Programmation Mathématique dans les ERP : deux cas d'utilisation au sein de l'ERP Copilote

Groleaz Lucas

14-12-2023

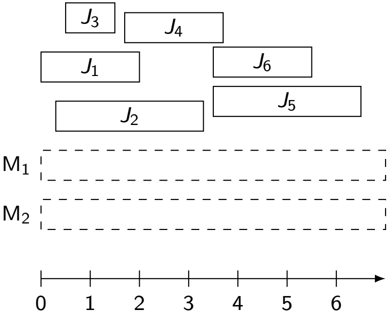
Société Infologic

- Entreprise fondée en 1982
- Développe et intègre son propre ERP, Copilote, spécialisé dans l'agro-alimentaire
- Plus de 600 sites équipés, 20.000 postes informatiques
- Un seul et même logiciel, paramétré pour répondre aux besoins spécifiques de chaque client

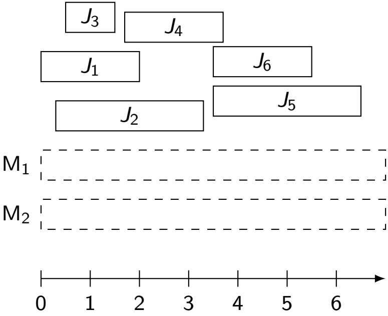
Cas d'utilisation

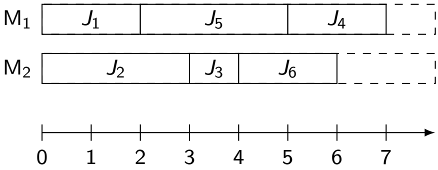
- Gestion de la production (GP)
- Préparation de commandes
- Déplacement des caristes

Problème de base



Problème de base

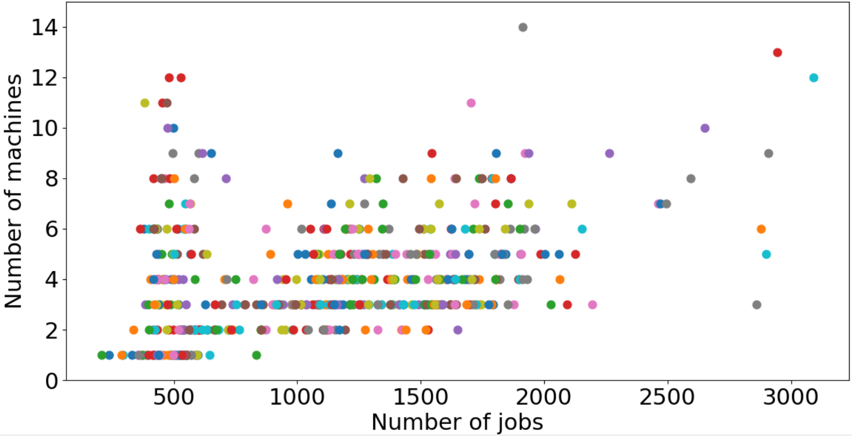




Étude comparative de méthodes de résolutions

- Récupération des données d'un client
- Modélisation de plusieurs contraintes
- Comparaisons de méthodes

Jeu de données



- 548 instances
- Durée des jobs: entre 3 secondes et 1h30min (moyenne de 1 min 30s)

Jeu de données

	Mean	Median	Standard deviation	Min	Max
Number of jobs	1112.7	1136.5	546.42	207	3092
Number of machines	4.07	4	2.22	1	14
Number of groups	163.4	169	70.2	55	462
Jobs duration (sec)	97.35	36.0	250.5	3	5400
Release date	06:35:21	09:01:00	04:49:17	00:00:00	15:24:00
Due dates	14:38:58	14:00:00	03:02:34	09:00:00	23:45:00
Machines speed	1.08	1.1	0.05	0.7	1.4

Méthodes comparées

- **MIP**: CPLEX version 12.9.0 [1]
- **Programmation par contraintes**: CPOptimizer 12.9.0 [2]
- **Programmation par contraintes / SAT** : CP-SAT de OrTools, version 9.7 [3]
- Recherche tabou
- Algorithmes de colonies de fourmis
- Méthodes hybrides (ACO - Tabou, ACO - CPO)

[1] Manual, "Ibm Ilog Cplex Optimization Studio", 1987
[2] Laborie et al., "IBM ILOG CP Optimizer for Scheduling", 2018
[3] Perron and Furnon, OR-Tools, 2019

Mesure de performance

- $x \in \mathcal{I}$ une instance, $a \in \mathcal{A}$ un algorithme, et t une limite de temps.

Mesure de performance

- $x \in \mathcal{I}$ une instance, $a \in \mathcal{A}$ un algorithme, et t une limite de temps.
- $x_a^t =$ la valeur de la meilleure solution trouvée par a pour x en t secondes.

Mesure de performance

- $x \in \mathcal{I}$ une instance, $a \in \mathcal{A}$ un algorithme, et t une limite de temps.
- $x_a^t =$ la valeur de la meilleure solution trouvée par a pour x en t secondes.
- $x^* = \min_{a \in \mathcal{A}} x_a^{3600}$ (Solution de référence)

Mesure de performance

- $x \in \mathcal{I}$ une instance, $a \in \mathcal{A}$ un algorithme, et t une limite de temps.
- $x_a^t =$ la valeur de la meilleure solution trouvée par a pour x en t secondes.
- $x^* = \min_{a \in \mathcal{A}} x_a^{3600}$ (Solution de référence)

$$ir_{a,x}^t = \begin{cases} 1 & \text{si } x_a^t = x^* \\ x^*/x_a^t & \text{sinon} \end{cases}$$

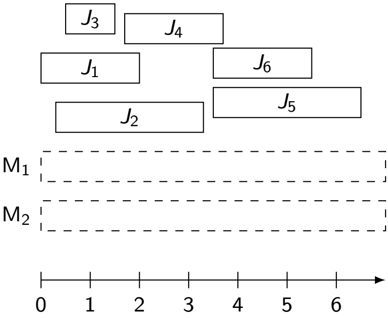
Mesure de performance

- $x \in \mathcal{I}$ une instance, $a \in \mathcal{A}$ un algorithme, et t une limite de temps.
- $x_a^t =$ la valeur de la meilleure solution trouvée par a pour x en t secondes.
- $x^* = \min_{a \in \mathcal{A}} x_a^{3600}$ (Solution de référence)

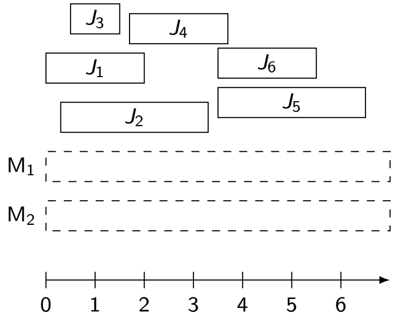
$$ir_{a,x}^t = \begin{cases} 1 & \text{si } x_a^t = x^* \\ x^*/x_a^t & \text{sinon} \end{cases}$$

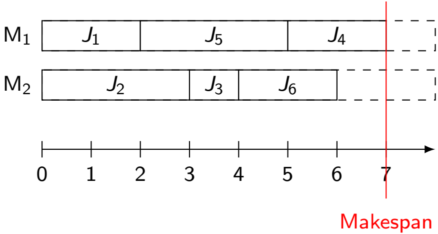
- $ir_{a,x}^t \in [0, 1]$

Problème d'assignation (Problème de Partition) $P||C_{max}$



Problème d'assignation (Problème de Partition) $P||C_{max}$

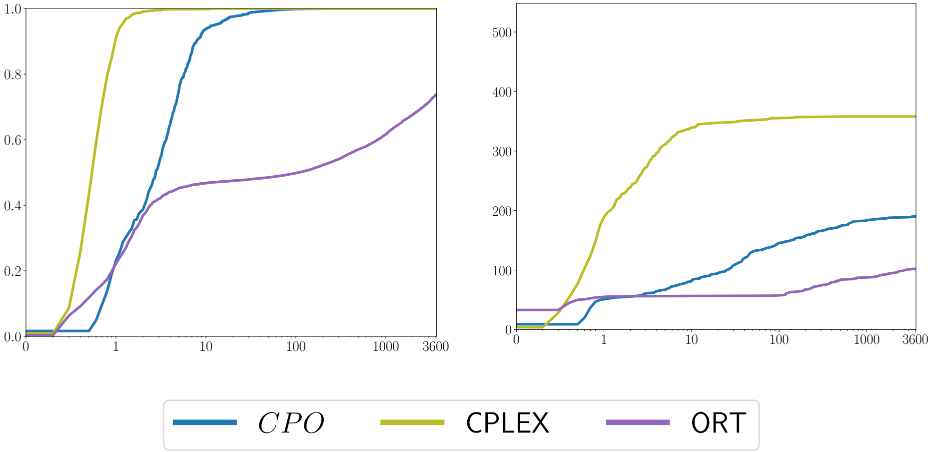




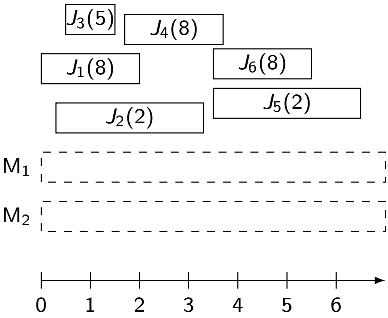
Problème d'assignation (Problème de Partition) $P||C_{max}$

$ir^t_{a,x}$ moyen

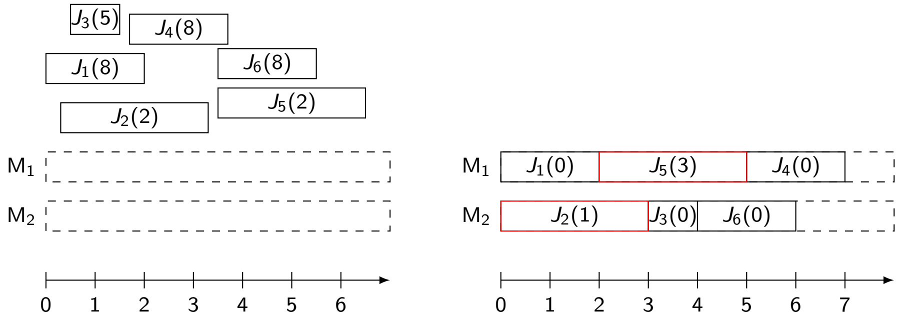
Optimalité des solutions



Problème de minimisation de retard $P|r_j|\sum T_j$

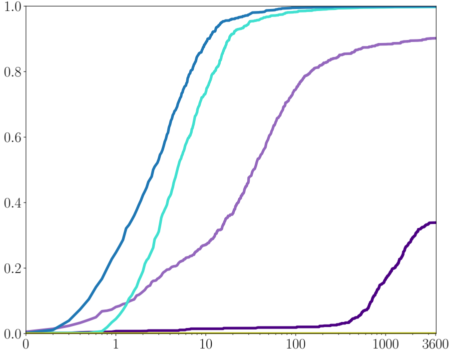


Problème de minimisation de retard $P|r_j|\sum T_j$

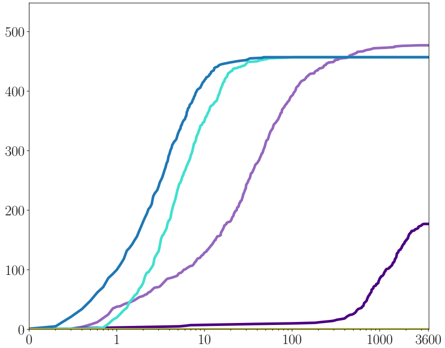


Problème de minimisation de retard $P|r_j|\sum T_j$

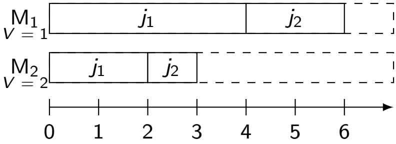
$ir^t_{a,x}$ moyen



Optimalité des solutions

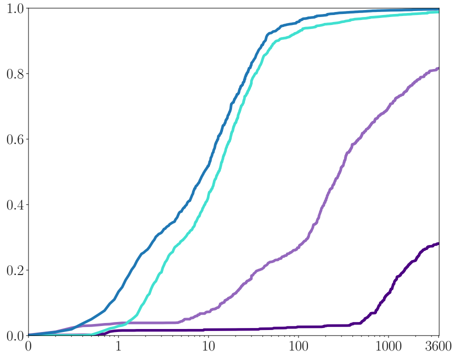


Vitesses différentes sur les machines $Q|r_j|\sum T_j$

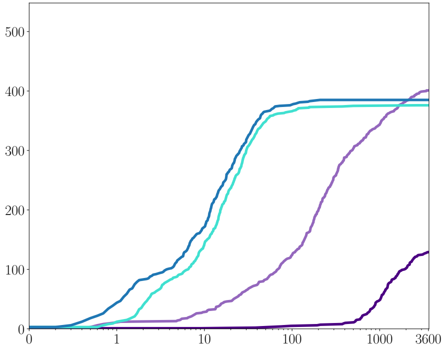


Vitesses différentes sur les machines $Q|r_j|\sum T_j$

$ir^t_{a,x}$ moyen

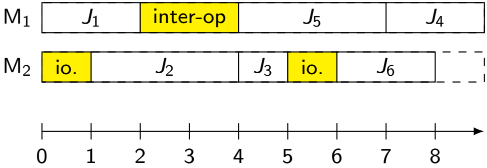


Optimalité des solutions

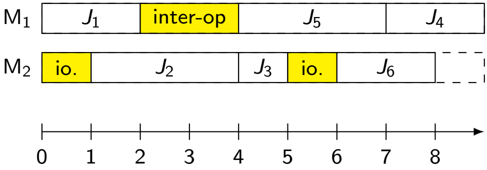


CPO_{16T} ORT_{16T} ORT_{1T}
CPO_{1T}

Sequence-dependent setup-times $Q|r_j, s_{jk}|\sum T_j$



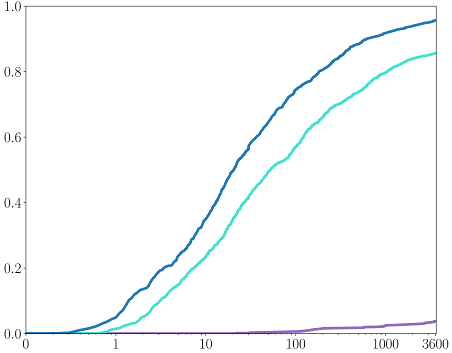
Sequence-dependent setup-times $Q|r_j, s_{jk}|\sum T_j$



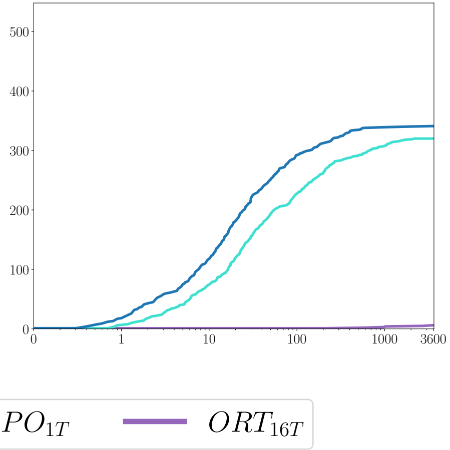
$$
\begin{array}{c}
 \\
j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \\ j_6
\end{array}
\begin{array}{c}
\begin{array}{ccccccc}
1^{er} & j_1 & j_2 & j_3 & j_4 & j_5 & j_6
\end{array} \\
\begin{pmatrix}
0 & \emptyset & 2 & 3 & 1 & 0 & 0 \\
1 & 0 & \emptyset & 2 & 4 & 1 & 0 \\
1 & 6 & 0 & \emptyset & 1 & 3 & 2 \\
0 & 2 & 0 & 1 & \emptyset & 0 & 2 \\
2 & \color{red}{2} & 3 & 1 & 1 & \emptyset & 2 \\
2 & 3 & 4 & 1 & 0 & 0 & \emptyset
\end{pmatrix}
\end{array}
$$

Sequence-dependent setup-times $Q|r_j, s_{jk}|\sum T_j$

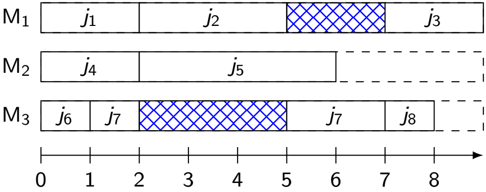
$ir^t_{a,x}$ moyen



Optimalité des solutions

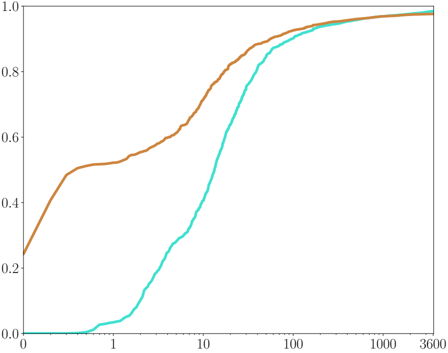


Calendrier machines $Q|r_j, brkdwn|\sum T_j$

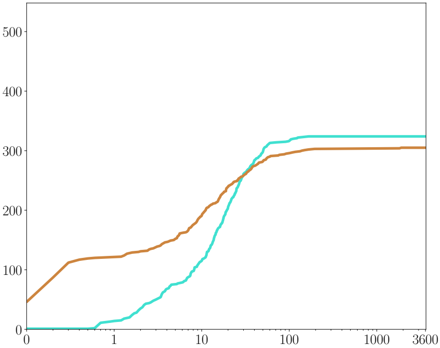


Calendrier machines $Q|r_j, brkdwn|\sum T_j$

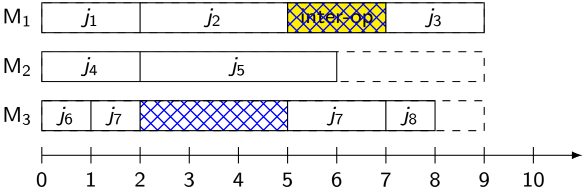
$ir^t_{a,x}$ moyen



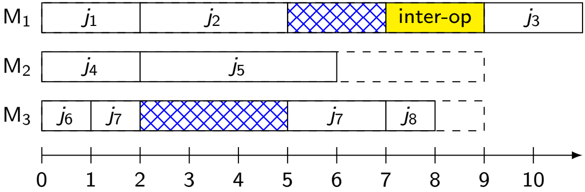
Optimalité des solutions



Pauses et temps inter-opérations $Q|r_j, brkdwn, s_{jk}|\sum T_j$

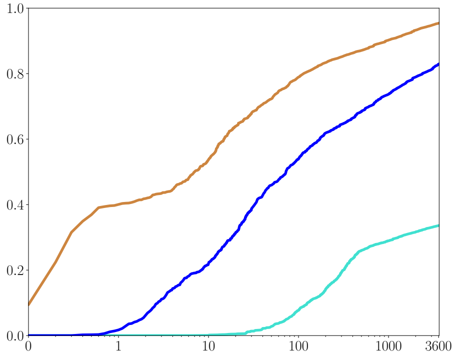


Pauses et temps inter-opérations $Q|r_j, brkdwn, s_{jk}|\sum T_j$

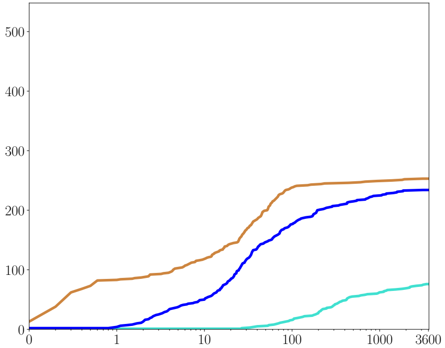


Pauses et temps inter-opérations $Q|r_j, brkdwn, s_{jk}|\sum T_j$

$ir^t_{a,x}$ moyen



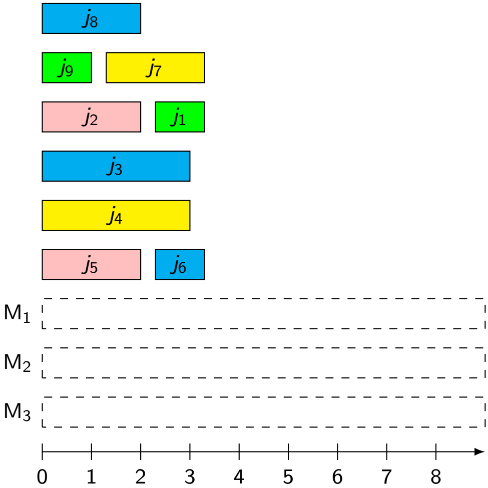
Optimalité des solutions



$CPO - HEUR$ CPO_{1T} ACO-Tabu

Contrainte cumulative de groupe $Q|r_j, s_{jk}, GC|\sum T_g$ [4]

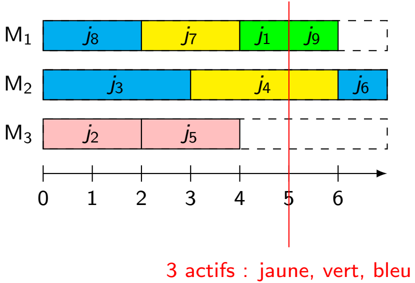
- Les jobs sont partitionnés en groupes



[4] Groleaz, "The Group Cumulative Scheduling Problem", 2021

Contrainte cumulative de groupe $Q|r_j, s_{jk}, GC|\sum T_g$ [4]

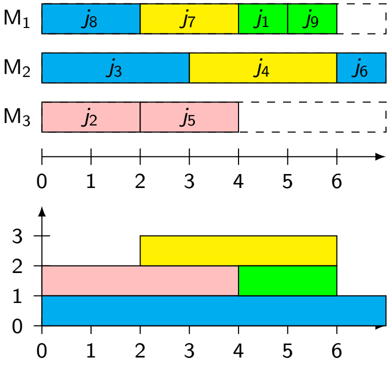
- Les jobs sont partitionnés en groupes
- Un groupe est actif à un instant t si au moins un de ses jobs est commencé et au moins un n'est pas fini



[4] Groleaz, "The Group Cumulative Scheduling Problem", 2021

Contrainte cumulative de groupe $Q|r_j, s_{jk}, GC|\sum T_g$ [4]

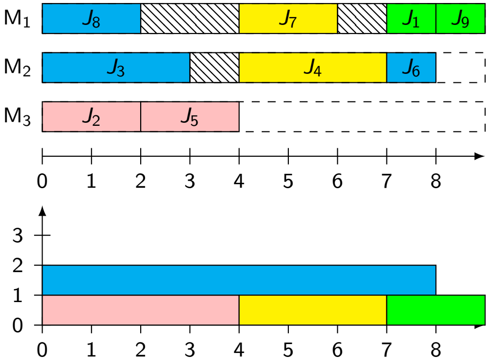
- Les jobs sont partitionnés en groupes
- Un groupe est actif à un instant t si au moins un de ses jobs est commencé et au moins un n'est pas fini
- La contrainte GC assure $\forall t, \#active(t) \leq L$



[4] Groleaz, "The Group Cumulative Scheduling Problem", 2021

Contrainte cumulative de groupe $Q|r_j, s_{jk}, GC|\sum T_g$ [4]

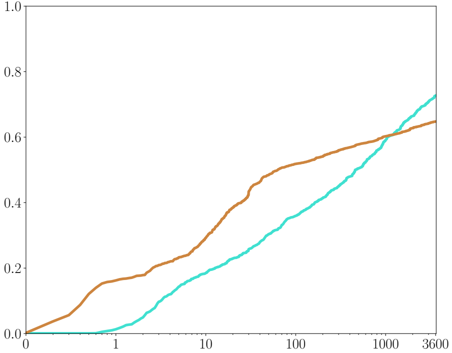
- Les jobs sont partitionnés en groupes
- Un groupe est actif à un instant t si au moins un de ses jobs est commencé et au moins un n'est pas fini
- La contrainte GC assure $\forall t, \#active(t) \leq L$



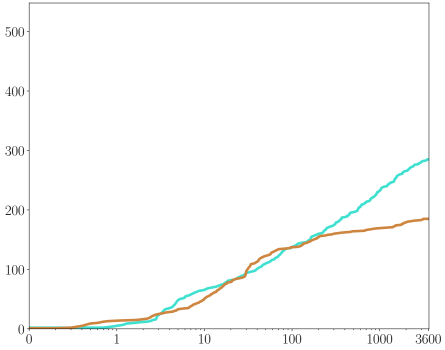
[4] Groleaz, "The Group Cumulative Scheduling Problem", 2021

Contrainte cumulative de groupe $Q|r_j, s_{jk}, GC|\sum T_g$

$ir^t_{a,x}$ moyen

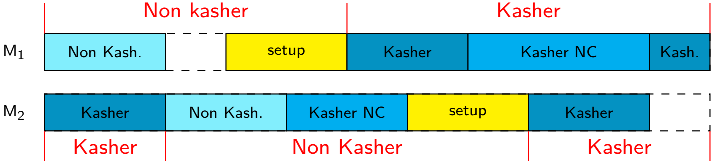


Optimalité des solutions



CPO_{1T} ACO-Tabu

Etats de la machine



Conclusion ordonnancement

- Utilisation en production
  - Heuristique
  - Recherche taboue
  - Post process
- Raisons
  - Contraintes exotiques à gérer
  - Performances

Type de roll

- Roll CC (à étages variables)

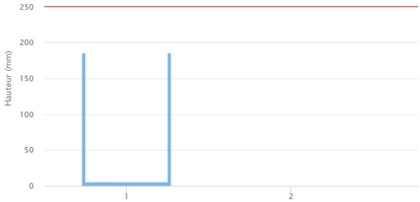
- Roll bois (à étages fixes)

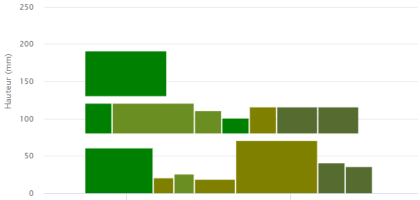
Plantes

- Quantité commandée
- Quantité max par étage
- Hauteur

Roll CC (à étages variables)

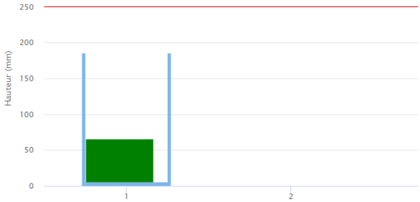
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

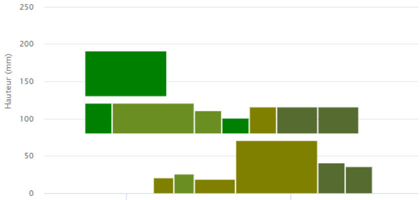




Roll CC (à étages variables)

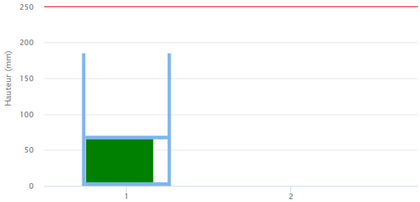
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

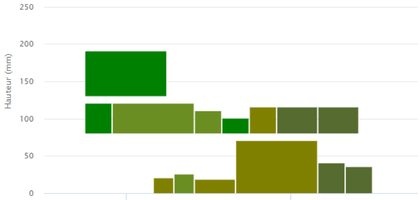




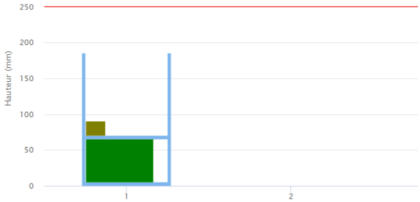
Roll CC (à étages variables)

- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

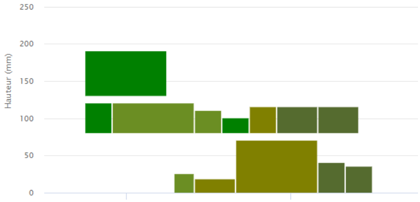




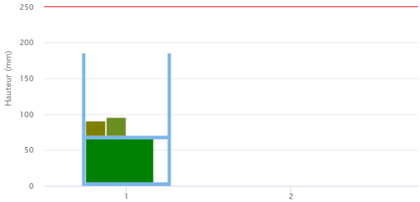
Roll CC (à étages variables)



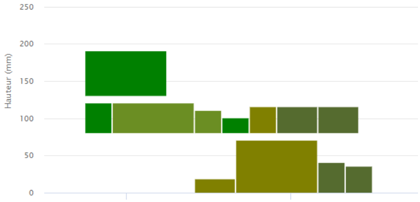
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage



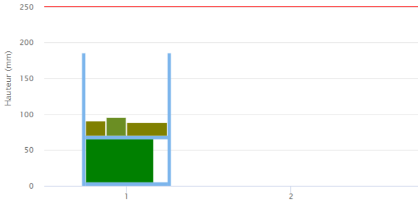
Roll CC (à étages variables)



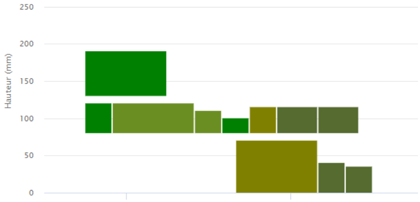
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage



Roll CC (à étages variables)

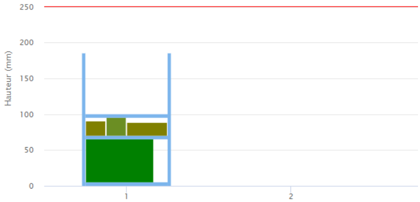


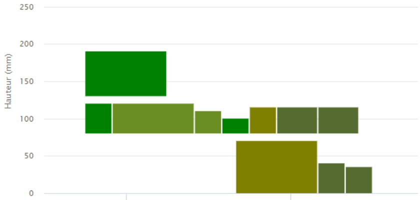
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage



Roll CC (à étages variables)

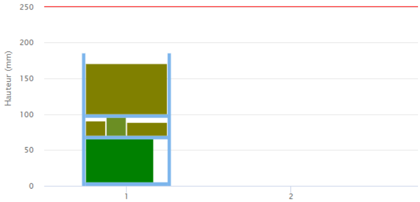
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

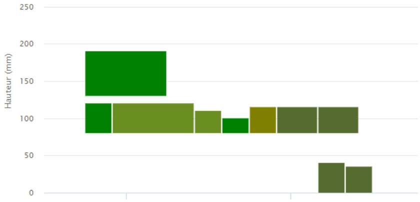




Roll CC (à étages variables)

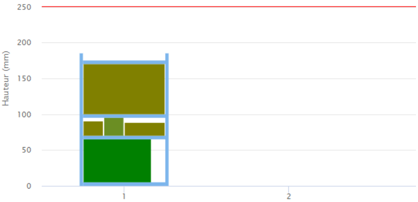
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

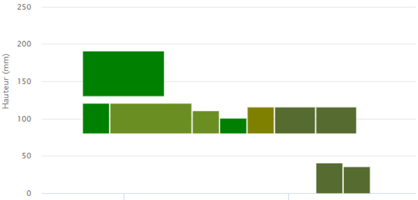




Roll CC (à étages variables)

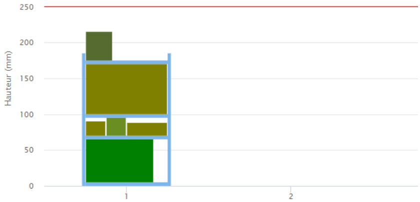
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

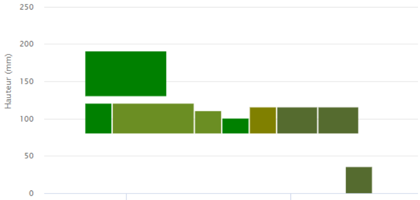




Roll CC (à étages variables)

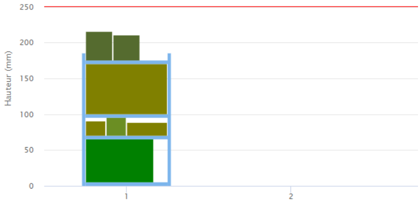
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

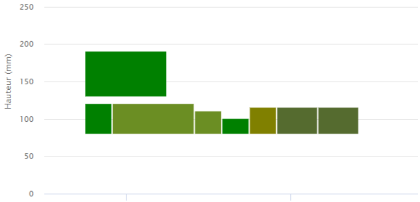




Roll CC (à étages variables)

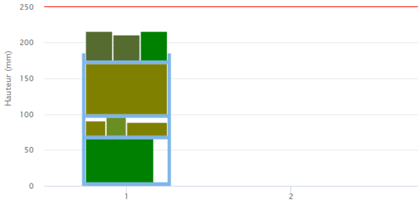
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

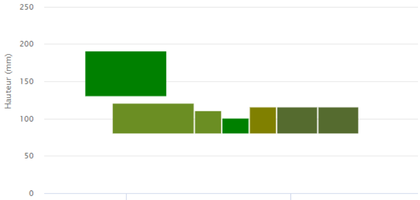




Roll CC (à étages variables)

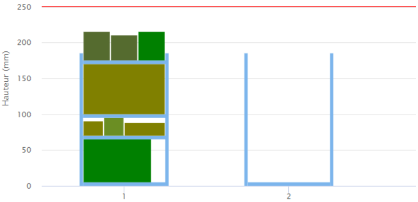
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

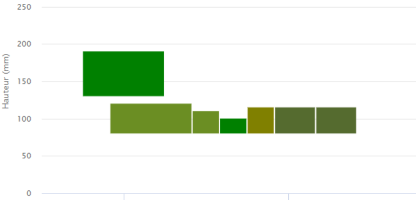




Roll CC (à étages variables)

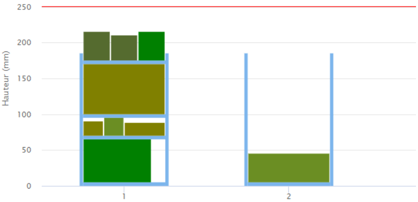
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

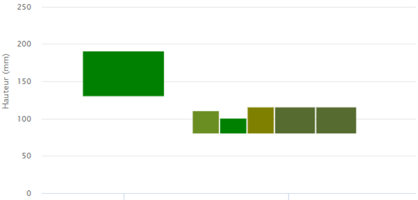




Roll CC (à étages variables)

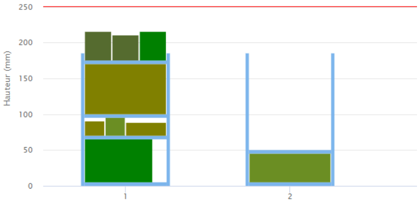
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

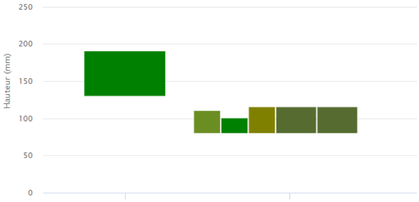




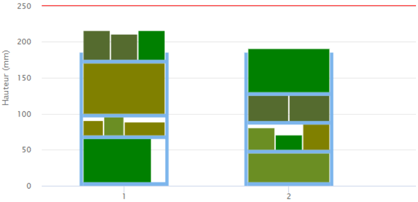
Roll CC (à étages variables)

- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage





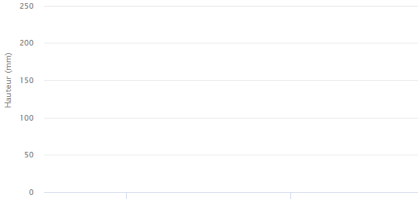
Roll CC (à étages variables)

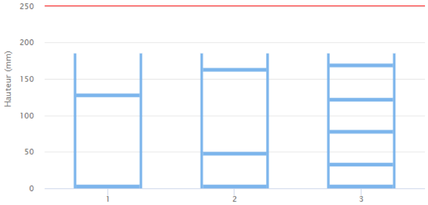


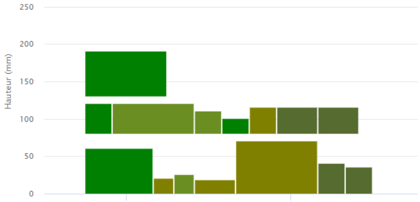
- Hauteur maximale
- Position maximale dernier étage
- Épaisseur d'étage
- Pas d'étage

Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

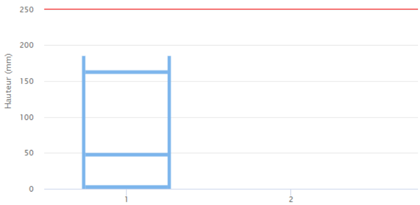


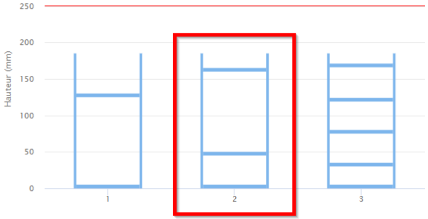


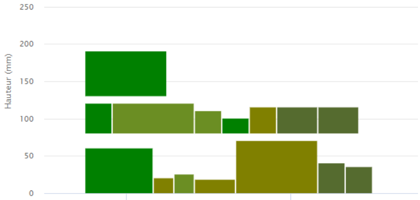


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

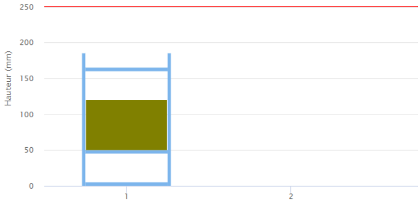


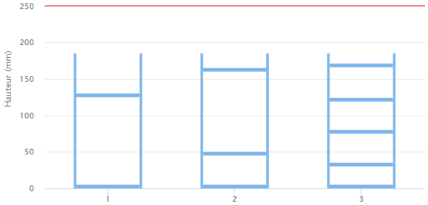


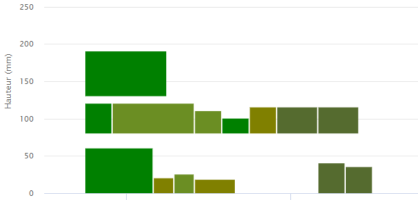


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

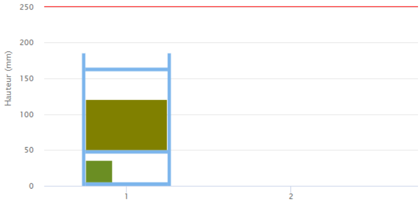


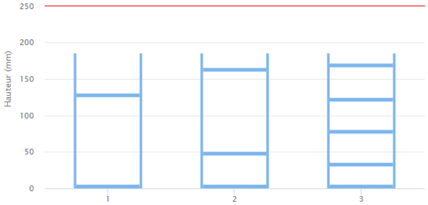


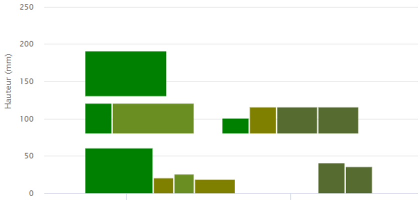


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

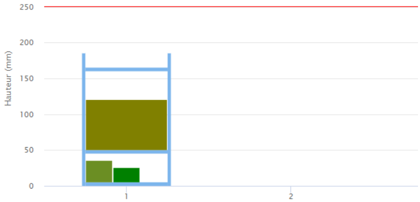


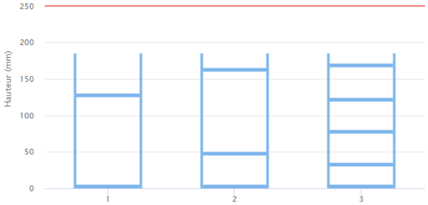


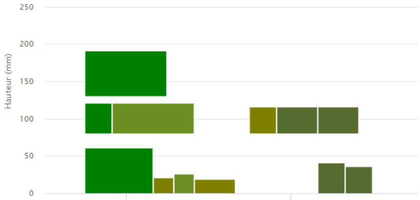


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

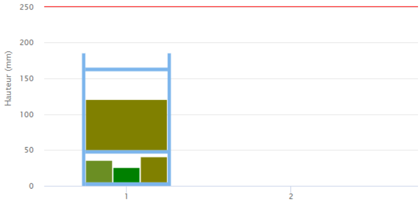


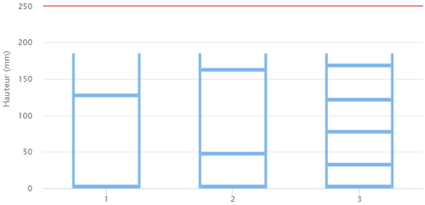


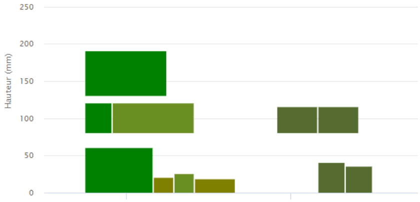


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

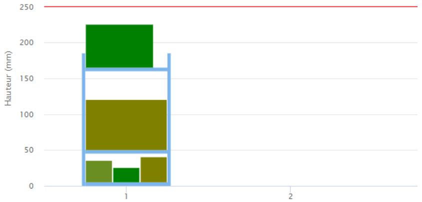


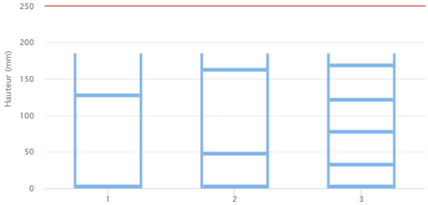


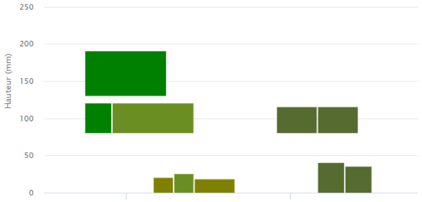


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

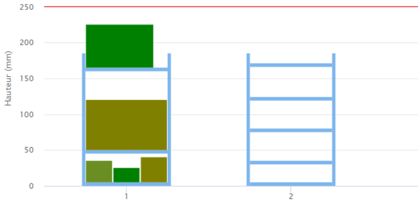


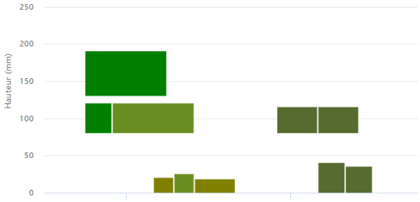




Roll Bois (à étages fixes)

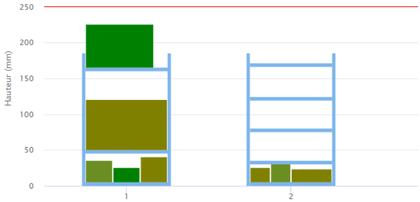
- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

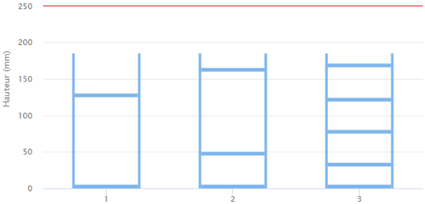


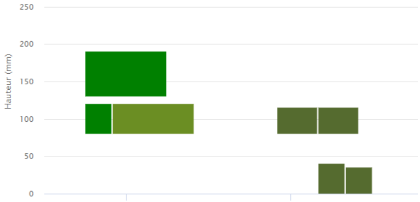


Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage

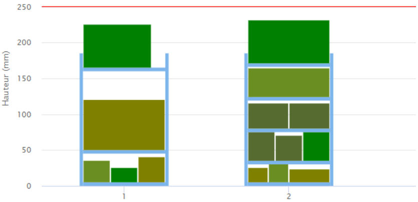


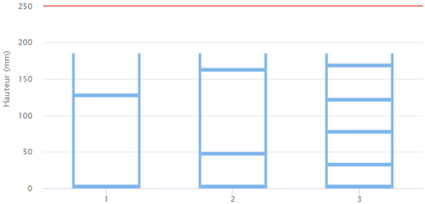




Roll Bois (à étages fixes)

- Différents configurations disponibles
- Position des étages spécifiées pour chaque config
- Épaisseur d'étage





Contraintes

- Les plantes de plus de 15kg sont sur le premier étage
- Les plantes de plus de 5kg sont sur les quatre premiers étages
- Les plantes BIO sont au-dessus des non-BIO
- Les articles de type MAF (marché au fleur) ne sont pas sur le même roll que les articles PEPI (pépinières)
- Il ne faut pas disperser sur plusieurs étages des plantes identiques qui occupent un étage complet

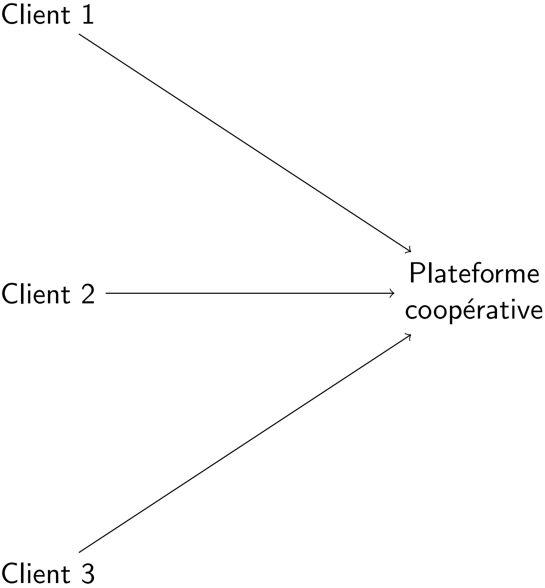
Plateforme coopérative

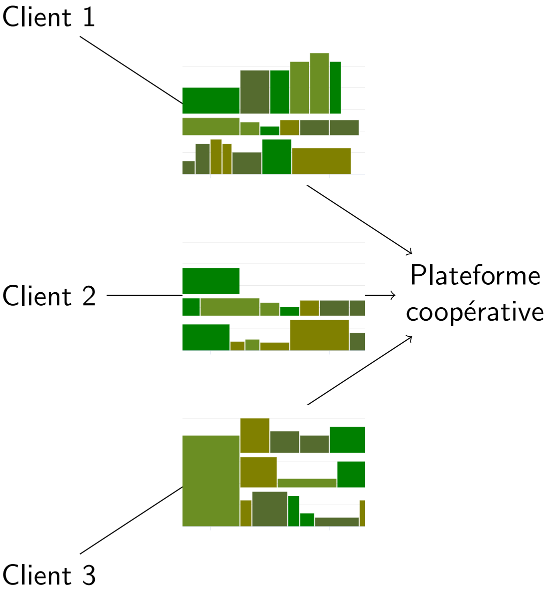
Client 1

Client 2

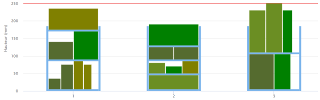
Plateforme coopérative

Client 3

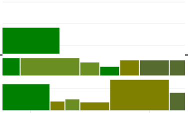

Client 1
Client 2
Plateforme
coopérative
Client 3


Client 1
Client 2
Client 3
Plateforme coopérative

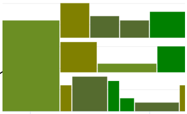
Client 1



Client 2

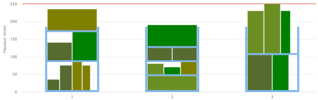


Plateforme coopérative

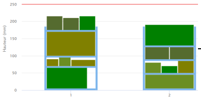


Client 3

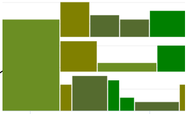
Client 1



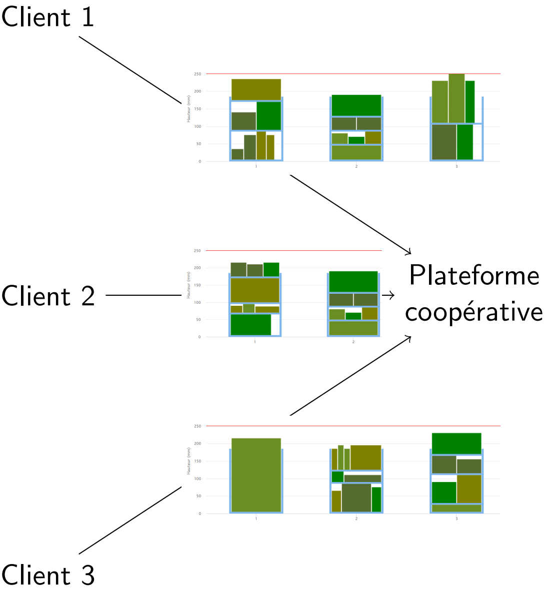
Client 2

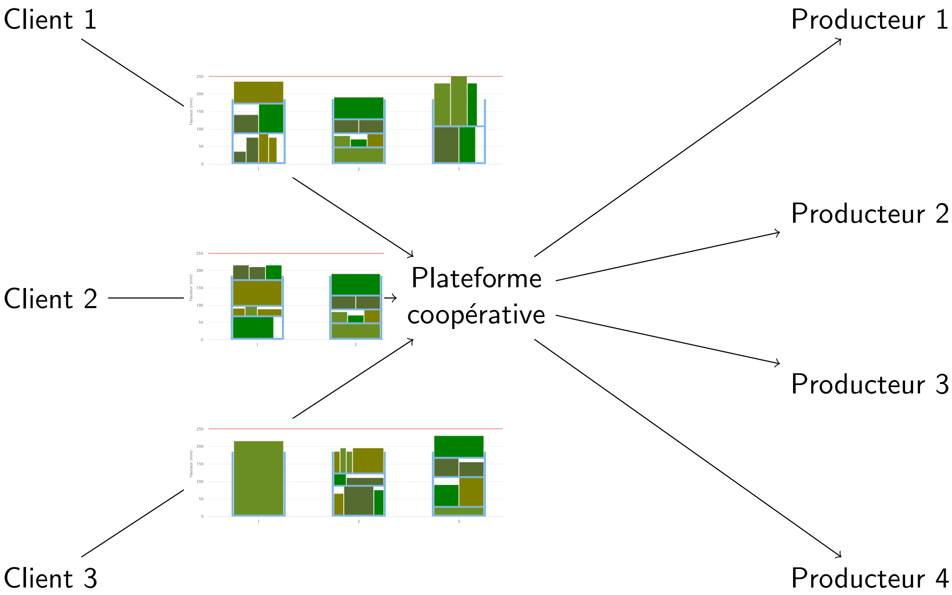


Plateforme coopérative

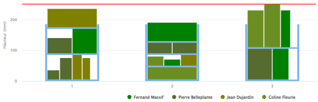


Client 3

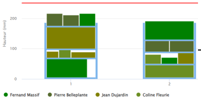

Client 1
Client 2
Client 3
Plateforme coopérative


Client 1
Client 2
Client 3
Plateforme coopérative
Producteur 1
Producteur 2
Producteur 3
Producteur 4

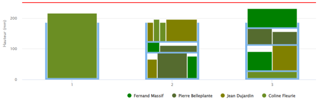
Client 1



Client 2



Client 3



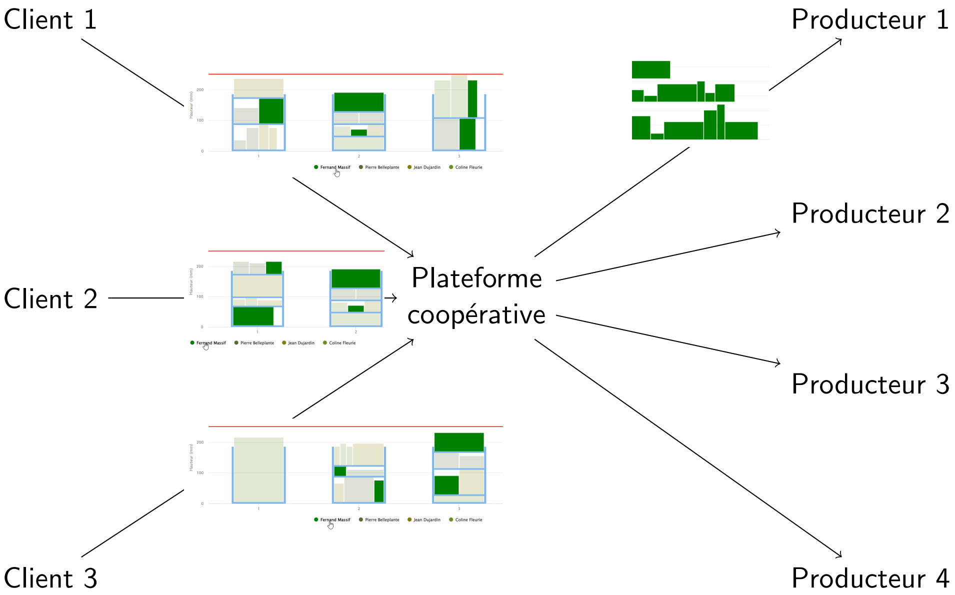
Plateforme coopérative

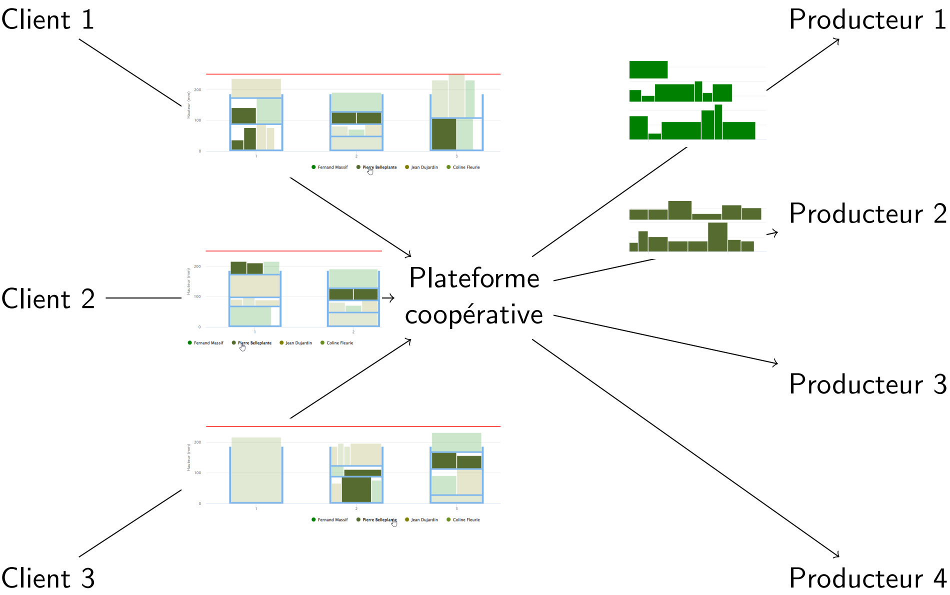
Producteur 1

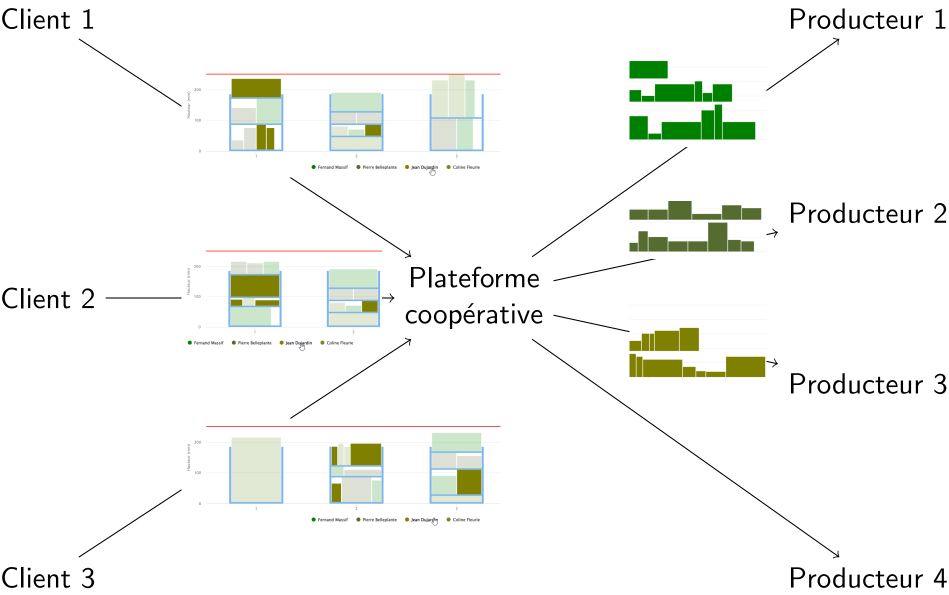
Producteur 2

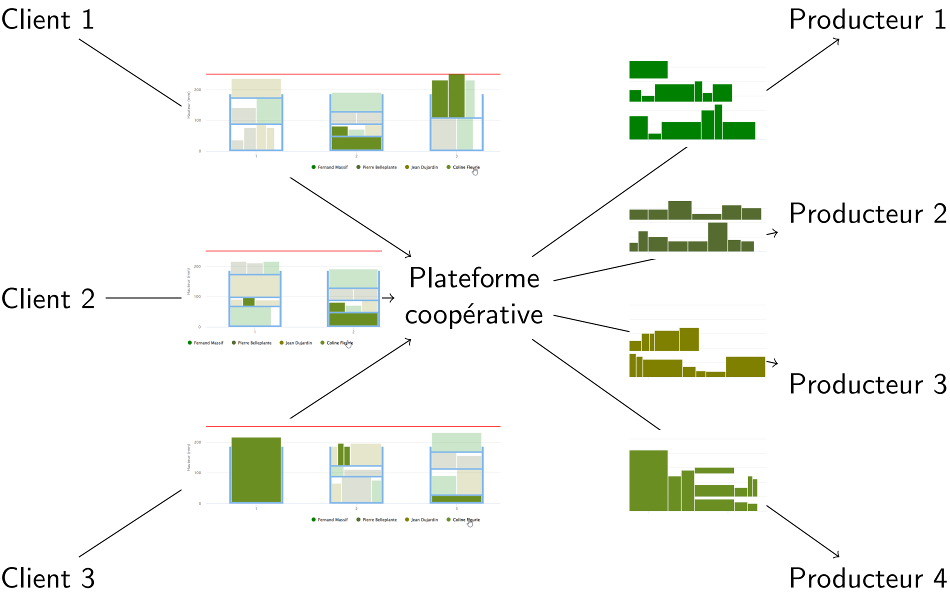
Producteur 3

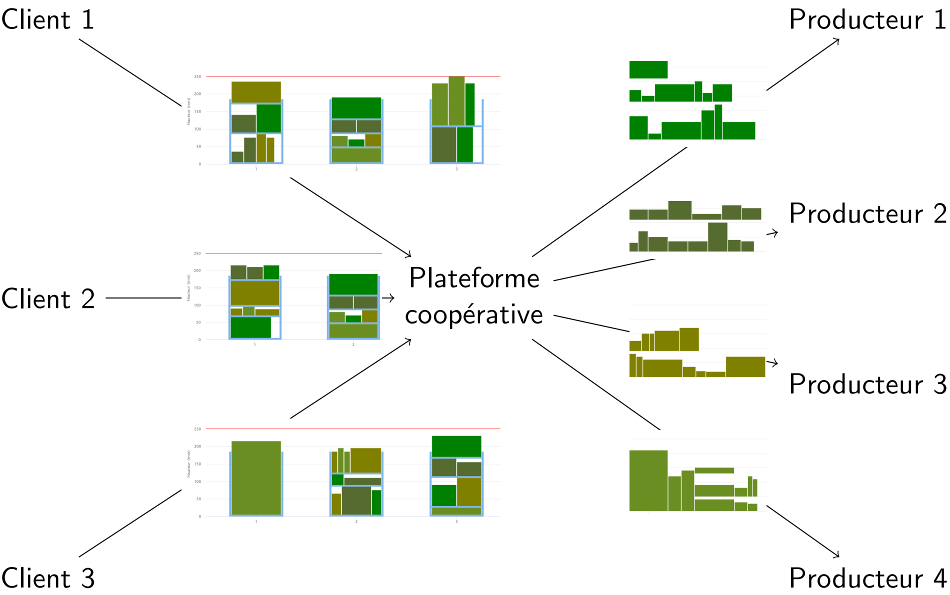
Producteur 4


Client 1
Client 2
Client 3
Hauteur (mm)
200
100
0
1
2
3
Fernand Massif
Pierre Belleplante
Jean Dujardin
Coline Fleurie
Plateforme coopérative
Producteur 1
Producteur 2
Producteur 3
Producteur 4

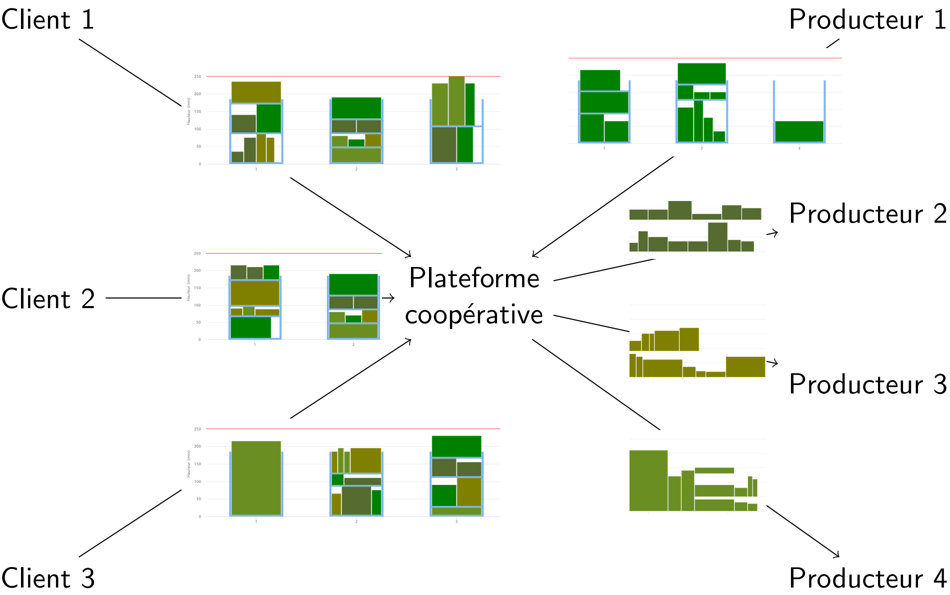

Client 1
Client 2
Client 3
Plateforme coopérative
Producteur 1
Producteur 2
Producteur 3
Producteur 4
Hauteur (mm)
Fernand Massif
Pierre Belleplante
Jean Dujardin
Coline Fleurie

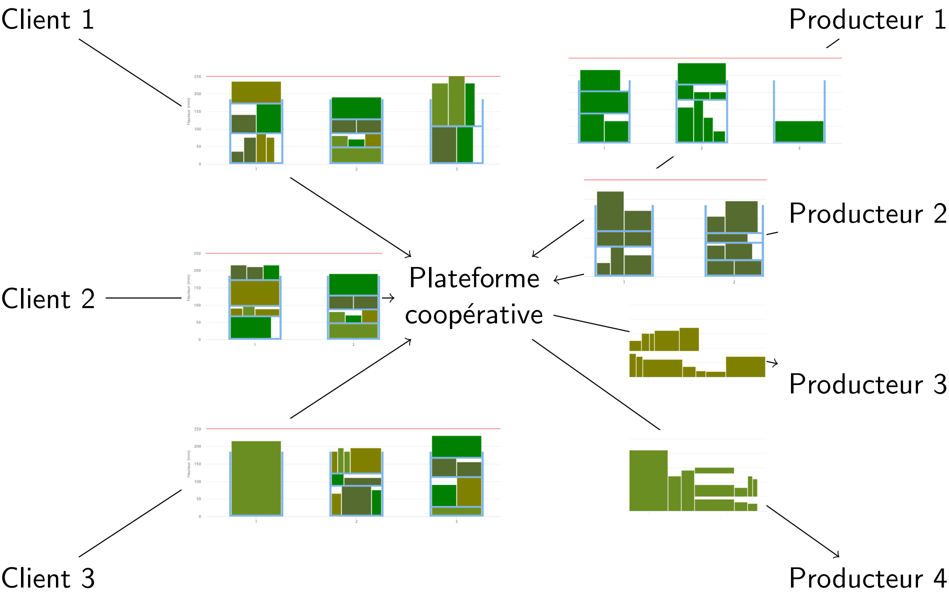

Client 1
Client 2
Client 3
Plateforme coopérative
Producteur 1
Producteur 2
Producteur 3
Producteur 4
Fernand Massif
Pierre Belleplante
Jean Dujardin
Coline Fleurie

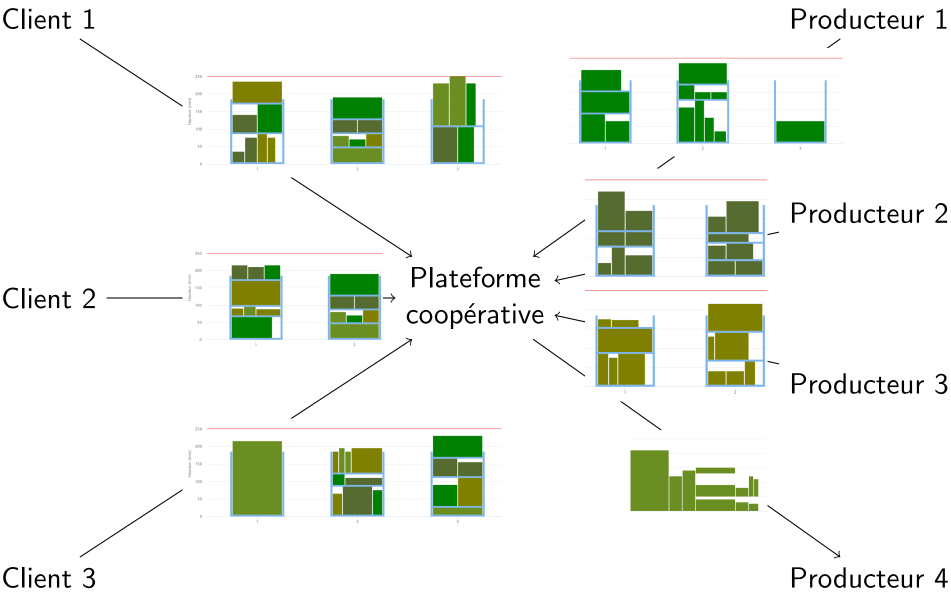

Client 1
Client 2
Client 3
Plateforme coopérative
Producteur 1
Producteur 2
Producteur 3
Producteur 4
Hauteur (mm)
Fernand Massif
Pierre Belleplante
Jean Dujardin
Coline Fleurie

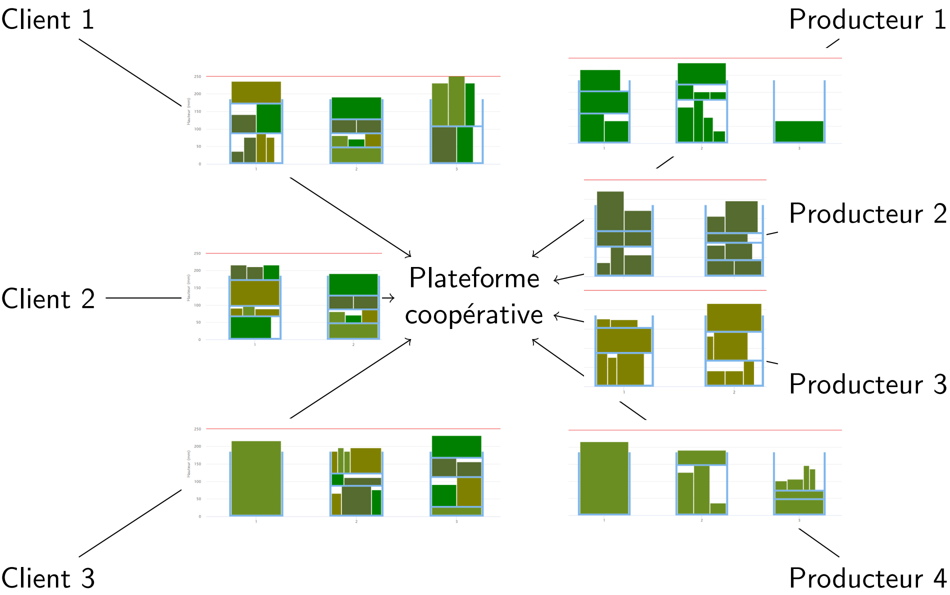


Client 1
Client 2
Client 3
Plateforme coopérative
Producteur 1
Producteur 2
Producteur 3
Producteur 4


Client 1
Producteur 1
Producteur 2
Client 2
Plateforme
coopérative
Producteur 3
Client 3
Producteur 4


Client 1
Producteur 1
Producteur 2
Plateforme coopérative
Client 2
Producteur 3
Client 3
Producteur 4


Client 1
Producteur 1
Producteur 2
Plateforme
coopérative
Client 2
Producteur 3
Client 3
Producteur 4

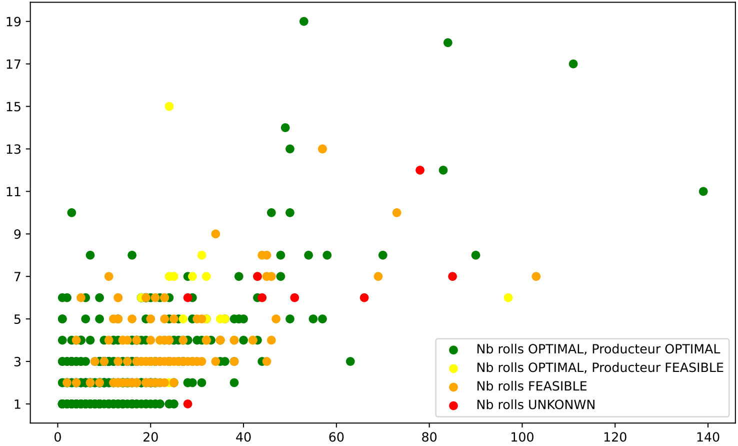

Client 1
Producteur 1
Producteur 2
Client 2
Plateforme coopérative
Producteur 3
Client 3
Producteur 4

Objectifs

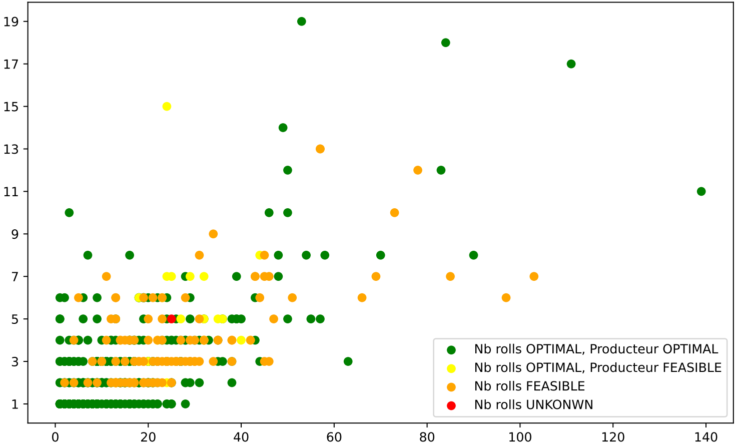
1. Utiliser le moins de rolls possibles pour placer toutes les plantes
2. Regrouper au maximum les plantes d'un même producteur sur un même roll
3. Regrouper au maximum chaque article sur les même étages

$$
\begin{aligned}
& Min \sum_{r \in R} u_r \\
& s.t.\ exactlyOne(\{x_{i,r,e} \mid \forall r \in R, \forall e \in E_r\}) && \forall i \in I_{complet} && (1) \\
& \sum_{r \in R, e \in E_r} t_{i,r,e} = q_i && \forall i \in I_{incomplet} && (2) \\
& \sum_{i \in I_{incomplet}} \frac{t_{i,r,e}}{QteEtage_i} + \sum_{i \in I_{complet}} x_{i,r,e} \leq 1 && \forall r \in R, \forall e \in E_r && (3) \\
& not(x_{i,r,e}) \implies t_{i,r,e} = 0 && \forall i \in I_{incomplet},\ \forall r \in R,\ \forall e \in E_r && (4) \\
& not(x_{i,r,e}) \implies t_{i,r,e} = 0 && \forall r \in R,\ \forall e \in E_r z_{r,e}\ \forall i \in I_{incomplet} && (5) \\
& x_{i,r,e} \implies u_r && \forall i \in I,\ \forall r \in R,\ \forall e \in E_r && (6) \\
& u_{r+1} \implies u_r && \forall r \in R && (7) \\
& x_{i,r,e} = 1 \implies x_{j,r,e^-} = 0 && \forall k \in \{1, ..., k_{hautBas}\}\ \forall r \in R && \\
& && \forall e \in E_r\ \forall e^+ \in E_r\ t.q.\ e^- \leq e && \\
& && \forall i \in I_{hautBas_k,B}\ \forall j \in I_{hautBas_k,J} && (8) \\
& exactlyOne(\{y_{r,c} \mid \forall c \in C\}) && x_{i,r,e} = 1 \implies x_{j,r,e^-} = 0 && (9) \\
& y_{r,c} \implies not(x_{i,r,e}) && \forall r \in R,\ \forall c \in C, && \\
& && \forall e \in E_r \text{ t.q. } e > e_c,\ \forall i \in I && (10) \\
& y_{r,c} \implies not(x_{i,r,e}) && \forall i \in I,\ \forall r \in R, && \\
& && \forall e \in E_r \text{ t.q. } h_i + ep > h_{c,e} + tol && (11)
\end{aligned}
$$

Résultats CP-SAT sans solution initiale



Résultats CP-SAT avec solution initiale



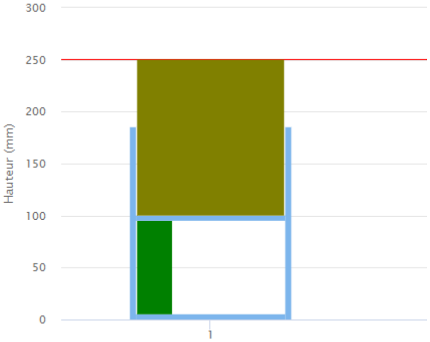
Résultats CP-SAT sans solution initiale

	Nb instances	Nb Rolls OPTIMAL Nb rolls par prod. OPTIMAL	Nb Rolls OPTIMAL Nb rolls par prod. FEASIBLE	Nb Rolls FEASIBLE	Nb Rolls UNKNOWN
CP-SAT Sans sol. initiale	593	455	22	107	8
CP-SAT Avec sol. initiale	593	454	22	115	1

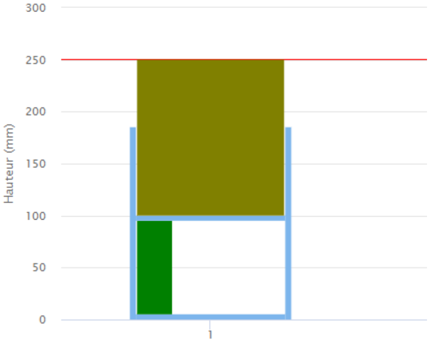
Conclusion calcul de roll

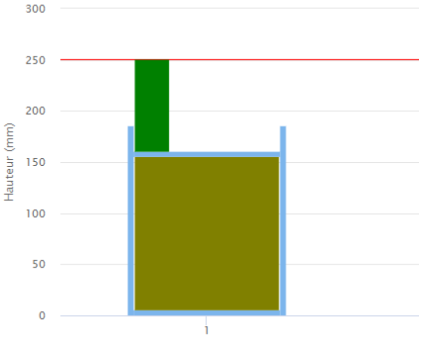
- Utilisation en production
  - Heuristique
  - Programmation par contrainte : solveur CP-SAT
  - Descente locale
  - Post-Process

Post process



Post process





Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique
3. Mise en place d'un programme mathématique minimale (moins de contraintes possibles)

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique
3. Mise en place d'un programme mathématique minimale (moins de contraintes possibles)
   - Si résultats satisfaisant : ajout de contraintes petit à petit
   - Sinon arrêt

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique
3. Mise en place d'un programme mathématique minimale (moins de contraintes possibles)
    - Si résultats satisfaisant : ajout de contraintes petit à petit
    - Sinon arrêt
4. Mise en place de voisinage et descente locale

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique
3. Mise en place d'un programme mathématique minimale (moins de contraintes possibles)
    - Si résultats satisfaisant : ajout de contraintes petit à petit
    - Sinon arrêt
4. Mise en place de voisinage et descente locale
5. Si modèle mathématique non performant, mise en place de méta-heuristique (tabou, recuit)

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique
3. Mise en place d'un programme mathématique minimale (moins de contraintes possibles)
    - Si résultats satisfaisant : ajout de contraintes petit à petit
    - Sinon arrêt
4. Mise en place de voisinage et descente locale
5. Si modèle mathématique non performant, mise en place de méta-heuristique (tabou, recuit)
6. Post-Process

Notre approche pour les problèmes d'optimisation combinatoire

1. Récupération d'un jeu de données
2. Mise en place d'une heuristique
3. Mise en place d'un programme mathématique minimale (moins de contraintes possibles)
    - Si résultats satisfaisant : ajout de contraintes petit à petit
    - Sinon arrêt
4. Mise en place de voisinage et descente locale
5. Si modèle mathématique non performant, mise en place de méta-heuristique (tabou, recuit)
6. Post-Process
7. **Itération avec le client**

Notre approche pour les problèmes d'optimisation combinatoire

Méthode	Avantages
Heuristique	• Rapide à calculer • Assure d'avoir toujours une solution • Réutilisable pour d'autres méthodes
Programmation mathématique	• Assez rapide à mettre en place • Calcule de bonnes solutions • Fournit des preuves d'optimalité
Recherche locale	• Proche du raisonnement humain • Calcule de bonnes solution • Réutilisable pour le post-process
Post process	• Être au plus proche du besoin "métier"

Annexes

Modèles MIP Assignation $P||C_{max}$

$$
\begin{aligned}
&\text{Minimize } y \\
&\text{subject to } \sum_{i \in \mathcal{M}} X_{ij} \geq 1 && \forall j \in \mathcal{J} && (1)\\
&y \geq \sum_{j \in \mathcal{J}} p_j * X_{ij} && \forall i \in \mathcal{M} && (2)\\
&X_{ij} \in \{0,1\} && \forall j \in \mathcal{M}\ \forall i \in \mathcal{J} && (3)\\
&y \in \mathbb{N} && && (4)
\end{aligned}
$$

Modèles MIP problème de base $P|r_j|\sum T_j$ (Flot)

$$\text{Minimize} \sum_{j \in \mathcal{J}} T_j$$

$$\text{subject to} \sum_{j \in \mathcal{J}} F_{0j} = \sum_{j \in \mathcal{J}} F_{j,n+1} = m \qquad (1)$$

$$\sum_{k \in \mathcal{J}^0 \setminus \{j\}} F_{kj} = \sum_{k \in \mathcal{J}^{n+1} \setminus \{j\}} F_{jk} = 1 \qquad \forall j \in \mathcal{J} \qquad (2)$$

$$r_k F_{kj} \leq B_{kj} \leq h F_{kj} \qquad \forall k, j \in \mathcal{J},\ k \neq j \qquad (3)$$

$$\sum_{k \in \mathcal{J}^{n+1} \setminus \{j\}} B_{jk} - \sum_{l \in \mathcal{J}^{n+1} \setminus \{j\}} (B_{lj} + p_l F_{lj}) \geq 0 \qquad \forall j \in \mathcal{J} \qquad (4)$$

$$T_j \geq \Big(\sum_{k \in \mathcal{J}^{n+1} \setminus \{j\}} B_{jk} \Big) + p_j - d_j \qquad \forall k \in \mathcal{J} \qquad (5)$$

$$F_{jk}, F_{0j}, F_{j,n+1} \in \{0,1\},\ B_{jk} \geq 0 \qquad \forall j, k \in \mathcal{J}, j \neq k \qquad (6)$$

Modèles MIP problème de base $P|r_j|\sum T_j$ (ordo)

$$\text{Minimize} \quad \sum_{j \in \mathcal{J}} T_j$$

$$
\begin{aligned}
\text{subject to} \quad & \sum_{i \in \mathcal{M}} \sum_{l=1}^{n} X_{jil} = 1 && \forall j \in \mathcal{J} && (1)\\
& \sum_{j \in \mathcal{J}} X_{jil} \leq 1 && \forall i \in \mathcal{M} \ \forall l \in [1, n] && (2)\\
& B_{i,l+1} - B_{il} \geq \sum_{j \in \mathcal{J}} p_j X_{jil} && \forall i \in \mathcal{M} \ \forall l \in [1, n-1] && (3)\\
& B_{il} \geq \sum_{j \in \mathcal{J}} r_j X_{jil} && \forall i \in \mathcal{M} \ \forall l \in [1, n] && (4)\\
& T_j \geq B_{il} + (X_{jil} - 1)M + p_j - d_j && \forall j \in \mathcal{J} \ \forall i \in \mathcal{M} \ \forall l \in [1, n] && (5)\\
& X_{jil} \in \{0, 1\}, \ \ B_{il}, T_j \geq 0 && \forall j \in \mathcal{J} \ \forall i \in \mathcal{M} \ \forall l \in [1, n] && (6)
\end{aligned}
$$

Modèles MIP avec vitesse $Q|r_j|\sum T_j$

$$\text{Minimize} \quad \sum_{j \in \mathcal{J}} T_j$$

$$\begin{aligned}
\text{subject to} \quad & \sum_{i \in \mathcal{M}} \sum_{l=1}^{n} X_{jil} = 1 & \forall j \in \mathcal{J} \quad & (1)\\
& \sum_{j \in \mathcal{J}} X_{jil} \leq 1 & \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad & (2)\\
& B_{i,l+1} - B_{il} \geq \sum_{j \in \mathcal{J}} \frac{p_j}{s_i} X_{jil} & \forall i \in \mathcal{M}\ \forall l \in [1, n-1] \quad & (3)\\
& B_{il} \geq \sum_{j \in \mathcal{J}} r_j X_{jil} & \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad & (4)\\
& T_j \geq B_{il} + (X_{jil} - 1)M + p_j - d_j & \forall j \in \mathcal{J}\ \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad & (5)\\
& X_{jil} \in \{0, 1\},\ \ B_{il}, T_j \geq 0 & \forall j \in \mathcal{J}\ \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad & (6)
\end{aligned}$$

Modèles MIP avec setup-times $Q|r_j, s_{jk}|\sum T_j$

Minimize $\sum_{j \in \mathcal{J}} T_j$

subject to

$$\sum_{i \in \mathcal{M}} \sum_{l=1}^{n} X_{jil} = 1 \qquad \forall j \in \mathcal{J} \quad (1)$$

$$\sum_{j \in \mathcal{J}} X_{jil} \leq 1 \qquad \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad (2)$$

$$B_{i,l+1} - B_{il} \geq \sum_{j \in \mathcal{J}} \left(\frac{p_j}{s_i} X_{jil} + \sum_{k \in \mathcal{J}} Y_{jkil} s_{jk} \right) \qquad \forall i \in \mathcal{M}\ \forall l \in [1, n-1] \quad (3)$$

$$B_{il} \geq \sum_{j \in \mathcal{J}} r_j X_{jil} \qquad \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad (4)$$

$$T_j \geq B_{il} + (X_{jil} - 1)M + p_j - d_j \qquad \forall j \in \mathcal{J}\ \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad (5)$$

$$Y_{jkil} = X_{jil} \cdot X_{ki,l+1} \qquad \forall j, k \in \mathcal{J}\ \forall i \in \mathcal{M}\ \forall l \in [1, n-1] \quad (6)$$

$$X_{jil}, Y_{jkil} \in \{0, 1\},\ B_{il}, T_j \geq 0 \qquad \forall j \in \mathcal{J}\ \forall i \in \mathcal{M}\ \forall l \in [1, n] \quad (7)$$

Modèles CP makespan $P||C_{max}$

$$
\begin{aligned}
\text{Minimize} \quad & \max_{j \in \mathcal{J}} \; endOf(a_j) \\
\text{subject to} \quad & a_j^i = interval\left(p_j\right) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (1) \\
& optional(a_j^i) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (2) \\
& a_j = alternative(\{a_j^i : i \in \mathcal{M}\}) && \forall j \in \mathcal{J} && (3) \\
& S^i = intervalSequence(\{a_j^i : j \in \mathcal{J}\}) && \forall i \in \mathcal{M} && (4) \\
& noOverlap(S^i) && \forall i \in \mathcal{M} && (5) \\
& && && (6)
\end{aligned}
$$

Modèles CP retard $P|r_j|\sum T_j$

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j \in \mathcal{J}} \max(0, \mathit{endOf}(a_j) - d_j) \\
\text{subject to} \quad & a_j^i = \mathit{interval}\,(p_j) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (1) \\
& \mathit{optional}(a_j^i) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (2) \\
& a_j = \mathit{alternative}(\{a_j^i : i \in \mathcal{M}\}) && \forall j \in \mathcal{J} && (3) \\
& S^i = \mathit{intervalSequence}(\{a_j^i : j \in \mathcal{J}\}) && \forall i \in \mathcal{M} && (4) \\
& \mathit{noOverlap}(S^i) && \forall i \in \mathcal{M} && (5) \\
& \mathit{startMin}(a_j^i, r_j) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (6)
\end{aligned}
$$

Modèles CP vitesse $Q|r_j|\sum T_j$

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j \in \mathcal{J}} \max(0, endOf(a_j) - d_j) \\
\text{subject to} \quad & a_j^i = interval\left(\frac{p_j}{s_i}\right) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (1) \\
& optional(a_j^i) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (2) \\
& a_j = alternative(\{a_j^i : i \in \mathcal{M}\}) && \forall j \in \mathcal{J} && (3) \\
& S^i = intervalSequence(\{a_j^i : j \in \mathcal{J}\}) && \forall i \in \mathcal{M} && (4) \\
& noOverlap(S^i) && \forall i \in \mathcal{M} && (5) \\
& startMin(a_j^i, r_j) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (6)
\end{aligned}
$$

Modèles CP setup $Q|r_j, s_{jk}|\sum T_j$

$$\text{Minimize} \quad \sum_{j \in \mathcal{J}} \max(0, endOf(a_j) - d_j)$$

$$\begin{aligned}
\text{subject to } & a_j^i = interval\left(\frac{p_j}{s_i}\right) & \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \quad (1)\\
& optional(a_j^i) & \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \quad (2)\\
& a_j = alternative(\{a_j^i : i \in \mathcal{M}\}) & \forall j \in \mathcal{J} \quad (3)\\
& S^i = intervalSequence(\{a_j^i : j \in \mathcal{J}\}, jobTypes) & \forall i \in \mathcal{M} \quad (4)\\
& noOverlap(S^i, jobTypes, setupTimes) & \forall i \in \mathcal{M} \quad (5)\\
& startMin(a_j^i, r_j) & \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \quad (6)
\end{aligned}$$

Modèles CP break $Q|r_j, brkdwn, s_{jk}|\sum T_j$

$$\text{Minimize} \quad \sum_{j\in\mathcal{J}} \max(0, endOf(a_j) - d_j)$$

$$\begin{aligned}
\text{subject to } & a_j^i = interval\left(\frac{p_j}{s_i}\right) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (1)\\
& optional(a_j^i) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (2)\\
& a_j = alternative(\{a_j^i : i \in \mathcal{M}\}) && \forall j \in \mathcal{J} && (3)\\
& S^i = intervalSequence(\{a_j^i : j \in \mathcal{J}\}, jobTypes) && \forall i \in \mathcal{M} && (4)\\
& noOverlap(S^i, jobTypes, setupTimes) && \forall i \in \mathcal{M} && (5)\\
& startMin(a_j^i, r_j) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (6)\\
& intensity(a_j^i, openPeriod_i) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (7)
\end{aligned}$$

Modèles Complet CP break and setup $Q|r_j, brkdwn, s_{jk}|\sum T_j$

$$\text{Min} \sum_{j \in \mathcal{J}} \max(0, endOf(a_j) - d_j)$$

$$\text{s.t. } a_j^i = interval\left(\frac{p_j}{s_i}\right) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{1}$$

$$intensity(a_j^i, openPeriod_i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{2}$$

$$optional(a_j^i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{3}$$

$$a_j = alternative(\{a_j^i : i \in \mathcal{M}\}) \quad \forall j \in \mathcal{J} \tag{4}$$

$$setup_j^i = interval() \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{5}$$

$$intensity(setup_j^i, openPeriod_i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{6}$$

$$startMin(setup_j^i, r_j) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{7}$$

$$optional(setup_j^i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{8}$$

$$presenceOf(setup_j^i) = presenceOf(a_j^i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{9}$$

$$startAtEnd(a_j^i, setup_j^i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{10}$$

$$cover_j^i = interval() \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{11}$$

$$optional(cover_j^i) \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{M} \tag{12}$$

Modèles Complet CP break and setup $Q|r_j, brkdwn, s_{jk}|\sum T_j$

$$
\begin{aligned}
\text{s.t. } & presenceOf(cover_j^i) = presenceOf(a_j^i) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (13)\\
& span(cover_j^i, \{a_j^i, setup_j^i\}) && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (14)\\
& S^i = intervalSequence(\{cover_j^i : j \in \mathcal{J}\}, jobTypes) && \forall i \in \mathcal{M} && (15)\\
& noOverlap(S^i) && \forall i \in \mathcal{M} && (16)\\
& lengthOf(setup_j^i) = \\
& \quad setupTimes[typeOfPrevious(S^i, cover_j^i, type[j])][type[j]] && \forall j \in \mathcal{J}, \forall i \in \mathcal{M} && (17)
\end{aligned}
$$

Contrainte Group Cumulative

$$span(F_g, \{a_j : j \in g\}) \qquad \forall g \in \mathcal{P} \tag{1}$$

$$Active = \sum_{g \in \mathcal{P}} pulse(F_g, 1) \tag{2}$$

$$lowerOrEqual(Active, L) \tag{3}$$