Algorithme de Bellman

Remarques :

- Complexité : $O(m)$
- Nécessite de trier les sommets par niveaux (en $O(n+m)$ avec un algorithme de parcours en profondeur)
- Variante : utiliser les niveaux pour éviter les prédécesseurs.

```
Initialiser π à +∞, Père à 0
π[r] := 0, Père[r] := r
Pour tout x successeur de r (ie. x ∈ N[1])
    π[x] := w(rx)
    Père[x] := r
FinPour
Pour i := 1 à L /* L est le nombre total de niveau */
    Pour tout x dans N[i]
       Pour tout successeur y de x tel que π[x]+w(xy)<π[y]
          π[y] := π[x]+w(xy)
          Père[y] := x
       FinPour
    FinPour
FinPour
```

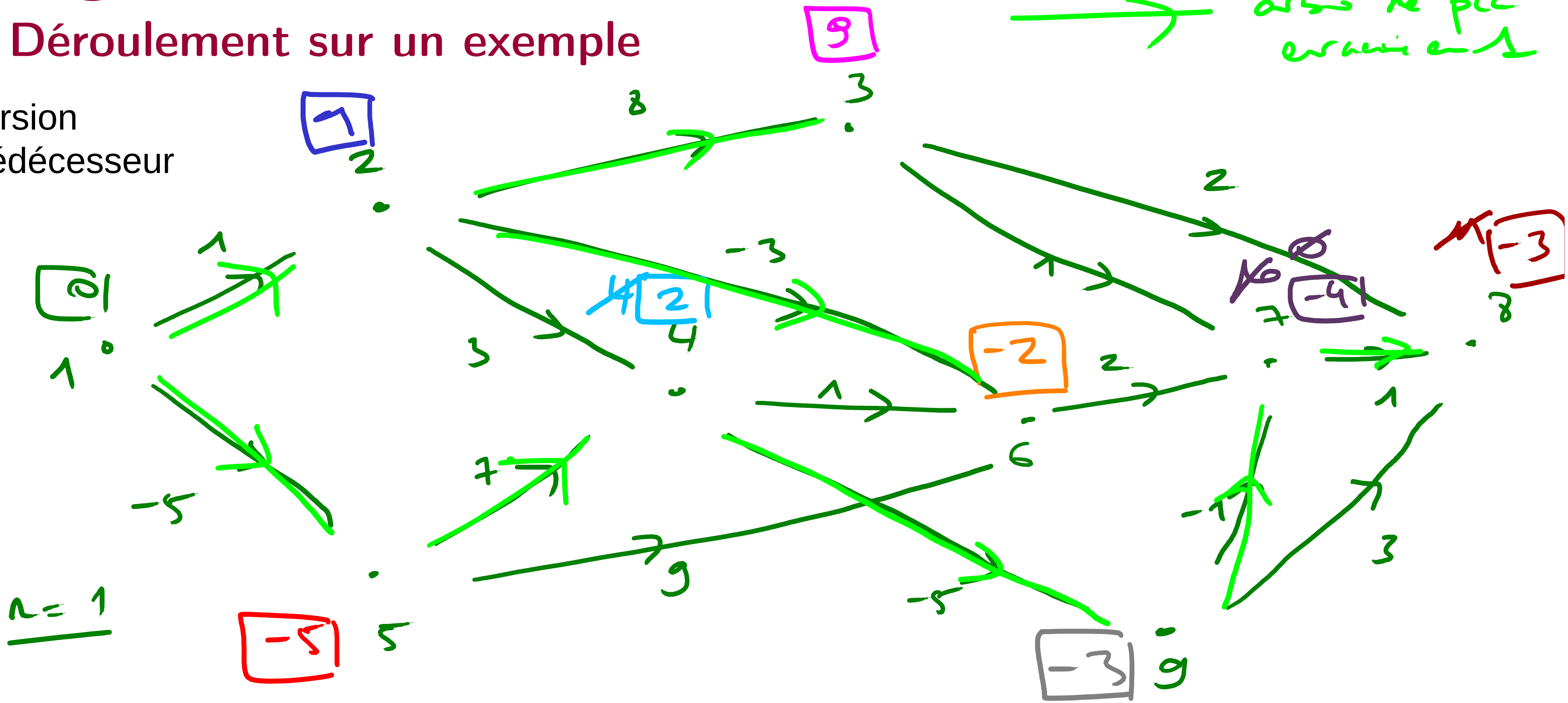
Algorithme de Bellman

Déroulement sur un exemple

$N_2 = ?\ 3, \cancel{6}, 4$

→ arbre de pcc enraciné en 1

Version prédécesseur



Etape 1 Tri topologique des sommets

$N_0 = \{1\}$
$N_1 = \{2;5\}$
$N_2 = \{3;4\}$
$N_3 = \{6,9\}$
$N_4 = \{7\}$
$N_5 = \{8\}$

Etape 2 : Sorted $(=\pi)\{\underbrace{1}_{N_0}, \underbrace{2, 5}_{N_1}, \underbrace{3, 4}_{N_2}, \underbrace{6, 9}_{N_3}, \underbrace{7}_{N_4}, \underbrace{8}_{N_5}\}$

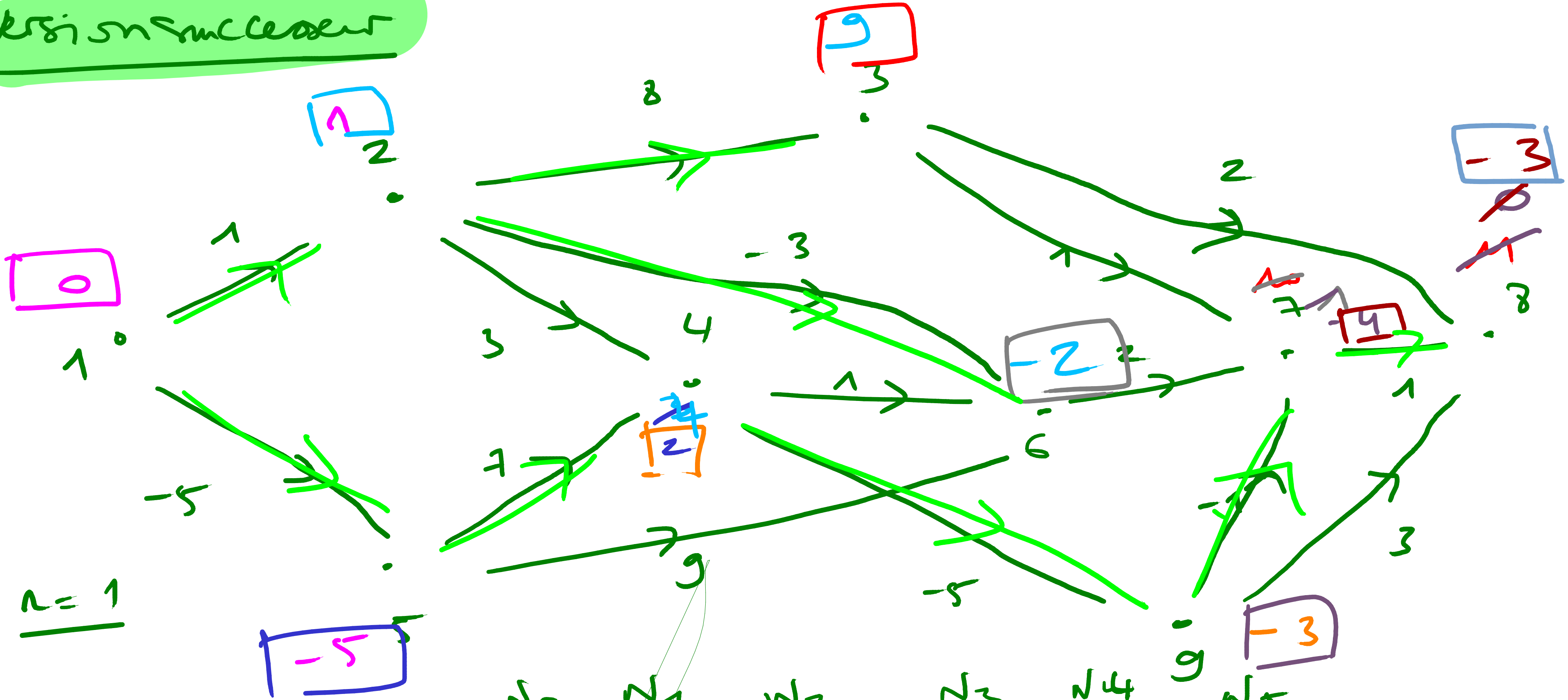
Père

1	2	3	4	5	6	7	8	9
1	1	2	5	1	2	9	7	4

π

0	1	9	2	-5	-2	-4	-3	-3

Version successeur



n = 1

Sorted/x : 1, 2, 5, 3, 4, 6, 9, 7, 8

Sorted/x : N_0: 1 | N_1: 2, 5 | N_2: 3, 4 | N_3: 6, 9 | N_4: 7 | N_5: 8

	1	2	3	4	5	6	7	8	9
π	0	1	9	~~4~~ 2	-5	-2	~~10~~ ~~7~~ -4	~~11~~ ~~0~~ -3	-3
Père	1	1	2	~~2~~ 5	1	2	~~3~~ ~~6~~ 9	~~3~~ ~~8~~ 7	4

→ arborescence des pcc

Application II

Problème d'ordonnancement

Problème d'ordonnancement (1)

La conception et la gestion d'un projet complexe composé de multiples travaux élémentaires pose des problèmes délicats :

- de planification : définition du calendrier, gestion de l'enchaînement des tâches, ...
- de contrôle de l'exécution du projet (coordination).

Il est donc nécessaire de faire appel à des techniques d'ordonnancement qui permettent de déterminer dans quel ordre doivent être exécutées les tâches de façon à *optimiser* un objectif (minimiser les coûts, minimiser le temps, ...).

Définition

On se trouve face à un **problème d'ordonnancement** lorsque, en vue de la réalisation d'un objectif quelconque, il faut accomplir de multiples tâches, elles-mêmes soumises à un ensemble de contraintes :

- *contraintes de précédence / succession* qui imposent qu'une tâche j ne peut commencer avant une autre tâche i.
- *contraintes disjonctives* qui empêchent 2 tâches i et j de s'exécuter en même temps (parce qu'elles partagent une même ressource, par exemple).
- *contraintes cumulatives* qui permettent de tenir compte des moyens humains et matériels disponibles. Les contraintes cumulatives sont très difficiles à prendre en compte.

Problème d'ordonnancement (2)

Exemple et visualisation

Exemple

Une entreprise de jouets fabrique des jouets en bois. Chaque pièce doit être découpée, poncée puis teintée (dans cet ordre ! !). L'atelier comporte des machines qui découpent (M1), poncent (M2) et teintent (M3). Voici pour trois types de pièces, les temps d'usinage en secondes pour chaque opération.

	T1	T2	T3
M1	40	80	70
M2	50	60	80
M3	70	60	50

Objectif : rechercher la durée minimale pour fabriquer ces trois pièces.

Pour cela on visualise l'avancement des travaux (occupation des machines et usinages des pièces) par un diagramme de GANTT dans lequel :

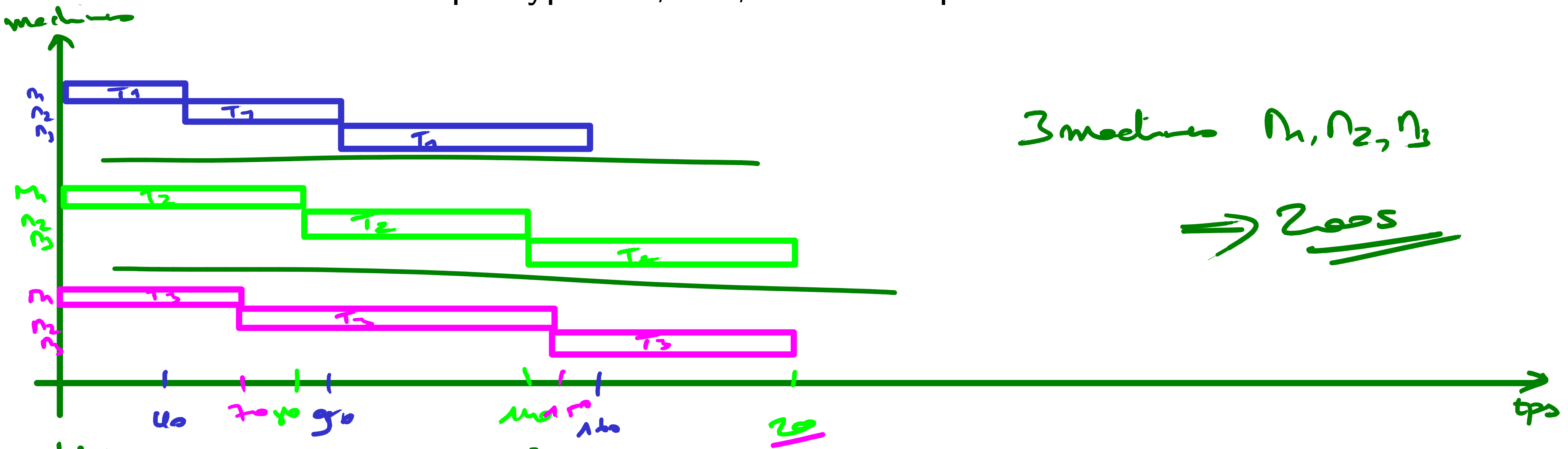
- l'axe horizontal représente le temps,
- l'axe vertical représente les ressources (ici les machines),
- une tâche élémentaire est représentée par un rectangle.

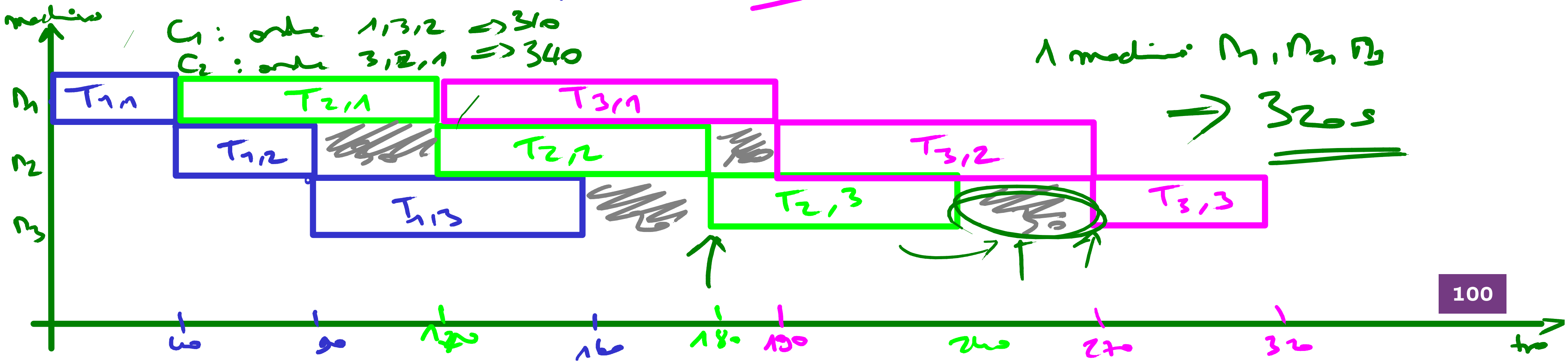
Problème d'ordonnancement (3)

Exemple et visualisation

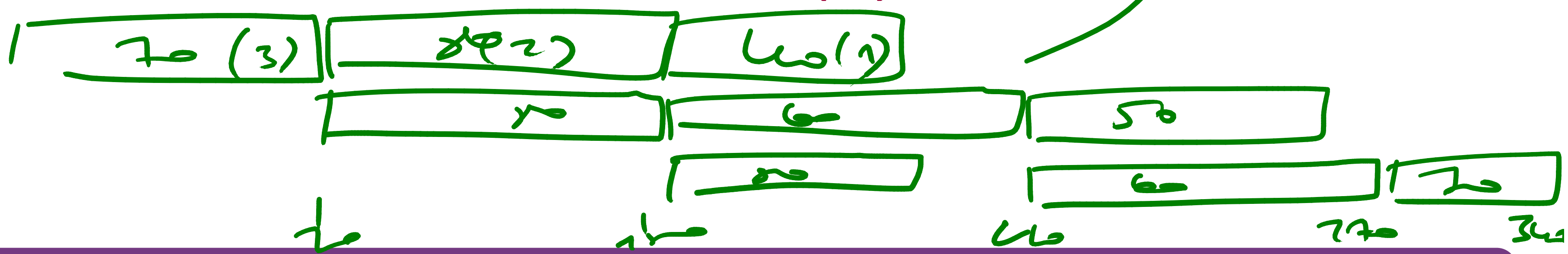
Visualisation : 2 cas

- 3 machines de chaque type M1, M2, M3 sont disponibles
- 1 machine de chaque type M1, M2, M3 est disponible





Problème d'ordonnancement (4)



Définition

On appelle **ordonnancement simple** des problèmes d'ordonnancement pour lesquels :

- le nombre de ressources n'est pas limité,
- les tâches ont une durée d'exécution connue,
- seules des contraintes de précédence existent.

En général, l'**objectif** est de rechercher la **durée minimale du projet** ie. la durée minimale de l'exécution de l'ensemble des tâches.

Modélisation potentiel-tâche : les tâches sont représentées par les sommets

Date au plus tôt, date au plus tard

Plaçons nous dans le cas d'une minimisation de la longueur totale d'un projet.

Il est alors possible de définir pour chaque tâche x :

- La date au plus tôt, $\pi(x)$: c'est la date minimum à laquelle on peut démarrer l'exécution de la tâche x en fonction des données et des contraintes du problème (notamment en fonction des contraintes de précédence).
- La date au plus tard, $\eta(x)$: c'est la date limite à laquelle on doit commencer la tâche x sous peine de retarder l'exécution totale du projet.
- La marge, $m(x)$, qui représente le délai dont on dispose pour lancer la tâche x. $m(x) = \eta(x) - \pi(x)$.

Les tâches ayant une marge nulle ($\eta(x) = \pi(x)$) sont appelées **tâches critiques**, car pour ces tâches, un retard pris dans l'exécution entraîne un retard dans la réalisation du projet.

Méthode Potentiel-tâche

Exemple

La construction des fondations d'une maison se compose de 5 tâches élémentaires :
- T : Terrassement - Durée 3 jours
- G : Mise en place de la grue - Durée 1 jour
- R : Branchement au réseau EDF et d'eau - Durée 2 jours
- B : Coulage de la dalle en béton - Durée 3 jours
- S : Mise en place de la fosse septique - Durée 4 jours

Les tâches T, G, R peuvent commencer tout de suite.
Par contre, les tâches S et B nécessitent la tâche T.
B a besoin de l'eau et de la grue qui a besoin de l'EDF pour fonctionner.

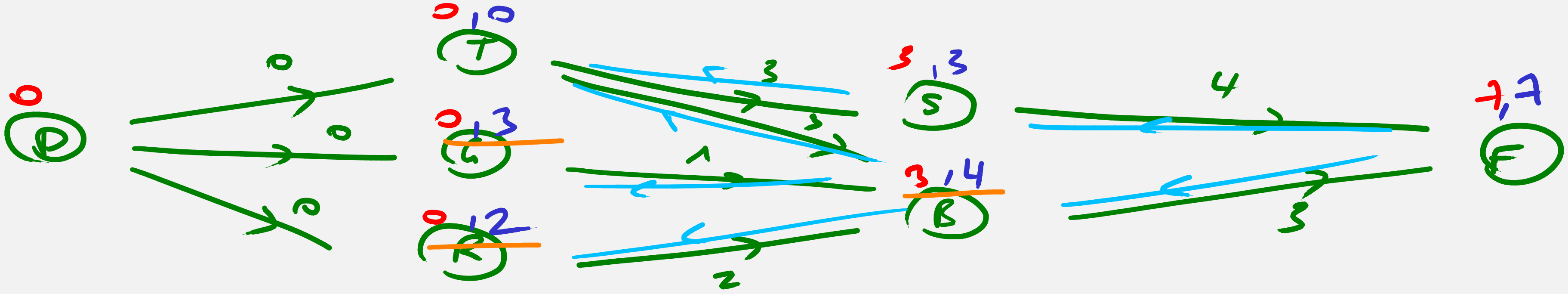
Le problème se résume sous la forme d'un tableau :

Tâche	durée	précédence
T	3	-
G	1	-
R	2	-
B	3	T, G, R
S	4	T

Méthode Potentiel-tâche

Modélisation

Une tâche est représentée par un sommet. Un arc représente une contrainte de précédence.



- Ajout des sommets D et F pour modéliser resp. le début et la fin des travaux.
- Sur les arcs, on indique la durée de la tâche correspondant au sommet initial.

Résolution

- On calcule les valeurs $\pi(x)$, en utilisant l'algorithme de Bellman à partir du sommet de début. Ici, on cherche le potentiel maximum de chaque tâche.
- On calcule les valeurs $\eta(x)$ en appliquant l'algorithme de Bellman sur l'antiarborescence issue du sommet de fin. Ici, on cherche le potentiel minimum de chaque tâche par rapport à la durée minimale trouvée $\pi(F)$.

Durée des travaux : 7
Tâches critiques : T, S
Tâches retardables : G (3j), R (2j), B (1j)