LE PROBLEME CENTRAL DE L'ORDONNANCEMENT

Dans cette leçon, nous introduisons le problème central de l'ordonnancement de tâches et le modélisons par un problème de plus long chemin dans un graphe. Il devient alors possible de résoudre le problème central de l'ordonnancement en s'appuyant sur les résultats concernant la résolution des problèmes de plus long chemin dans un graphe.

I Définition du problème

Considérons l'exemple suivant :

La construction d'un bâtiment peut être décomposée de manière très schématique dans les activités ou tâches suivantes :

Fondation et maçonnerie
Plan des aménagements intérieurs
Toiture
Installations électriques et sanitaires
Façade
Peintures intérieures

L'exemple est volontairement très simplifié.

Il s'agit de planifier ces différentes activités et plus précisément de déterminer pour chaque tâche la date de début de son exécution. Pour cela, on dispose pour chaque tâche des informations suivantes :

- sa durée
- les tâches qui doivent être terminées afin qu'elle puisse commencer

Les données du problème peuvent être résumées dans le tableau suivant :

Tâches		Durée	Prédécesseurs
Fondation et maçonnerie	A	7	//
Plan des aménagements intérieurs	B	8	//
Toiture	C	2	A
Installations électriques et sanitaires	D	4	A et C
Façade	E	3	C et D
Peintures intérieures	F	1	C et D

Par exemple, la tâche maçonnerie est prévue pour durer 7 semaines et peut commencer dès le début alors que la toiture qui dure 2 semaines ne peut être entreprise que si la maçonnerie est terminée.

Le problème est de déterminer un calendrier d'exécution de ces tâches de manière à terminer les travaux dans les meilleurs délais.

Définition générale du problème central de l'ordonnancement

Un projet est découpé en un ensemble T de tâches pour lesquelles on dispose des informations suivantes :

- chaque tâche $i \in T$ a une durée d_i supposée connue avec certitude,
- ces tâches sont liées entre elles par des contraintes de succession,

- les tâches peuvent être affectées de contraintes de localisation temporelle : par exemple, date de début imposée pour une tâche.

Il s'agit de déterminer le calendrier d'exécution de ces tâches, compatible avec les contraintes, de manière à ce que toutes les tâches soient réalisées en un minimum de temps.

Remarque
Dans le problème central de l'ordonnancement, on ne tient pas compte de l'utilisation éventuelle de ressources humaines ou matérielles pour réaliser les tâches.

Définitions
Un **ordonnancement** est une solution réalisable du problème.
C'est donc un calendrier possible.

Un **ordonnancement optimal** est une solution ...optimale !
C'est un calendrier d'exécution qui conduit à la durée totale la plus courte.

II Modélisation du problème

On dispose d'un ensemble T de tâches.
Pour chacune d'elles, on a sa durée d_i.
Les différentes contraintes de succession ou de localisation temporelle ont été recensées.

Comme pour tout problème d'optimisation, on peut mettre en évidence les trois phases :
- que doit-on faire, quelles sont les décisions à prendre ?
- que peut-on faire, quelles sont les contraintes qui limitent ces décisions ?
- quel critère choisit-on entre plusieurs décisions ?

Les décisions à prendre
On souhaite déterminer pour chaque tâche la date à laquelle elle doit débuter.
On associe à chaque tâche $i \in T$ une **variable de décision** t_i représentant sa date de début.
On introduit deux tâches fictives : α représentant le début des travaux et ω représentant la fin des travaux. Ces deux tâches ont une durée nulle.

La date de début des travaux correspond à t.
On prend $t\alpha = 0$, c'est-à-dire que l'origine du temps est fixée à la date de début des travaux.
La date de fin des travaux sera mesurée par $t\omega$, date d'exécution de la tâche fictive ω.
Si $t\alpha = 0$, $t\omega$ représente aussi la durée des travaux.

Les décisions possibles
Il s'agit de respecter un certain nombre de contraintes portant sur les dates de début des tâches.
Dans le cadre du problème central de l'ordonnancement, on peut prendre en compte, par exemple, les contraintes suivantes :

1 - Contraintes de succession :
La tâche j ne peut commencer avant la fin de i.
2 - Contraintes de succession partielle :
La tâche j peut commencer dès qu'un pourcentage p_i de la tâche i est exécuté.
3 - Contraintes de succession immédiate :
La tâche j doit commencer dès que i est terminée.
4 - Date de disponibilité :

La tâche i ne peut commencer avant la date r_i.

5 - Date de livraison:

La tâche i doit être terminée avant la date l_i.

Contraintes représentant les décisions possibles :

Les contraintes précédentes ont comme propriété de pouvoir toutes être mises sous la forme :

$t_j \geq t_i + l_{ij}$ avec i et j $\in$T et l_{ij} un nombre réel.

Cas 1 : $t_j \geq t_i + d_i$

Cas 2 : $t_j \geq t_i + p_i d_i$

Cas 3 : $t_j = t_i + d_i$ soit $t_j \geq t_i + d_i$ et $t_j \leq t_i + d_i$ donc $t_j \geq t_i + d_i$ et $t_i \geq t_j - d_i$

Cas 4 : $t_i \geq r_i$ soit $t_i \geq t\alpha + r_i$ ($t\alpha = 0$)

Cas 5 : $t_i + d_i \leq l_i$ soit $t\alpha \geq t_i + d_i - l_i$

Toute contrainte entre les dates de début de tâches qui peut se mettre sous la forme $t_j \geq t_i + l_{ij}$ peut être prise en compte dans le cadre du problème central de l'ordonnancement.

Remarque

Parmi les contraintes du problème, il faut introduire celles traduisant que la tâche ω représentant la fin des travaux est postérieure à la fin de toutes les autres. De même, il faut traduire que les tâches sans prédécesseur ne peuvent commencer ... avant le début.

Le critère

Dans le cadre du problème central de l'ordonnancement, le critère retenu est la recherche d'un ordonnancement conduisant à la durée totale minimale.

Fonction objectif

ω représentant la tâche "fin" du projet, le critère peut être traduit par :

Min ($t\omega$)

La problème central de l'ordonnancement est alors modélisé par le problème d'optimisation suivant :

$$
(P) \left\{ \begin{array}{l} \text{Min } (t\omega), \\ t_j - t_i \geq l_{ij}, \\ t\alpha = 0 \end{array} \right.
$$

Les contraintes portent sur tous les couples (i, j) de tâches de T associées aux contraintes du problème.

Ecrit sous cette forme, le problème central de l'ordonnancement est modélisé par un problème de programmation linéaire que nous aborderons dans les prochaines leçons.

Cette modélisation, qui semble naturelle, n'est cependant pas la mieux adaptée pour la résolution de ce type de problème.

A la fin des années 50, des modélisations s'appuyant sur les graphes ont été proposées.

L'une conduit à la méthode PERT (1959 - Program Evaluation Research Task), l'autre à la méthode CPM (1960 - Méthode du chemin critique - Critical Path Method).

Nous développons ici cette dernière. Elle est encore appelée "Méthode des potentiels".

III Représentation des données du problème par un graphe : le graphe potentiel-tâche

Les données du problème vont être représentées par un graphe valué, pour lequel il faut définir ce que représentent les sommets, les arcs et les longueurs des arcs.

A chaque **tâche** de T on associe un **sommet** du graphe. Le sommet associé à α correspond au début des travaux et le sommet associé à ω correspond à la fin des travaux.

A chaque **contrainte** on associe un **arc :**

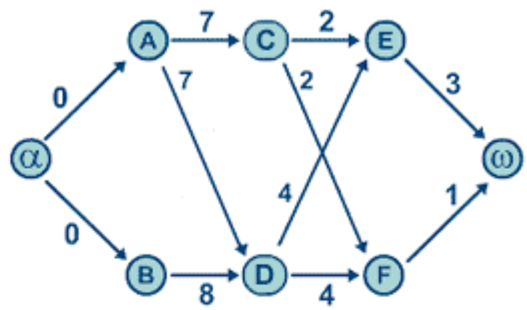
A la contrainte $t_j \geq t_i + l_{ij}$ est associé un arc (i, j) de **longueur** l_{ij}

Le graphe ainsi obtenu porte le nom de **graphe potentiel - tâche.**

Exemple

Dans l'exemple présenté, on n'a que des contraintes de succession.

	Durée	Prédécesseurs
A	7	//
B	8	//
C	2	A
D	4	A et C
E	3	C et D
F	1	C et D



A chaque tâche on associe un sommet, y compris aux tâches de début α et de fin ω.

La tâche A n'a a priori pas de prédécesseur mais elle ne peut commencer avant le début des travaux, d'où la contrainte $t_A \geq t\alpha$ représentée par un arc (α , A) de longueur 0. Il en est de même pour B.

La tâche C ne peut commencer avant la fin de A ce qui se traduit par :

$t_C \geq t_A + 7$ puisque A a une durée de 7.
Cette contrainte est modélisée par l'arc (A, C) de longueur 7.

Il faut aussi indiquer que la tâche ω ne peut intervenir avant que toutes le tâches ne soient terminées en particulier E et F.
On a, par exemple, $t_\omega \geq t_E + 3$ contrainte représentée par l'arc (E, ω) de longueur 3.

IV Résolution du problème

A - Ordonnancement au plus tôt

On construit un premier ordonnancement optimal appelé "ordonnancement au plus tôt".

Définition
On appelle **date de début au plus tôt** d'une tâche la plus petite date à laquelle elle peut débuter si toutes les contraintes sont respectées.
Le calendrier de l'ensemble des tâches est appelé **"ordonnancement au plus tôt".**

Soit un graphe associé à un problème d'ordonnancement et soit $C_{\alpha i}$ un chemin du sommet α au sommet associé à la tâche i.
Pour tous les arcs (h, k) de ce chemin, on a, par construction, la propriété : $t_k \geq t_h + l_{hk}$.
Si on additionne ces inégalités pour tous les arcs du chemin, on arrive à :
$t_i \geq t_\alpha +$ somme des longueurs des arcs du chemin $C_{\alpha i}$
Comme $t_\alpha = 0$, on en déduit que la date de début de la tâche i est au moins égale à la longueur du chemin.
Ce résultat est valable pour tous les chemins de α au sommet i, la date de début au plus tôt de la tâche i est égale à la longueur du plus long chemin de α à i.

Si on prend en particulier le sommet ω associé à la fin des travaux, la date t_ω qui représente la durée des travaux est au moins égale à la longueur de n'importe quel chemin de α à ω, donc au plus long d'entre eux.

D'où le résultat :

Proposition
La date de début au plus tôt d'une tâche est égale à la longueur du plus long chemin de α au sommet représentant cette tâche dans le graphe potentiel-tâche.
La durée minimale des travaux est égale à la longueur du plus long chemin de α à ω dans le graphe potentiel-tâche.

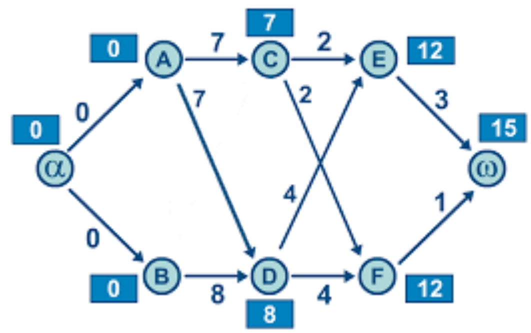
Calcul des dates au plus tôt
Ce calcul revient à celui de la longueur d'un plus long chemin. On peut donc utiliser les algorithmes adaptés à la détermination de plus longs chemins.

En particulier, si les seules contraintes sont des contraintes de succession, le graphe potentiel-tâche est sans circuit ; on peut donc utiliser l'algorithme de Bellman.
On prend les sommets dans un ordre tel que chacun n'est examiné qu'après que chacun de ses prédécesseurs a été examiné (tri topologique)

Par application de l'algorithme de Bellman, la date de début au plus tôt t_i^* de la tâche i est calculée par :

$$t_i^* = \mathrm{Max}(t_j^* + d_j) \text{ avec } j \in \mathrm{Pred}(i)$$

Exemple



Les dates de début au plus tôt sont calculées dans l'ordre :

$t_\alpha^* = 0$, $t_A^* = 0$ et $t_B^* = 0$, $t_C^* = 7$ et $t_D^* = 8$, $t_E^* = 12$ et $t_F^* = 12$, $t_\omega^* = 15$

Par exemple pour le graphe ci-dessus on a :

$t_E^* = \mathrm{Max}(t_C^* + 2, t_D^* + 4) = \mathrm{Max}(7 + 2, 8 + 4) = 12$

La durée minimale des travaux est égale à 15 date de début au plus tôt t_ω^* de ω.

Le calcul des dates au plus tôt repose sur la formule $t_i^* = \mathrm{Max}(t_j^* + d_j)$ avec $j \in \mathrm{Pred}(i)$

Si on interprète cette formule de calcul de l'algorithme de Bellman en terme de date au plus tôt, on obtient que la date de début au plus tôt d'une tâche est égale à la date de début au plus tôt des tâches qui la précèdent immédiatement, augmentée de la durée de ces tâches ; ce qui se justifie évidemment a posteriori.

B - Ordonnancement au plus tard

Le calendrier précédent conduit à une durée minimale de 15.

Il s'agit maintenant de déterminer la date à laquelle chacune des tâches doit impérativement avoir commencé si on veut que la durée totale des travaux soit respectée.

Définition

On appelle **date de début au plus tard** d'une tâche la date à laquelle elle doit impérativement avoir commencé afin que la date de fin de travaux soit respectée.

Le calendrier correspondant est l'**ordonnancement au plus tard.**

Principe de calcul
Sur l'exemple précédent, considérons par exemple la tâche C. Pour déterminer la date à laquelle elle doit impérativement commencer pour que la fin des travaux intervienne à la date 15, il faut s'intéresser à ses deux successeurs E et F.

L'interprétation des arcs du chemin $CE\omega$ nous indique qu'il faut au minimum 5 unités de temps avant la fin (2 pour C et 3 pour E).
De même, l'interprétation des arcs du chemin $CF\omega$ indique qu'il faut 2 unités de temps pour C et une unité pour F soit 3 unités de temps.
La tâche F pouvant être faite en parallèle avec E, l'analyse de ces résultats montre qu'il faut au minimum 5 unités de temps après le début de C avant la fin des travaux. Cette durée correspond à la longueur du plus long chemin de C à ω.

Proposition
La date de début au plus tard d'une tâche est égale à la différence entre la date de fin des travaux et la longueur du plus long chemin du sommet représentant cette tâche dans le graphe potentiel-tâche au sommet ω.

Le calcul des dates au plus tard revient donc à un calcul de plus long chemin dans un graphe.

On peut, en s'inspirant des algorithmes développés pour l détermination de plus longs chemins, mettre en place un algorithme permettant d'affecter à chaque sommet une étiquette $\lambda(i)$ de valeur égale à la longueur du plus long chemin de ce sommet à ω et en déduire la date de début au plus tard, et ceci quelles que soient les propriétés du graphe potentiel-tâche.

Soit $T\omega$ la date de fin des travaux.
La date de début au plus tard Ti * d'une tâche i est égale à $T\omega$ - longueur du plus long chemin de i à ω.
$Ti^* = T\omega - \lambda(i)$.
Il faut donc calculer la longueur $\lambda(i)$ d'un plus long chemin de i à ω.

Nous allons ici considérer le cas particulier d'un graphe sans circuit et, dans ce cas, établir une formule permettant de déterminer directement les dates de début au plus tard.

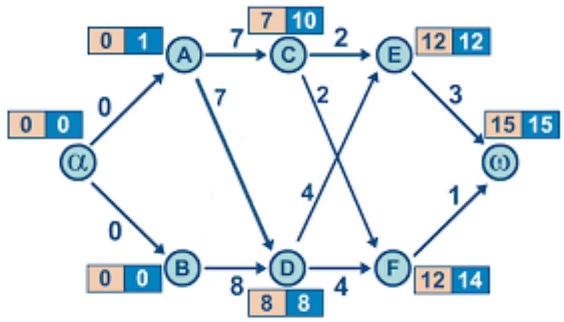
Calcul des dates au plus tard dans un graphe sans circuit
Par analogie avec l'algorithme de Bellman pour le calcul de la longueur des plus courts chemins, le calcul de la longueur d'un plus long chemin repose sur le résultat :

$\lambda(i) = \text{Max}\,(l(i, j) + \lambda(j))$ le max étant prit sur les successeurs de i et $\lambda(j)$ étant égal à la longueur d'un plus long chemin de j à ω.

On a donc :
$Ti^* = T\omega - \lambda(i) = T\omega - \text{Max}\,(l(i, j) + \lambda(j))$
$= \text{Min}(T\omega - \lambda(j) - l(i, j))$, le min étant pris sur les successeurs j de i.
d'où le résultat :
$T_i^* = \text{Min}\,(T_j^* - l(i,j))$ le min étant pris sur les successeurs j de i.

Cette formule permet de calculer directement les dates de début au plus tard de chaque tâche.
Il faut prendre les tâches dans un ordre tel que chaque tâche n'est examinée qu'après que tous ses successeurs l'aient été (ce qui est possible si le graphe est sans circuit).

Exemple



Les tâches sont examinées dans l'ordre suivant :
$T\omega = 15$, $T_E^* = 12$ ou $T_F^* = 14$, $T_C^* = 10$ ou $T_D^* = 8$, $T_A^* = 1$ ou $T_B^* = 0$, $T\alpha = 0$

Par exemple, pour D on calcule :
$\mathrm{Min}(T_E^* - l(D, E), T_F^* - l(D, F)) = \mathrm{Min}(12 - 4, 14 - 4) = 8$

Ce tableau récapitule les 2 calendriers particuliers, ordonnancement au plus tôt et au plus tard, conduisant à la durée totale minimale de 15.

	Durée	Date de début au plus tôt	Date de début au plus tard
α	0	0	0
A	7	0	1
B	3	0	0
C	2	7	10
D	4	8	8
E	3	12	12
F	1	12	14
α	0	15	15

C - Tâches critiques, chemin critique

Dans l'exemple précédent, la tâche D a une date de début au plus tôt égale à sa date de début au plus tard : la date de début est donc impérative si on veut que les travaux soient réalisés dans une durée minimale.

Définition
Une **tâche critique** est une tâche dont les dates de début au plus tôt et au plus tard coïncident.

Tout retard sur les tâches critiques retarde la fin des travaux.

Définition

La **marge totale** d'une tâche est égale à la différence entre la date début au plus tard et la date de début au plus tôt.

Les tâches critiques sont donc les tâches de marge totale nulle.

Exemple

Les tâches critiques sont ici α, B, D, E et ω.

Ces tâches critiques sont situées sur le chemin α B D E ω de longueur 15 qui est le plus long chemin de α à ω.

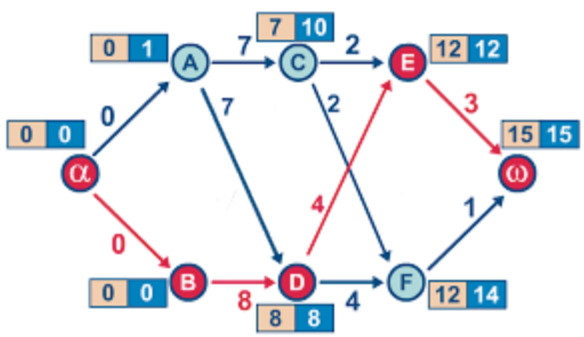
Définition

Le **chemin critique** est le plus long chemin du sommet "début" au sommet "fin".

Proposition

Les sommets du chemin critique sont les tâches critiques.

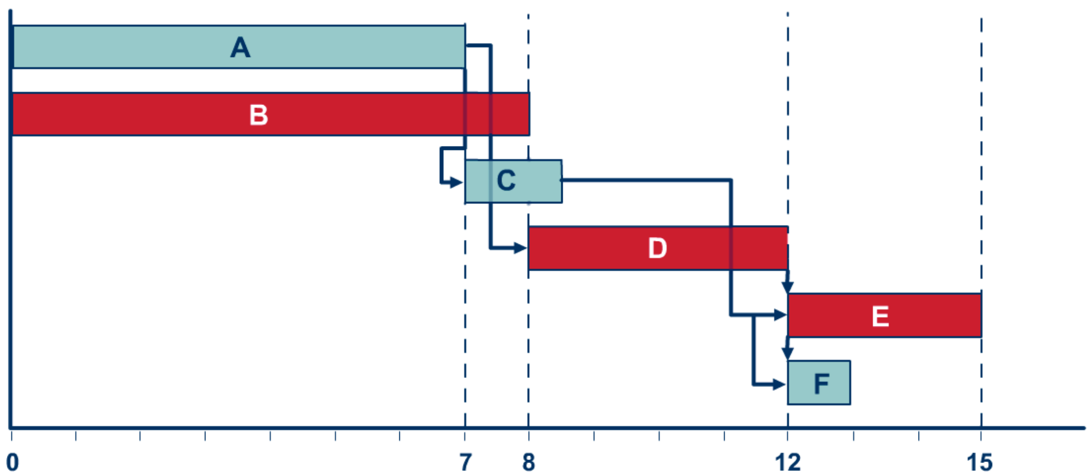
	Durée	Date de début au plus tôt	Date de début au plus tard	Marge totale
α	0	**0**	**0**	**0**
A	7	0	1	1
B	3	**0**	**0**	**0**
C	2	7	10	3
D	4	**8**	**8**	**0**
E	3	**12**	**12**	**0**
F	1	12	14	2
ω	0	**15**	**15**	**0**



D - Diagramme de Gantt

On peut représenter le calendrier d'exécution des tâches par un diagramme appelé "diagramme de Gantt".
Chaque tâche est représentée par une barre de longueur proportionnelle à sa durée.
Les contraintes de succession entre tâche sont matérialisées par des flèches.

Si les tâches sont positionnées à leur date de début au plus tôt, on dit que les tâches sont calées à gauche.
On constate sur ce diagramme qu'il est possible de déplacer les tâches A, C et F tout en respectant les contraintes de succession sans retarder la fin des travaux.



Ordonnancement au plus tôt

E - Marge libre

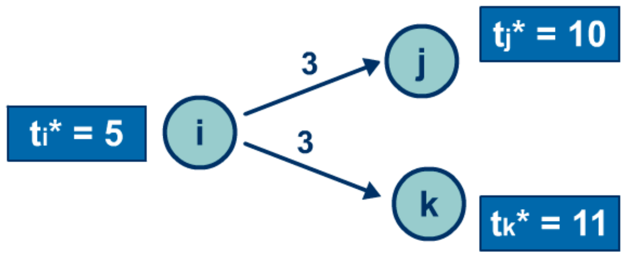
Toutes les tâches non-critiques ne sont pas équivalentes.
Parmi les tâches non-critiques, il en existe qui, si elles ne commencent pas à leur date au plus tôt, les suivantes ne peuvent pas non plus commencer à leur date au plus tôt.

On dira que leur marge libre est nulle.

Définition
La **marge libre** d'une tâche est le délai dont on dispose pour la retarder par rapport à sa date au plus tôt, en laissant la possibilité aux suivantes de commencer au plus tôt.

$$\text{Marge libre de } i = \min(t_j^* - l(i, j) - t_i^*) \text{ pour } j \in succ(i)$$

Exemple



Sur cet exemple, on connaît les dates au plus tôt des 3 tâches i, j, k.
On peut retarder i de 2 par rapport à sa date au plus tôt, tout en permettant à j de commencer à la date 10 et à k de commencer à la date 11.

L'ensemble des résultats est regroupé dans le tableau ci-dessous :

	Durée	Date de début au plus tôt	Date de début au plus tard	Marge totale	Marge libre
α	0	0	0	0	//
A	7	0	1	1	0
B	3	0	0	0	//
C	2	7	10	3	3
D	4	8	8	0	//
E	3	12	12	0	//
F	1	12	14	2	2
ω	0	15	15	0	//

F - Prise en compte de contraintes de localisation temporelle

Dans les paragraphes précédents, l'exemple traité ne comportait que des contraintes de succession.
Les résultats concernant les ordonnancements au plus tôt et au plus tard utilisent uniquement le fait qu'un arc de longueur l_{ij} représente une contrainte $t_j \geq t_i + l_{ij}$.
Ce type de représentation peut être associée à d'autres types de contraintes comme nous l'avons vu §II.

A une contrainte de succession partielle : la tâche j peut commencer dès qu'un pourcentage p_i de la tâche i est exécuté :

$$t_j \geq t_i + p_i d_i \quad \text{on associe un arc (i, j) de longueur } p_i d_i$$

A une contrainte de succession immédiate : la tâche j doit commencer dès que i est terminée

$$t_j = t_i + d_i \quad \text{soit} \quad t_j \geq t_i + d_i \quad \text{et} \quad t_j \leq t_i + d_i$$

on associe 2 arcs : l'arc (i, j) de longueur d_i et l'arc (j, i) de longueur - d_i

A une contrainte portant sur une date de disponibilité : la tâche i ne peut commencer avant la date r_i

$t_i \geq r_i$ soit $t_i - t\alpha \geq r_i$ on associe un arc (α , i) de longueur r_i

A une contrainte de date de livraison : la tâche i doit être terminée avant la date l_i

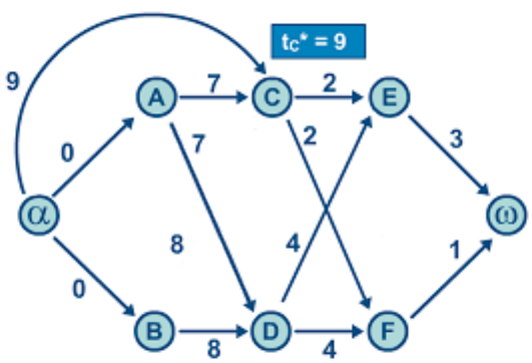
$t_i + d_i \leq l_i$ soit $t\alpha - t_i \geq d_i - l_i$ on associe un arc (i , α) de longueur $d_i - l_i$

Exemple 1

Actuellement la tâche C a pour date de début au plus tôt 7 et pour date de début au plus tard 10.
On ajoute la contrainte : la tâche C ne peut pas commencer avant la date 9.
$t_C \geq 9$ et en introduisant la tâche α $t_C - t\alpha \geq 9$ ou $t_C \geq t\alpha + 9$

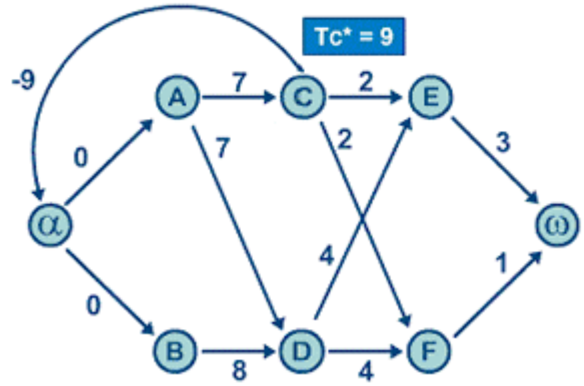
A cette contrainte, on associe sur le graphe un arc (α , C) de longueur 9.

Le calcul de la date au plus tôt de C revient à chercher le plus long chemin de α à C : c'est le chemin constitué de l'unique arc α C. Il a pour longueur 9.
On constate que l'introduction de cet arc permet de traduire correctement la contrainte imposée.
La marge totale de C étant égale à 3, l'adjonction de cette contrainte ne remet pas en cause la durée totale des travaux.



Exemple 2

Considérons la contrainte suivante :
La tâche C doit être commencée avant la date 9 alors que sa date de début au plus tard est actuellement de 10.
$t_C \leq 9$ soit $t_C - t\alpha \leq 9$ soit $t\alpha \geq t_C - 9$
Cette contrainte se traduit par un arc de C à α de longueur -9.



On constate que le graphe possède maintenant un circuit α A C α, et qu'il existe des arcs de longueur positive et négative.
Le calcul des dates au plus tôt et au plus tard nécessite l'utilisation d'un algorithme général déduit de l'algorithme de Ford.
Le plus long chemin de α à ω n'est pas changé.
La date de début au plus tôt de C est inchangée mais il existe maintenant un nouveau chemin de C à ω :
C α B D E ω de longueur -9 + 15 = 6.
La date de début au plus tard de C, qui est égale à la durée totale diminuée de la longueur du plus long chemin de C à ω, passe à 15 - 6 soit 9 ce qui était souhaitée.

V Le Graphe PERT : graphe potentiel - étapes

Une autre modélisation du problème central de l'ordonnancement par un graphe a été proposée. Ce graphe porte le nom de graphe PERT ou graphe potentiel-étape. Nous donnons ici quelques éléments sur cette modélisation.

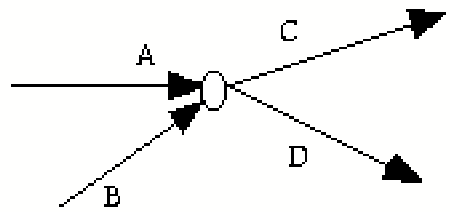
Dans cette représentation, les arcs sont associés aux tâches; ils sont valués par la durée des tâches, et les sommets représentent certains événements qui regroupent en général la fin de certaines tâches et le début d'autres.

Le graphe est moins facile à représenter ; il faut définir les événements correspondant aux sommets.
Certaines contraintes de succession nécessitent l'introduction de tâches fictives.
Enfin, la prise en compte de contraintes qui ne sont pas des contraintes de succession peut être plus délicates.

Supposons par exemple que l'on ait les tâches suivantes A, B, C, D avec :
A précède C et D
B précède D

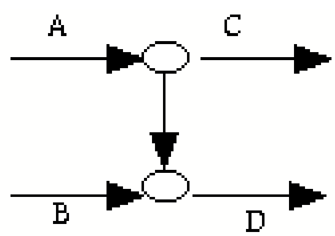
A ces 4 tâches sont associés 4 arcs : A, B, C, D.

A précède C et D se traduit par : l'extrémité de l'arc correspondant à la tâche A coïncide avec l'origine de l'arc correspondant à la tâche C et avec l'origine de l'arc correspondant à la tâche D.
B précède D est traduit par : l'extrémité de l'arc correspondant à la tâche B coïncide avec l'origine de l'arc correspondant à la tâche D. Cela peut conduire à la représentation suivante :



Dans cette représentation, C et D ont même origine ce qui impose la contrainte : B précède également C qui n'était pas dans les données du problème.

Il faut alors introduire une tâche fictive de longueur 0.



Dans cette représentation, le calcul des dates de début au plus tôt et au plus tard revient également à un calcul de plus long chemin.