Benchmarking de methodes exactes et heuristiques sur des problemes d'ordonnancement industriels complexes

Christian Artigues[1]   Guillaume Poveda[2]

[1] LAAS-CNRS, Université de Toulouse, France

[2] AIRBUS AI Research, Toulouse, France

KC@ANITI, 11/03/2022

Présentation basée sur les papiers

- Nahum Alvarez, Christian Artigues, Guillaume Poveda, Multi Skill Scheduling with Preemption and Generalized Precedence Constraints : an Industrial Use Case, Working Paper.
- Oliver Polo Mejia, Christian Artigues, Pierre Lopez, Lars Mönch, Virginie Basini. Heuristic and metaheuristic methods for the multi-skill project scheduling problem with partial preemption. International Transactions in Operational Research, Wiley, 2021, https ://doi.org/10.1111/itor.13063.
- Oliver Polo-Mejía, Christian Artigues, Pierre Lopez, Virginie Basini : Mixed-integer/linear and constraint programming approaches for activity scheduling in a nuclear research facility. International Journal of Production Research 58(23) : 7149-7166 (2020)
- Tamara Borreguerro Sanchidrián, Tom Portoleau, Christian Artigues, Alvaro García Sánchez, Miguel Ortega Mier, et al.. Exact and heuristic methods for an aeronautical assembly line time-constrained scheduling problem with multiple modes and a resource leveling objective. 2021. ⟨hal-03344445⟩

Exemple de contexte industriel 1

- Chaine d'assemblage de l'Airbus A330 MRT

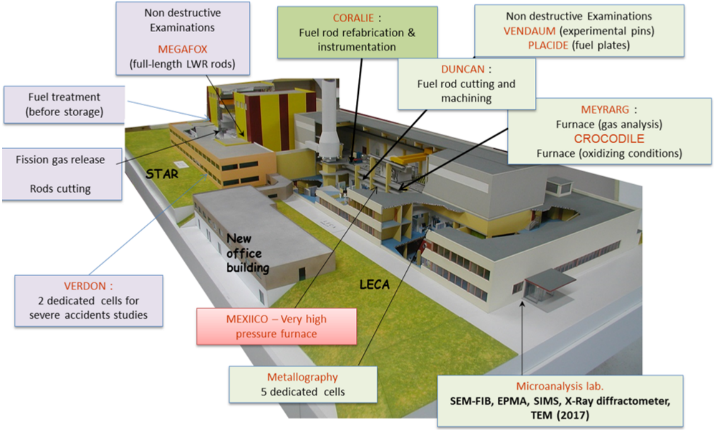
Borreguerro, Portoleau et al. 2021]

Exemple de contexte industriel 1

- Planning à date de fin imposée (*takt time*), objectif minimiser le nombre maximal d'opérateurs utilisés
- Environ 700 tâches
- Ressources : opérateurs multi-compétences
- Modélisation comme un problème multi-modes

Exemple de contexte industriel 2

- Installation Nucléaire de Base du CEA



- Laboratoire d'Examens des Combustibles Actifs (LECA) : examens post-irradiatoires des éléments combustibles et des matériaux de structure
- Station de Traitement, d'Assainissement et de Reconditionnement (STAR) : traitement et reconditionnement du combustible sans emploi

[Polo et al, 2020, 2021]

Exemple de contexte industriel 2

- 100 activités par semaine
- 180 personnes (opérateurs, expérimentateurs et agents de maintenance)
- 6 laboratoires

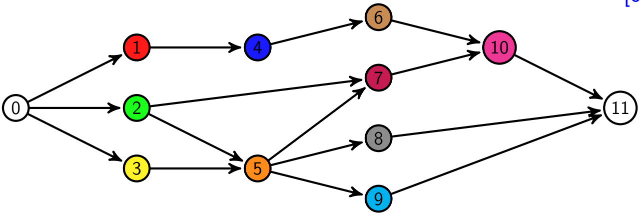
- Planning = élément important pour la sûreté et la sécurité de l'installation
- Assurer l'exécution la plus rapide de l'ensemble des activités
- Caractéristiques des activités et réglementations nucléaires $\Rightarrow$ ordonnancement complexe

Caractéristique des problèmes d'ordonnancement industriels complexes

- Ressources renouvelables à capacités limitées : cumulatives et disjonctives
- Précédences entre activités : simples ou généralisées
- Activités requièrent des opérateurs/techniciens avec des compétences particulières et des accréditations
- Les exécutions d'activités sont soumises à des fenêtres temporelles
- Certaines activités requièrent un nombre minimum d'opérateurs pour leur exécution
- Certaines activités doivent être exécutées sans interruption ; d'autres peuvent être interrompues
- Pour des raisons de sûreté, pendant les périodes de préemption, certaines ressources utilisées pour l'exécution d'une activité ne peuvent pas être libérées (ex : chambres de confinement) ; les autres oui (ex : opérateurs)

RCPSP : données du problème

- R ensemble de ressources, disponibilité limitée $B_k \geq 0, k \in R$,
- A ensemble d'activités (tâches), durée $p_i \geq 0, i \in A$, demande $b_{ik} \geq 0$ pour $k \in R$,
- E ensemble de contraintes de précédence (i, j), $i, j \in A$, $i < j$
- $\mathcal{T} = [0, T]$ intervalle de temps (horizon d'ordonnancement)



$|R| = 1, B = 4, \mathcal{T} = [0, 30]$

i	p_i	b_i
1	3	2
2	5	3
3	1	3
4	3	1
5	2	1
6	4	2
7	5	3
8	6	1
9	4	1
10	4	1

RCPSP : variables, objectif et contraintes

- $S_i \geq 0$ date de début de l'activité i
- $C_{\max} = S_{n+1}$ durée totale du projet

RCPSP (formulation conceptuelle)

$$\min_{S \in \mathcal{S}_T} S_{n+1}$$

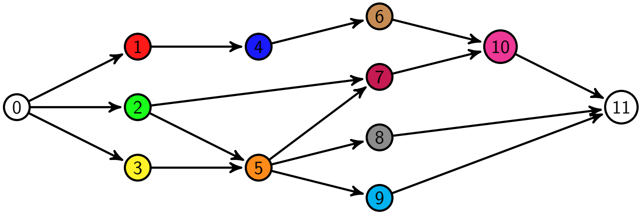
où $\mathcal{S}_T =$

$$\begin{cases} S_j \geq S_i + p_i & (i,j) \in E & \textit{Contraintes de précédence} \\ \sum_{j \in A(t)} b_{jk} \leq B_k & t \in \mathcal{T}, k \in R & \textit{Contraintes de ressources} \\ 0 \leq S_j \leq T - p_j & i \in A & \end{cases}$$

avec $A(t) = \{j \in A | t \in [S_j, S_j + p_j)\}$, $\forall t \in \mathcal{T}$

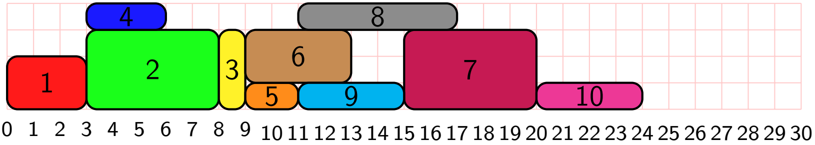
$\mathcal{S}_T^{\emptyset}$: ensemble des ordonnancements respectant les contraintes de précédence et l'horizon de temps T.

RCPSP : exemple de solution



$|R| = 1, B = 4, \mathcal{T} = [0, 30]$

i	p_i	b_i
1	3	2
2	5	3
3	1	3
4	3	1
5	2	1
6	4	2
7	5	3
8	6	1
9	4	1
10	4	1



RCPSP : complexité, variantes and méthodes

- NP difficile au sens fort
- Généralisation des problèmes à une machine, machines parallèles, job-shop, open-shop, flow-shop
- Multitude de variantes
  - Autres objectifs : $\min \sum_{i \in A} w_i(S_i + p_i)$
  - Contraintes de précédence généralisées $S_j \geq S_i + l_{ij}$
  - Temps de préparation, modes multiples, ressources consommables, tâches à intensités variables ...
  - Incertitude $p_i \in [p_i^{\min}, p_i^{\max}]$, $p_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$
- Méthodes exactes et approchées
  - Heuristiques et Metaheuristiques [Kolisch & Hartmann 2006, A. & Rivreau 2008]
  - Méthodes spécifiques de séparation et évaluation
  - Programmation linéaire en nombres entiers (MILP)
  - Programmation par contraintes (CP)
  - hybridations SAT/CP

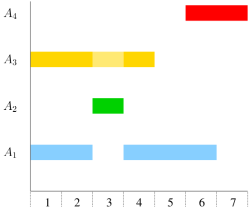
Multi-Skill Project Scheduling Problem with Partial Preemption = **MSPSP-PP**

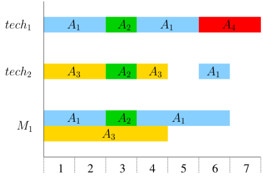
- RCPSP multi-compétences = MSPSP (*Multi-Skill Project Scheduling Problem*)
- Proposé par Emmanuel Néron (2002) dans le cadre de la planification de projets de développement
- Nombreux domaines d'applications
  - Pharmaceutique
  - Chimique
  - Nucléaire
- NP-difficile
- MSPSP-PP NP-difficile (RCPSP = cas particulier du MSPSP-PP avec toutes ressources mono-compétences et non préemptives)

MSPSP avec Préemption Partielle : Exemple

Activity	Duration	(Required skill, Quantity)	(Required resource, Quantity)	Deadline	Release date	Type
A_1	5	$(c_1, 1)$	$(M_1, 1)$	–	–	$\overline{P}$
A_2	1	$(c_3, 1), (c_4, 1)$	$(M_1, 1)$	4	3	$\overline{NP}$
A_3	3	$(c_2, 1)$	$(M_1, 1)$	–	–	$\overline{PP}$ (M_1 cannot be released)
A_4	2	$(c_1, 1)$	–	–	6	$\overline{NP}$

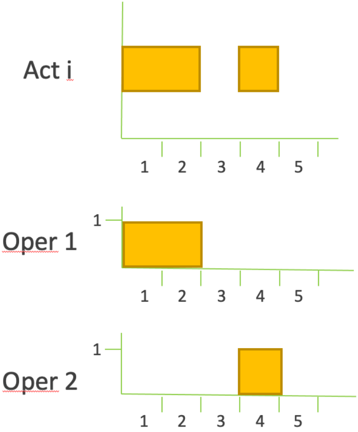
Technician	Mastered skills	Resource	Capacity
$tech_1$	$\{c_1, c_3\}$	M_1	2
$tech_2$	$\{c_1, c_2, c_4\}$		





Modèle PLNE – Variables globales

$Y_{i,t} = 1$ si activité i en exécution pendant la période t
$O_{j,i,t} = 1$ si opérateur j alloué à l'activité i pendant la période t
$Pp_{i,t} = 1$ si activité i partiellement préemptive est interrompue pendant la période t

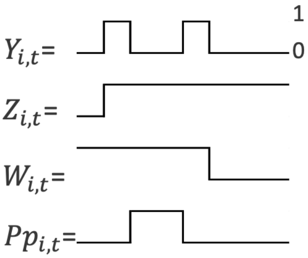


$$\begin{cases} Y_{i,t} & : 1\ 1\ 0\ 1\ 0 \\ Pp_{i,t} & : 0\ 0\ 1\ 0\ 0 \\ O_{1,i,t} & : 1\ 1\ 0\ 0\ 0 \\ O_{2,i,t} & : 0\ 0\ 0\ 1\ 0 \end{cases}$$

Modèle PLNE 1 – Variables binaires

$Z_{i,t} = 1$ pour les périodes $t \geq$ à la date de début de l'activité i

$W_{i,t} = 1$ pour les périodes $t \leq$ à la date de fin de l'activité i



$$Z_{i,t} \geq Y_{i,t'} \qquad \forall i \notin \overline{P}, \forall t' \leq t \qquad (1)$$

$$W_{i,t} \geq Y_{i,t'} \qquad \forall i \notin \overline{P}, \forall t' \geq t \qquad (2)$$

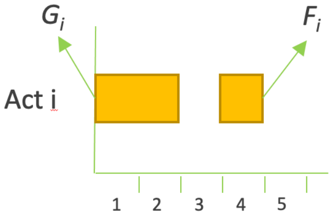
$$Pp_{i,t} = Z_{i,t} + W_{i,t} - Y_{i,t} - 1 \qquad \forall i \in \overline{PP} \qquad (3)$$

Modèle PLNE 2 – Variables continues

G_i : date de début de l'activité i

F_i : date de fin de l'activité i

$F_i - G_i - D_i = \#$ périodes préemptées



$$Pp_{i,t} \leq 1 - Y_{i,t} \qquad \forall i \in \overline{PP} \qquad (4)$$

$$Pp_{i,t} \leq \sum_{t'=1}^{t} Y_{i,t'} \qquad \forall i \in \overline{PP} \qquad (5)$$

$$Pp_{i,t} \leq \sum_{t'=t}^{T} Y_{i,t'} \qquad \forall i \in \overline{PP} \qquad (6)$$

$$F_i - G_i + 1 \leq D_i + \sum_{t=1}^{T} Pp_{i,t} \qquad \forall i \in \overline{PP} \qquad (7)$$

Modèle 1 ("binaire") *vs.* modèle 2 ("continu")

Théorème

La formulation "binaire" est plus forte que la formulation "continue" (meilleure relaxation linéaire).

Démonstration.

(fragment)

$$G_i = T - \sum_{t=1}^{T} Z_{i,t} + 1 \quad \text{et} \quad F_i = \sum_{t=1}^{T} W_{i,t}$$

$$\begin{aligned} F_i - G_i + 1 &= \sum_{t=1}^{T} W_{i,t} + \sum_{t=1}^{T} Z_{i,t} - T \\ &= \sum_{t=1}^{T} Pp_{i,t} + \sum_{t=1}^{T} Y_{i,t} \\ &= \sum_{t=1}^{T} Pp_{i,t} + D_i \end{aligned}$$

$$Pp_{i,t} = Z_{i,t} + W_{i,t} - Y_{i,t} - 1$$

$\square$

Jeu de données

- 4 ensembles de 50 instances : Fort. P, Fort. PP, Fort. NP, Unif.

	Fort. P	Fort. PP	Fort. NP	Unif.
$\overline{NP}$	10%	10%	**80%**	33,3%
$\overline{PP}$	10%	**80%**	10%	33,3%
$\overline{P}$	**80%**	10%	10%	33,3%

- 30 activités ; $5 \leq D_i \leq 10$
- Jusqu'à 15 compétences
- 8 techniciens, chacun maîtrisant entre 5 et 10 compétences
- 8 ressources cumulatives
- 20% des activités avec fenêtres temporelles
- 10% des activités avec contraintes de précédence
- Horizon de planification entre 70 et 90 unités de temps (heure)

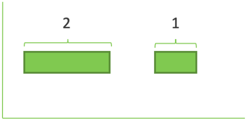
Résultats numériques

- IBM ILOG CPLEX
- Time out = 10 mn
- “warm start” (algorithme glouton)

	Modèle binaire			**Modèle continu**		
Jeu de données	Nombre d’instances optimales	Temps moyen à l’optimum	Ecart moyen	Nombre d’instances optimales	Temps moyen à l’optimum	Ecart moyen
Fort. P	47	110.85 s	0.01 %	46	87.39 s	0.05 %
Fort. PP	19	262.99 s	1.68 %	15	154.12 s	2.69 %
Fort. NP	**0**	–	**9.43 %**	**0**	–	**9.45 %**
Unif.	18	289.35 s	1.85 %	19	216.12 s	1.99 %
Global	84	183.51 s	3.24 %	80	130.48 s	3.55 %

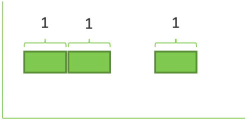
Modèle PPC (1)

- IBM ILOG CP Optimizer : variables d'intervalle
- Les variables d'intervalle peuvent être optionnelles
- Deux façons de modéliser les ordonnancements préemptifs avec les variables d'intervalle :
  - Nombre variable de morceaux avec durée variable : le nombre et la taille de chaque intervalle est à décider au moment de résoudre le modèle



Activité de durée 3

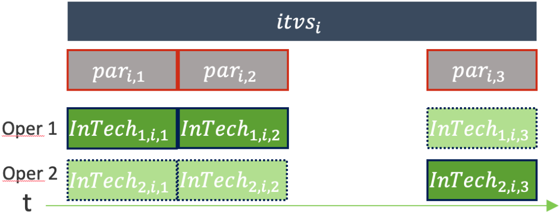
  - Nombre fixe de morceaux de durée unitaire : le nombre d'intervalles pour chaque activité est égal à sa durée



- Collaboration avec Philippe Laborie (IBM)

Modèle PPC (2)

- $itvs_i$: variable d'intervalle entre le début et la fin de l'activité i
- $par_{i,v}$: variable d'intervalle indiquant le début et la fin de chaque unité de durée v de l'activité i. Une seule partie pour les activités non-préemptives et D_i parties pour les autres
- $InTech_{j,i,v}$: variable d'intervalle optionnelle indiquant les périodes sur lesquelles chaque opérateur travaille (ou pas) sur chaque activité



comparaison PPC/PLNE

Table 4. Distribution of preemption types per set of instances.

	Set A1	Set B1	Set C1	Set D1
Non-preemptive	10%	10%	80%	33.3%
Partially preemptive	10%	80%	10%	33.3%
Preemptive	80%	10%	10%	33.3%

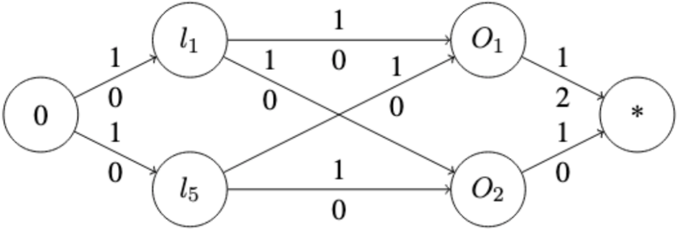
Table 6. Results of MILP and CP models after 10 min of computation using warm start

	MILP			CP		
	Number of instances solved to optimality	Average time to optimality	Average gap	Number of instances solved to optimality	Average time to optimality	Average gap
Set A1	46	87.39 s	0.05%	39	67.17 s	0.18%
Set B1	15	154.12 s	2.69%	40	88.01 s	0.15%
Set C1	0	-	9.45%	41	108.73 s	0.39%
Set D1	19	216.12 s	1.99%	40	76.14 s	0.21%
All	80	130.48 s	3.55%	160	85;27 s	0.23%

Heuristiques : Algorithme glouton + flot

$$g_{io} = |\{l \in \mathcal{L} | m_{ol} = 1\} \cap \{l \in \mathcal{L} | a_{il} > 0\}|.$$

$$c_o = \frac{\sum_{j \in \mathcal{Q} \setminus \{i\}} p_j g_{jo}}{g_{io}}.$$



Heuristiques : GRASP

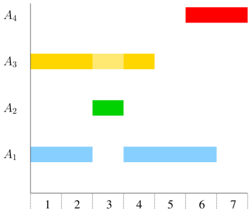
Algorithm 3: GRASP for the MSPSP-PP

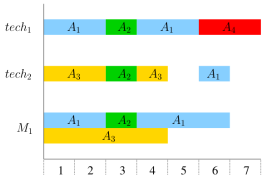
$git \leftarrow 0$; $fail \leftarrow 0$; $it \leftarrow 0$; $fit \leftarrow 0$; $\varepsilon \leftarrow \emptyset$; $OLDSIM \leftarrow \infty$; $OLDFAIL \leftarrow -\infty$
Choose randomly α and initialise β, δ and γ to 0.33;
while $git \leq MAXITER$ **do**
    $it \leftarrow it + 1$ `// Iterations counter for parameters`
    `// Generate initial solution`
    Run greedy SGS (algorithm 1) with parameters α, β, δ and γ to update the RCL
    if *Algorithm 1 found a feasible solution with sequence* σ **then**
        Run tree-based local search (Algorithm 2) using sequence σ for selecting activities;
        Update ε with σ;
        $git \leftarrow git + 1$
    else
        $fail \leftarrow fail + 1$ `// Fails counter`
    end
    `// Update` α, β, δ `and` γ
    if $it = NITER$ **then**
        Update the probability of each α;
        if $|\Theta| \neq 0$ **then**
            $AVSIM \leftarrow \frac{\sum_{\sigma \in \Theta} SIM(\sigma, \Theta)}{|\Theta|}$;
            if $AVSIM < OLDSIM$ **and** $\delta \leq 0.9$ **then**
                $\delta \leftarrow \delta + 0.1$
            else if $\delta \geq 0.1$ **then**
                $\delta \leftarrow \delta - 0.1$
            end
        end
        if $fail < OLDFAIL$ **then**
            $\gamma \leftarrow \gamma - 0.1$; $fit \leftarrow 0$
        else if $\gamma \geq 0.9$ *and* $fit = NINF$ **then**
            exit `// fail`
        else if $\gamma \geq 0.9$ **then**
            $fit \leftarrow fit + 1$
        else
            $\gamma \leftarrow \gamma + 0.1$
        end
        $\beta \leftarrow 1 - \delta - \gamma$; $it \leftarrow 0$; $OLDFAIL \leftarrow fail$; $fail \leftarrow 0$; $OLDSIM \leftarrow AVSIM$
    end
end

MSPSP avec Préemption Partielle : Exemple

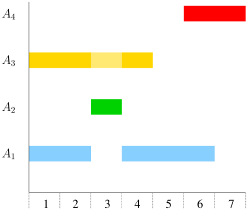
Activity	Duration	(Required skill, Quantity)	(Required resource, Quantity)	Deadline	Release date	Type
A_1	5	$(c_1, 1)$	$(M_1, 1)$	–	–	$\overline{P}$
A_2	1	$(c_3, 1), (c_4, 1)$	$(M_1, 1)$	4	3	$\overline{NP}$
A_3	3	$(c_2, 1)$	$(M_1, 1)$	–	–	$\overline{PP}$ (M_1 cannot be released)
A_4	2	$(c_1, 1)$	–	–	6	$\overline{NP}$

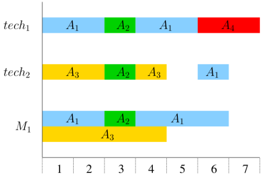
Technician	Mastered skills	Resource	Capacity
$tech_1$	$\{c_1, c_3\}$	M_1	2
$tech_2$	$\{c_1, c_2, c_4\}$		

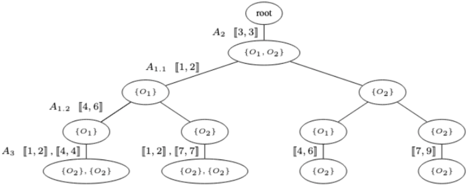




Heuristiques : Tree based local search



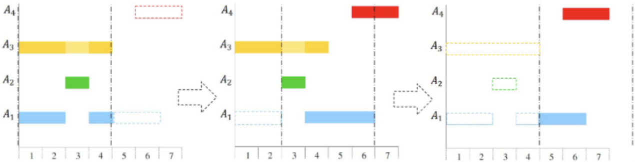




Heuristiques : LNS

Algorithm 4: LNS for the MSPSP-PP

Generate initial solution using the multi-pass greedy SGS (Algorithm 1);
if *a feasible solution has been found* **then**
    $Improvement \leftarrow True$;
    Define initial time window;
    while *Improvement* **do**
        Select activities for the subproblem;
        Construct subproblem;
        Solve subproblem using the CP or MILP exact method (Polo-Mejía et al., 2020);
        if *Subproblem solution is improved* **then**
            Include subproblem solution in the global solution;
            Reschedule the activities to the right of the current time window with multi-pass greedy Algorithm 1
        end
        if $C_{\max}$ *is inside the current time window* **then**
            if *Current* $C_{\max}$ *is equal than previous one* **then**
                $Improvement \leftarrow False$;
            else
                Return the time window to period t where first change happened (compared to previous solution)
            end
        else
            Shift time window to start at the middle of the previous one;
        end
    end
end



Résultats sur des instances de la littérature non-préemptives

	Gap for GRASP	Execution time (sec) *GRASP*	*Young et al. (2017)*
Set 1A	2.8%	40.3	0.5
Set 1B	2.3%	151.1	536.3
Set 2A	4.7%	67.6	196.6
Set 2B	4.3%	58.9	122.8
Set 2C	5.21%	68.6	1.2

56 improved solutions