

Compte Rendu du TP4

Florian Robin Vincent Palacio

Sujet : Optimisation du temps d'exécution d'une application critique

1-Proposer un modèle de graphe pour représenter le cycle d'exécution de l'application. En tenant compte des conventions précédentes, le modèle proposé peut être un graphe orienté $G = (S, A)$ défini comme suit :

$S = \{\text{DEBUT}, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, \text{FIN}\}$

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

Chaque sommet du modèle de graphe porte 3 données :

- le code du thread,
- sa date d'activation au plus tôt, (qui sera calculée en 4)
- et sa date d'activation au plus tard. (qui sera calculée en 5)

-les arcs représentent les relations de précédence entre les threads : l'arc (x, y) signifie que le thread y est précédé immédiatement du thread x .

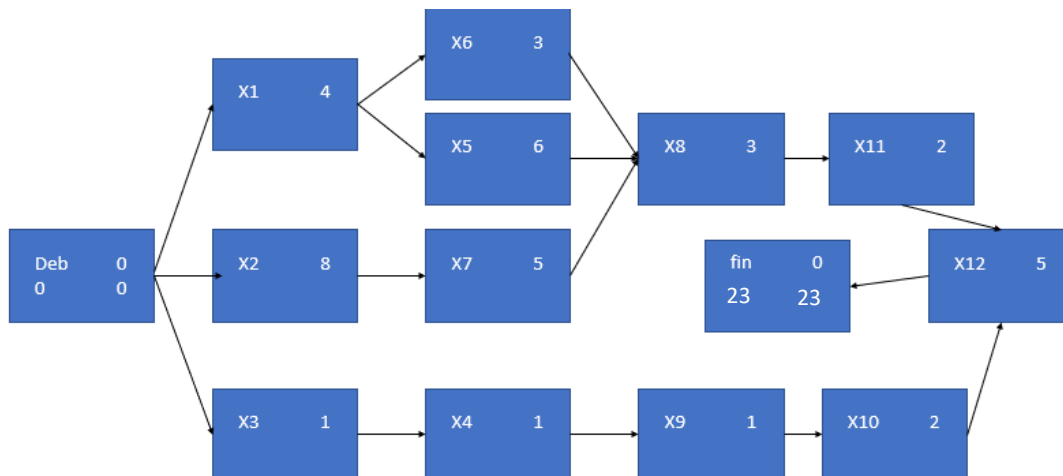
-le coût associé à un arc (x, y) correspond à la durée maximum d'exécution du thread y .

-deux sommets particuliers sont à distinguer :

- l'un, appelé Début, représente un thread virtuel qui initialise le cycle : sa durée est nulle ; il n'est précédé d'aucun thread. Sa date au plus tôt et sa date au plus tard sont égales à 0.

- l'autre, appelé Fin, représente un thread virtuel qui termine le cycle : sa durée est nulle ; il ne précède aucun thread

Le modèle de graphe est visualisé ci-après :



2-Montrer, sur la base du modèle précédent, que le problème de calcul de la date d'activation au plus tôt d'un thread T se ramène au problème de recherche d'un 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 :

- ne peut être exécuté qu'une fois les précédents effectués.

Donc, la détermination de la date au plus tôt d'un thread se ramène au calcul :

- de la longueur L_x du chemin le plus long

- entre les sommets DEBUT et X

D'où la formule suivante de calcul de la date au plus tôt d'une décision :

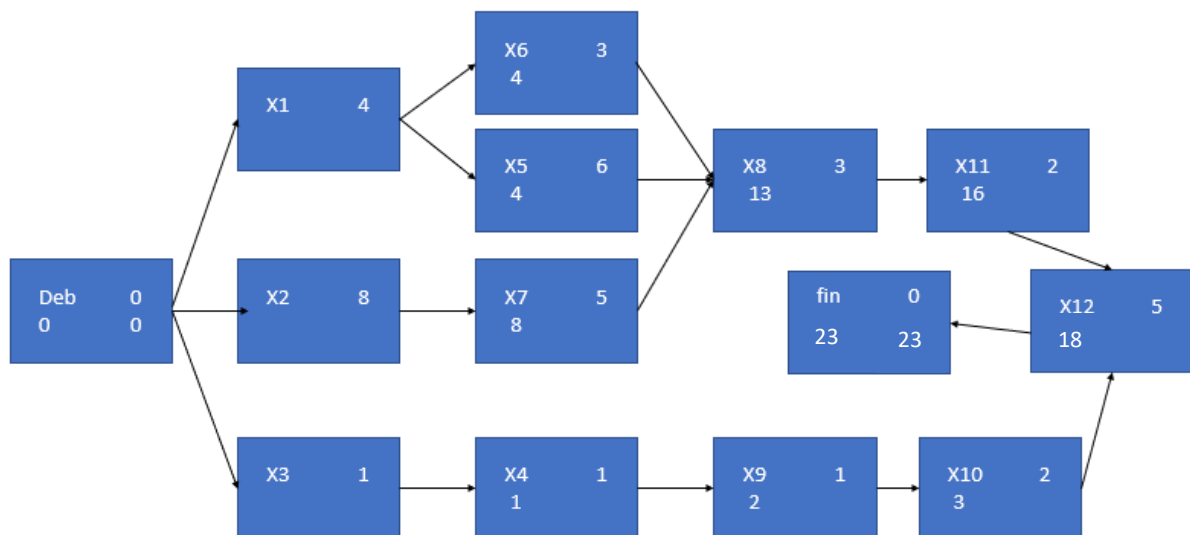
$$D+tôt(X) = LX$$

3-Compte tenu de ce qui précède, proposer une solution pour calculer les dates au plus tôt d'activation des threads et la durée optimale du cycle d'exécution.

En appliquant l'algorithme de Bellman pour le calcul du plus long chemin, on obtient avec la formule précédente, le tableau suivant :

Date au plus tôt
X1 : 0
X2 : 0
X3 : 0
X4 : 1
X5 : 4
X6 : 4
X7 : 8
X8 : 13
X9 : 2
X10 : 3
X11 : 16
X12 : 18

Ce qui donne donc le graphe suivant :



La durée optimal de la décision globale est obtenue en calculant la date au plus tôt de la tâche FIN. D'après 2) elle est égale à longueur du chemin le plus long partant du sommet DEBUT et atteignant le sommet FIN :

Durée optimale de la décision globale = Date+tot(Fin)

= 23 (unités de temps)

4 et 5- calculer, pour chaque thread, la date d'activation au plus tard pour cela il faut montrer que le problème de calcul des dates au plus tard passe par le problème de recherche d'un chemin optimal entre chaque sommet et le sommet Fin :

« L'objectif est d'estimer la durée maximale : dont il est possible retarder le démarrage d'un thread sans pour autant augmenter la durée optimale de la décision globale »

Pour cela, on doit garantir :

- la date de fin d'un thread, doit précéder la date de début de tous les thread suivants.

D'où le calcul pour chaque thread de décision X du chemin le plus long :

- entre le sommet Fin et le sommet qui représentant X.

6-Proposer une solution pour calculer les dates au plus tard :

Notant LXF, la longueur du plus long chemin entre :

- le sommet représentant le thread X et la tâche FIN

La date de début au plus tard du thread X est calculée à l'aide de la formule :

$$\text{Date+tard}(T) = (23 - \text{LXT})$$

La longueur L est calculée en appliquant l'algorithme de

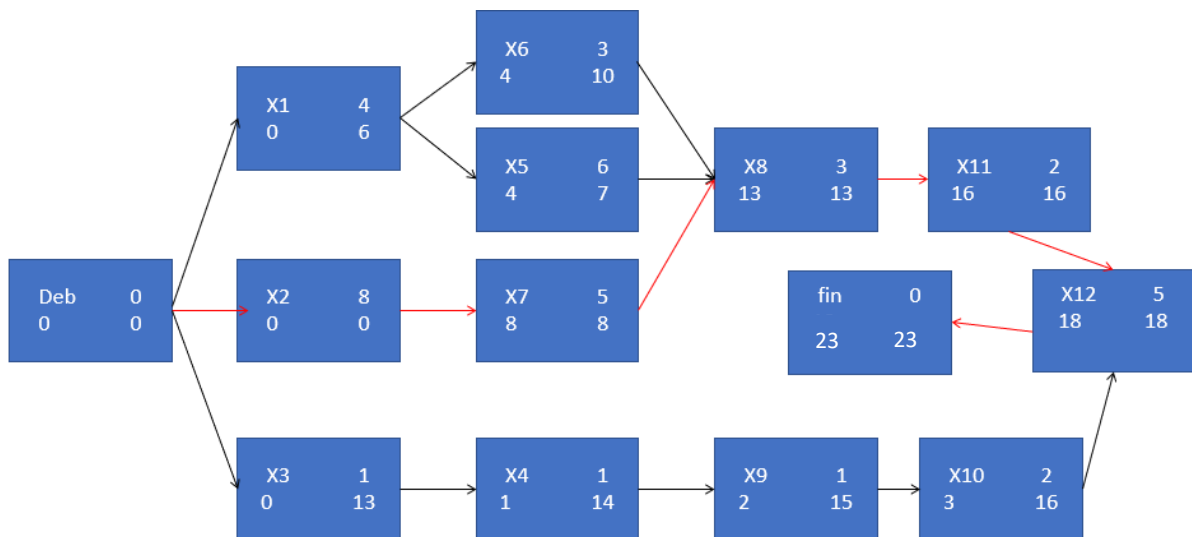
Bellman dual dont la procédure est exposée ci-dessous :

```
Bellman_Duale(G, t)  
Début  
  /*initialisation */  
  S' ← {t}; S ← S-S';  
  Pour tout u ∈ S Faire  
    L(u) ← ∞;  
    Suc(u) ← nil  
  FinPour;  
  L(t) ← 0;  
  
  /*relaxation de l'arc (u,v) */  
  Tant que ∃ u ∈ S-S' | SucG(u) ⊆ S' Faire  
    S' ← S' ∪ {u};  
    S ← S - {u};  
    Pour tout (u,v) | v ∈ S' Faire  
      Si L(u) > L(v) + cout(u,v)  
      Alors  
        L(u) ← L(v) + cout(u,v);  
        Suc(u) ← v  
      FinSi;  
    FinPour  
  FinTq  
Fin
```

Est considérée comme critique toute décision partielle X telle que : $\text{Date+tôt}(X) = \text{Date+tard}(X)$

Tout retard au niveau d'un tel thread entraîne la remise en cause de la durée optimale de la tâche globale calculée en 2).

Le chemin traversant les sommets correspondant aux thread critiques est appelé chemin critique : c'est le chemin le long du graphe partant du sommet début et atteignant le sommet Fin. Ici ce chemin est modélisé par les flèches rouges.



Bien que nous ayons compris tout le principe du TP ainsi que les tâches demandées, nous n'avons pas réussi à produire un code correct car nous n'arrivons pas à implémenter correctement l'algorithme de Bellman.