Recherche Opérationnelle :
TP 5 et 6

Sandra U. Ngueveu (ngueveu@laas.fr), Arthur Claviere, Aloïs Duguet,
Alexandre Dupaquis, Quentin Fabry

2020

Table des matières

L'objectif de ce TP est de résoudre le problème du job-shop par une *procédure de séparation et évaluation (PSE)* basée sur une *relaxation* équivalente à un calcul de plus long chemin dans un graphe particulier. Vous utiliserez vos codes de calcul de plus long chemin dans un graphe (cf TP3-4) pour résoudre ces relaxations.

Les notions de *procédures de séparation et évaluation = PSE*, *relaxations*, *relaxations linéaires*, *tests de sondabilité* (*TA*, *TO*, *TR*) ont été définies dans le cours.

1 Prise en main : ordonnancement avec contraintes de précédence

Problème (P1) : Soit un ensemble de tâches T, l'objectif est de déterminer les dates d'exécution de chaque tâche de manière à minimiser la durée totale d'exécution tout en respectant les durées et les contraintes de précédence entre les tâches.

Tâche i	A	B	C	D	E
Durée d_i	2	3	1	4	1
Condition de début	/	Après fin de A	Après fin de A	Après fin de B Après fin de C	Après fin de C

Tableau 1 – Données utilisées dans l'exemple du notebook P1

1.1 Modélisation classique par programmation linéaire (voir notebook P1)

Variables de décision :

— $t_i \in \mathbb{R}^+, \forall i \in T$: date de début de la tâche i

— $t_{\text{fin}} \in \mathbb{R}^+$: date de fin du projet

Le programme linéaire résultant fourni en Figure 1 est déjà implémenté dans le notebook P1.

1.2 Modélisation classique par graphe potentiel-tache (à faire)

Principes :

— chaque contrainte $t_j - t_i \geq a_{ij}$ est représentée par un arc de i à j et de valeur/longueur a_{ij}.

— le potentiel t_i correspond au début de la tâche i

— une tâche fictive de début peut précéder toutes les tâches sans prédécesseur

— une tâche fictive de fin succède à toutes les tâches sans successeur

$$\min t_{\text{fin}}$$

sous contraintes (s.c)

$$\begin{aligned}
t_2 - t_1 &\geq d_1 \\
t_3 - t_1 &\geq d_1 \\
t_4 - t_2 &\geq d_2 \\
t_4 - t_3 &\geq d_3 \\
t_5 - t_3 &\geq d_3 \\
t_{\text{fin}} - t_4 &\geq d_4 \\
t_{\text{fin}} - t_5 &\geq d_5 \\
t_i &\geq 0, \forall i \in 1..5
\end{aligned}$$

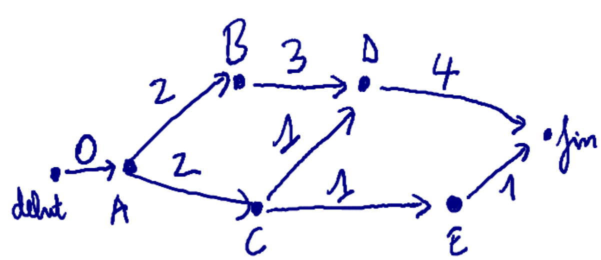

Figure 1 – Modèle linéaire et graphe potentiel-tâche associés aux données du Tableau 1

Appliquez votre algorithme de calcul de plus long chemin issu du TP3-4 sur le graphe potentiel-tâche correspondant au Tableau 1. Comparez la solution obtenue avec celle du programme linéaire.

2 Job-shop : ordonnancement avec contraintes de précédence et contraintes de ressources

Le problème du job-shop consiste à planifier un ensemble de travaux pour minimiser la durée totale d'exécution tout en respectant des contraintes de précédence (chaque travail est décomposé en opérations à réaliser dans l'ordre) et de ressources (chaque opération utilise une machine et chaque machine ne peut traiter qu'une opération à la fois).

Travaux	Opérations		
	1	2	3
1	6	7	/
2	3	5	1

(a)

Travaux	Opérations		
	1	2	3
1	M1	M3	/
2	M1	M2	M3

(b)

Tableau 2 – Exemple de données : (a) durées des opérations, (b) machines associées aux opérations

2.1 Relaxation des contraintes de ressources (à faire)

Soit une relaxation (R) qui consiste à ignorer les contraintes de ressources du job-shop. Montrer que (R) est équivalente au problème (P1). Construire et résoudre le programme linéaire et le graphe potentiel-tâche associés aux données du Tableau 2(a) pour (R). Montrer que la solution obtenue ne respecte pas les contraintes du Tableau 2(b).

2.2 Modélisation des contraintes de ressources

Principe : Si deux opérations ij et kl nécessitent une même ressource, alors il faut imposer que l'une se termine avant que l'autre ne commence. L'ordre n'étant pas connu à l'avance, il faut donc réussir à modéliser la contrainte $\boxed{\text{“}t_{ij} - t_{kl} \geq d_{kl} \text{ OU } t_{kl} - t_{ij} \geq d_{ij}\text{”}}$ qui n'est pas linéaire.

2.2.1 Méthode du bigM

L'idée est d'utiliser une variable binaire x_{kl_ij} qui vaudra 1 si kl est exécuté avant ij et 0 sinon.

$$t_{ij} - t_{kl} \geq d_{kl} - M(1 - x_{kl\_ij}) \quad \text{(C1)}$$

$$t_{kl} - t_{ij} \geq d_{ij} - Mx_{kl\_ij} \quad \text{(C2)}$$

$$0 \leq x_{kl\_ij} \leq 1 \quad \text{(C3)}$$

$$x_{kl\_ij} \in \mathbb{N} \quad \text{(C4)}$$

La Figure 2 présente le modèle mathématique correspondant aux Tableaux 2(a) et 2(b). Ce modèle n'est pas purement linéaire à cause des contraintes (15).

2.2.2 Méthode des graphes disjonctifs

L'idée est d'utiliser un graphe $G(X, U, D)$ appelé graphe disjonctif, qui est un outil pratique de modélisation pour ressources non partageables où :

- L'ensemble des sommets est X
- L'ensemble des arcs est composé de :
  - U : partie conjonctive représentant les gammes opératoires d'un travail
    - $\forall (ij, ik) \in U, t_{ik} - t_{ij} \geq p_{ij}$, représenté par un arc orienté de ij vers ik
  - D : partie disjonctive associée aux conflits d'utilisation d'une ressource non partageable
    - $\forall (ij, kl) \in D$, on a soit $t_{kl} - t_{ij} \geq p_{ij}$, soit $t_{ij} - t_{kl} \geq p_{kl}$, représenté par une arête ou paire de disjonction

Le graphe disjonctif correspondant aux données des Tableaux 2(a) et 2(b) est fourni ci-dessous. Les arêtes en pointillés à double tête y représentent les paires de disjonction.

$$\min t_{\text{fin}} \qquad (1)$$

sous contraintes (s.c)

$$\begin{aligned}
t_{1,2} - t_{1,1} &\geq d_{1,1} && (2)\\
t_{1,3} - t_{1,2} &\geq d_{1,2} && (3)\\
t_{2,2} - t_{2,1} &\geq d_{2,1} && (4)\\
t_{2,3} - t_{2,2} &\geq d_{2,2} && (5)\\
t_{\text{fin}} - t_{1,3} &\geq d_{1,3} && (6)\\
t_{\text{fin}} - t_{2,3} &\geq d_{2,3} && (7)\\
t_{2,1} - t_{1,1} + Mx_{2_1__1_1} &\geq d_{1,1} && (8)\\
t_{1,1} - t_{2,1} - Mx_{2_1__1_1} &\geq -M + d_{1,1} && (9)\\
t_{2,3} - t_{1,2} + Mx_{2_3__1_2} &\geq d_{1,2} && (10)\\
t_{1,1} - t_{2,1} - Mx_{2_3__1_2} &\geq -M + d_{1,2} && (11)\\
t_i &\geq 0 \quad \forall i \in 1..5 && (12)\\
0 \leq x_{2_1__1_1} &\leq 1 && (13)\\
0 \leq x_{2_3__1_2} &\leq 1 && (14)\\
x_{2_1__1_1}, x_{2_3__1_2} &\in \mathbb{N} && (15)
\end{aligned}$$

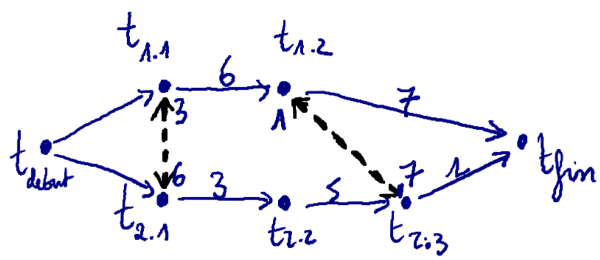

Figure 2 – Modèle et graphe disjonctif associés aux données du Tableau 2

2.3 Résolution (à faire)

En partant du notebook fourni, implémenter une PSE pour résoudre le problème du job-shop sur les données du Tableau 2, celles du Tableau 3 et si possible d'autres données de votre choix.

↓ Travail / Opération →	1	2	3
1	6 / M1	7 / M3	4 / M2
2	3 / M1	5 / M2	1 / M3

Tableau 3 – Autre jeu de données

2.3.1 PSE basée sur la relaxation linéaire du modèle avec bigM (voir notebook JS)

Principe : utiliser le solveur linéaire Clp (et non pas Cbc) pour résoudre les relaxations linéaires

Paramètres proposés

- Critère de séparation
  - une variable fractionnaire
- Tests de sondabilité
  - TA : réussi si la relaxation linéaire (RL) n'a pas de solution admissible
  - TO : réussi si la solution de la RL est pire que la meilleure solution connue
  - TR : réussi si la solution de la RL est entière
- Stratégie d'exploration : priorité à la profondeur

2.3.2 PSE basée sur le graphe disjonctif (à faire)

Principe : utiliser votre algorithme de calcul de plus long chemin pour résoudre les relaxations du graphe disjonctif

Paramètres proposés

- Principe de séparation
  - Choix d'un arc dans une paire de disjonction
- Fonction d'évaluation (borne inférieure)
  - Recherche du chemin critique dans un graphe possédant (1) tous les arcs conjonctifs et (2) les arcs disjonctifs des conflits déjà arbitrés
    - En déduire les tests TA, TO, TR résultants
- Stratégie d'exploration : priorité à la profondeur

3 BONUS

Tester différents critères de séparation et de stratégies d'exploration (ordre de choix des arcs disjonctifs à arbitrer) et analyser l'impact sur le temps de calcul et le nombre de noeuds créés par l'arborescence.

4 ANNEXE : Exemples d'arborescences résultant de PSE

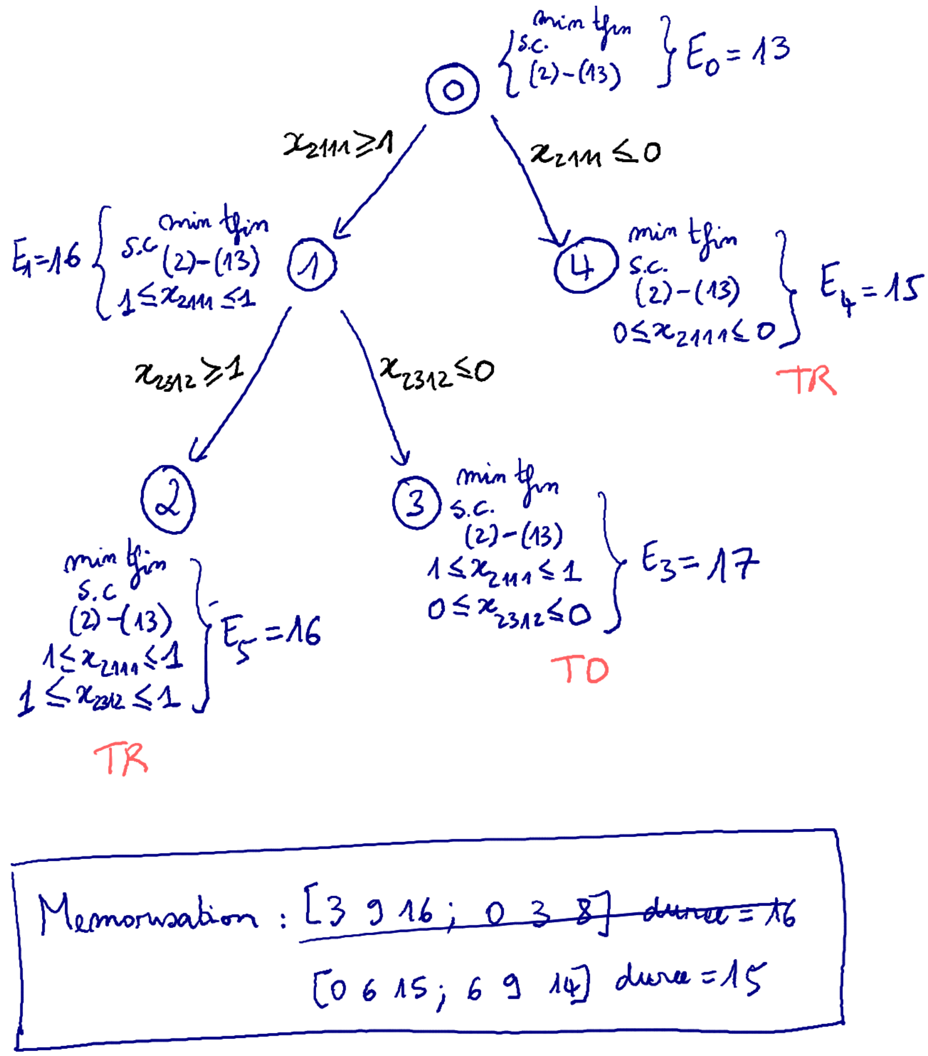

Figure 3 – PSE basée sur la relaxation linéaire avec les données du Tableau 2

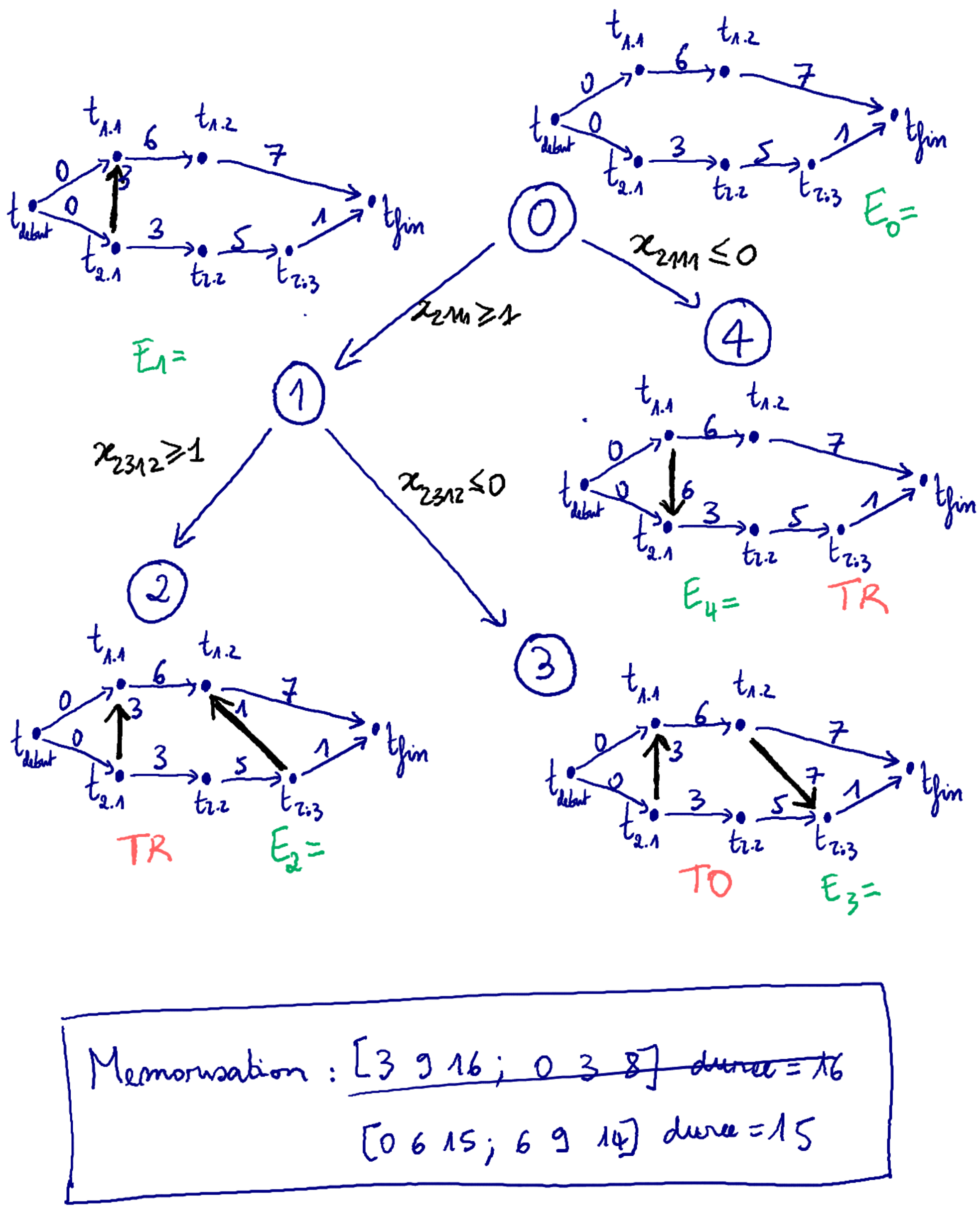

FIGURE 4 – PSE basée sur le graphe disjonctif avec les données du TABLEAU 2