ICO

- Ordonnancement
  (Optimisation) : différentes méthodes
  Métaheuristique

Fil rouge

Problème de Tournée de véhicule avec fenêtre de temps

Groupe de 6 → chaque binôme développe une métaheuristique

- Système multi-agents (SMA)
  → intégration des métaheuristique
  collaboration ou concurrence

- Apprentissage par renforcement

Rapport

Méta | Méta + SMA | Méta + SMA + Apprentissage

Ordonnancement

Réalisation d'un ensemble de tâches en considérant les contraintes temporelles et dues aux ressources

-> Entreprise : impératif de productivité, flexibilité, réactivité
-> Théorique : fortement combinatoire, pas de méthodes générales, -> complexité exp

5 questions
1) Quel produit
2) Où
3) Quand
4) Qui
5) Combien de temps

→ Définir les critères à optimiser

⇒ arbitrage

1 critère à optimiser : mono-objectif

x critères multi-objectifs

↳ Intégrale de Choquet

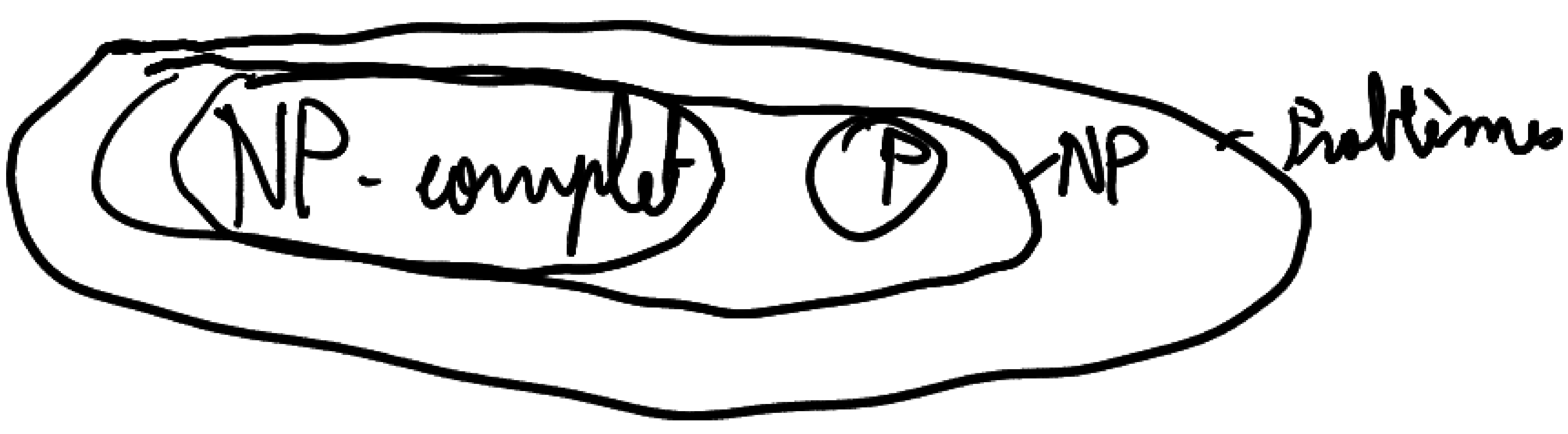
La complexité : évaluer le nombre d'opération en fonction du nombre d'opération

NP-difficile

Classe P : ensemble de tous les problèmes de reconnaissance polynomiaux

Classe NP : $P \subseteq NP$

Classe NP-complets : NP-complets (on sait qu'on existe pas de solution polynomial



On essaie de classer le problème

Problèmes d'ordonnancement

- open-shop (même opération mais pas dans le même ordre)
  <-> hôpital
- flow-shop (les pièces suivent le même chemin)
- job-shop (les pièces ne suivent pas le même chemin)

↳ Pour connaître la complexité : comparer

On sait quelles sont les tâches à exécuter

pb statique

on connaît à partir de quel moment peut débuter à n'importe quel moment

pb dynamique

non déterministe mais durée des tâches connues

pb stochastique

la durée des tâches ne sont pas connues (probabilité)

I : ensemble des tâches   Soit $i \in I$

$n = |I|$

p_i : durée d'exécution

r_i : date début au plus tôt

d_i : date de fin au plus tard

t_i : date de début réel

c_i : date de fin réel

$\boxed{r_i \leq t_i < c_i \leq d_i}$

$t_j - t_i \geq p_i \Leftrightarrow t_j \geq t_i + p_i$ i précède j

$\boxed{c_i = t_i + p_i}$

Si $c_i > d_i$: coût associé à une pénalité de retard

Voir si préemption

Ressources : renouvelable ou consommable

Critères : fonction économique

. variables : date c_i

retard $T_i = \max(0, c_i - d_i)$

indicateur de retard

$\begin{cases} U_i = 0 \text{ si } c_i \leq d_i \\ U_i = 1 \text{ sinon} \end{cases}$

. minimiser le makespan C_{max}

. respect des dates au plus tard

. minimisation d'un coût

. nombre d'interruptions

PDP : Plan directeur de production

OF : Ordre de fabrication

Planification

Ordonnancement

Pilotage

Temps opératoire p_{ijk} dépend de la machine choisie

décision d'affectation

décision d'attribution

Représenter les solutions : Gantt ressource

Machine

Produit

Gamme

Exemple 1 : évolution du cumul de moyens au cours du temps

Exemple 2 : ordonnancement à 5 tâches

Tâches	Durée	R1	R2	Instant
1	6	4	8	0
2	3	1	7	3
3	4	3	10	6
4	5	2	10	8
5	5	3	4	10

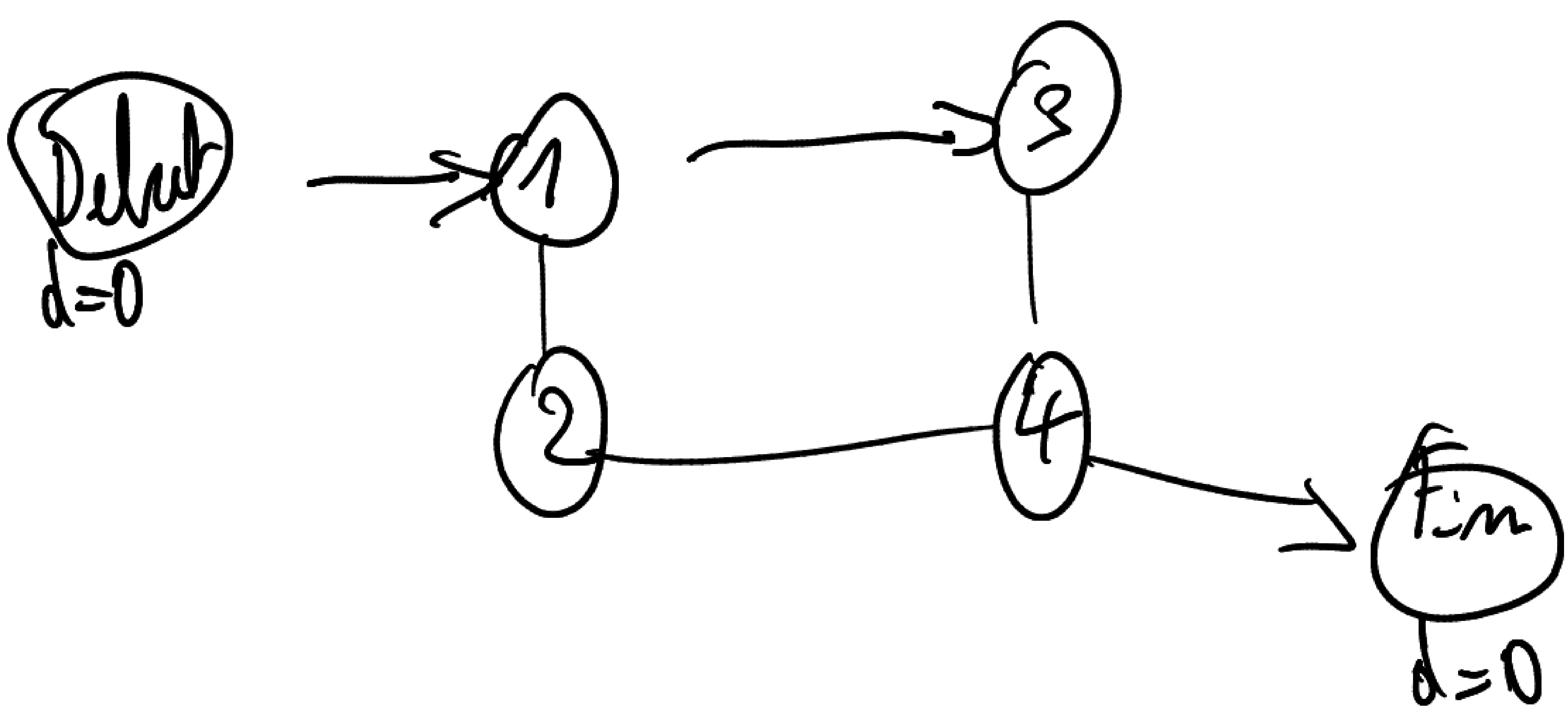
(i, j, k) tâche j du travail i qui s'exécute sur k

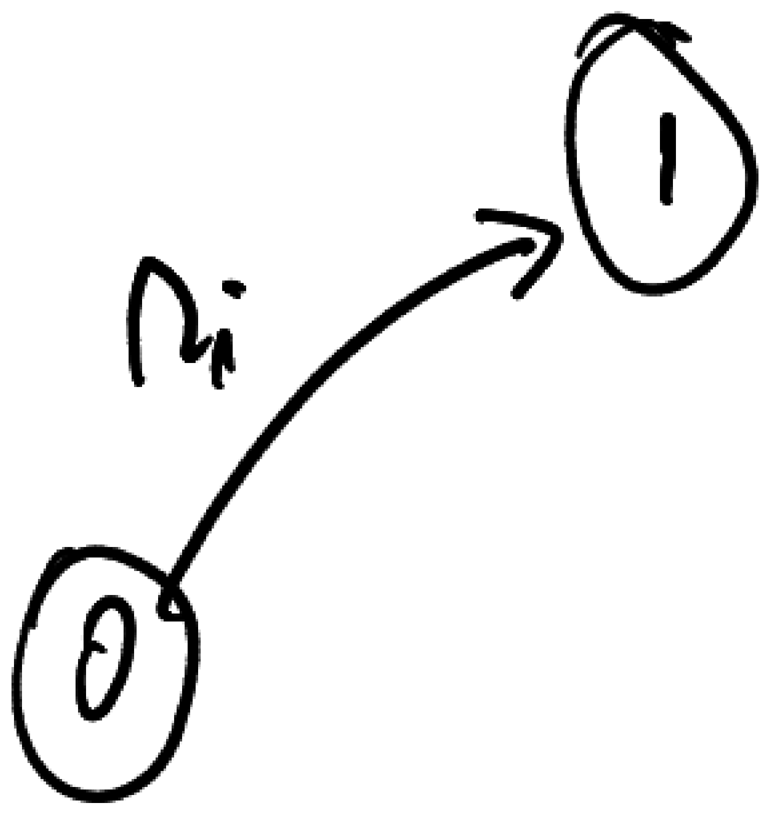
Graphe potentiels tâches

arcs : représentent les contraintes de potentiels

sommet : tâche

2 tâches fictives : début et fin de durée nulle



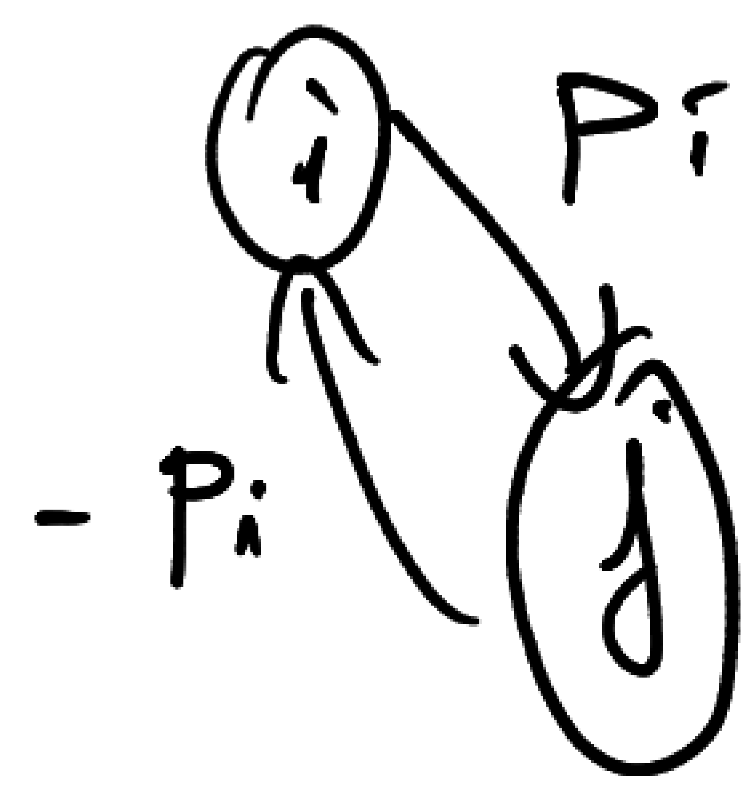


r_i : date de début au plus tôt (disponibilité)

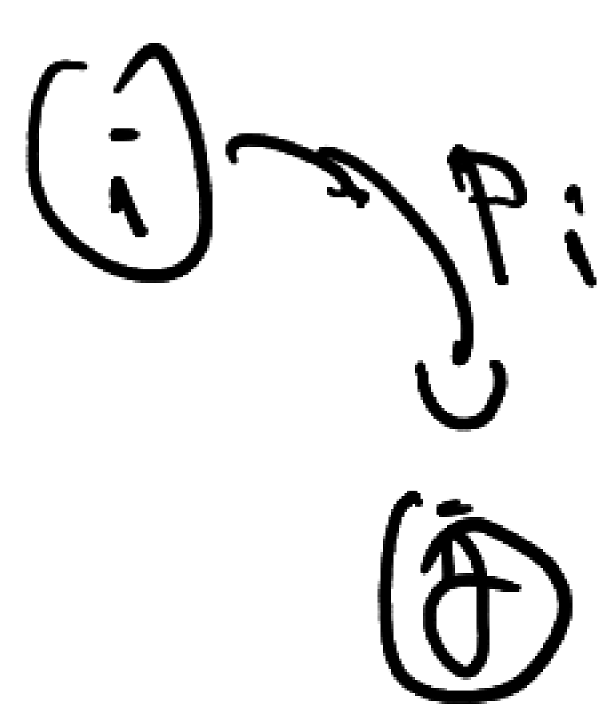
$$t_i - t_0 \geq r_i$$

"date au plutôt"

Succession immédiate



Succession large



Exemple 2 (Slide 45)

1) "Tache 2 commence à la date 3"

$$t_2 - t_0 \geq t_3 \quad \text{et} \quad t_0 - t_2 \geqslant -3$$

2) "Les taches 3 et 4 se chevauchent sur au moins une unité de temps"

$$t_3 \leq t_4 + p_4 - 1 \quad \text{et} \quad t_4 \leq t_3 + p_3 + 1$$

3) "La tache 4 ne peut commencer qu'après la fin des taches 1 et 2

$$t_4 \geq C_1 \quad \text{et} \quad t_4 \geqslant C_2$$

ou $t_4 - t_2 \geq p_2$ et $t_4 - t_1 \geq p_1$

4) "La tache 4 ne peut commencer qu'après le début de la tache 3"

$$t_5 - t_3 \geq 0$$

Solution optimale : chemin critique

Pour prendre en compte les ressources

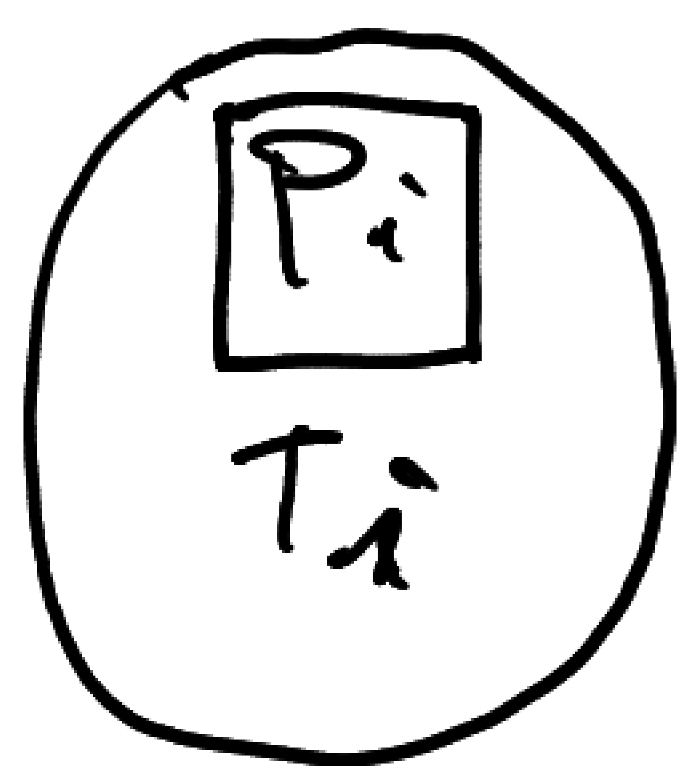
graphe PERT

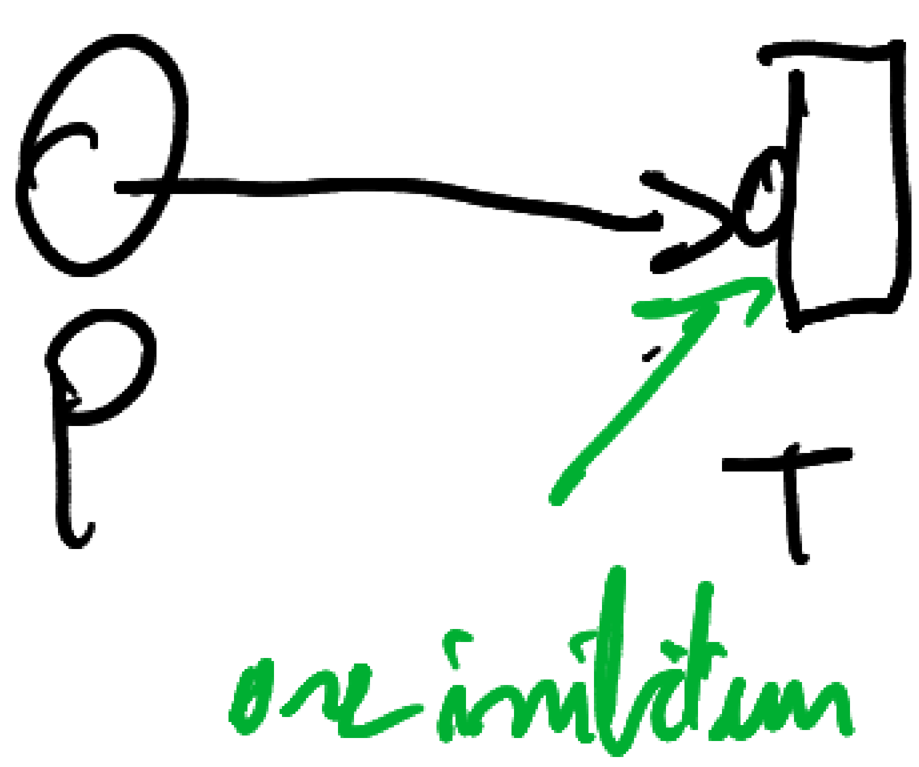
arc : tâches réel/fictives

sommet : évènements D1, F1 par ex.

Réseau de pétri RdPT

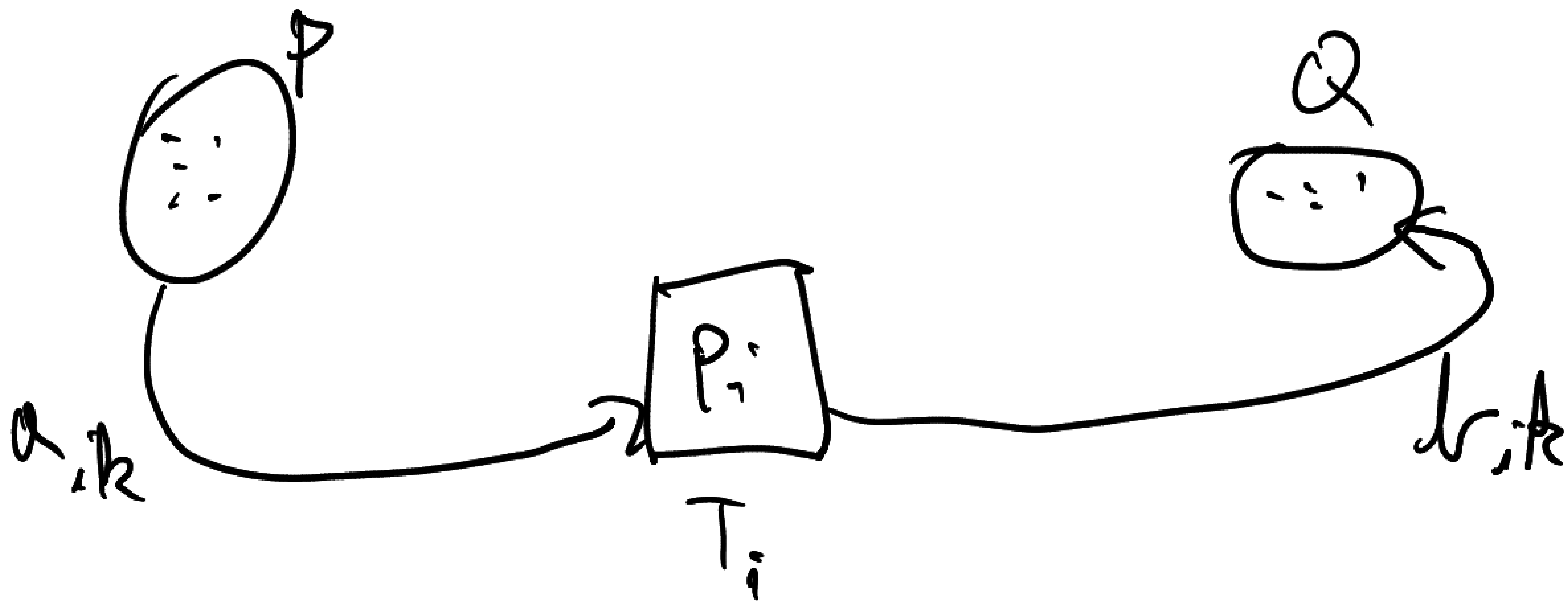
Règle 4





Voir diapo

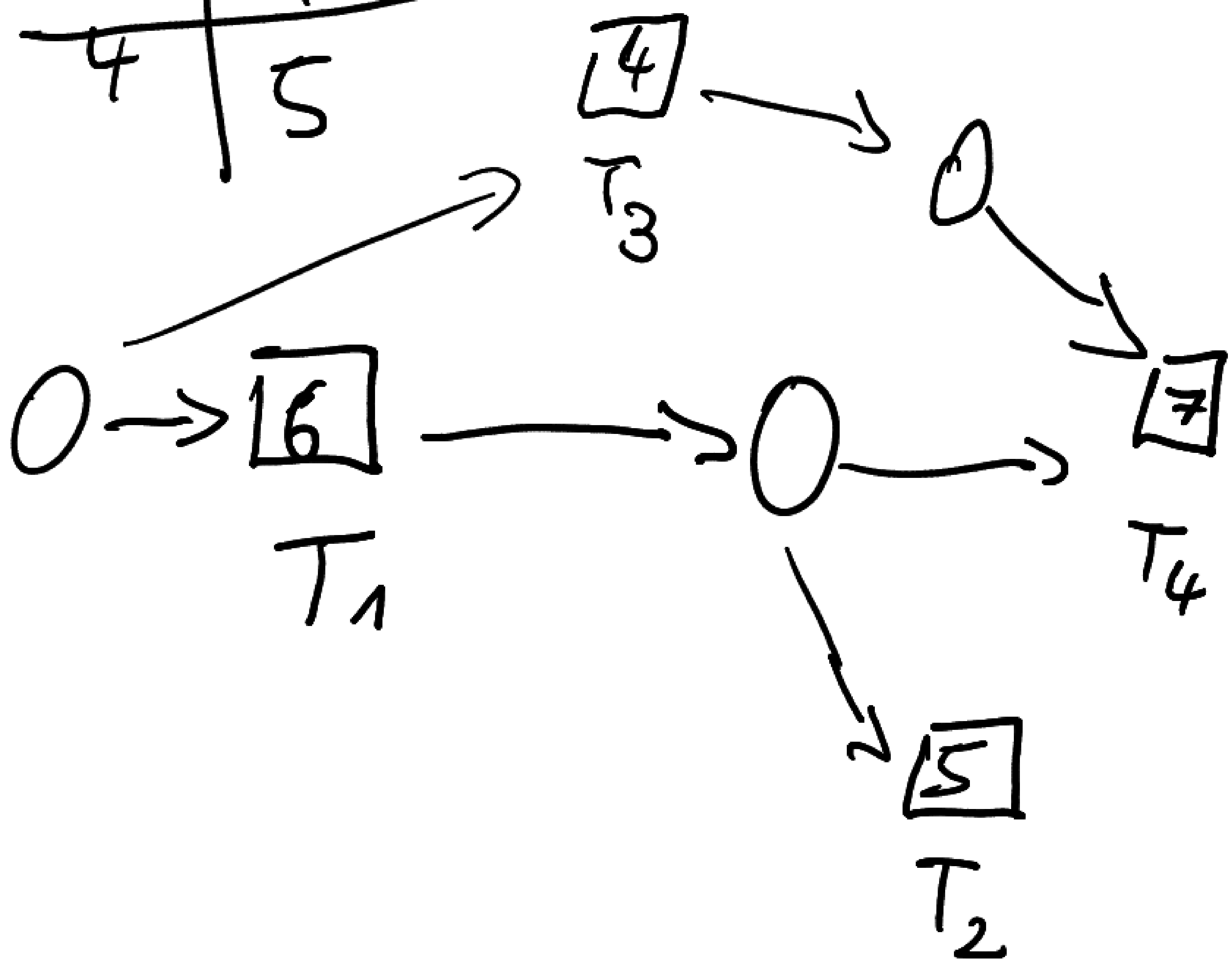
(très mal expliqué par le prof)


P
Q
p_i
a_{ik}
b_{ik}
T_i

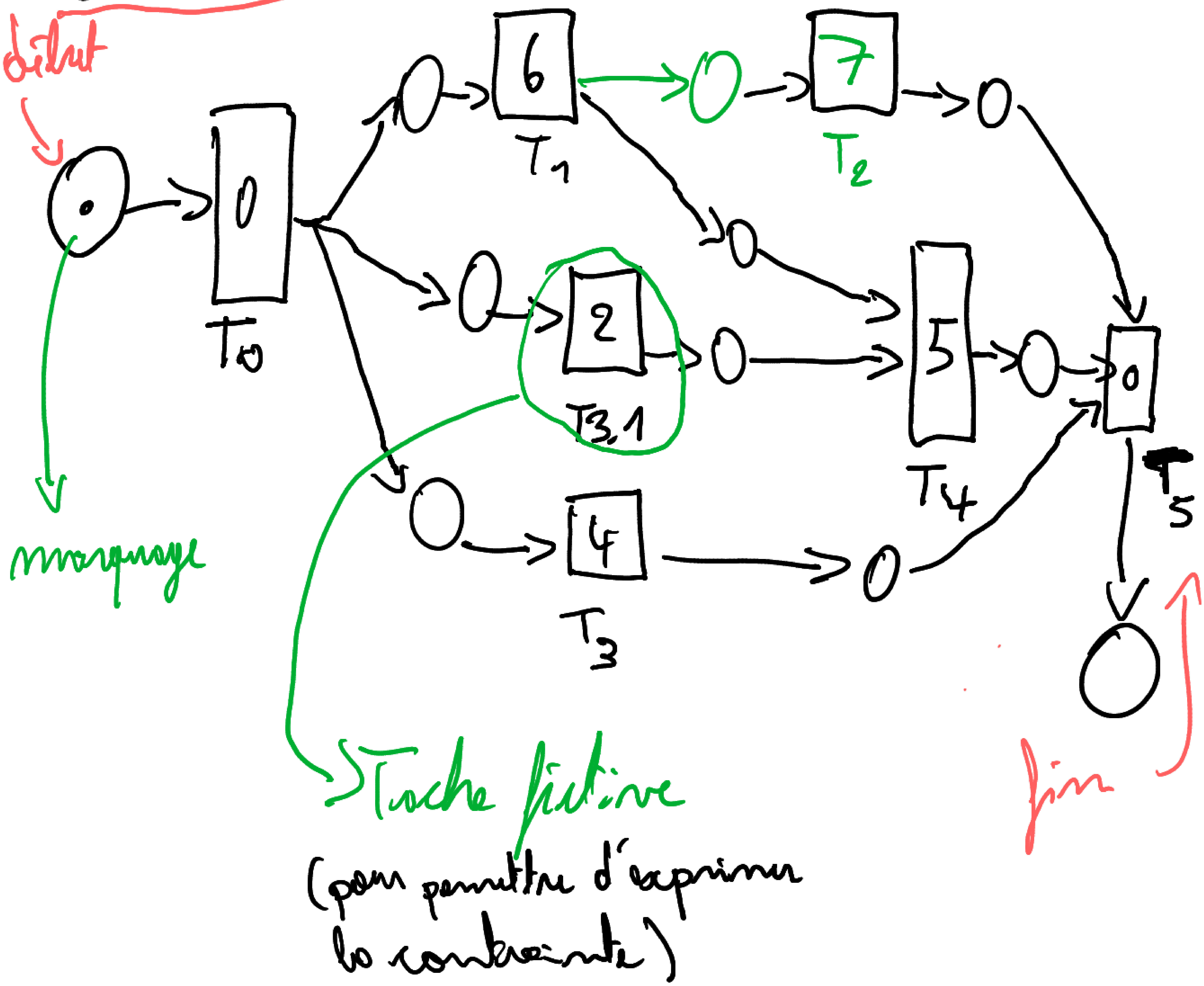

T_1

Exemple 1

Tâches	Durée
1	6
2	7
3	4
4	5



Correction



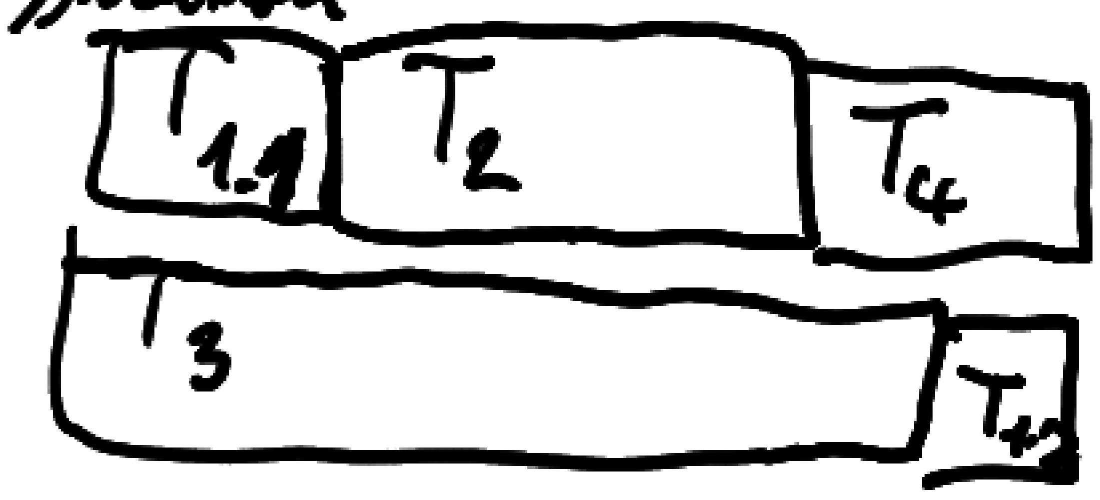
Calcul de C_{max} = date de début de fin de T_5

Ainsi:

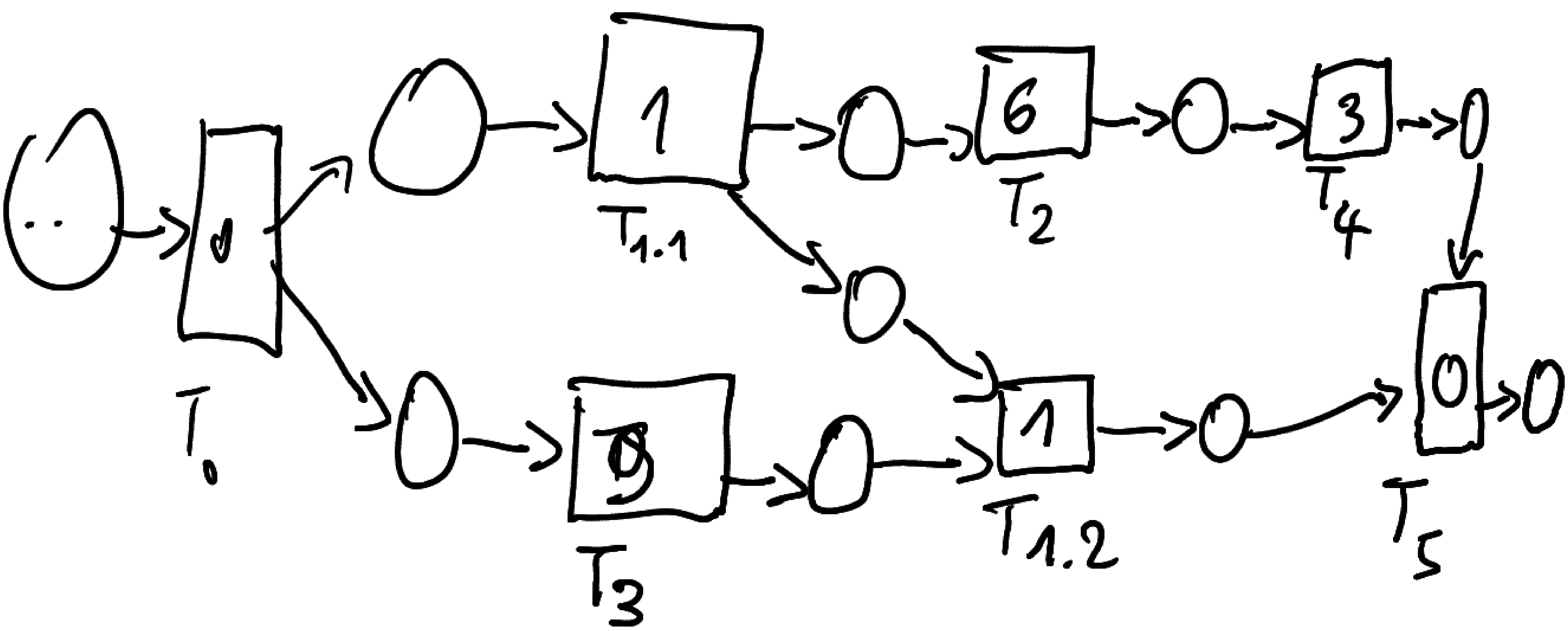
$$t(T_5) = \max\left(t(T_2)+7,\ t(T_4)+5,\ t(T_3)+4\right)$$

Exemple 2

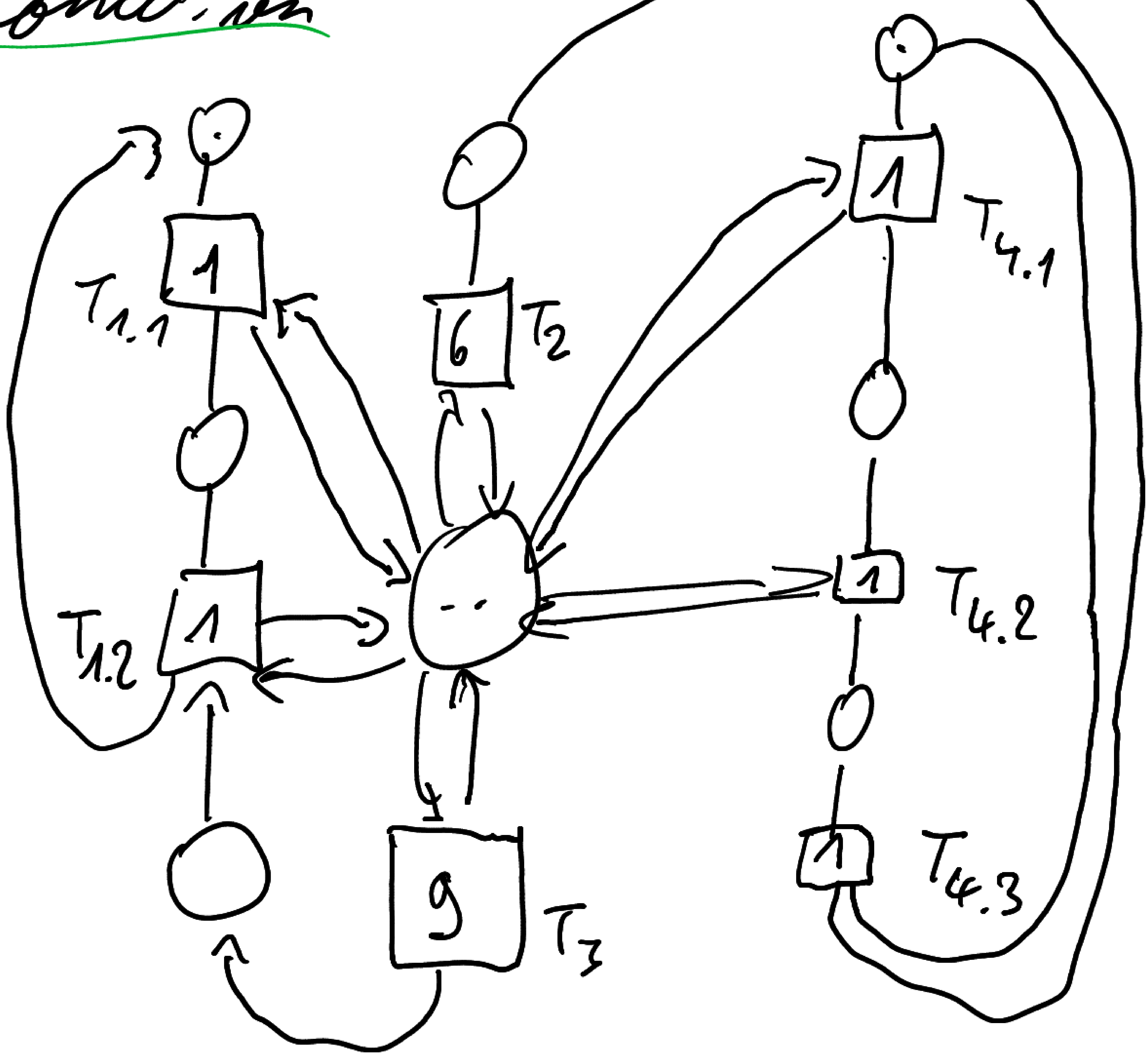
Brouillon



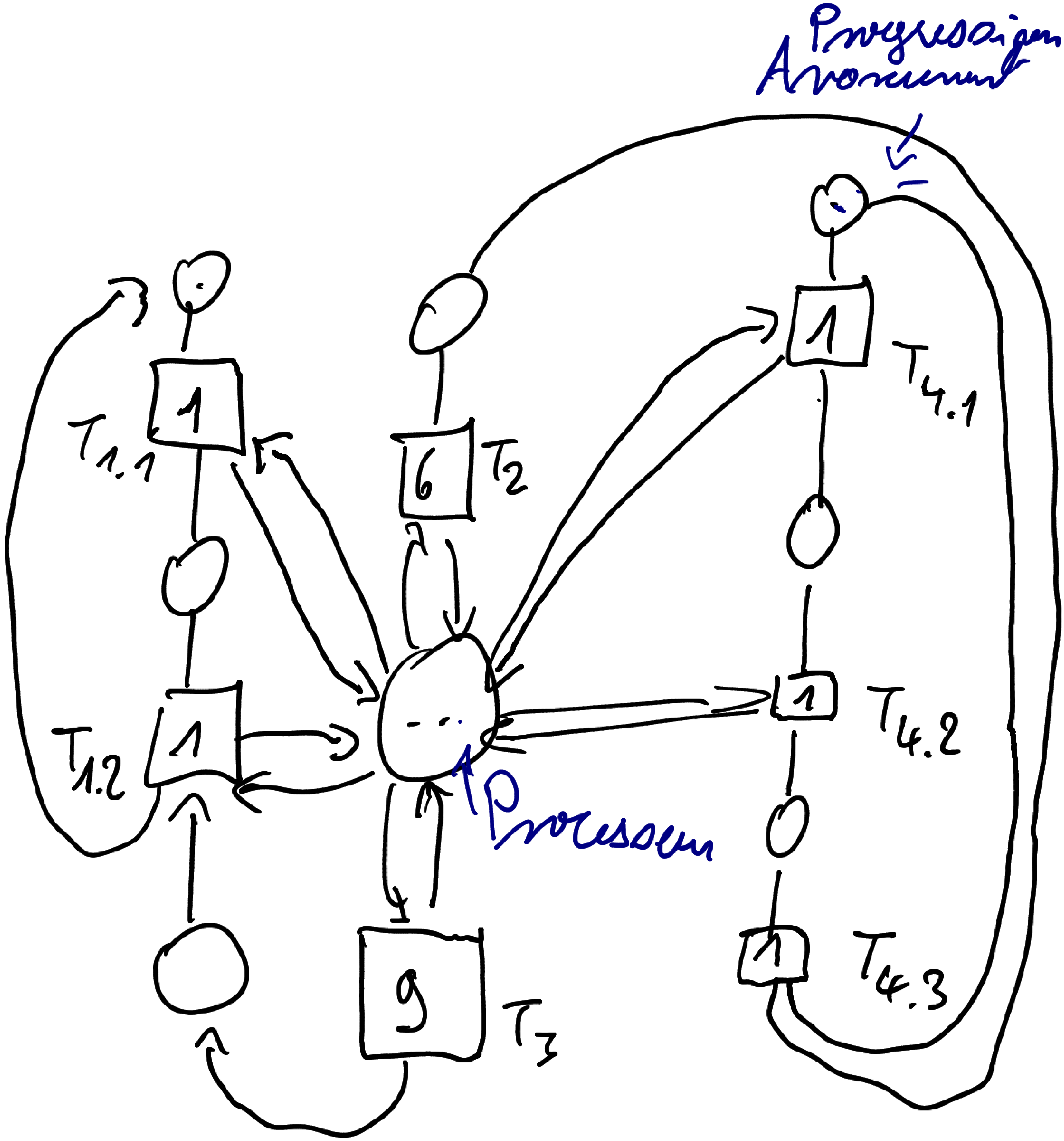
Tâches	Délai	Préemptive
1	2	Oui
2	6	
3	9	
4	3	Oui



Correction



Pour les taches préemptives
lier la dernière à la première


Avancement
1
T1.1
6
T2
1
T4.1
1
T1.2
1
T4.2
Processeur
9
T3
1
T4.3

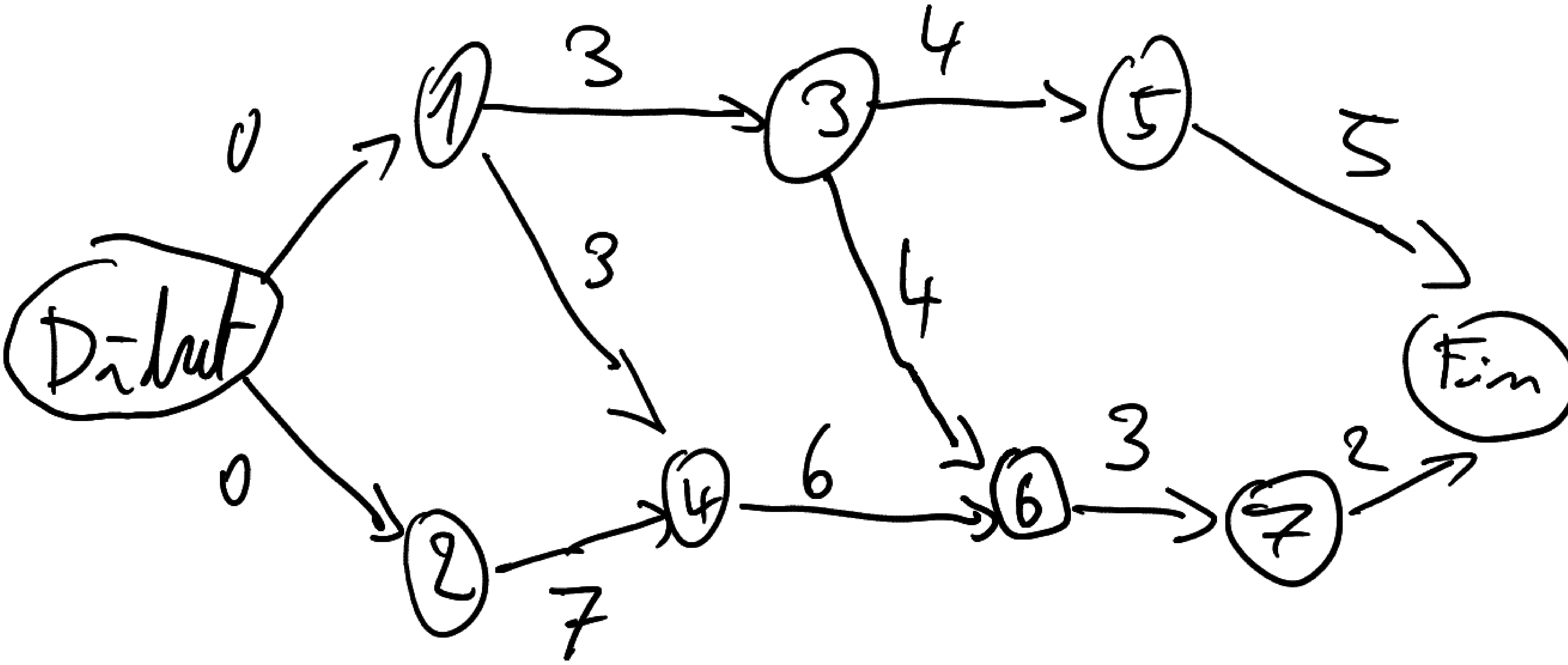
Partie 2

Méthode de chemin critique
escort
avancé

Chemin critique : - localisation temporelle
- contraintes de succession

R : au plus tôt : vers l'avant
F : au plus tard : vers l'arrière

Algorithme de Bellman



$r_i \leqslant t_i \leqslant f_i$

date au plus tôt

date d'exécution

date au plus tard

donc

$$r_i = I(0,i)$$

$$f_i = I(0,n+1) - I(i,n+1)$$

L'existence de circuit n'entraine pas l'inexistence de solution

Bellman $O(m)$ où $m = |\text{contraintes potentielles}|$

Méthode de chemin critique

Marge : délai dont on peut retarder une tache sans "trop modifier" l'ordonnancement

→ libre, total, certaine

→ fin au plus tôt
très peu utilisé

→ rend la tâche critique lorsqu'elle s'annule : fin au plus tard

→ réserve de sécurité attachée à la tache
fin au plus tôt

Méthode exacte

↳ Algo de Johnson-1 et 2 $O(n^2)$

↳ Méthode arborescente "Branch and Bound"

minimiser un coût

→ flowshop