut être atteinte au plus tard à partir de la durée calculée le long du chemin le plus long

De plus Le chemin critique fixe une date au plus tard pour la fin du cycle d'exécution.

La durée du chemin critique détermine la durée optimale cycle d'exécution.

La durée optimale du cycle d'exécution est alors calculée le long du chemin critique