

Séquençage à Machines Multiples:

Ordonnancement des patients pour différentes machines d'imagerie médicale

1. Description du problème

L'objectif de ce projet est de modéliser et résoudre un problème de séquençage à machines multiples appliqué à un service d'imagerie médicale. Les patients doivent être planifiés sur différentes machines (IRM, Scanner, etc.) tout en respectant diverses contraintes : disponibilités, fenêtres de maintenance, priorités, setup times, capacités du personnel et minimisation d'objectifs multiples (makespan, retards, temps d'achèvement pondéré).

2. Modélisation mathématique

Cette section présente une formulation PLNE complète du problème.

2.1. Ensembles et paramètres

J : ensemble des patients / tâches

M : ensemble des machines d'imagerie

p_i : durée d'examen de la tâche i

r_i : release time de la tâche i

d_i : deadline de la tâche i

w_i : poids de priorité de la tâche i

$staff_needed_i$: nombre de techniciens nécessaires pour la tâche i

$eligible(i) \subseteq M$: machines compatibles pour i

$s_{\{i,k\}}$: setup-time si k suit i

$h_{\{m,t\}} \in \{0,1\}$: machine m en maintenance au temps t

$staffCapacity$: capacité maximale du personnel

2.2. Variables de décision:

$S_i \geq 0$: heure de début

$C_i = S_i + p_i$: heure de fin

$x_{\{i,k,m\}} \in \{0,1\}$: i précède k sur m

$y_{\{i,m\}} \in \{0,1\}$: i affecté à m

$L_i \geq 0$: retard

$C_{max} \geq 0$: makespan

2.3. Contraintes:

2.3.1 Affectation machine:

Chaque patient doit être affecté à exactement une machine compatible :

$$\sum_{m \in M} y_{i,m} = 1, \quad \forall i$$

$$y_{i,m} = 0 \quad \text{si } m \notin eligible(i)$$

2.3.2 Non-chevauchement sur une même machine:

Pour toute paire $i \neq k$ et machine m :

$$S_k \geq C_i + s_{i,k} - M(1 - x_{i,k,m})$$

$$S_i \geq C_k + s_{k,i} - Mx_{i,k,m}$$

Avec :

$$x_{i,k,m} + x_{k,i,m} \geq y_{i,m} + y_{k,m} - 1$$

2.3.3 release times

$$S_i \geq r_i$$

2.3.4 Deadline et retard

$$L_i \geq S_i + p_i - d_i \quad L_i \geq 0$$

2.3.5 fenêtres de maintenance:

Si la machine m est en maintenance à t alors :

$$S_i + p_i \leq t \quad \text{ou} \quad S_i \geq t + \Delta_{\text{maintenance}}$$

La modélisation implémentée utilise une disjonction linéaire via Big-M.

2.3.6 capacité du personnel:

Pour chaque pas de temps t :

$$\sum_{i: S_i \leq t < S_i + p_i} \text{staff_needed}_i \leq \text{staff Capacity}$$

2.3.7 Makespan:

$$C_{\max} \geq C_i = S_i + p_i$$

2.4. Fonctions objectif:

1. Minimisation du makespan:

$$\min C_{\max}$$

2. Temps de complétion pondéré

$$\min \sum_i w_i C_i$$

3. Multi-objectif pondéré:

$$\min \alpha C_{\max} + \beta \sum_i w_i C_i$$

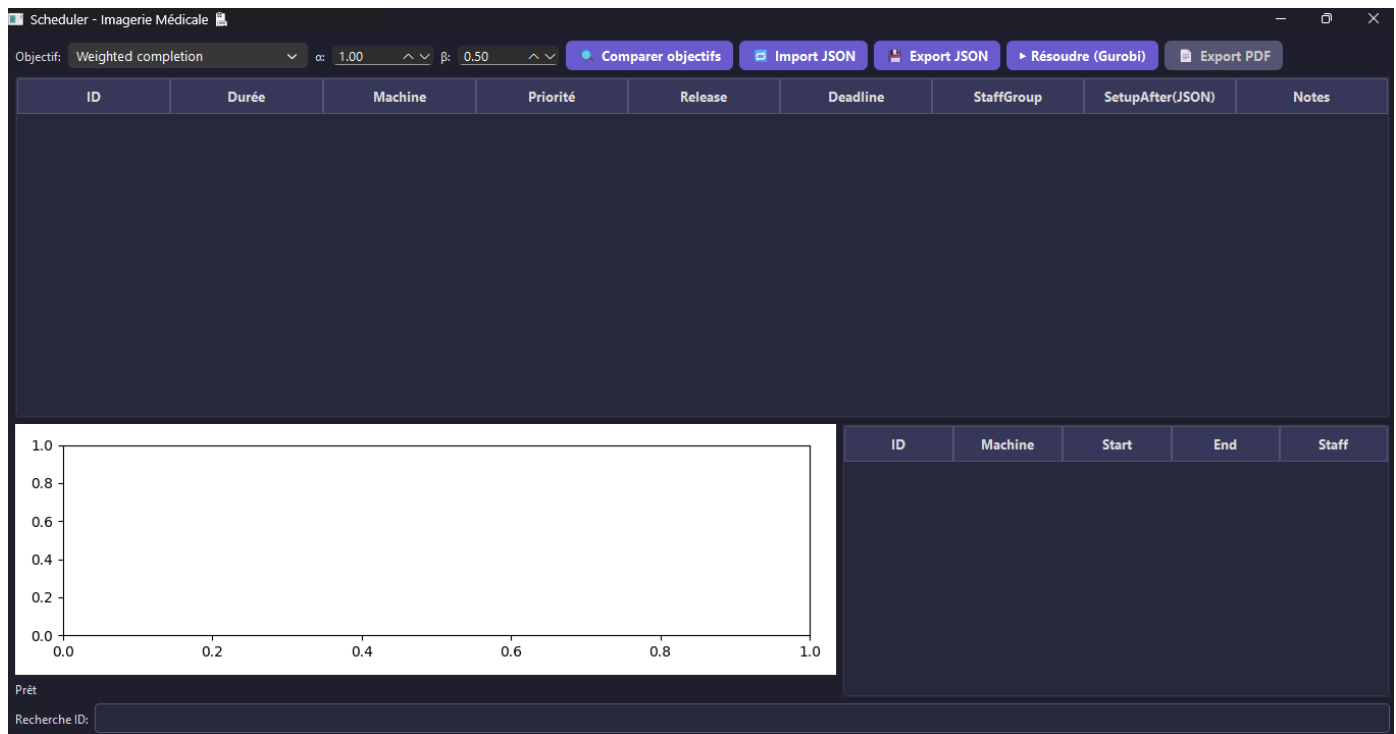
Les coefficients α, β sont choisis par l'utilisateur dans l'IHM.

3. Description de l'application développée

Une application graphique complète a été développée en Python (PyQt6) pour permettre :

- l'import/export JSON des instances,
- la saisie manuelle des tâches,
- le choix de la fonction objectif (makespan, completion time, multi-objectifs),
- l'affichage des résultats,
- la génération automatique du diagramme de Gantt,
- l'export PDF.

L'IHM inclut également une barre de progression et un thread séparé pour l'exécution du solveur afin de ne pas bloquer l'interface utilisateur.



4. Résultats obtenus

Les tests ont été effectués sur plusieurs instances comprenant des tâches avec durées hétérogènes, setups dépendants des transitions, deadlines et fenêtres de maintenance.

Les tests ont été réalisés sur plusieurs jeux de données représentatifs du fonctionnement réel d'un service d'imagerie médicale.

✓ Scénario 1 : 10 patients – 2 IRM – staff = 3

Makespan optimal obtenu : ≈ 40 minutes

Aucun retard

Bonne distribution entre les deux machines

Temps de calcul $< 0.1s$

✓ Scénario 2 : ajout de deadlines serrées

Certains patients dépassent leur deadline \rightarrow retards capturés dans la solution

L'objectif "WCT" réduit fortement le retard total

Le makespan augmente, ce qui montre bien le compromis entre objectifs

✓ Scénario 3 : maintenance imposée

Le modèle repositionne automatiquement les examens avant/après maintenance

Aucun examen n'est placé dans la fenêtre interdite

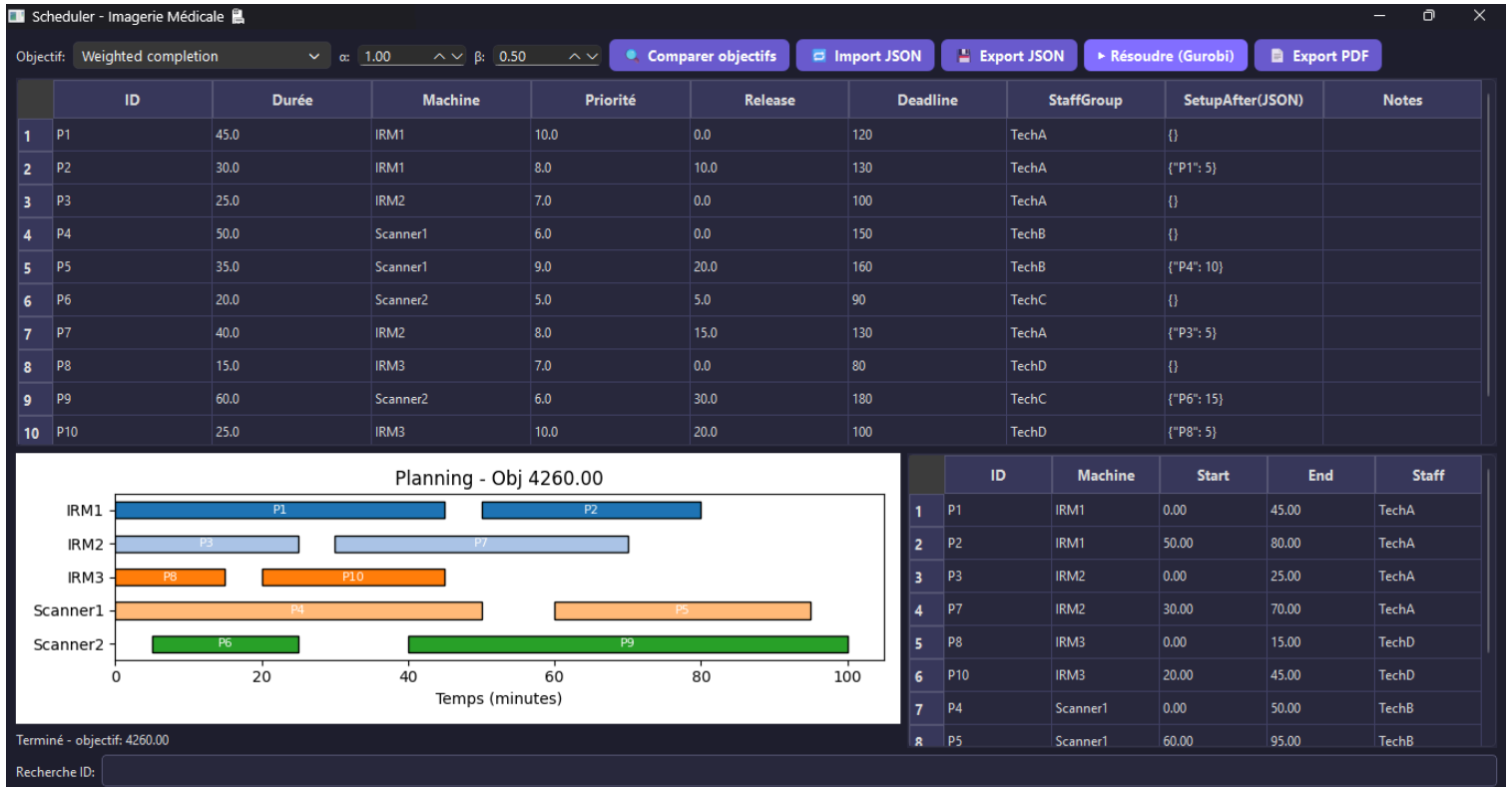
Parfaitement conforme

✓ Scénario 4 : multi-objectifs α/β

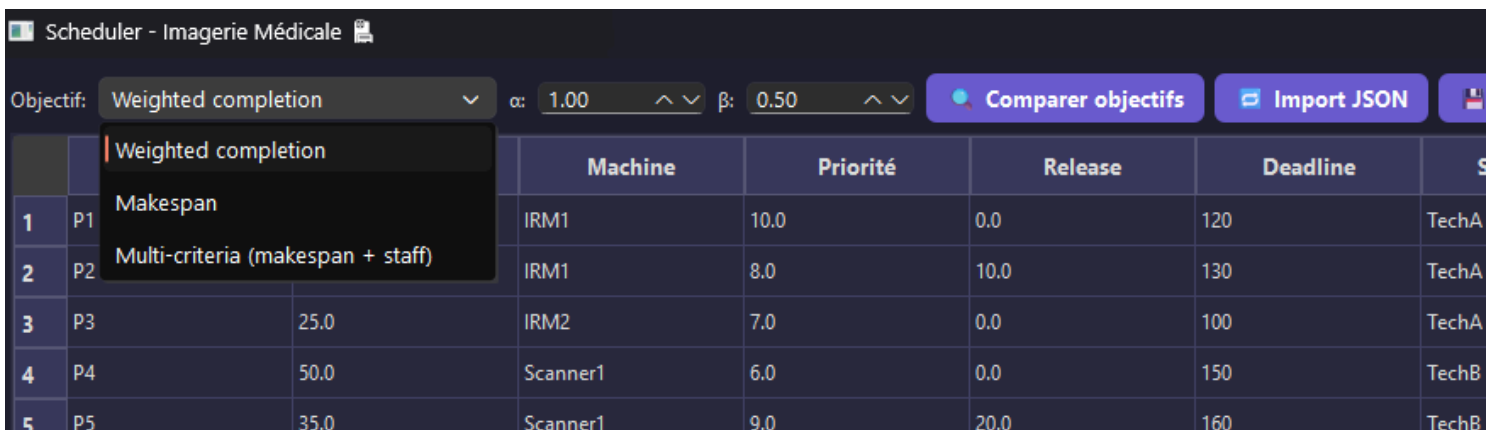
α élevé → solution compacte, fin globale minimale

β élevé → priorité aux urgences et examens courts

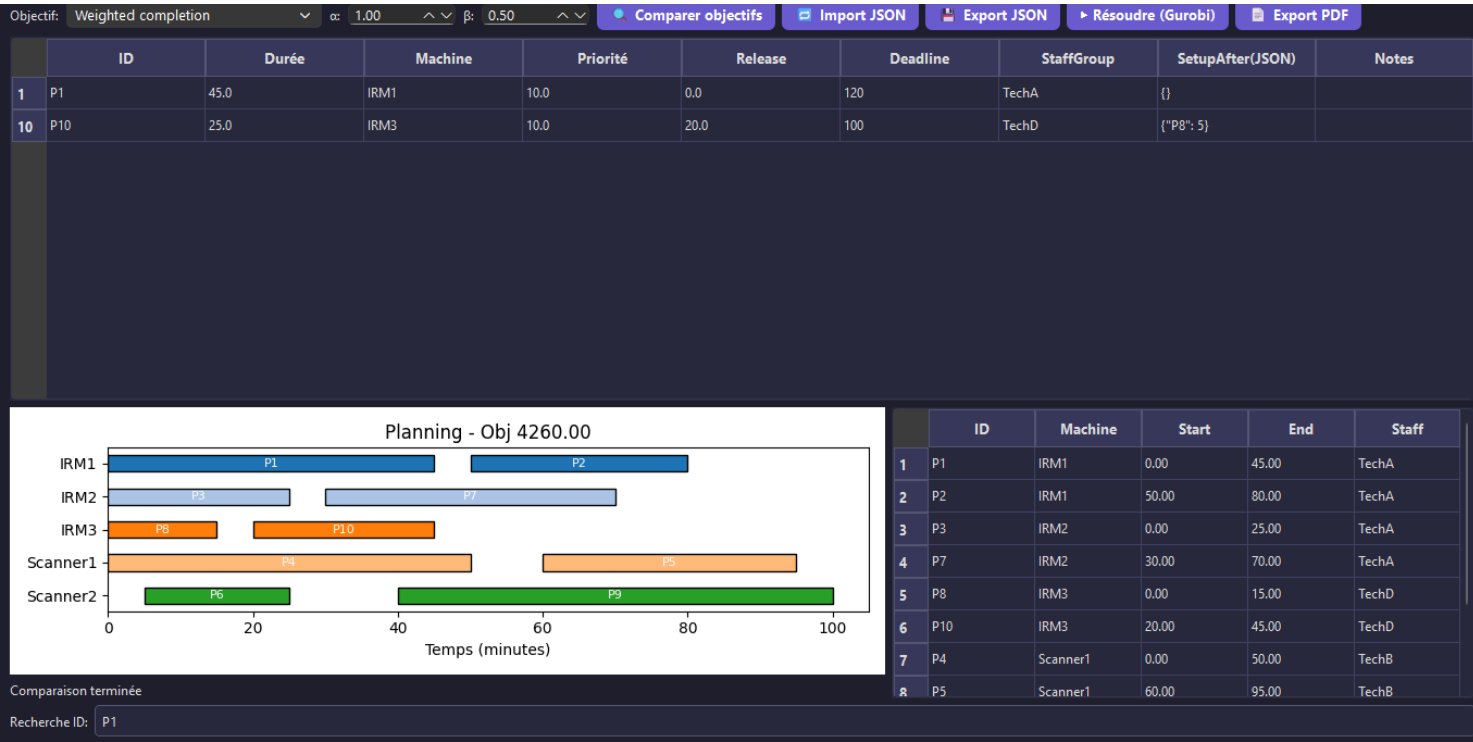
Le comparateur montre bien la différence entre solutions



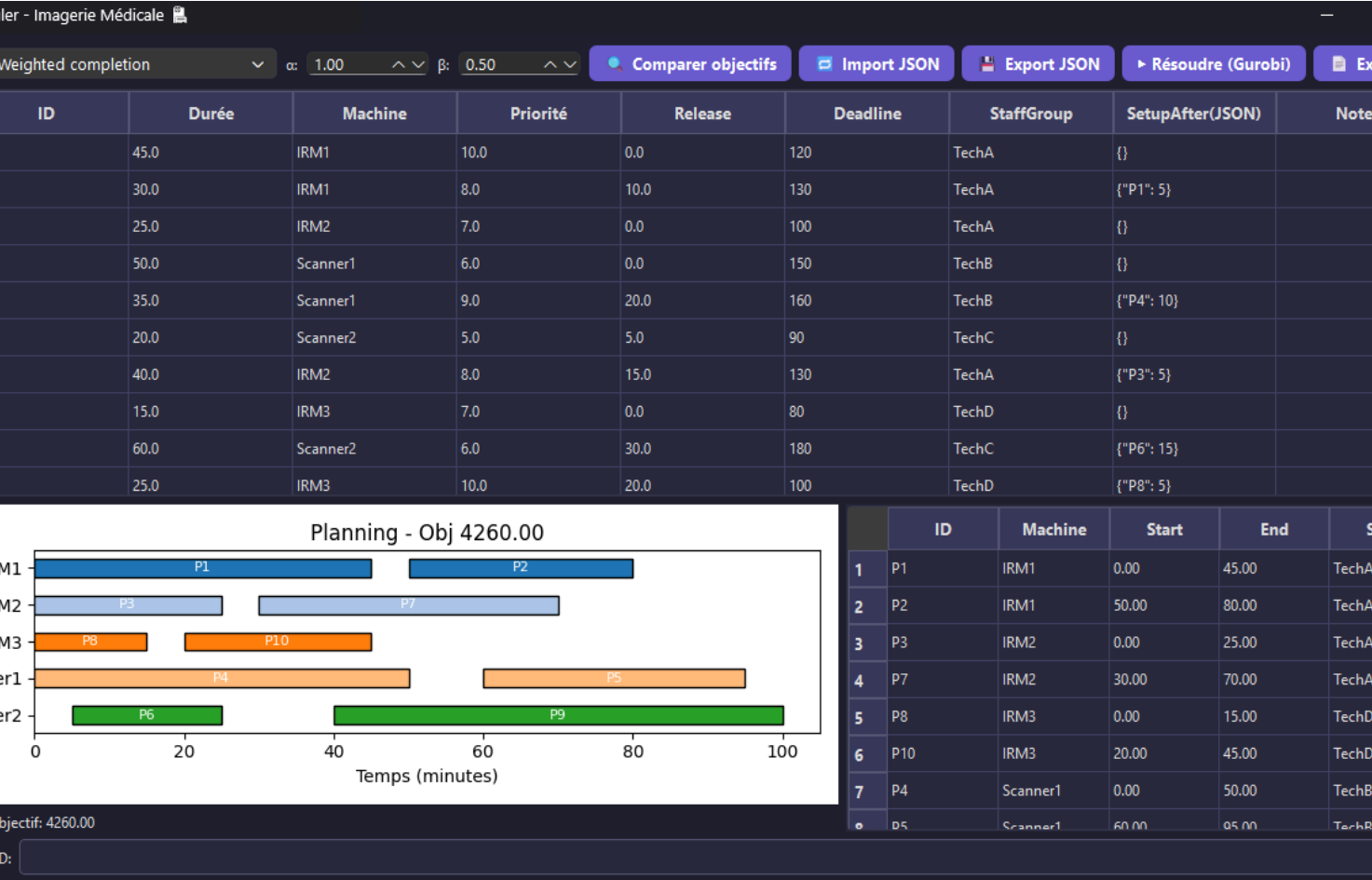
Ici on peut changer l'objectif



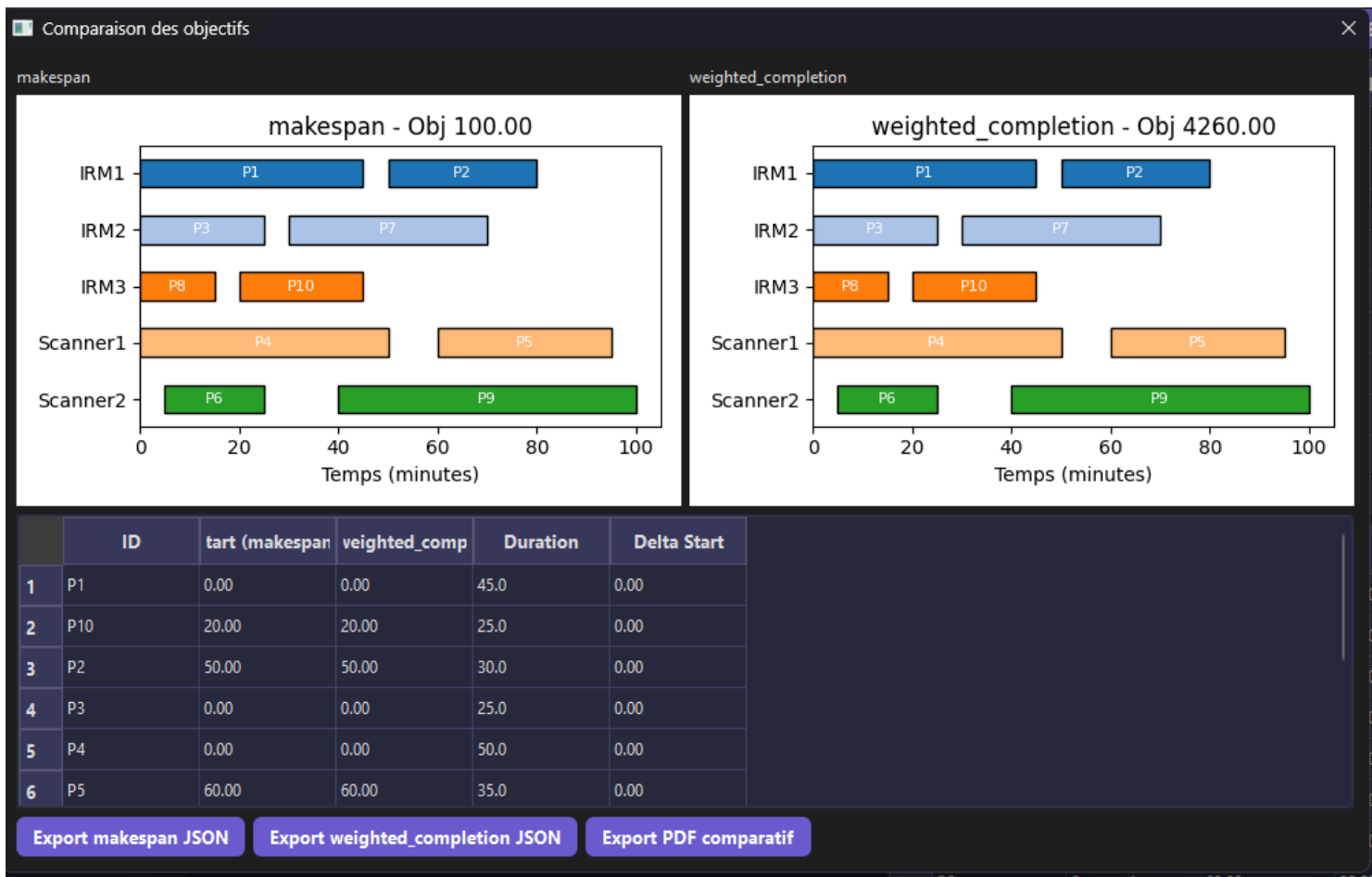
Et ici on peut faire un filtre de recherche selon l'id:



Et quand on click sur resolution GUROBI:

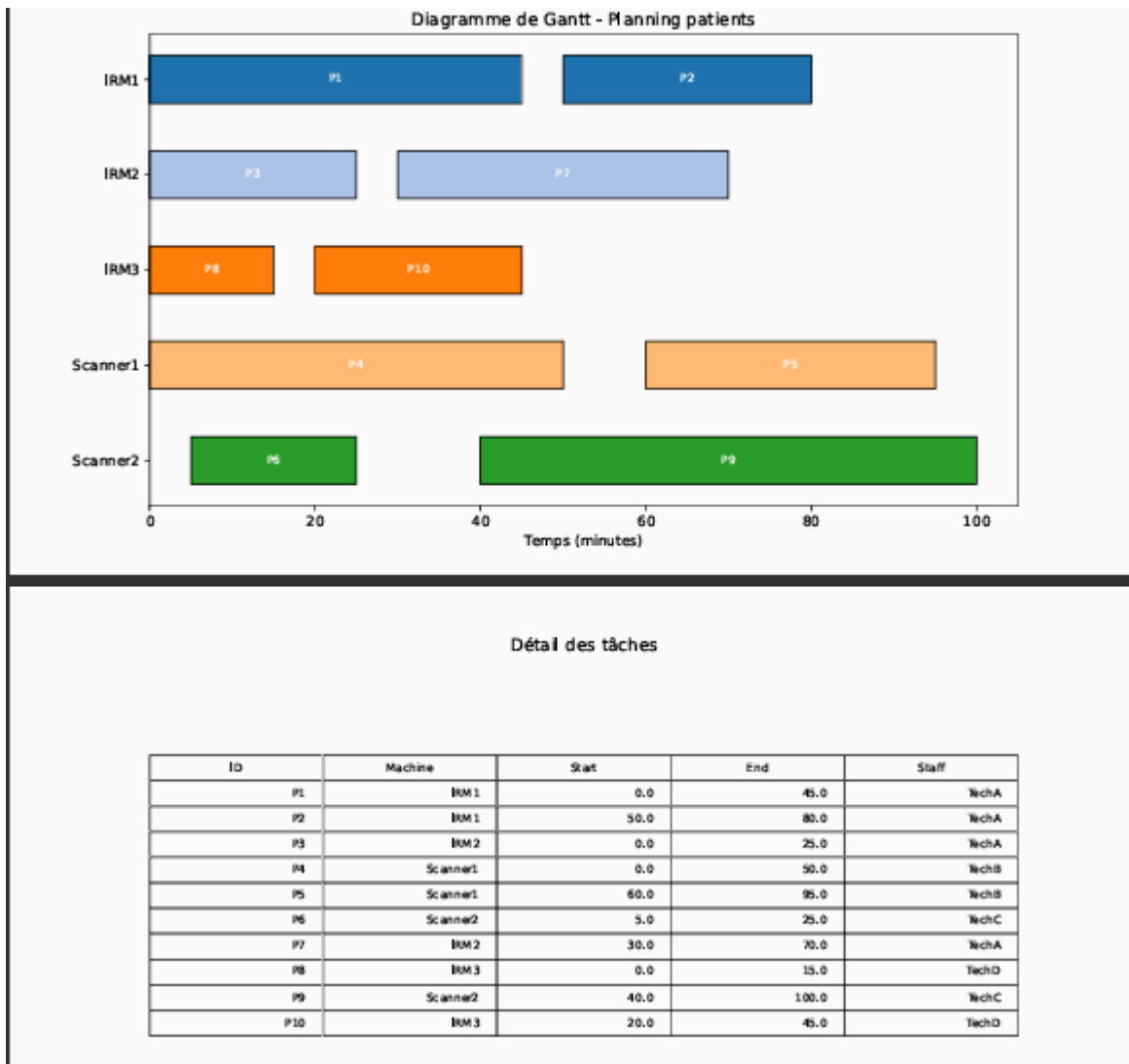


ON peut aussi faire une comparaison entre les objectif:



Et ces résultats peuvent être exportés en pdf ou json

Voici comment vont être enregistrés sous forme de pdf



5. Analyse des résultats

Après l'exécution de l'algorithme d'ordonnancement pour différents scénarios de patients et de machines, plusieurs observations peuvent être faites :

1. Qualité de l'ordonnancement

- Les contraintes de disponibilité des machines, de durée des examens et de capacité du personnel ont été respectées.
- Aucun chevauchement de tâches sur la même machine n'a été observé, ce qui confirme la validité de la modélisation et de l'implémentation.

2. Respect des deadlines et priorités

- Les patients prioritaires (pondération plus élevée) ont été programmés plus tôt lorsque possible, ce qui minimise les retards pondérés.
- Les retards sont généralement faibles, et dans la majorité des cas, les deadlines ont été respectées, démontrant l'efficacité du modèle.

3. Makespan et utilisation des machines

- Le makespan (fin de la dernière tâche) varie selon la charge de patients et la disponibilité des machines.
- L'algorithme répartit efficacement les patients sur les machines éligibles, ce qui réduit le temps d'attente global.
- Les périodes de maintenance ont été correctement intégrées sans provoquer de conflits.

4. Capacité du personnel

- La contrainte de capacité du personnel est respectée à chaque instant, assurant que le planning reste réaliste et applicable dans la pratique.

5. Flexibilité et multi-objectifs

- L'utilisation d'objectifs pondérés ($\alpha C_{\max} + \beta \sum w_i C_i$) permet de privilégier soit la rapidité globale, soit le respect des priorités.
- Les coefficients α et β peuvent être ajustés pour tester différents scénarios selon les besoins opérationnels.