Graphes

9. Ordonnancement

Solen Quiniou

solen.quiniou@univ-nantes.fr

IUT de Nantes

Année 2021-2022 – BUT 1 (Semestre 2)

[Mise à jour du 23 février 2022]

Plan du cours

Introduction

- Soit un projet découpé en n tâches $A_1, A_2, .., A_n$ ayant les propriétés suivantes :
  - tâches indivisibles ;
  - tâches de durée fixe d_i (pour la tâche A_i) ;
  - tâches soumises à un ensemble de contraintes.

$\rightarrow$ La problématique à résoudre est généralement la même : **déterminer un calendrier d'exécution des tâches respectant les contraintes**.

Plan du cours

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

- Une **contrainte potentielle** est une contrainte temporelle portant sur la date de début ou la date de fin de l'exécution d'une tâche :

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

- Une **contrainte potentielle** est une contrainte temporelle portant sur la date de début ou la date de fin de l'exécution d'une tâche :
  - $t_i \geq T$: la tâche A_i ne peut pas commencer avant la date T

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

- Une **contrainte potentielle** est une contrainte temporelle portant sur la date de début ou la date de fin de l'exécution d'une tâche :
  - $t_i \geq T$: la tâche A_i ne peut pas commencer avant la date T
  - $t_i \leq T$: la tâche A_i ne peut pas commencer après la date T

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

- Une **contrainte potentielle** est une contrainte temporelle portant sur la date de début ou la date de fin de l'exécution d'une tâche :
  - $t_i \geq T$: la tâche A_i ne peut pas commencer avant la date T
  - $t_i \leq T$: la tâche A_i ne peut pas commencer après la date T
  - $t_i + d_i \leq T$: la tâche A_i ne peut pas se terminer après la date T

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

- Une **contrainte potentielle** est une contrainte temporelle portant sur la date de début ou la date de fin de l'exécution d'une tâche :
  - $t_i \geq T$: la tâche A_i ne peut pas commencer avant la date T
  - $t_i \leq T$: la tâche A_i ne peut pas commencer après la date T
  - $t_i + d_i \leq T$: la tâche A_i ne peut pas se terminer après la date T
  - $t_i \geq t_j + d_j$: la tâche A_i ne peut pas commencer avant la fin de la tâche A_j

Contraintes potentielles

Notations

- Date de début de la tâche A_i : t_i
- Date de fin de la tâche A_i : $t_i + d_i$
- Tâche fictive de début de projet : A_0 (t_0 : date de début du projet)
- Tâche fictive de fin de projet : A_{n+1} (t_{n+1} : date de fin du projet)
- Durée du projet : $t_{n+1} - t_0$ (en général, on a $t_0 = 0$)

Définition : contrainte potentielle

- Une **contrainte potentielle** est une contrainte temporelle portant sur la date de début ou la date de fin de l'exécution d'une tâche :
  - $t_i \geq T$: la tâche A_i ne peut pas commencer avant la date T
  - $t_i \leq T$: la tâche A_i ne peut pas commencer après la date T
  - $t_i + d_i \leq T$: la tâche A_i ne peut pas se terminer après la date T
  - $t_i \geq t_j + d_j$: la tâche A_i ne peut pas commencer avant la fin de la tâche A_j

→ **Ensemble des contraintes potentielles** : ensemble conjonctif de contraintes exprimées sous la forme d'un système d'inéquations $t_j \geq t_i + a_{i,j}$ ($i, j \in \{0, .., n+1\}$)

Contraintes disjonctives et contraintes implicites

Définition : contrainte disjonctive

- Une **contrainte disjonctive** est une contrainte temporelle exprimant des incompatibilités entre des tâches :
  - $t_i \geq t_j + d_j$ ou $t_j \geq t_i + d_i$: les tâches A_i et A_j ne peuvent pas être réalisées simultanément (une seule des deux contraintes doit être vérifiée)

Définition : contraintes implicites

Contraintes disjonctives et contraintes implicites

Définition : contrainte disjonctive

- Une **contrainte disjonctive** est une contrainte temporelle exprimant des incompatibilités entre des tâches :
  - $t_i \geq t_j + d_j$ ou $t_j \geq t_i + d_i$: les tâches A_i et A_j ne peuvent pas être réalisées simultanément (une seule des deux contraintes doit être vérifiée)

→ **Ensemble des contraintes disjonctives** : ensemble disjonctif de contraintes sous la forme d'un système d'inéquations

Définition : contraintes implicites

Contraintes disjonctives et contraintes implicites

Définition : contrainte disjonctive

- Une **contrainte disjonctive** est une contrainte temporelle exprimant des incompatibilités entre des tâches :
  - $t_i \geq t_j + d_j$ ou $t_j \geq t_i + d_i$: les tâches A_i et A_j ne peuvent pas être réalisées simultanément (une seule des deux contraintes doit être vérifiée)

→ **Ensemble des contraintes disjonctives** : ensemble disjonctif de contraintes sous la forme d'un système d'inéquations

Définition : contraintes implicites

- **Contrainte implicite de début** :
  $(\forall\ i = 1, \ldots, n+1) \quad t_i \geq t_0 \Rightarrow (x_0, x_i) \in \Gamma \wedge v(x_0, x_i) = 0$

Contraintes disjonctives et contraintes implicites

Définition : contrainte disjonctive

- Une **contrainte disjonctive** est une contrainte temporelle exprimant des incompatibilités entre des tâches :
  - $t_i \geq t_j + d_j$ ou $t_j \geq t_i + d_i$: les tâches A_i et A_j ne peuvent pas être réalisées simultanément (une seule des deux contraintes doit être vérifiée)

→ **Ensemble des contraintes disjonctives** : ensemble disjonctif de contraintes sous la forme d'un système d'inéquations

Définition : contraintes implicites

- **Contrainte implicite de début** :
  $(\forall\, i = 1, \ldots, n+1) \;\; t_i \geq t_0 \Rightarrow (x_0, x_i) \in \Gamma \wedge v(x_0, x_i) = 0$
- **Contrainte implicite de fin** :
  $(\forall\, i = 1, \ldots, n) \;\; t_{n+1} \geq t_i + d_i \Rightarrow (x_i, x_{n+1}) \in \Gamma \wedge v(x_i, x_{n+1}) = d_i$

Contraintes disjonctives et contraintes implicites

Définition : contrainte disjonctive

- Une **contrainte disjonctive** est une contrainte temporelle exprimant des incompatibilités entre des tâches :
  - $t_i \geq t_j + d_j$ ou $t_j \geq t_i + d_i$: les tâches A_i et A_j ne peuvent pas être réalisées simultanément (une seule des deux contraintes doit être vérifiée)

→ **Ensemble des contraintes disjonctives** : ensemble disjonctif de contraintes sous la forme d'un système d'inéquations

Définition : contraintes implicites

- **Contrainte implicite de début** :
  $(\forall\ i = 1, \ldots, n+1) \ \ t_i \geq t_0 \Rightarrow (x_0, x_i) \in \Gamma \wedge v(x_0, x_i) = 0$
- **Contrainte implicite de fin** :
  $(\forall\ i = 1, \ldots, n) \ \ t_{n+1} \geq t_i + d_i \Rightarrow (x_i, x_{n+1}) \in \Gamma \wedge v(x_i, x_{n+1}) = d_i$

→ Seules les **contraintes implicites non redondantes** avec les autres contraintes sont conservées dans le graphe

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
- **Contraintes implicites non-redondantes de début** :
  $\rightarrow$ A_0 : tâche fictive de début de projet
- **Contraintes implicites non-redondantes de fin** :
  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
- **Contraintes implicites non-redondantes de début** :
  - $\rightarrow$ A_0 : tâche fictive de début de projet
- **Contraintes implicites non-redondantes de fin** :
  - $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
- **Contraintes implicites non-redondantes de début** :
  - $\rightarrow$ A_0 : tâche fictive de début de projet
- **Contraintes implicites non-redondantes de fin** :
  - $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
- **Contraintes implicites non-redondantes de début** :
  - $\rightarrow$ A_0 : tâche fictive de début de projet
- **Contraintes implicites non-redondantes de fin** :
  - $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$

- **Contraintes implicites non-redondantes de début** :
  $\rightarrow$ A_0 : tâche fictive de début de projet

- **Contraintes implicites non-redondantes de fin** :
  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$

- **Contraintes implicites non-redondantes de début** :

  $\rightarrow$ A_0 : tâche fictive de début de projet

- **Contraintes implicites non-redondantes de fin** :

  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
- **Contraintes implicites non-redondantes de début** :
  - $\rightarrow$ A_0 : tâche fictive de début de projet
- **Contraintes implicites non-redondantes de fin** :
  - $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
  - $t_7 \geq t_6 + 3$

- **Contraintes implicites non-redondantes de début** :

  $\rightarrow$ A_0 : tâche fictive de début de projet

- **Contraintes implicites non-redondantes de fin** :

  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
  - $t_7 \geq t_6 + 3$

- **Contraintes implicites non-redondantes de début** :

  $\rightarrow$ A_0 : tâche fictive de début de projet
  - $t_1 \geq t_0 + 0$

- **Contraintes implicites non-redondantes de fin** :

  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
  - $t_7 \geq t_6 + 3$
- **Contraintes implicites non-redondantes de début** :

  $\rightarrow$ A_0 : tâche fictive de début de projet
  - $t_1 \geq t_0 + 0$
  - $t_2 \geq t_0 + 0$
- **Contraintes implicites non-redondantes de fin** :

  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
  - $t_7 \geq t_6 + 3$
- **Contraintes implicites non-redondantes de début** :

  $\rightarrow$ A_0 : tâche fictive de début de projet
  - $t_1 \geq t_0 + 0$
  - $t_2 \geq t_0 + 0$
  - $t_3 \geq t_0 + 0$
- **Contraintes implicites non-redondantes de fin** :

  $\rightarrow$ A_8 : tâche fictive de fin de projet

Exemple : contraintes potentielles et implicites

Soit le **problème d'ordonnancement** défini par le tableau suivant.

Tâches	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Durées	6	3	6	2	4	3	1
Contraintes	-	-	-	A_2 finie	A_2 finie	A_1 finie A_4 finie	A_3 finie A_5 finie A_6 finie

- **Contraintes potentielles** :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
  - $t_7 \geq t_6 + 3$

- **Contraintes implicites non-redondantes de début** :

  $\rightarrow$ A_0 : tâche fictive de début de projet
  - $t_1 \geq t_0 + 0$
  - $t_2 \geq t_0 + 0$
  - $t_3 \geq t_0 + 0$

- **Contraintes implicites non-redondantes de fin** :

  $\rightarrow$ A_8 : tâche fictive de fin de projet
  - $t_8 \geq t_7 + 1$

Plan du cours

Problème central de l'ordonnancement

Définition : ordonnancement compatible

Un **ordonnancement compatible** est un vecteur $T = (t_0, t_1, \ldots, t_{n+1})$, avec $t_i \in \mathbb{R}^+$, pour lequel les contraintes de chaque tâche sont vérifiées et où

- t_i est la date de début de la tâche A_i ($i = 1, \ldots, n$)
- t_0 est la date de début du projet
- t_{n+1} est la date de fin du projet

Problème central de l'ordonnancement

Définition : ordonnancement compatible

Un **ordonnancement compatible** est un vecteur $T = (t_0, t_1, \ldots, t_{n+1})$, avec $t_i \in \mathbb{R}^+$, pour lequel les contraintes de chaque tâche sont vérifiées et où

- t_i est la date de début de la tâche A_i ($i = 1, \ldots, n$)
- t_0 est la date de début du projet
- t_{n+1} est la date de fin du projet

Problème central de l'ordonnancement

- Parmi tous les ordonnancements compatibles, en trouver un de durée F donnée
  $\rightarrow$ Utilisation de la **méthode des potentiels**

Plan du cours

Modélisation à l'aide d'un graphe

Définition : graphe potentiels-tâches

Un **graphe potentiels-tâches** $G = (X, \Gamma, v)$ est un graphe orienté tel que

- $X = \{x_0, x_1, \ldots, x_{n+1}\}$ représente l'ensemble des sommets du graphe, chaque sommet x_i étant associé à une tâche A_i

Modélisation à l'aide d'un graphe

Définition : graphe potentiels-tâches

Un **graphe potentiels-tâches** $G = (X, \Gamma, v)$ est un graphe orienté tel que

- $X = \{x_0, x_1, \ldots, x_{n+1}\}$ représente l'ensemble des sommets du graphe, chaque sommet x_i étant associé à une tâche A_i
- Γ et v représentent l'ensemble des contraintes, chaque contrainte $t_j \geq t_i + a_{i,j}$ étant représentée par l'arc $(x_i, x_j) \in \Gamma$ de valuation $v(x_i, x_j) = a_{i,j}$

Modélisation à l'aide d'un graphe

Définition : graphe potentiels-tâches

Un **graphe potentiels-tâches** $G = (X, \Gamma, v)$ est un graphe orienté tel que

- $X = \{x_0, x_1, \dots, x_{n+1}\}$ représente l'ensemble des sommets du graphe, chaque sommet x_i étant associé à une tâche A_i
- Γ et v représentent l'ensemble des contraintes, chaque contrainte $t_j \geq t_i + a_{i,j}$ étant représentée par l'arc $(x_i, x_j) \in \Gamma$ de valuation $v(x_i, x_j) = a_{i,j}$

Remarques

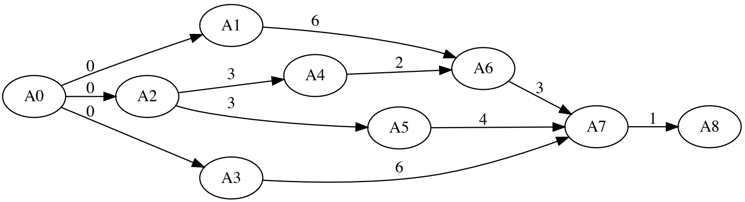
- $a_{i,j} \geq 0$ représente une **contrainte au plus tôt** sur la tâche A_j
- $a_{i,j} \leq 0$ représente une **contrainte au plus tard** sur la tâche A_i

Exemple : graphe potentiels-tâches

- Contraintes potentielles :
  - $t_4 \geq t_2 + 3$
  - $t_5 \geq t_2 + 3$
  - $t_6 \geq t_1 + 6$
  - $t_6 \geq t_4 + 2$
  - $t_7 \geq t_3 + 6$
  - $t_7 \geq t_5 + 4$
  - $t_7 \geq t_6 + 3$
- Contraintes implicites non-redondantes :
  - $t_1 \geq t_0 + 0$
  - $t_2 \geq t_0 + 0$
  - $t_3 \geq t_0 + 0$

  - $t_8 \geq t_7 + 1$

- **Graphe potentiels-tâches correspondant aux contraintes**



Existence d'ordonnancements compatibles

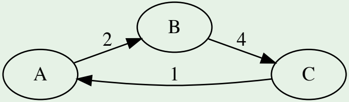
Théorème

Le problème admet des ordonnancements compatibles ssi le graphe potentiels-tâches correspondant n'admet aucun circuit de valeur strictement positive.

Contre-exemple : circuit de valuation positive

Soit les contraintes suivantes :

1. $t_B \geq t_A + 2$
2. $t_C \geq t_B + 4$
3. $t_A \geq t_C - 1 \;\Leftrightarrow\; t_C \leq t_A + 1$



On se retrouve avec une incompatibilité sur la tâche C. En effet, en fixant par exemple $t_A = 0$, on obtient à la fois

- Contraintes 1 et 2 : $t_B \geq 2$ et donc $t_C \geq 6$
- Contrainte 3 : $t_C \leq 1$

Cela est impossible.

Plan du cours

Ordonnancement au plus tôt

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive. L'**ordonnancement au plus tôt** est un ordonnancement compatible, de durée minimale, correspondant au vecteur $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ défini par :

- $\lambda_0 = 0$
- λ_i **= valeur maximale d'un chemin de** x_0 **à** x_i $(i = 1, \ldots, n+1)$

Remarques

Définitions : chemin critique, tâche critique

Ordonnancement au plus tôt

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive. L'**ordonnancement au plus tôt** est un ordonnancement compatible, de durée minimale, correspondant au vecteur $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ défini par :

- $\lambda_0 = 0$
- λ_i **= valeur maximale d'un chemin de x_0 à x_i** $(i = 1, \ldots, n+1)$

Remarques

- λ_i correspond à la **date de début au plus tôt** de la tâche A_i

Définitions : chemin critique, tâche critique

Ordonnancement au plus tôt

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive. L'**ordonnancement au plus tôt** est un ordonnancement compatible, de durée minimale, correspondant au vecteur $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ défini par :

- $\lambda_0 = 0$
- λ_i **= valeur maximale d'un chemin de** x_0 **à** x_i $(i = 1, \ldots, n+1)$

Remarques

- λ_i correspond à la **date de début au plus tôt** de la tâche A_i
- L'ordonnancement Λ est tel que, $\forall\ T = (t_0 = 0, t_1, \ldots, t_{n+1})$ ordonnancement compatible, on a $\lambda_i \leq t_i$ $(\forall\ i = 0, 1, \ldots, n+1)$

Définitions : chemin critique, tâche critique

Ordonnancement au plus tôt

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive. L'**ordonnancement au plus tôt** est un ordonnancement compatible, de durée minimale, correspondant au vecteur $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ défini par :

- $\lambda_0 = 0$
- λ_i **= valeur maximale d'un chemin de** x_0 **à** x_i $(i = 1, \ldots, n+1)$

Remarques

- λ_i correspond à la **date de début au plus tôt** de la tâche A_i
- L'ordonnancement Λ est tel que, $\forall\ T = (t_0 = 0, t_1, \ldots, t_{n+1})$ ordonnancement compatible, on a $\lambda_i \leq t_i$ $(\forall\ i = 0, 1, \ldots, n+1)$

Définitions : chemin critique, tâche critique

- Un **chemin critique** est un chemin de valeur maximale de x_0 à x_{n+1}

Ordonnancement au plus tôt

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive. L'**ordonnancement au plus tôt** est un ordonnancement compatible, de durée minimale, correspondant au vecteur $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ défini par :

- $\lambda_0 = 0$
- λ_i **= valeur maximale d'un chemin de x_0 à x_i** $(i = 1, \ldots, n+1)$

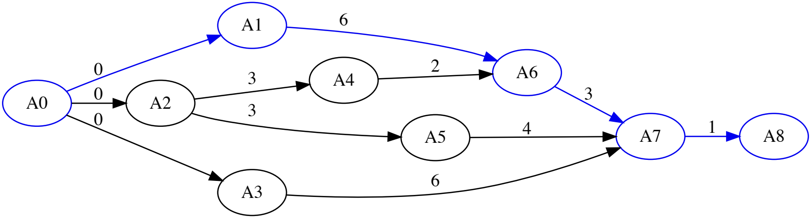
Remarques

- λ_i correspond à la **date de début au plus tôt** de la tâche A_i
- L'ordonnancement Λ est tel que, $\forall\ T = (t_0 = 0, t_1, \ldots, t_{n+1})$ ordonnancement compatible, on a $\lambda_i \leq t_i$ $(\forall\ i = 0, 1, \ldots, n+1)$

Définitions : chemin critique, tâche critique

- Un **chemin critique** est un chemin de valeur maximale de x_0 à x_{n+1}
- Une **tâche critique** est une tâche appartenant à un chemin critique
  - $\rightarrow$ Tout retard pris sur une tâche critique augmente la durée du projet

Exemple : ordonnancement au plus tôt



Tâches	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Ordonnancement au plus tôt (λ_i)	0	0	0	0	3	3	6	9	**10**

La durée minimale du projet est donc égale à 10 (longueur du chemin critique, en bleu).

Plan du cours

Ordonnancement au plus tard

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive, F une durée fixée et λ_{n+1} la durée minimale du projet.

- Si $\lambda_{n+1} > F$ alors il n'existe pas d'ordonnancement compatible de durée F

Remarques

Ordonnancement au plus tard

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive, F une durée fixée et λ_{n+1} la durée minimale du projet.

- Si $\lambda_{n+1} > F$ alors il n'existe pas d'ordonnancement compatible de durée F
- Si $\lambda_{n+1} \leq F$ alors l'**ordonnancement au plus tard de durée F** correspond au vecteur $\Lambda' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_{n+1})$ défini par :
  - $\lambda'_0 = 0$
  - $\lambda'_i = F - \alpha_i$ avec α_i la valeur maximale d'un chemin de x_i à x_{n+1} $(i = 1, \ldots, n+1)$

Remarques

Ordonnancement au plus tard

Théorème

Soit G un graphe ne possédant pas de circuit de valeur strictement positive, F une durée fixée et λ_{n+1} la durée minimale du projet.

- Si $\lambda_{n+1} > F$ alors il n'existe pas d'ordonnancement compatible de durée F
- Si $\lambda_{n+1} \leq F$ alors l'**ordonnancement au plus tard de durée F** correspond au vecteur $\Lambda' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_{n+1})$ défini par :
  - $\lambda'_0 = 0$
  - $\lambda'_i = F - \alpha_i$ avec α_i la valeur maximale d'un chemin de x_i à x_{n+1} $(i = 1, \ldots, n+1)$

Remarques

- λ'_i correspond à la **date de début au plus tard** de la tâche A_i, relativement à la durée F

Ordonnancement au plus tard

Théorème

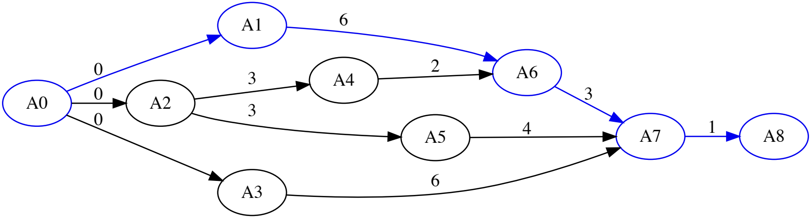
Soit G un graphe ne possédant pas de circuit de valeur strictement positive, F une durée fixée et λ_{n+1} la durée minimale du projet.

- Si $\lambda_{n+1} > F$ alors il n'existe pas d'ordonnancement compatible de durée F
- Si $\lambda_{n+1} \leq F$ alors l'**ordonnancement au plus tard de durée F** correspond au vecteur $\Lambda' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_{n+1})$ défini par :
  - $\lambda'_0 = 0$
  - $\lambda'_i = F - \alpha_i$ avec α_i la valeur maximale d'un chemin de x_i à x_{n+1} $(i = 1, \ldots, n+1)$

Remarques

- λ'_i correspond à la **date de début au plus tard** de la tâche A_i, relativement à la durée F
- L'ordonnancement Λ' est tel que, $\forall\ T = (t_0 = 0, t_1, \ldots, t_{n+1})$ ordonnancement compatible avec $t_{n+1} = \lambda'_{n+1} = F$, on a $t_i \leq \lambda'_i$ $(\forall\ i = 1, \ldots, n+1)$

Exemple : ordonnancement au plus tard de durée 10



Tâches	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Ordonnancement au plus tard (λ'_i)	0	0	1	3	4	5	6	9	10

Plan du cours

Marges

Définitions : marges totales

- Soit $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ l'ordonnancement au plus tôt et $\Lambda' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_{n+1})$ l'ordonnancement au plus tard de durée $F \geq \lambda_{n+1}$. La **marge totale de la tâche A_i**, relativement à F, correspond à $m_i = \lambda'_i - \lambda_i$ $(i = 1, .., n)$

$\rightarrow$ C'est le retard maximum que peut prendre l'achèvement de A_i sans retarder la date F de fin de projet

Définitions : marges libres

Marges

Définitions : marges totales

- Soit $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ l'ordonnancement au plus tôt et $\Lambda' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_{n+1})$ l'ordonnancement au plus tard de durée $F \geq \lambda_{n+1}$. La **marge totale de la tâche A_i**, relativement à F, correspond à $m_i = \lambda'_i - \lambda_i$ $(i = 1, .., n)$

$\rightarrow$ C'est le retard maximum que peut prendre l'achèvement de A_i sans retarder la date F de fin de projet

Définitions : marges libres

- Soit $T = (t_0, t_1, \ldots, t_{n+1})$ un ordonnancement compatible. La **marge libre de la tâche A_i**, relativement à T, correspond à $\mu_i = \min_{x_j \in \Gamma(x_i)} \; t_j - t_i - a_{i,j}$ $(i = 1, .., n)$

Marges

Définitions : marges totales

- Soit $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n+1})$ l'ordonnancement au plus tôt et $\Lambda' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_{n+1})$ l'ordonnancement au plus tard de durée $F \geq \lambda_{n+1}$. La **marge totale de la tâche A_i**, relativement à F, correspond à $m_i = \lambda'_i - \lambda_i$ $(i = 1, .., n)$

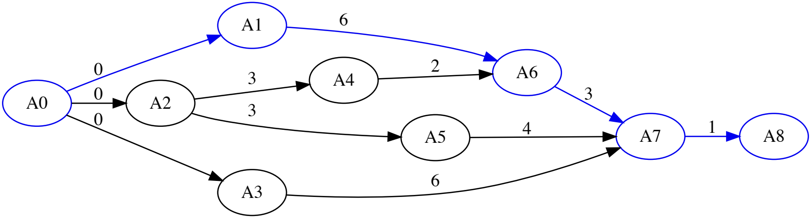
$\rightarrow$ C'est le retard maximum que peut prendre l'achèvement de A_i sans retarder la date F de fin de projet

Définitions : marges libres

- Soit $T = (t_0, t_1, \ldots, t_{n+1})$ un ordonnancement compatible. La **marge libre de la tâche A_i**, relativement à T, correspond à $\mu_i = \min_{x_j \in \Gamma(x_i)} \; t_j - t_i - a_{i,j}$ $(i = 1, .., n)$

$\rightarrow$ C'est le retard maximum que peut prendre l'achèvement de A_i sans retarder les dates de début t_j prévues pour les autres tâches

Exemple : marges totales et libres



Tâches	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Ordonnancement au plus tôt (λ_i)	0	0	0	0	3	3	6	9	10
Ordonnancement au plus tard (λ'_i)	0	0	1	3	4	5	6	9	10
Marges totales (m_i)	-	0	1	3	1	2	0	0	-
Marges libres pour ordo. au plus tôt (μ_i)	-	0	0	3	1	2	0	0	-