

GOPP

Rapport

ABOUDOU Hanrifani
LENTALI Thomas
Master 2 MIMSE
Université de Bordeaux

Résumé

Lors de ce projet, nous allons présenter et mettre en œuvre les méthodes de configurations interdites minimales, de formulation de flot et de modèle avec conflits pour résoudre le problème de RCPSP. La première partie du rapport a pour but d'étudier les principes de résolution de ces méthodes ainsi que les améliorations que nous avons apporté afin de réduire les temps d'exécutions. Dans un second temps, nous allons analyser les résultats issus nos implémentations.

1 Méthodes de Résolutions

1.1 Preprocessing

A chaque fois qu'on lit une instance, on la rentre dans notre structure de données, et on lui rajoute des informations qui vont aider à l'exécution du modèle. On a implémenté une heuristique gloutonne (algorithme de liste en série) qui consiste à placer chaque tâche dans l'ordre de la liste au plus tôt en respectant les contraintes de capacité et de précédences. Cette borne supérieure va nous permettre de majorer l'horizon de temps dans nos modèles. Nous avons aussi implémenté la méthode PERT qui nous permet d'avoir les dates au plus tôt et au plus tard des tâches. Ces dates au plus tôt et au plus tard vont nous permettre d'avoir les valeurs d_{ij} qui correspondent au plus grand écart d'ordonnancement entre i et j .

1.2 Formulation flot

1.2.1 Modèle

$$\begin{aligned} \min \quad & S_{n+1} \\ \text{s.c : } \quad & x_{ij} = 1 \quad \forall (i, j) \in E \\ & x_{ij} + x_{ji} \leq 1 \quad \forall (i, j) \in V^2, i \leq j \\ & x_{ik} \geq x_{ij} + x_{jk} - 1 \quad \forall (i, j, k) \in V^3 \\ & S_j - S_i \geq -M + (p_i + M)x_{ij}, \quad \forall (i, j) \in V^2 \\ & \sum_{j \in V} f_{ij}^k = b_{ik}, \quad \forall i \in V, \forall k \in \mathcal{R} \\ & \sum_{j \in V} f_{ji}^k = b_{ik}, \quad \forall i \in V, \forall k \in \mathcal{R} \\ & f_{ij}^k \leq \min(b_{ik}, b_{jk})x_{ij} \quad \forall (i, j) \in V^2, \forall k \in \mathcal{R} \\ & x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in V^2, i \neq j \\ & S_i \geq 0, \quad \forall i \in V \\ & f_{ij}^k \geq 0 \quad \forall (i, j) \in V^2, \forall k \in \mathcal{R} \end{aligned}$$

Les défauts de la formulation sont un grand nombre de symétries, mn^2 variables de flot, des grandes valeurs M .

1.2.2 Améliorations

Nous avons amélioré le modèle :

- $asap[i] \leq S[i] \leq alap[i]$, $\forall i$, avec $asap$ les dates au plus tôt et $alap$ les dates au plus tard des tâches,
- Préprocessing pour limiter la valeur des M : on affine le M en affectant $M = |asap[j] - alap[i]|$.

1.3 Configurations interdites minimales

1.3.1 Modèle

$$\begin{aligned}
& \min S_{n+1} \\
& \text{s.c : } x_{ij} = 1 \quad \forall (i, j) \in E \\
& \quad x_{ij} + x_{ji} \leq 1 \quad \forall (i, j) \in V^2, i \leq j \\
& \quad x_{ik} \geq x_{ij} + x_{jk} - 1 \quad \forall (i, j, k) \in V^3 \\
& \quad S_j - S_i \geq -M + (p_i + M)x_{ij}, \quad \forall (i, j) \in V^2 \\
& \quad \sum_{(i,j) \in F^2} x_{ij} \geq 1, \quad \forall F \in \mathcal{F} \\
& \quad x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in V^2, i \neq j \\
& \quad S_i \geq 0, \quad \forall i \in V
\end{aligned}$$

- \mathcal{F} : ensemble des configurations interdites minimales,
- M : une grande valeur.

Ce modèle comporte comme défaut un nombre exponentiel de contraintes (autant que de configurations interdites minimales) et de grandes constantes M .

1.3.2 Coupe

Vérification du respect des contraintes de ressource :

```

L = {Cj : Aj ∈ V}
trier L par valeurs croissantes
pour t ∈ L faire
|   p ← 0, F ← ∅
|   pour Aj ∈ A faire
|   |   si Sj < t, Cj ≥ t et bjk > 0 alors
|   |   |   o ← o + bjk, F ← F ∪ Aj
|   |   fin
|   |   si o > Bk alors
|   |   |   retourner coupe
|   |   fin
|   fin
|   retourner vrai
fin

```

Cet algorithme sert à vérifier si la solution trouvée, en relâchant la contrainte : $\sum_{(i,j) \in F^2} x_{ij} \geq 1, \quad \forall F \in \mathcal{F}$, est réalisable.

Si elle n'est pas réalisable, on rajoute toutes les coupes violées par cette configuration, sinon cette solution est optimale.

1.3.3 Améliorations

- $asap[i] \leq S[i] \leq alap[i], \forall i$, avec $asap$ les dates au plus tôt et $alap$ les dates au plus tard des tâches,
- Préprocessing pour limiter la valeur des M : on affine le M en affectant $M = |asap[j] - alap[i]|$,
- on a utilisé les LAZYCALLBACK pour créer la coupe $\sum_{i \in F} \sum_{j \in F} x_{ij} \geq 1$.

1.4 Modèle indexé avec le temps

1.4.1 Modèle

$$\begin{aligned}
& \min \sum_{t \in T} t \cdot y_{n+1,t} \\
& \text{s.c : } \sum_{t=0}^T y_{it} = 1 \quad \forall i \in V \\
& \sum_{t=0}^T t(y_{jt} + y_{it}) \geq p_i \quad \forall (i, j) \in E \\
& \sum_{i \in V} b_{ik} \sum_{r=t-P_i+1}^t y_{ir} \leq B_k, \quad \forall k \in \mathcal{R}, \forall t = 1, \dots, T \\
& y_{it} \in \{0, 1\}, \quad \forall i \in V, \forall t = 1, \dots, T
\end{aligned}$$

Les défauts de la formulation sont un nombre pseudo-polynomial de variables et une relaxation linéaire très faible.

1.4.2 Améliorations

Améliorations :

- borne la plus précise possible pour T par l'heuristique de preprocessing.
- $y_{it} = 0$ si $t < asap[i]$ ou $t > alap[i]$

2 Implémentation et Résultats

Des trois méthodes étudiées, la plus efficace en terme de temps d'exécution est le modèle avec conflits. Néanmoins pour certaines instances, le modèle de configurations interdites minimales surpasse le modèle avec conflits en temps, dû au faible nombre de configurations interdites à générer dans certains cas. Le modèle de formulation de flot est de façon générale la plus lente des trois méthodes.

Sans amélioration, nous obtenons les résultats suivants :

Nom instance	sol. flot	temps flot	sol. min	temps min	sol. conflit	temps conflits
j301 1.sm	43	9.33s	43	3.51s	43	0.89s
j301 2.sm	47	26.63s	47	1.6s	47	0.72s
j301 3.sm	47	6.41s	47	0.66s	47	0.49s
j301 4.sm	62	215.74s	62	6.86s	62	1.79s
j301 5.sm	39	89.18s	39	4.7s	39	2.92s
j301 6.sm	48	24.2s	48	2.14s	48	1.46s
j301 7.sm	60	1.07s	60	0.88s	60	0.18s
j301 8.sm	53	14.88s	53	1.74s	53	0.23s
j301 9.sm	49	42.79s	49	3.07s	49	7.42s
j301 10.sm	45	10.4s	45	2.62s	45	0.85s

Ce premier tableau nous donne une idée sur le temps gagné par les améliorations même si toutes les instances n'ont pas la même complexité.

Résultats obtenus avec l'ajout des améliorations :

Précisons que le temps d'exécution de chaque méthode est limité à 10 minutes par instances.

Si l'optimal n'est pas atteint, le résultat affiché est la meilleure solution réalisable.

Si aucune solution réalisable n'est atteinte, le résultat affiché est -1.

Nom instance	sol. flot	temps flot	sol. min	temps min	sol. conflit	temps conflits
j301 1.sm	43	2.71s	43	3.71s	43	0.06s
j301 2.sm	47	4.2s	47	1.67s	47	0.06s
j301 3.sm	47	0.88s	47	0.55s	47	0.08s
j301 4.sm	62	43.98s	62	3.4s	62	0.25s
j301 5.sm	39	14.85s	39	5.99s	39	0.44s
j301 6.sm	48	9.21s	48	1.87s	48	0.48s
j301 7.sm	60	0.44s	60	0.79s	60	0.05s
j301 8.sm	53	1.52s	53	0.97s	53	0.04s
j301 9.sm	49	19.77s	49	2.68s	49	0.78s
j301 10.sm	45	7.16s	45	1.88s	45	0.29s
j302 1.sm	38	2.17s	38	1.22s	38	0.09s
j302 2.sm	51	0.59s	51	1.2s	51	0.06s
j302 3.sm	43	0.29s	43	0.33s	43	0.01s
j302 4.sm	43	0.19s	43	0.24s	43	0.03s
j302 5.sm	51	0.38s	51	0.84s	51	0.05s
j302 6.sm	47	0.22s	47	0.39s	47	0.02s
j302 7.sm	47	0.26s	47	0.48s	47	0.02s
j302 8.sm	54	24.06s	54	0.52s	54	0.08s
j302 9.sm	54	2.45s	54	0.32s	54	0.03s
j302 10.sm	43	0.98s	43	1.33s	43	0.14s

Nom instance	sol. flot	temps flot	sol. min	temps min	sol. conflit	temps conflits
j303 1.sm	72	0.47s	72	0.42s	72	0.05s
j303 2.sm	40	0.28s	40	0.32s	40	0.02s
j303 3.sm	57	0.27s	57	0.22s	57	0.03s
j303 4.sm	98	0.43s	98	0.2s	98	0.04s
j303 5.sm	53	0.18s	53	0.33s	53	0.02s
j303 6.sm	54	0.16s	54	0.2s	54	0.02s
j303 7.sm	48	0.2s	48	0.19s	48	0.02s
j303 8.sm	54	0.31s	54	0.39s	54	0.02s
j303 9.sm	65	0.6s	65	0.4s	65	0.03s
j303 10.sm	59	0.51s	59	0.31s	59	0.05s
j304 1.sm	49	0.1s	49	0.08s	49	0.02s
j304 2.sm	60	0.22s	60	0.11s	60	0.02s
j304 3.sm	47	0.25s	47	0.12s	47	0.02s
j304 4.sm	57	0.12s	57	0.08s	57	0.01s
j304 5.sm	59	0.33s	59	0.12s	59	0.02s
j304 6.sm	45	0.17s	45	0.09s	45	0.02s
j304 7.sm	56	0.18s	56	0.1s	56	0.02s
j304 8.sm	55	0.13s	55	0.09s	55	0.02s
j304 9.sm	38	0.33s	38	0.09s	38	0.01s
j304 10.sm	48	0.34s	48	0.1s	48	0.02s

Nom instance	sol. flot	temps flot	sol. min	temps min	sol. conflit	temps conflits
j305 1.sm	59	603.48s	53	8.73s	53	19.17s
j305 2.sm	-1	604.13s	82	13.03s	82	20.75s
j305 3.sm	-1	603s	76	40.71s	76	330.11s
j305 4.sm	-1	604.95s	63	53.64s	64	600.35s
j305 5.sm	77	603.11s	76	4.48s	76	11.97s
j305 6.sm	-1	600.07s	64	30.19s	64	4.36s
j305 7.sm	-1	600.07s	76	50.47s	76	438.08s
j305 8.sm	-1	600.07s	67	33.4s	67	155.23s
j305 9.sm	54	603.08s	49	12.61s	49	3.68s
j305 10.sm	-1	603.57s	70	6.2s	70	40s
j306 1.sm	59	243.34s	59	5.21s	59	2.96s
j306 2.sm	51	433.18s	51	12.71s	51	1.27s
j306 3.sm	48	11.54s	48	1.48s	48	0.38s
j306 4.sm	44	600.67s	42	9.65s	42	0.69s
j306 5.sm	67	110.99s	67	3.28s	67	0.79s
j306 6.sm	37	21.32s	37	1.8s	37	0.09s
j306 7.sm	46	13.15s	46	3.34s	46	0.24s
j306 8.sm	39	32.02s	39	4.08s	39	0.13s
j306 9.sm	51	32.8s	51	1.72s	51	0.21s
j306 10.sm	61	108.33s	61	5.45s	61	2.23s

Nom instance	sol. flot	temps flot	sol. min	temps min	sol. conflit	temps conflits
j307 1.sm	55	0.8s	55	0.34s	55	0.03s
j307 2.sm	42	1.81s	42	0.35s	42	0.03s
j307 3.sm	42	13.06s	42	1.17s	42	0.06s
j307 4.sm	44	1.02s	44	0.41s	44	0.02s
j307 5.sm	44	5.01s	44	1.3s	44	0.24s
j307 6.sm	35	0.96s	35	0.27s	35	0.02s
j307 7.sm	50	13.53s	50	0.82s	50	0.09s
j307 8.sm	44	11.78s	44	1.4s	44	0.07s
j307 9.sm	60	11.52s	60	0.55s	60	0.05s
j307 10.sm	49	31.98s	49	1.12s	49	0.05s
j308 1.sm	44	0.26s	44	0.1s	44	0.02s
j308 2.sm	51	0.22s	51	0.09s	51	0.02s
j308 3.sm	53	0.5s	53	0.12s	53	0.03s
j308 4.sm	48	0.41s	48	0.12s	48	0.03s
j308 5.sm	58	0.4s	58	0.11s	58	0.04s
j308 6.sm	47	0.56s	47	0.1s	47	0.03s
j308 7.sm	41	0.32s	41	0.09s	41	0.01s
j308 8.sm	51	1s	51	0.11s	51	0.03s
j308 9.sm	39	0.43s	39	0.09s	39	0.03s
j308 10.sm	67	0.68s	67	0.1s	67	0.03s

Nom instance	sol. flot	temps flot	sol. min	temps min	sol. conflit	temps conflits
j309 1.sm	-1	600.99s	83	184.53s	83	132.75s
j309 2.sm	-1	600.08s	92	600.07s	101	600.17s
j309 3.sm	-1	602.08s	68	331.03s	68	204.72s
j309 4.sm	-1	600.07s	71	44.36s	74	600.2s
j309 5.sm	-1	600.16s	70	8.55s	70	105.48s
j309 6.sm	-1	602.49s	59	69.17s	59	358.9s
j309 7.sm	-1	600.08s	63	108.66s	66	600.16s
j309 8.sm	-1	603.35s	91	34.19s	92	600.19s
j309 9.sm	-1	600.07s	63	149.87s	65	600.14s
j309 10.sm	-1	603.96s	88	600.06s	89	600.88s

Bien que le problème de RCPSP soit un problème classique bien connu, certaines petites instances restent dures à traiter car même si les méthodes étudiées ici trouvent des solutions réalisables en moins de 10 minutes (sauf pour la formulation flot qui parfois n'a pas le temps d'en trouver), elles n'arrivent pas toujours à atteindre l'optimum en moins de 10 minutes.