MARO 018 - Optimization & Operational Research

Arnaud Vandaele

Faculté Polytechnique
Service MAthématique et Recherche Opérationnelle

UMONS

Année académique 2024-2025

Séance 1 :
Ordonnancement de projets

Séance 1 – Ordonnancement de projets

Tables des matières

Plan

Description du problème et méthode MPM

Les données consistent en :

- une liste de n tâches à réaliser
- la durée p_i de chaque tâche i avec $i = 1, ..., n$
- une série de contraintes de posériorité stricte entre des paires de tâches

Exemble :

Tâches	Durées	Prédécesseur(s)
A	2	–
B	3	–
C	3	A

But : en tenant compte des données, comment planifier l'exécution des tâches afin de minimiser le délai d'exécution du projet.

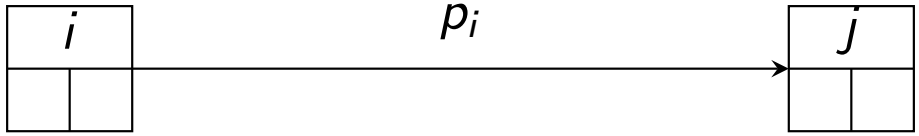
Algorithme : MPM (méthode des potentiels métra)

Représentation des données sous la forme d'un graphe

Cette méthode utilse la théorie des graphes et la présentation des données se fait à l'aide d'un *graphe-tâches* :

- les sommets représentent les tâches (un sommet par tâche)
- les arcs représentent les contraintes et les valeurs portées par ces arc indiquent les délais entre les instants de début de tâches.

Si x_i représente l'instant de début dune tâche i, la contrainte de postériorité stricte $j \succ i$ se formule $x_j \geq x_i + p_i$ et se représente par :



De plus, on ajoute deux sommets *fictifs* :

- un sommet `DEBUT`, sans prédécesseur, et qui est relié à toutes les tâche sans antécédents,
- un sommet `FIN`, sans successeur, et auquel sont reliées toutes les tâche sans successeurs.

Algorithme : MPM (méthode des potentiels métra)

Algorithme de propogation des dates début au plus tôt et au plus tard

- $\Gamma(i)$: ensemble des sommets sucesseurs directs du sommet i
- $\Gamma^{-1}(i)$: ensemble des sommets précédents directs du sommet i

Date de début au plus tôt (ES - Earliest Starting Time)

$ES(i)$ représente le premier instant auquel on peut commencer la tâche i.
Pour respecter les contraintes de postériorité, cette date correspond à la date de début au plus tôt des tâches qui la précédent additionnées du délai approprié :

$$\begin{aligned} ES(\texttt{DEBUT}) &= 0 \\ ES(i) &= \max_{j \in \Gamma^{-1}(i)} ES(j) + p_j \end{aligned}$$

Date de début au plus tard (LS - Latest Starting Time)

$LS(i)$ est le dernier instant auquel la tâche i peut commencer sans allonger la durée totale du projet.
Cette date est déterminée par l'ensemble des successeurs à i :

$$\begin{aligned} LS(\texttt{FIN}) &= ES(\texttt{FIN}) \\ LS(i) &= \min_{j \in \Gamma(i)} LS(j) - p_i \end{aligned}$$

Marges totales, libres et chemin critique

Les dates de début au plus tôt et au plus tard induisent certains degrés de liberté dans l'ordonnacement d'un projet, appelés marges.

Marge totale (MT)

$MT(i)$ représente le retard maximal que peut subir la tâche i, par rapport à son ordonnancement au plus tôt, sans allonger la durée totale du projet :

$$MT(i) = LS(i) - ES(i).$$

Marge libre (MT)

$ML(i)$ représente le retard maximal que peut subir la tâche i, par rapport à son ordonnancement au plus tôt, sans retarder l'exécution d'aucune tâche suivante :

$$ML(i) = \min_{j \in \Gamma(i)} \left(ES(j) - (ES(i) + p_i) \right).$$

De manière générale,

$$0 \leq ML(i) \leq MT(i),$$

et les tâches pour lesquelles $MT(i) = 0$ sont appelées **tâches critiques**.

Il existe toujours un chemin composé de tâches critiques entre `DEBUT` et `FIN`, appelé chemin critique.

Exercice

Exemple 3.1 Pour mener à bien un projet d'informatique, une équipe de développement prépare son planning. Pour ce faire, le responsable liste les différentes tâches à accomplir ainsi que les durées de celles-ci. Enfin, il décrit les contraintes de postériorité entre les tâches. Ainsi par exemple, la "conception graphique" ne peut être entamée qu'après la fin de "l'étude de marché" :

	Activité	**Temps nécessaire (semaines)**	**Tâches prédécesseurs**
a.	étude de marché	3	–
b.	conception graphique	4	a
c.	organigramme	2	a
d.	fenêtres, menus, i/o	6	b, c
e.	module 1	5	c
f.	module 2	3	c
g.	module 3	7	e
h.	module 4	5	f
i.	regroupement et validation	8	d, g, h

Plan

Comment faire quand les ressources sont limitées ?

Les ressources sont typiquement de trois types :

1. **Ressources renouvelables** (ex. machines, opérateurs)
   → On retrouve la même quantité après chaque usage,
   → Contrainte de quantité maximale disponible à chaque instant.
2. **Ressources consommables** (ex. matières premières, argent)
   → Diminue au fur et à mesure de son utilisation / consommation,
   → Contrainte de quantité disponible pour une période donnée.
3. **Ressources doublement contraintes** (ex. énergie, connexion internet)
   → Contrainte sur l'utilisation instantanée et sur la consommation cumulée.

Reprenons l'exemple initial + besoin d'un type de ressource renouvelable :

Tâches	Durées	Prédéc.
A	2	–
B	3	–
C	3	A

- A a besoin d'une unité de ressource renouvelable
- B a besoin d'une unité de ressource renouvelable
- C a besoin de deux unités de ressource renouvelable

L'ordonnacement proposé initialement ne convient plus... Comment faire ?

Description d'une heuristique (il en existe d'autres)

Schéma de génération série

L'approche "série" place une activité à la fois, aussi tôt que possible au regard des contraintes de précédence et de disponibilité des ressources.
A chaque étape (temporelle, par ex.), le mécanisme scanne la liste de priorité et choisit la première tâche éligible (i.e. qui n'a pas encore été planifiée et dont tous les prédécesseurs ont déjà été planifiés).
Celle-ci est alors positionnée au plus tôt, en respectant les contraintes de précédence et de disponibilité des ressources.

Exercice 1. (avec ressource renouv. max = 5, et liste de prio. = ordre alpha.)

Tâches	Durées	Prédécesseur(s)	Ressource
A	1	–	1
B	2	–	2
C	4	–	2
D	3	–	2
E	1	A	2
F	5	E	1
G	3	D	2

Exercice 2

Ordonnançons un projet avec une ressource de chaque type

- Ressource R1 : renouvelable, limitée à 4 unités
- Ressource R2 : consommable, 3 unités livrées aux instants 0, 4, 8, ...

Problème :

Tâches	Durées	Prédécesseur(s)	R1	R2
A	2	–	3	2
B	3	–	1	1
C	1	–	2	1
D	4	A, B	3	2
E	2	B, C	3	2
F	1	D	3	1

Proposer un planning (un diagramme de Gantt) qui respecte ces contraintes de ressources.

Exercice 3 : vers la complexité

Projet composé de 4 tâches et 5 ressources renouvelables (limite = 1) :

Tâches	Durées	Prédécesseur(s)	R12	R13	R24	R34	R23
A	1	–	1	1	0	0	0
B	1	–	1	0	1	0	1
C	1	–	0	1	0	1	1
D	1	–	0	0	1	1	0

Tenter de déterminer la solution optimale.

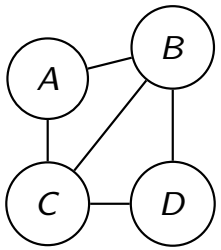
Plan

Reduction

GRAPH-COLORING est un problème NP-hard :

INSTANCE: Un graphe $G = (V, E)$ et un naturel c.
QUESTION: est-il possible d'attribuer une couleur à chacun des $|V|$ sommets, en utilisant au plus c couleurs, de façon à ce que 2 sommets partageant une arête aient des couleurs différentes ?

Exemple avec $c = 3$ et



Réponse : OUI avec A, B, C, D.

Que peut-on dire sur la complexité du problème de RCPSP ?

INSTANCE: n tâches avec des durées p_i, des contraintes de précédence, k ressources renouvelables avec des limites L_j et un entier C_{max}.
QUESTION: est-il possible d'ordonnancer les n tâches en respectant les contraintes de précédence et de ressource de façon à ce que la fin du projet $\leq C_{max}$?

- A partir de n'importe quelle instance de GRAPH-COLORING, on décrit une procédure qui va créer une instance de RCPSP en temps polynomial.
- Ensuite, il faut prouver :
  (instance-OUI $\Rightarrow$ instance-OUI) et (instance-OUI $\Leftarrow$ instance-OUI).