

# À la recherche de stratégies d'ordonnancement garanties pour les observations radioastronomiques

Frédéric Vivien

INRIA – Équipe ROMA – Lyon

Atelier ECLAT 2024  
29 novembre 2024

# Le problème (schématiquement)



---

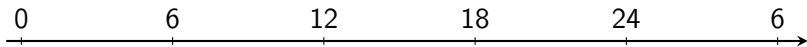
---

---

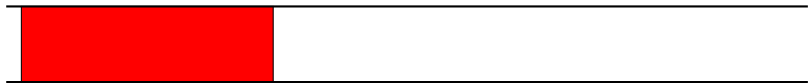
---

---

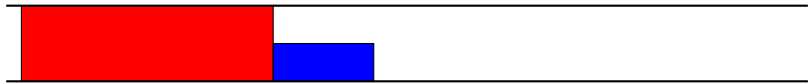
# Le problème (schématiquement)



# Le problème (schématiquement)



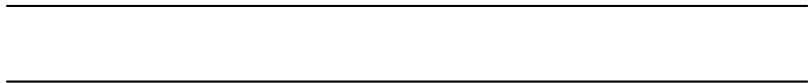
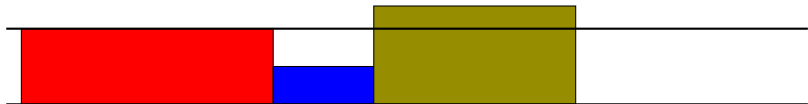
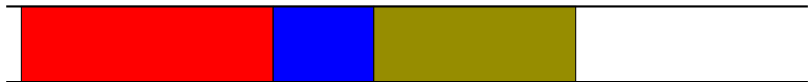
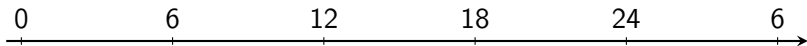
# Le problème (schématiquement)



# Le problème (schématiquement)



# Le problème (schématiquement)

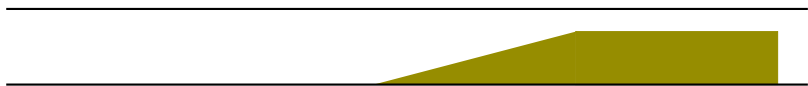
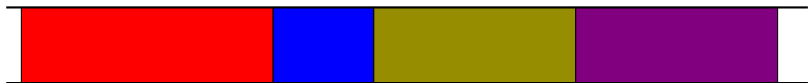


# Le problème (schématiquement)

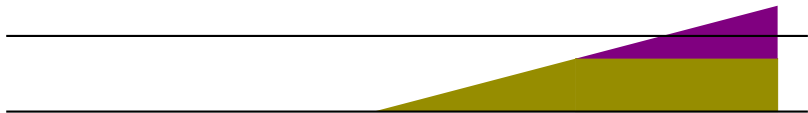
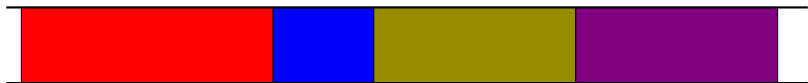




# Le problème (schématiquement)



# Le problème (schématiquement)



# Le problème (schématiquement)



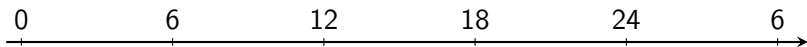
# Le problème (schématiquement)



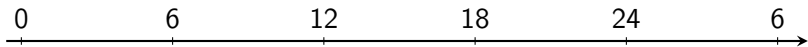
# Le problème (schématiquement)



# Le problème (schématiquement)



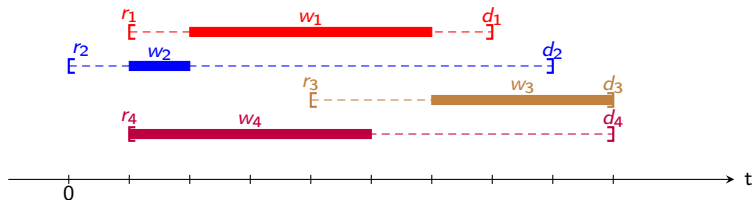
# Le problème (schématiquement)



~~SOLEIL~~



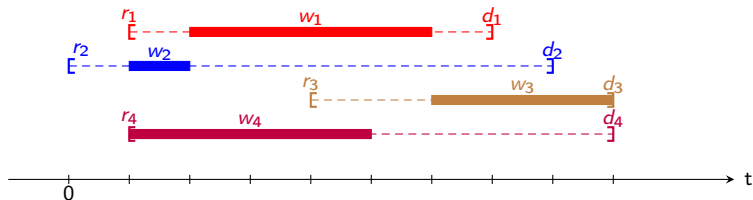
# Le problème (formellement)



- ▶ Un ensemble  $O_1, \dots, O_n$  de requêtes d'observation
- ▶ Chaque requête d'observation  $O_i$  est définie par :
  - ▶ Une durée d'observation  $w_i$
  - ▶ Une date de début d'observation au plus tôt  $r_i$
  - ▶ Une date de fin d'observation au plus tard  $d_i$
  - ▶ Une charge du calculateur  $l_i$
  - ▶ Un volume de données  $v_i$
- ▶ Objectif : minimiser la date de complétion de la dernière observation



# Le problème (formellement)



- ▶ Un ensemble  $O_1, \dots, O_n$  de requêtes d'observation
- ▶ Chaque requête d'observation  $O_i$  est définie par :
  - ▶ Une durée d'observation  $w_i$
  - ▶ Une date de début d'observation au plus tôt  $r_i$  ( $\neq$  release date)
  - ▶ Une date de fin d'observation au plus tard  $d_i$  ( $\neq$  deadline)
  - ▶ Une charge du calculateur  $l_i$
  - ▶ Un volume de données  $v_i$
- ▶ Objectif : minimiser la date de complétion de la dernière observation

Ne correspond à rien de connu en théorie de l'ordonnancement

# Est-il difficile de trouver une solution optimale ?

## Hypothèses simplificatrices

- ▶ Calculateur de puissance infinie
- ▶ Capacité de stockage infinie

## Complexité

- ▶ NP-complet au sens fort
- ▶ À la louche : (on croit que) trouver une solution optimale a un coût exponentiel en la taille du problème

# Est-il difficile de trouver une bonne solution ?

## Algorithmes d'approximation

- ▶ Algorithme de complexité polynomiale en la taille du problème
- ▶ Algorithme  $\mathcal{A}$  est une  $\rho$ -approximation :

$$\forall \text{Instance } I, \text{ Objectif}(\mathcal{A}(I)) \leq \rho \times \text{Objectif}(\text{Optimal}(I))$$

# Est-il difficile de trouver une bonne solution ?

## Algorithmes d'approximation

- ▶ Algorithme de complexité polynomiale en la taille du problème
- ▶ Algorithme  $\mathcal{A}$  est une  $\rho$ -approximation :

$$\forall \text{Instance } I, \text{ Objectif}(\mathcal{A}(I)) \leq \rho \times \text{Objectif}(\text{Optimal}(I))$$

## Exemple de l'ordonnancement sur machine parallèle

- ▶ Ordonnancement de liste  $\approx$  ne jamais laisser volontairement inactif un processeur
- ▶ Tout ordonnancement de liste est une 2-approximation

# Est-il difficile de trouver une bonne solution ?

## Algorithmes d'approximation

- ▶ Algorithme de complexité polynomiale en la taille du problème
- ▶ Algorithme  $\mathcal{A}$  est une  $\rho$ -approximation :

$$\forall \text{Instance } I, \text{ Objectif}(\mathcal{A}(I)) \leq \rho \times \text{Objectif}(\text{Optimal}(I))$$

## Exemple de l'ordonnancement sur machine parallèle

- ▶ Ordonnancement de liste  $\approx$  ne jamais laisser volontairement inactif un processeur
- ▶ Tout ordonnancement de liste est une 2-approximation

## Ordonnancement « raisonnable » d'observations

- ▶ Quand une nouvelle observation est ordonnancée, elle est exécutée au plus tôt
- ▶ Les ordonnancements « raisonnables » définissent-ils des algorithmes d'approximation ?

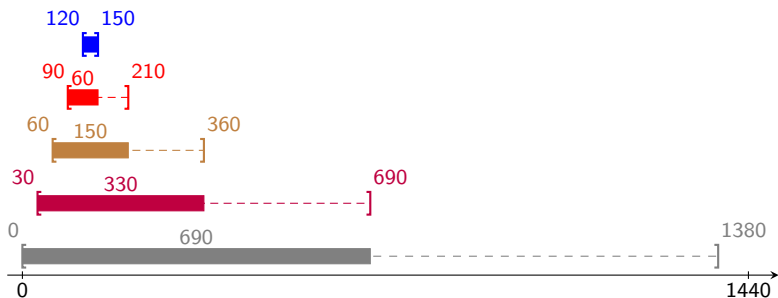
## 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



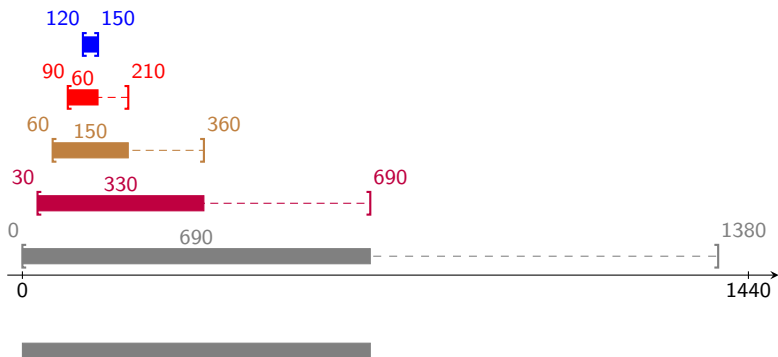
## 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



# 1<sup>er</sup> algorithme : *Earliest Start Time*

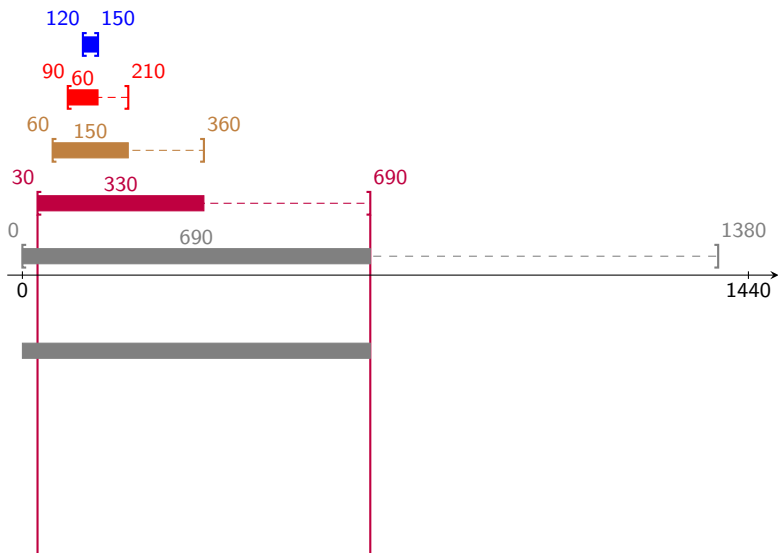
Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible





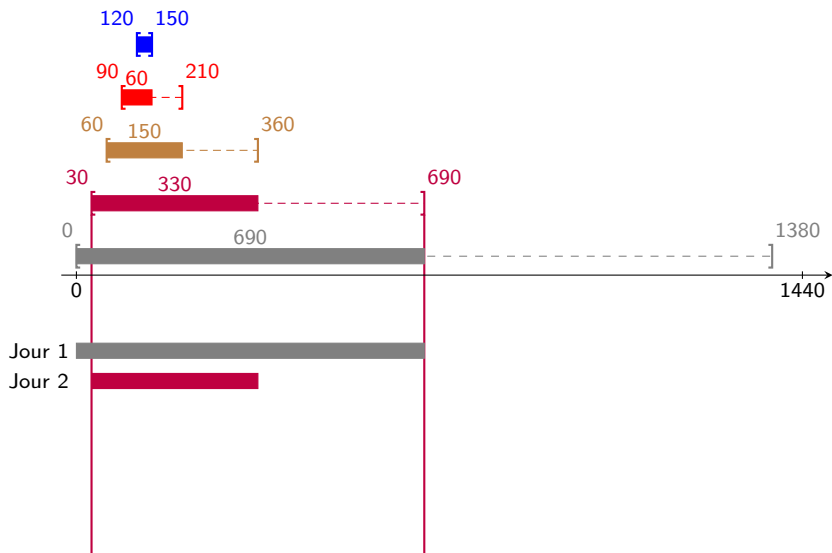
# 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



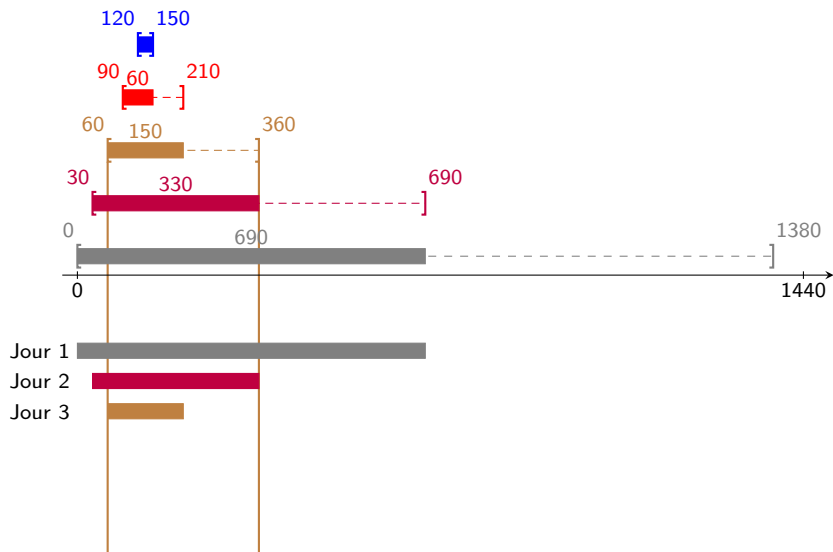
# 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



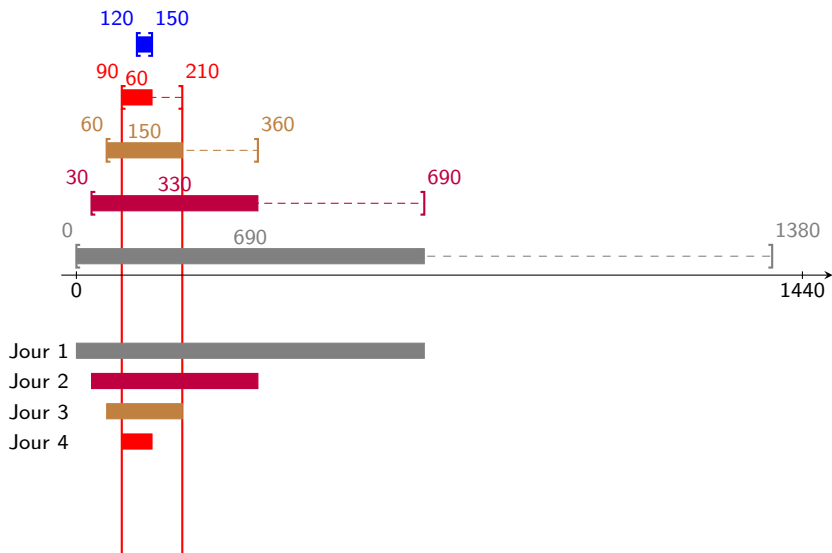
# 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



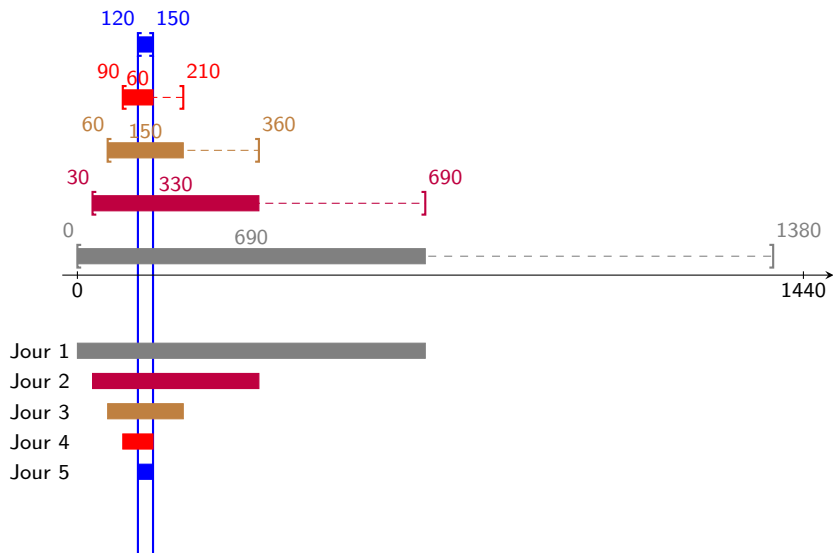
# 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



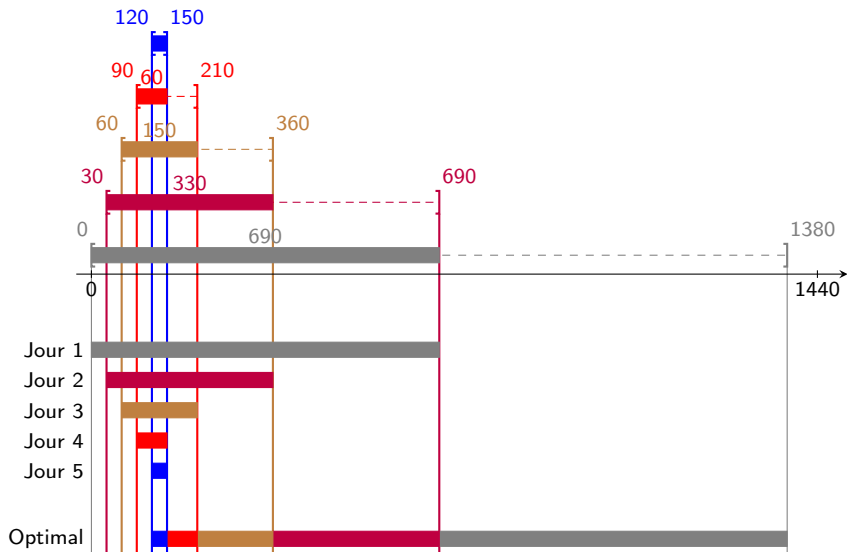
# 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



# 1<sup>er</sup> algorithme : *Earliest Start Time*

Prend une observation qui peut commencer le plus tôt  
et la commence le plus tôt possible



# 1<sup>er</sup> algorithme : *Earliest Start Time* : conclusion

## Exemple présenté

- ▶ Optimal : solution dure moins d'un jour
- ▶ *Earliest Start Time* : solution dure plus de 4 jours
- ▶ Si  $\rho$ -approximation :  $\rho > 4$

## Généralisation de l'exemple

- ▶ *Earliest Start Time* n'est pas un algorithme d'approximation (à facteur constant)
- ▶ Les ordonnancements « raisonnables » **ne** définissent **pas** des algorithmes d'approximation (à facteur constant)

## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

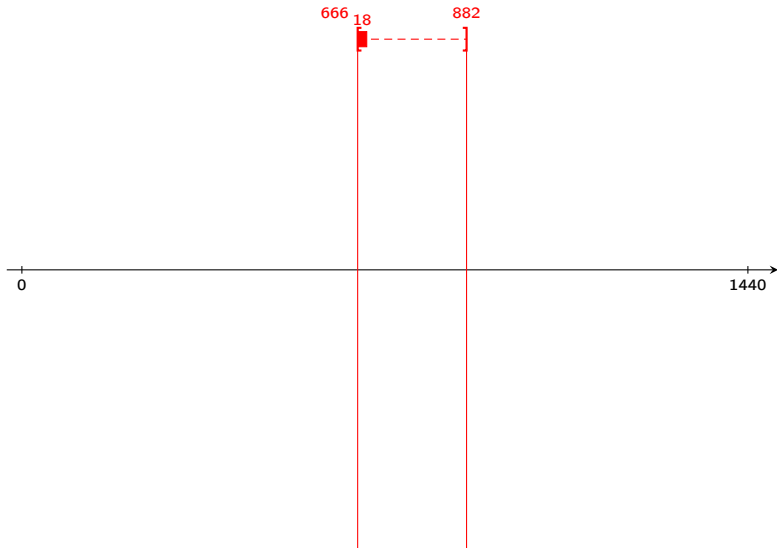
Les observations sont ordonnancées par date butoir croissante





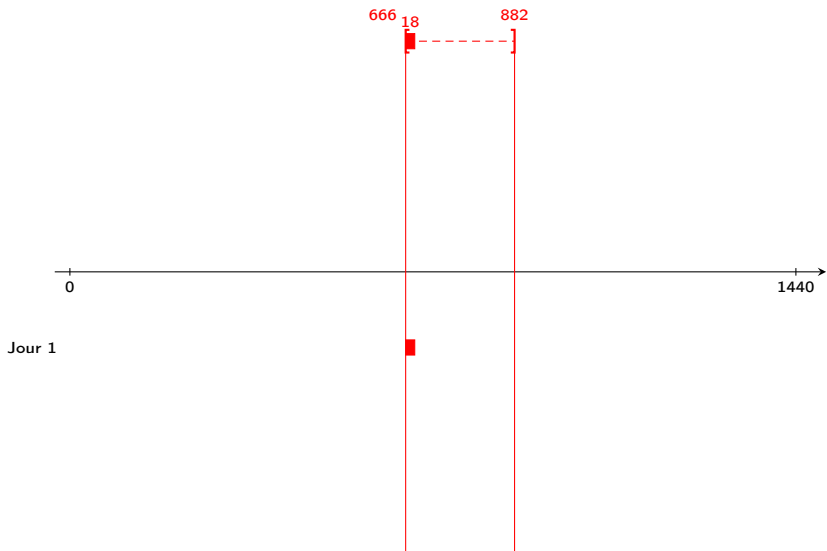
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



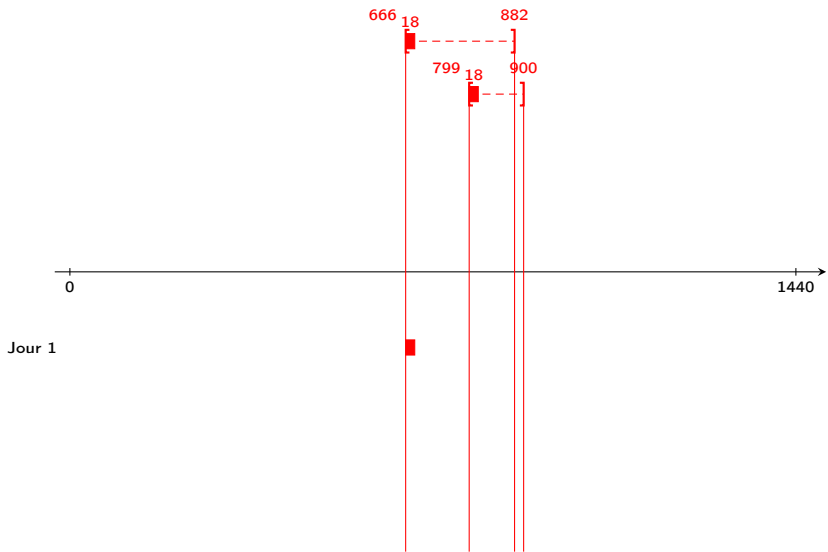
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



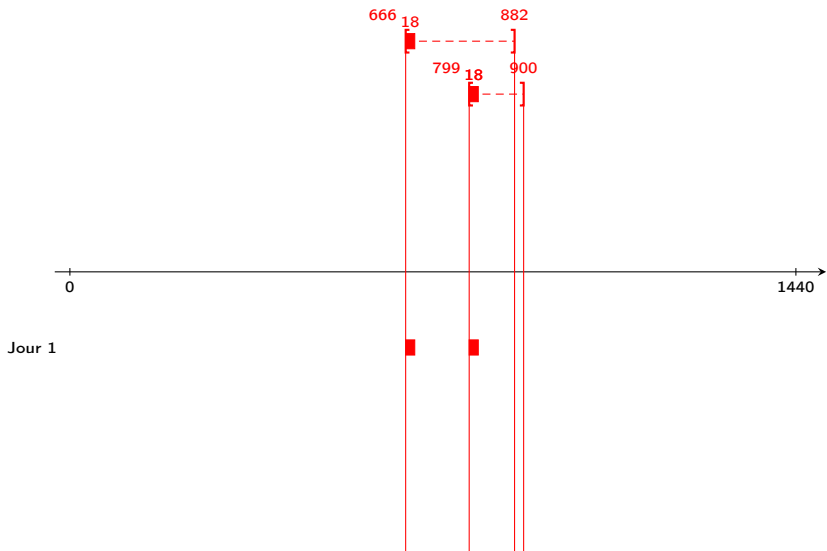
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



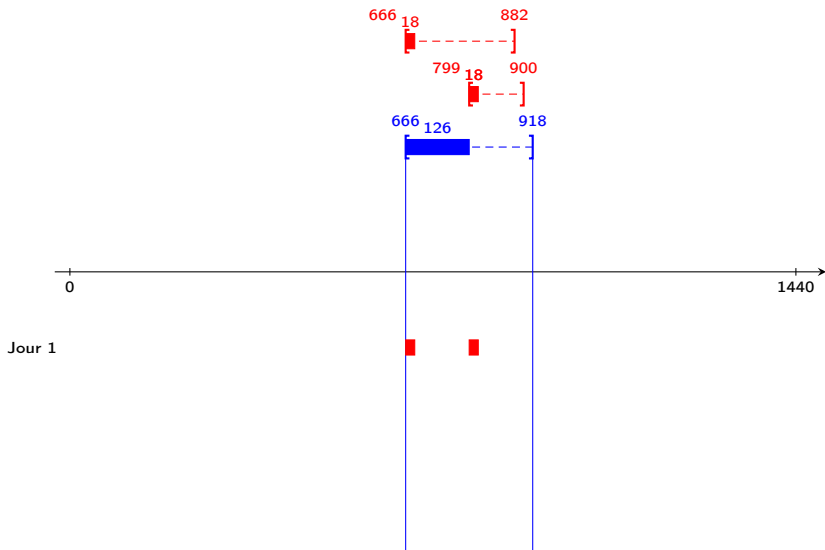
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



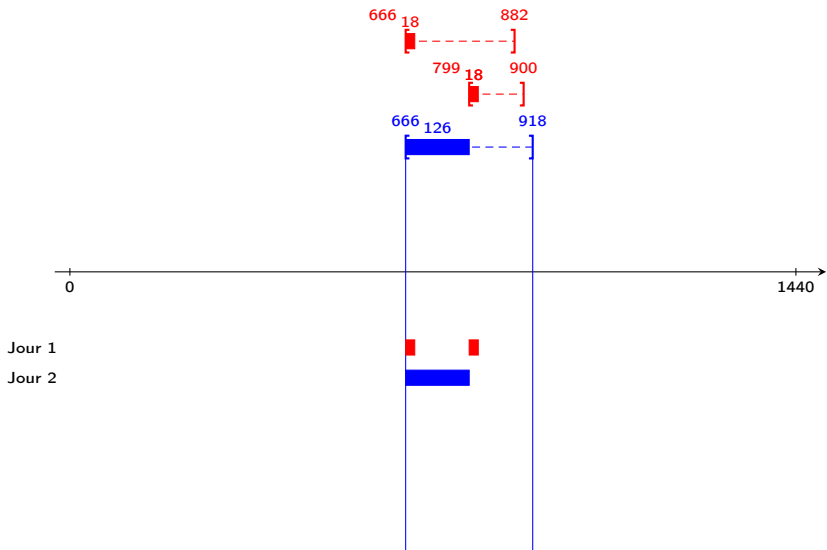
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



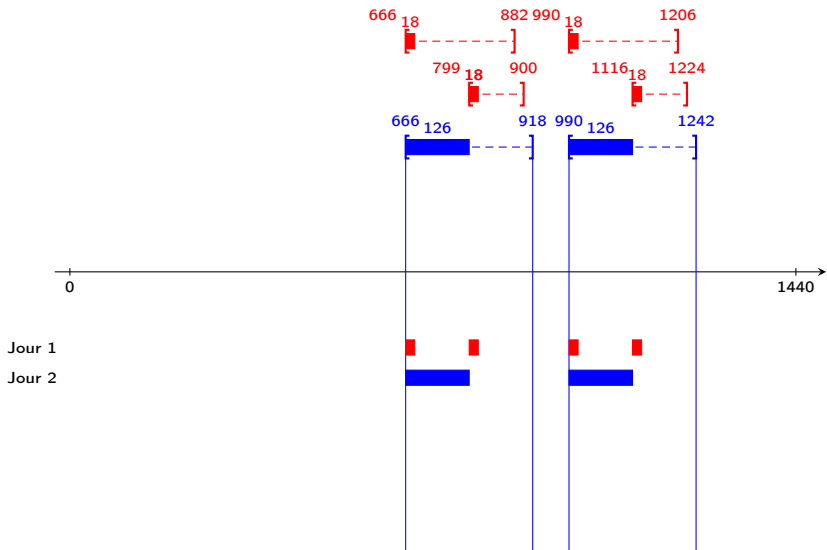
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



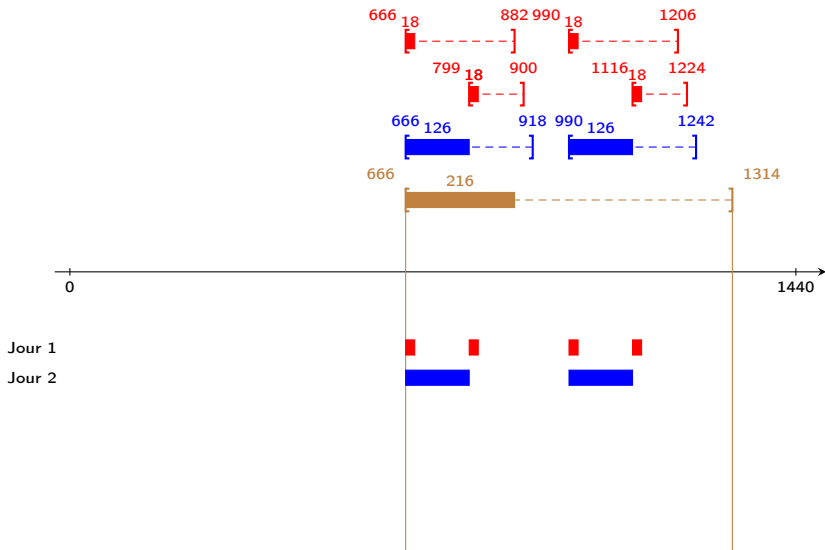
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

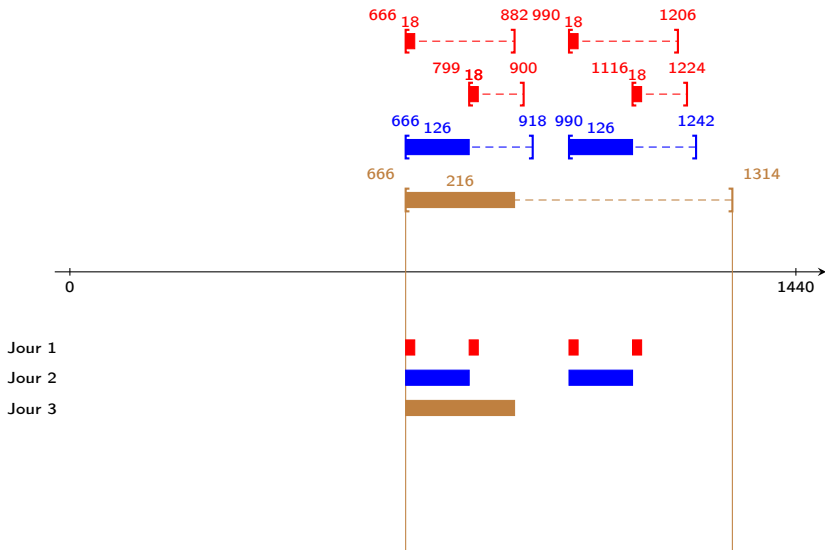
Les observations sont ordonnancées par date butoir croissante





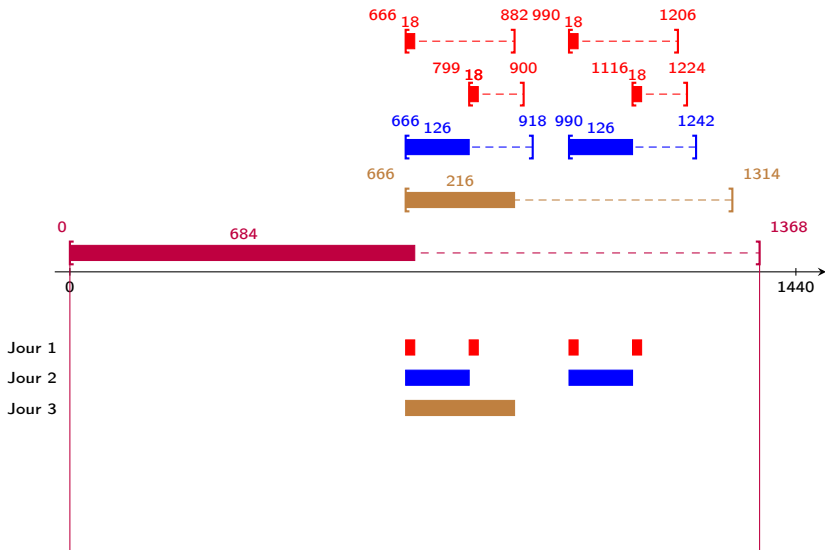
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



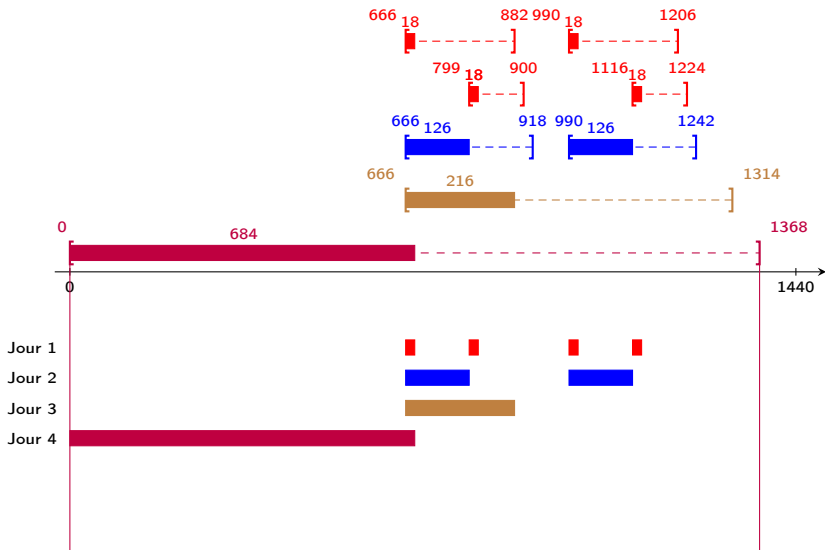
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



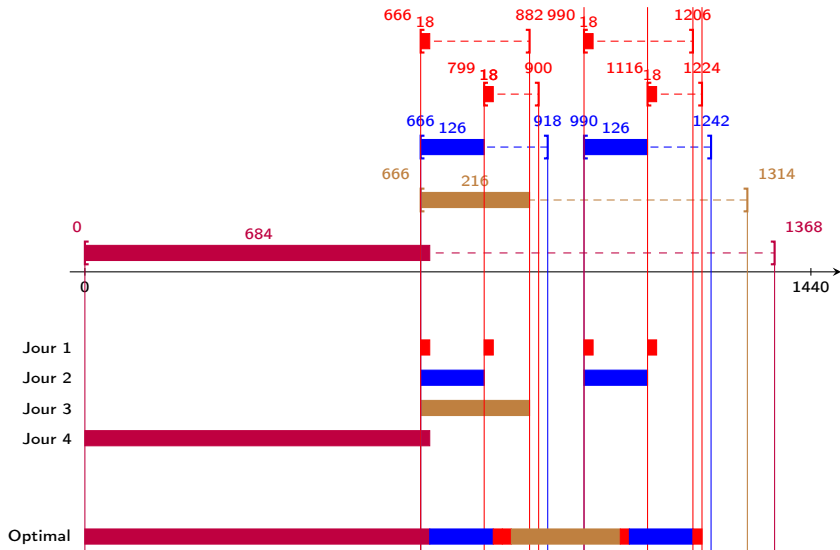
## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF)

Les observations sont ordonnancées par date butoir croissante



## 2<sup>e</sup> algorithme : *Earliest Deadline First* (EDF) : conclusion

### Généralisation de l'exemple

- ▶ *Earliest Deadline First* n'est pas un algorithme d'approximation (à facteur constant)

# Les instances des pires cas ne sont pas réalistes

## Constat

- ▶ En pratique les heuristiques testées sont proches de l'optimal (Observations ordonnancées sur NenuFAR en janvier 2024)
- ▶ Le problème est-il simple en pratique ?
- ▶ Une variabilité des durées des observations limitée...
- ▶ Un quantum de temps relativement grossier (15 minutes ?)...
- ▶ ... Une combinatoire limitée ?

## Piste de travail

- ▶ Recherche de solution optimale par algorithme paramétré

# Conclusion

- ▶ Difficile de faire des choses intelligentes pour le cas général si on ne comprend pas le cas simple
- ▶ Comprendre les spécificités des requêtes d'observations radioastronomiques et l'impact de ces spécificités
- ▶ Rechercher de préférence des solutions de qualité garantie théoriquement