

# Approximating Time-Dependent Transition Times in Constraint Programming for an Earth Observation Mission

Romain Barrault<sup>1,2</sup>, Cédric Pralet<sup>1</sup>, Gauthier Picard<sup>1</sup>, Eric Sawyer<sup>2</sup>

\*[romain.barrault@onera.fr](mailto:romain.barrault@onera.fr)

- 21<sup>th</sup> of july 2026 -

<sup>1</sup> ONERA, <sup>2</sup> CNES

# Context and purpose : *Earth Observation Satellite (EOS)*

---



**Uses :** Natural hazard management, defense, mapping

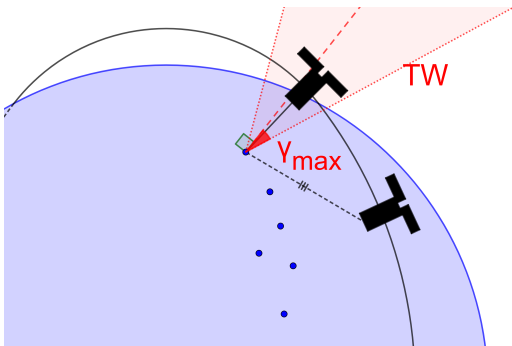
# Problem presentation

---

(i) Visibility time windows

# Problem presentation

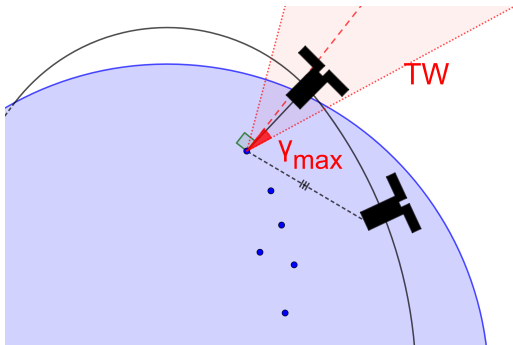
## (i) Visibility time windows



# Problem presentation

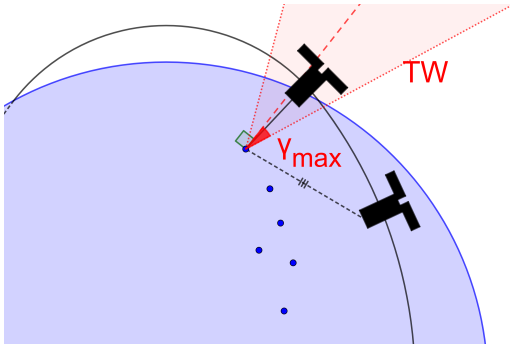
(i) Visibility time windows

(ii) *Time-dependent* transition times

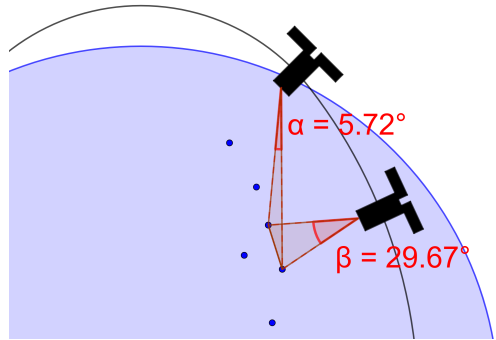


# Problem presentation

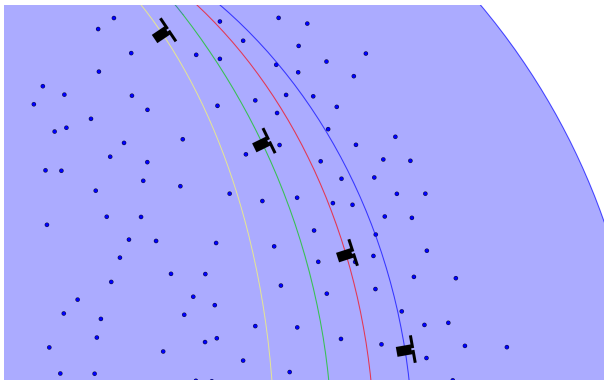
(i) Visibility time windows



(ii) *Time-dependent* transition times



# Problem illustration



# Problem illustration

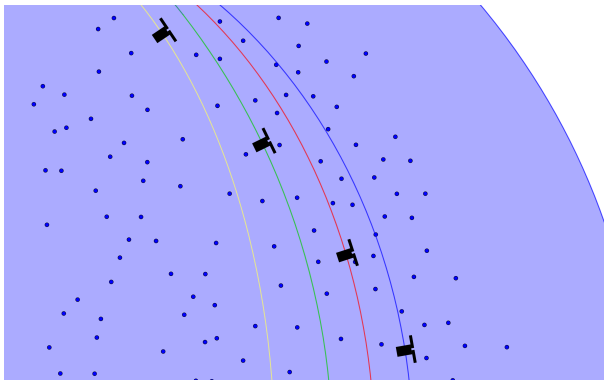


Figure – Questions : what selection ? What dispatching ? What orders ?

# Problem illustration

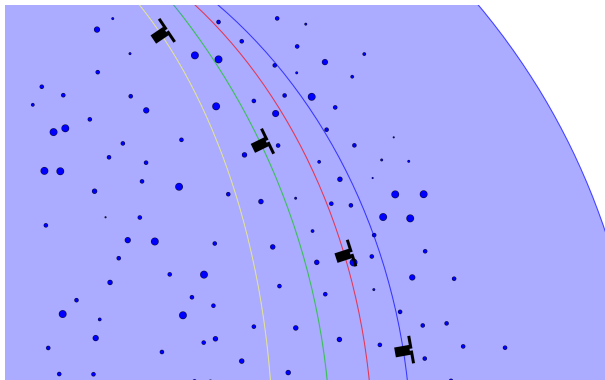


Figure – Questions : what selection ? What dispatching ? What orders ?

# Problem data

- A time horizon  $H$

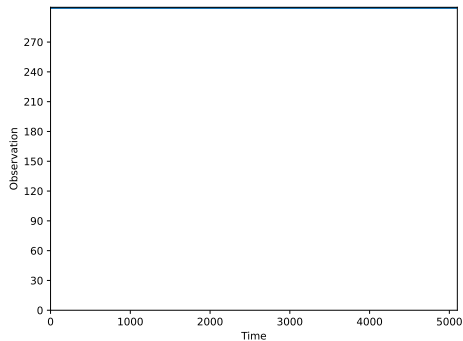


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit

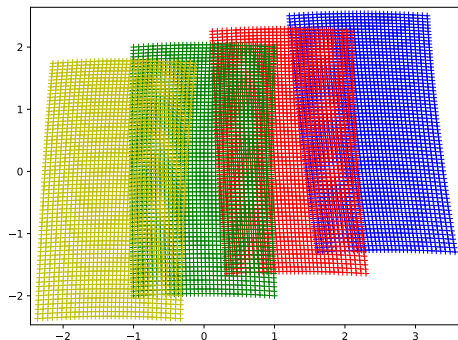


Figure – Areas flown over by satellites

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit

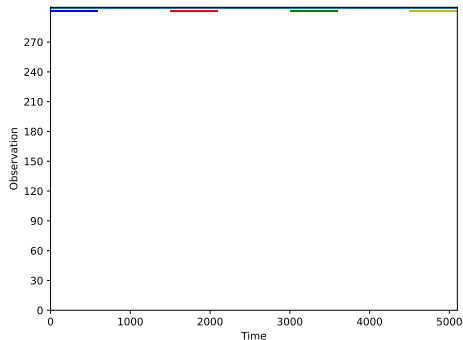


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites

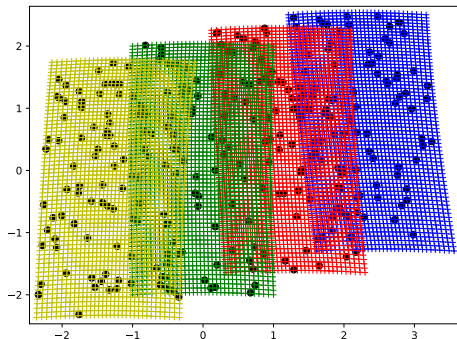


Figure – Areas flown over by satellites

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest (POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :

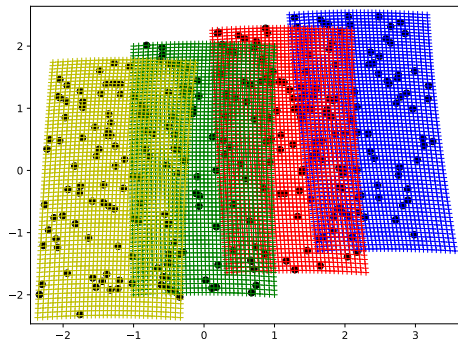


Figure – Areas flown over by satellites

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest (POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$ ;

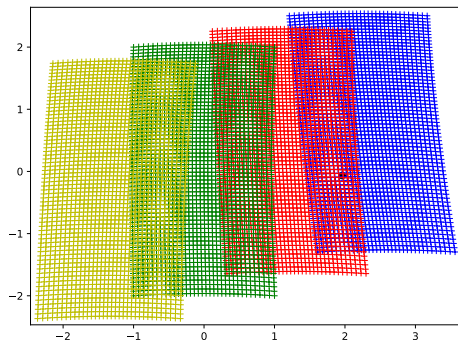


Figure – Areas flown over by satellites

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$ ;
  - a satellite  $S_o \in \mathcal{S}$ ;

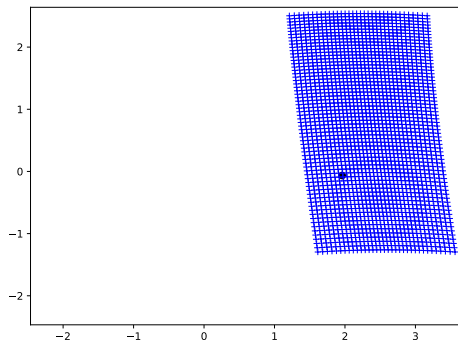


Figure – Areas flown over by satellites

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$  ;
  - a satellite  $S_o \in \mathcal{S}$  ;
  - A visibility time window  $[W_o^{sta}, W_o^{end}] \subset H$  ;

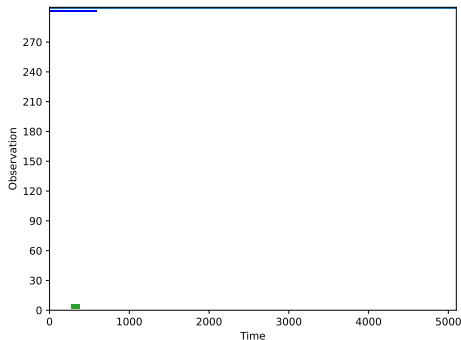


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$  ;
  - a satellite  $S_o \in \mathcal{S}$  ;
  - A visibility time window  $[W_o^{sta}, W_o^{end}] \subset H$  ;
  - an observation duration  $Dur_o$  ;

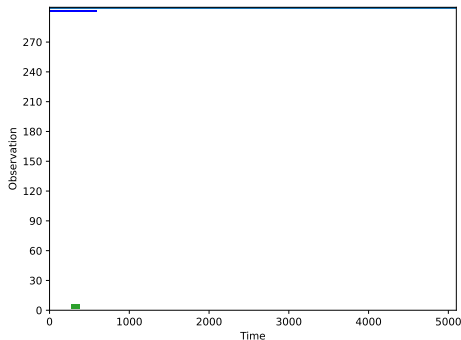


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$  ;
  - a satellite  $S_o \in \mathcal{S}$  ;
  - A visibility time window  $[W_o^{sta}, W_o^{end}] \subset H$  ;
  - an observation duration  $Dur_o$  ;
  - an individual reward  $\rho_o$ .

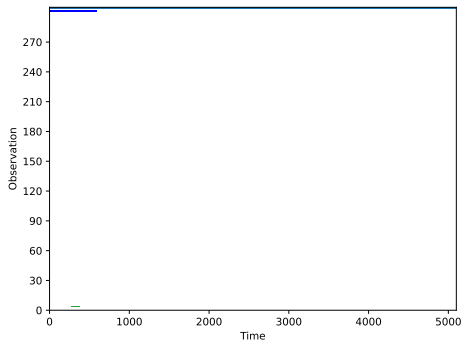


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$ ;
  - a satellite  $S_o \in \mathcal{S}$ ;
  - A visibility time window  $[W_o^{sta}, W_o^{end}] \subset H$ ;
  - an observation duration  $Dur_o$ ;
  - an individual reward  $\rho_o$ .

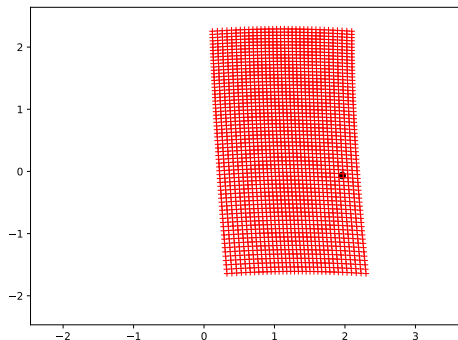


Figure – Areas flown over by satellites

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest(POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$  ;
  - a satellite  $S_o \in \mathcal{S}$  ;
  - A visibility time window  $[W_o^{sta}, W_o^{end}] \subset H$  ;
  - an observation duration  $Dur_o$  ;
  - an individual reward  $\rho_o$ .

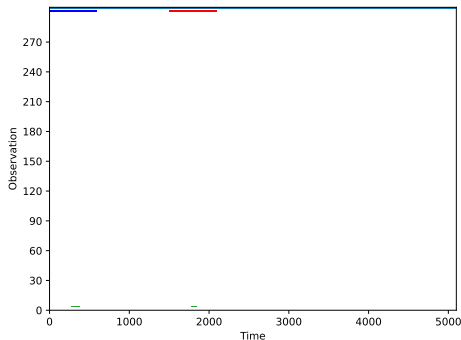


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest (POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$  ;
  - a satellite  $S_o \in \mathcal{S}$  ;
  - A visibility time window  $[W_o^{\text{sta}}, W_o^{\text{end}}] \subset H$  ;
  - an observation duration  $Dur_o$  ;
  - an individual reward  $\rho_o$ .

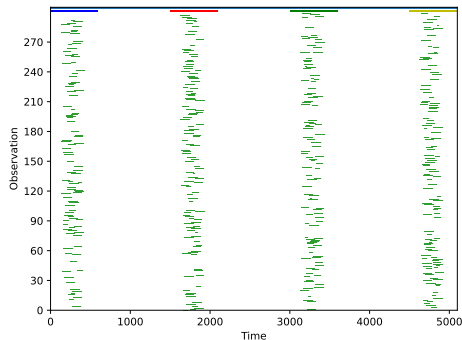


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest (POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$ ;
  - a satellite  $S_o \in \mathcal{S}$ ;
  - A visibility time window  $[W_o^{sta}, W_o^{end}] \subset H$ ;
  - an observation duration  $Dur_o$ ;
  - an individual reward  $\rho_o$ .
- A transition time function family : for  $s \in \mathcal{S}, (o, o') \in \mathcal{O}_s$  :

$$\Delta_{o,o'}^s : t \rightarrow \Delta_{o,o'}^s(t)$$

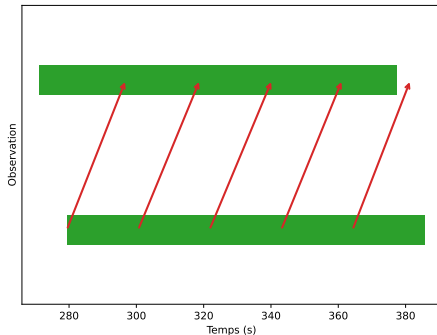


Figure – Gantt diagram

# Problem data

- A time horizon  $H$
- A set of satellites  $\mathcal{S}$  equally spread out on a same polar orbit
- A set of Points Of Interest (POI)  $\mathcal{P}$  visible by at least one of the satellites
- A set of observations  $\mathcal{O}$  defined by :
  - a POI  $P_o \in \mathcal{P}$ ;
  - a satellite  $S_o \in \mathcal{S}$ ;
  - A visibility time window  $[W_o^{\text{sta}}, W_o^{\text{end}}] \subset H$ ;
  - an observation duration  $Dur_o$ ;
  - an individual reward  $\rho_o$ .
- A transition time function family : for  $s \in \mathcal{S}, (o, o') \in \mathcal{O}_s$  :

$$\Delta_{o,o'}^s : t \rightarrow \Delta_{o,o'}^s(t)$$

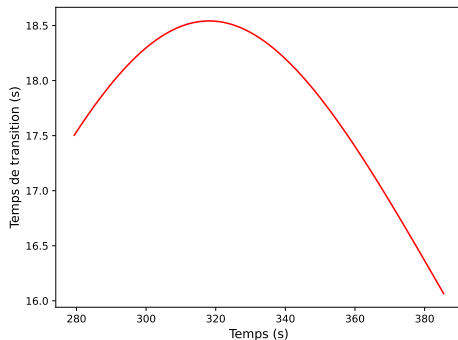


Figure – Time transition function

# Problem mathematical model

## Définition

The *Multi-Satellite Earth Observation Scheduling Problem* (MSEOSP), for  $\zeta = |\mathcal{S}|$  satellites, consists in finding  $\zeta$  observation sequences  $\sigma_s = [\sigma_{s,1}, \dots, \sigma_{s,K_s}]$  such that

# Problem mathematical model

## Définition

The *Multi-Satellite Earth Observation Scheduling Problem* (MSEOSP), for  $\zeta = |\mathcal{S}|$  satellites, consists in finding  $\zeta$  observation sequences  $\sigma_s = [\sigma_{s,1}, \dots, \sigma_{s,K_s}]$  such that

- each candidate POI in  $\mathcal{P}$  appears at most once in the set of  $(\sigma_s)_{s \in \mathcal{S}}$ ;

# Problem mathematical model

## Définition

The *Multi-Satellite Earth Observation Scheduling Problem* (MSEOSP), for  $\zeta = |\mathcal{S}|$  satellites, consists in finding  $\zeta$  observation sequences  $\sigma_s = [\sigma_{s,1}, \dots, \sigma_{s,K_s}]$  such that

- each candidate POI in  $\mathcal{P}$  appears at most once in the set of  $(\sigma_s)_{s \in \mathcal{S}}$  ;
- for each satellite  $s \in \mathcal{S}$  and observation  $o \in \sigma_s (o = \sigma_{s,i})$ , we define the observation instant of  $o$  by recurrence with respect to visibility window and transition times  $t_{\sigma_{s,i}} = \max(W_o^{\text{sta}}, t_{\sigma_{s,i-1}} + \text{Dur}_{\sigma_{s,i-1}} + \Delta_{\sigma_{s,i-1}, \sigma_{s,i}}^s (t_{\sigma_{s,i-1}} + \text{Dur}_{\sigma_{s,i-1}}))$

# Problem mathematical model

## Définition

The *Multi-Satellite Earth Observation Scheduling Problem* (MSEOSP), for  $\zeta = |\mathcal{S}|$  satellites, consists in finding  $\zeta$  observation sequences  $\sigma_s = [\sigma_{s,1}, \dots, \sigma_{s,K_s}]$  such that

- each candidate POI in  $\mathcal{P}$  appears at most once in the set of  $(\sigma_s)_{s \in \mathcal{S}}$  ;
- for each satellite  $s \in \mathcal{S}$  and observation  $o \in \sigma_s (o = \sigma_{s,i})$ , we define the observation instant of  $o$  by recurrence with respect to visibility window and transition times  $t_{\sigma_{s,i}} = \max(W_o^{\text{sta}}, t_{\sigma_{s,i-1}} + \text{Dur}_{\sigma_{s,i-1}} + \Delta_{\sigma_{s,i-1}, \sigma_{s,i}}^s (t_{\sigma_{s,i-1}} + \text{Dur}_{\sigma_{s,i-1}}))$
- the latter verifies  $t_{\sigma_{s,i}} \leq W_o^{\text{end}}$

# Problem mathematical model

## Définition

The *Multi-Satellite Earth Observation Scheduling Problem* (MSEOSP), for  $\zeta = |\mathcal{S}|$  satellites, consists in finding  $\zeta$  observation sequences  $\sigma_s = [\sigma_{s,1}, \dots, \sigma_{s,K_s}]$  such that

- each candidate POI in  $\mathcal{P}$  appears at most once in the set of  $(\sigma_s)_{s \in \mathcal{S}}$  ;
- for each satellite  $s \in \mathcal{S}$  and observation  $o \in \sigma_s (o = \sigma_{s,i})$ , we define the observation instant of  $o$  by recurrence with respect to visibility window and transition times  $t_{\sigma_{s,i}} = \max(W_o^{\text{sta}}, t_{\sigma_{s,i-1}} + \text{Dur}_{\sigma_{s,i-1}} + \Delta_{\sigma_{s,i-1}, \sigma_{s,i}}^s (t_{\sigma_{s,i-1}} + \text{Dur}_{\sigma_{s,i-1}}))$
- the latter verifies  $t_{\sigma_{s,i}} \leq W_o^{\text{end}}$
- the sum of individual collected rewards  $\sum_{s \in \mathcal{S}, o \in \sigma_s} \rho_o$  is maximized.

# Solving the problem in the SOTA

---

- [15] : *Dynamic discretization discovery for solving the Continuous Time Inventory Routing Problem with Out-and-Back Routes*, Lagos et al., 2022 : time-dependent warehouse supply problem solved in an exact way thanks to dynamic discretization of the temporal state, but does not scale well (up to 2h of computation for 30 nodes instances)

# Solving the problem in the SOTA

---

- [15] : *Dynamic discretization discovery for solving the Continuous Time Inventory Routing Problem with Out-and-Back Routes*, Lagos et al., 2022 : time-dependent warehouse supply problem solved in an exact way thanks to dynamic discretization of the temporal state, but does not scale well (up to 2h of computation for 30 nodes instances)
- [1] *A Time-Dependent No-Overlap Constraint : Application to Urban Delivery Problems*, Aguiar Melgarejo et al., 2015 : *Time-Dependent Traveling Salesman Problem* solving with time-dependent constraints, but unavailable implementation and does not scale well either.

# Constraint Programming and metaheuristics combination

Interval variable

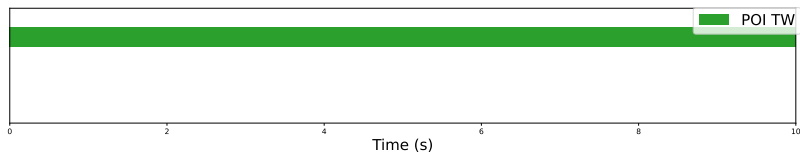


Figure – POI visibility windows

# CP Model

## Interval variable

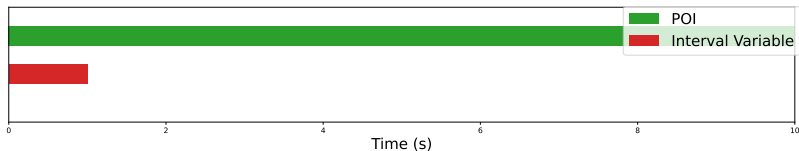


Figure – Interval variable for POI observation

# CP Model

## Interval variable

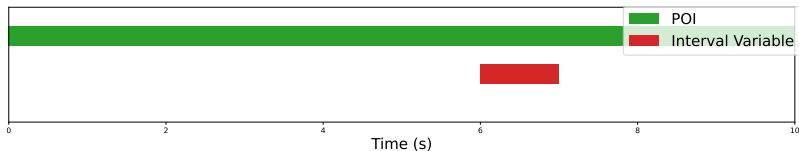


Figure – Interval variable for POI observation

# CP Model

## Interval variable

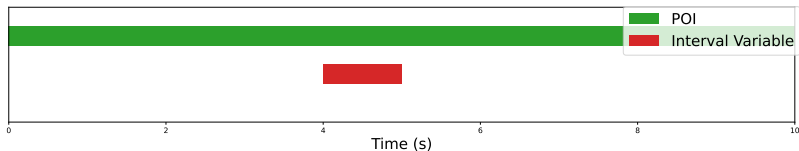


Figure – Interval variable for POI observation

# CP Model

## Interval variable

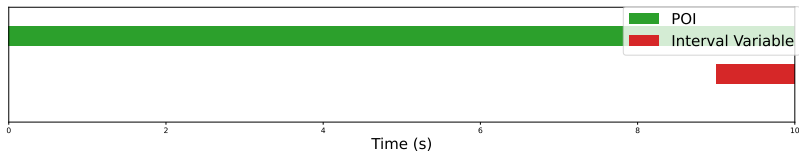


Figure – Interval variable for POI observation

Constant transition time matrix

$$\begin{pmatrix} 0 & 8 & 7 & 22 \\ 10 & 0 & 8 & 9 \\ 5 & 15 & 0 & 15 \\ 15 & 20 & 30 & 0 \end{pmatrix}$$

## Constant transition time matrix

$$\begin{pmatrix} 0 & 8 & 7 & 22 \\ 10 & 0 & 8 & 9 \\ 5 & 15 & 0 & 15 \\ 15 & 20 & 30 & 0 \end{pmatrix}$$

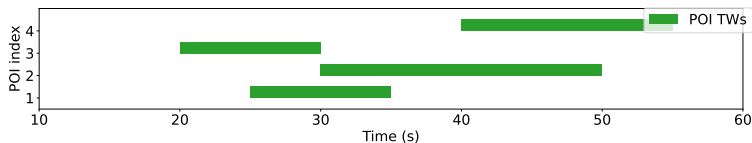


Figure – POI visibility window

## Constant transition time matrix

$$\begin{pmatrix} 0 & 8 & 7 & 22 \\ 10 & 0 & 8 & 9 \\ 5 & 15 & 0 & 15 \\ 15 & 20 & 30 & 0 \end{pmatrix}$$

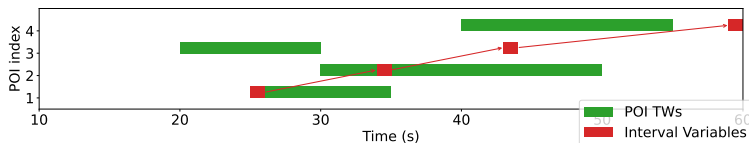


Figure – Incorrect observation instantiation

Implementation with `noOverlap(obsSequence, transitionTime)`

## Constant transition time matrix

$$\begin{pmatrix} 0 & 8 & 7 & 22 \\ 10 & 0 & 8 & 9 \\ 5 & 15 & 0 & 15 \\ 15 & 20 & 30 & 0 \end{pmatrix}$$

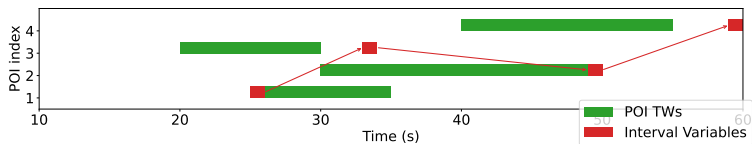


Figure – Incorrect observation instantiation

## Constant transition time matrix

$$\begin{pmatrix} 0 & 8 & 7 & 22 \\ 10 & 0 & 8 & 9 \\ 5 & 15 & 0 & 15 \\ 15 & 20 & 30 & 0 \end{pmatrix}$$

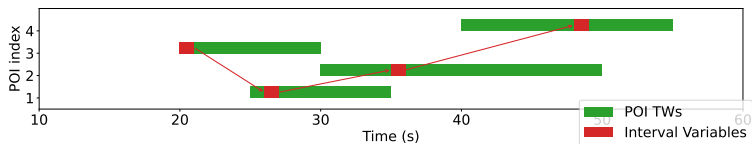


Figure – Correct observation instantiation

# CP Model

$$\text{maximize} \quad \sum_{o \in \mathcal{O}} \rho_o \cdot \text{presenceOf}(\mathbf{itvObs}_o) \quad (1)$$

$$\forall p \in \mathcal{P}, \quad \sum_{o \in \mathcal{O} \mid P_o = p} \text{presenceOf}(\mathbf{itvObs}_o) \leq 1 \quad (2)$$

$$\forall s \in \mathcal{S}, \text{noOverlap}(\{\mathbf{itvObs}_o \mid o \in \mathcal{O}_s\}, trTimeApprox^s) \quad (3)$$

$$\forall s \in \mathcal{S}, o \in \mathcal{O}_s, \mathbf{itvObs}_o = \text{intervalVariable}([W_o^{\text{sta}}, W_o^{\text{end}}], Dur_o) \quad (4)$$

$$\text{maximize} \quad \sum_{o \in \mathcal{O}} \rho_o \cdot \text{presenceOf}(\mathbf{itvObs}_o) \quad (1)$$

$$\forall p \in \mathcal{P}, \quad \sum_{o \in \mathcal{O} \mid P_o = p} \text{presenceOf}(\mathbf{itvObs}_o) \leq 1 \quad (2)$$

$$\forall s \in \mathcal{S}, \text{noOverlap}(\{\mathbf{itvObs}_o \mid o \in \mathcal{O}_s\}, trTimeApprox^s) \quad (3)$$

$$\forall s \in \mathcal{S}, o \in \mathcal{O}_s, \mathbf{itvObs}_o = \text{intervalVariable}([W_o^{\text{sta}}, W_o^{\text{end}}], Dur_o) \quad (4)$$

Transition time approximation strategy :

$$trTimeApprox_{o,o'}^s = \text{approx}\{trTime_{o,o'}^s(W_o^{\text{sta}} + k \cdot T) \mid k \in [0..n]\} \quad (5)$$

$$\text{where } \text{approx} \in \{\min, \max, \text{avg}\} \ \& \ T = (W_o^{\text{end}} - W_o^{\text{sta}})/n \quad (6)$$

# A finer CP model

Problem : temporal dependency ?

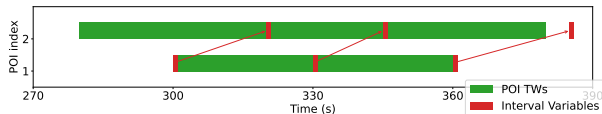
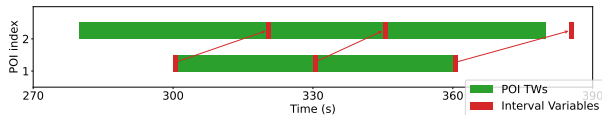


Figure – Time dependency's influence ?

# A finer CP model

Problem : temporal dependency ?



Transition time worst modeling error equal to :

$$\epsilon = \max_{s \in S} \max_{o \neq o' \in \mathcal{O}_s} \max_{t \in [W_o^{\text{sta}} + \text{Dur}_o, W_o^{\text{end}}]} |trTimeApprox_{o,o'}^s - trTime_{o,o'}^s(t)| \quad (7)$$

# A finer CP model

## TW discretization and corresponding matrix

$$\begin{pmatrix} 0 & 12 \\ 14 & 0 \end{pmatrix}$$

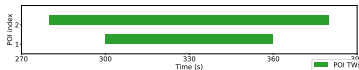


Figure – Time windows discretization : for  $o \in \mathcal{O}$ , we have the sub-observations  $\Omega(o) = \{\omega_1\}$

# A finer CP model

## TW discretization and corresponding matrix

$$\begin{pmatrix} 0 & 0 & 12 & 12 \\ 0 & 0 & 12.5 & 12.5 \\ 14 & 14 & 0 & 0 \\ 13 & 13 & 0 & 0 \end{pmatrix}$$



Figure – Time windows discretization : for  $o \in \mathcal{O}$ , we have the sub-observations  $\Omega(o) = \{\omega_1, \omega_2\}$

# A finer CP model

## TW discretization and corresponding matrix

$$\begin{pmatrix} 0 & 0 & \dots & 12 & 12 \\ 0 & 0 & \dots & 12.5 & 12.5 \\ & & \dots & & \\ 0 & 0 & \dots & 11.8 & 11.8 \\ 14 & 14 & \dots & 0 & 0 \\ & & \dots & & \\ 13.5 & 13.5 & \dots & 0 & 0 \\ 13.2 & 13.2 & \dots & 0 & 0 \end{pmatrix}$$

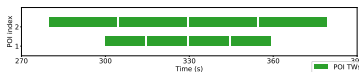


Figure – Time windows discretization : for  $o \in \mathcal{O}$ , we have the sub-observations  $\Omega(o) = \{\omega_1, \dots, \omega_4\}$

# Resulting CP model

The transition time worst modeling error is then equal to

$$\epsilon' = \max_{s \in \mathcal{S}} \max_{\substack{o \neq o' \in \mathcal{O}_s \\ \omega \in \Omega(o), \omega' \in \Omega(o')}} \max_{t \in [W_{\omega}^{\text{sta}} + \text{Dur}_o, W_{\omega'}^{\text{end}}]} |trTimeApprox_{\omega, \omega'}^s - trTime_{\omega, \omega'}^s(t)| \quad (7)$$

→ compromise between model precision and induced additional complexity

# Resulting CP model

$$\text{maximize} \quad \sum_{\omega \in \Omega} \rho_{\omega} \cdot \text{presenceOf}(\mathbf{itvObs}_{\omega}) \quad (7)$$

$$\forall p \in \mathcal{P}, \quad \sum_{\omega \in \Omega \mid P_{\omega}=p} \text{presenceOf}(\mathbf{itvObs}_{\omega}) \leq 1 \quad (8)$$

$$\forall s \in \mathcal{S}, \text{ noOverlap}(\{\mathbf{itvObs}_{\omega} \mid \omega \in \Omega_s\}, trTimeApprox^s) \quad (9)$$

$$\forall s \in \mathcal{S}, o \in \mathcal{O}, \omega \in \Omega_o, \mathbf{itvObs}_{\omega} = \text{intervalVariable}([W_{\omega}^{\text{sta}}, W_{\omega}^{\text{end}}], Dur_o) \quad (10)$$

$$(11)$$

$$\text{where } trTimeApprox_{\omega, \omega'}^s = \text{approx}\{trTime_{o, o'}^s(W_{\omega}^{\text{sta}} + k \cdot T) \mid k \in [0..n]\}, \quad (12)$$

$$\text{approx} \in \{\min, \max, \text{avg}\} \text{ and } T = (W_{\omega}^{\text{end}} - W_{\omega}^{\text{sta}})/n \quad (13)$$

# Discretization strategies

## Reference discretization

No induced sub-observation :  $\forall o \in \mathcal{O}, \Omega(o) = \{o\}$

→ method F (for **Full window**)

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :



Figure – Constant timestep discretization : strategy T

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :

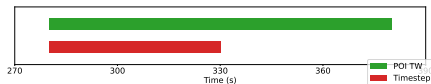


Figure – Constant timestep discretization : strategy T

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :

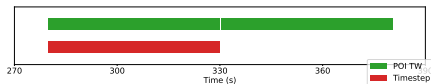


Figure – Constant timestep discretization : strategy T

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :

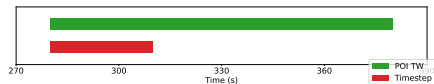


Figure – Constant timestep discretization : strategy T

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :

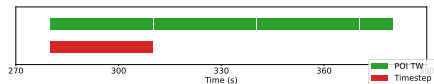


Figure – Constant timestep discretization : strategy T

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :



Figure – Constant timestep discretization : strategy T

# Discretization strategies

## Constant timestep discretization

For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$  and visibility window  $[W_o^{sta}, W_o^{end}]$ , and a constant timestep  $\tau$ , we define the corresponding sub-observations :

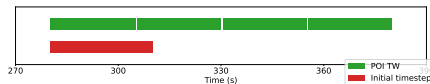


Figure – Constant timestep discretization : strategy T

- Number of sub-observations :  $n_o = \lceil (W_o^{end} - W_o^{sta} - Dur_o) / \tau \rceil$
- Individual timestep :  $\tau_o = (W_o^{end} - W_o^{sta} - Dur_o) / n_o$
- Induced sub-windows :  $([W_o^{sta} + k \cdot \tau_o, W_o^{sta} + Dur_o + (k + 1) \cdot \tau_o])_{k \in \{0..n_o-1\}}$

# Discretization strategies

## Constant angular step discretization

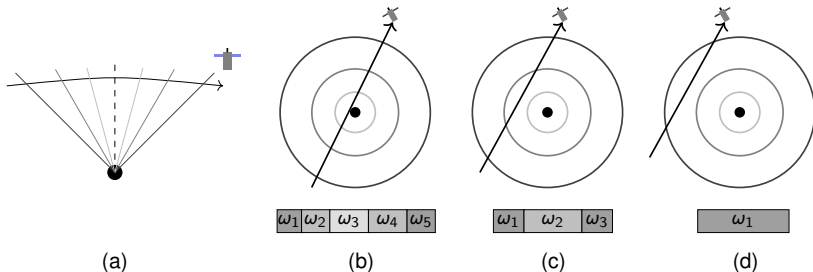


Figure – Sub-windows computation with angular step

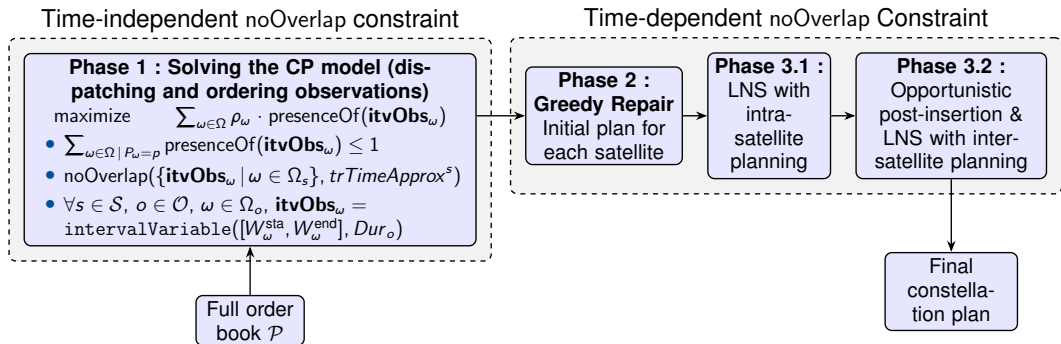
# Discretization strategies

## Constant angular step discretization

Maximum observation angle :  $\alpha_o^{\max} = 30^\circ$ . For an observation  $o \in \mathcal{O}$  of duration  $Dur_o$ , visibility window  $[W_o^{\text{sta}}, W_o^{\text{end}}]$ , best observation angle  $\alpha_o^{\text{prox}}$  reached at instant  $t_o^{\text{prox}}$ , and an angular step equal to  $\theta$  :

- Number of angular regions  $n_o = \lceil (\alpha_o^{\max} - \alpha_o^{\text{prox}}) / \theta \rceil$  ;
- Individual angular step  $\theta_o = (\alpha_o^{\max} - \alpha_o^{\text{prox}}) / n_o$  for an equal spreading of the interval  $[\alpha_o^{\text{prox}}, \alpha_o^{\max}]$  ;
- $\forall k \in [1..n_o]$ , time window  $[s_{o,k}, e_{o,k}] \subseteq [W_o^{\text{sta}}, W_o^{\text{end}}]$  during which the satellite traverses the cone of aperture  $\alpha_{o,k} = \alpha_o^{\max} - (k - 1) \cdot \theta_o$
- Induced sub-windows :  
 $[s_{o,1}, s_{o,2}], \dots [s_{o,n_o-1}, s_{o,n_o}], [s_{o,n_o}, e_{o,n_o}], [e_{o,n_o}, e_{o,n_o-1}], \dots [e_{o,3}, e_{o,2}], [e_{o,2}, e_{o,1}]$

# Solving algorithm workflow



# Experimental protocol

---

Problem instances



# Experimental protocol

---

## Problem instances

- 4 satellites

# Experimental protocol

## Problem instances

- 4 satellites
- 300 POIs seen by at most 2 different satellites

# Experimental protocol

## Problem instances

- 4 satellites
- 300 POIs seen by at most 2 different satellites
- 72 instances with real cloud cover data for computing rewards

# Experimental protocol

## Problem instances

- 4 satellites
- 300 POIs seen by at most 2 different satellites
- 72 instances with real cloud cover data for computing rewards

## Algorithm parameters

# Experimental protocol

## Problem instances

- 4 satellites
- 300 POIs seen by at most 2 different satellites
- 72 instances with real cloud cover data for computing rewards

## Algorithm parameters

- 2 minutes of computation : 1m30 for solving CP, 15s for intra-satellite LNS and 15s for inter-satellite LNS

# Experimental protocol

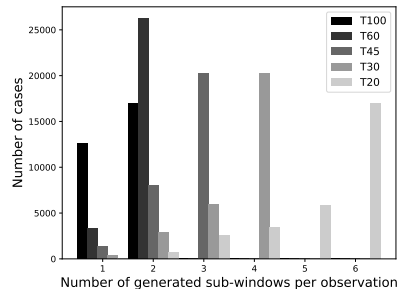
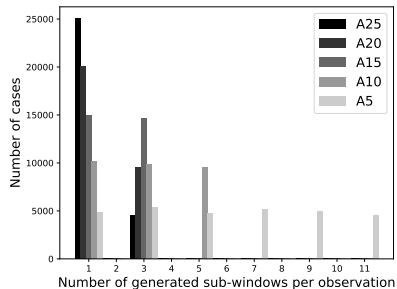
## Problem instances

- 4 satellites
- 300 POIs seen by at most 2 different satellites
- 72 instances with real cloud cover data for computing rewards

## Algorithm parameters

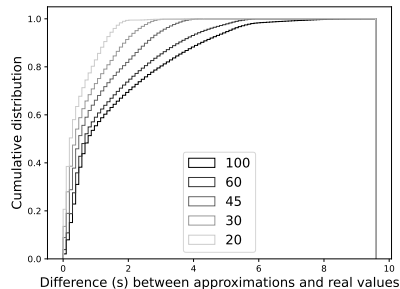
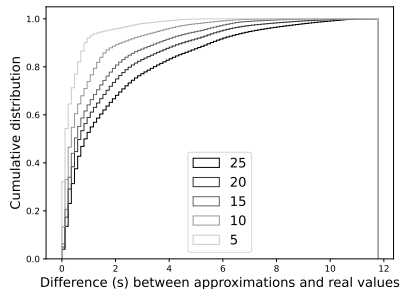
- 2 minutes of computation : 1m30 for solving CP, 15s for intra-satellite LNS and 15s for inter-satellite LNS
- Algorithms implemented in Java and executed on 20-core Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 62GB RAM, Ubuntu 18.04.5 LTS, with an OpenJDK 11.0.9 JVM; 22.1.1 version of CP Optimizer, and noOverlap constraint propagation set to value Extended

# Induced complexity



**Figure** – Distribution of the number of generated sub-windows per discretization strategy : angular step (A) on the left, and time step (T) on the right. Results for 5 different parameter values ( $^{\circ}$  et  $s$ ).

# Induced complexity



**Figure** – Cumulative distribution functions with different parameter values for both strategies (strategy A on left, strategy T on right).

# Performances

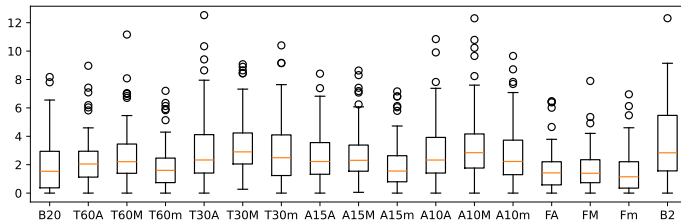


Figure – Optimality gaps (%) for each algorithm variant and baselines for every instance

# Performances

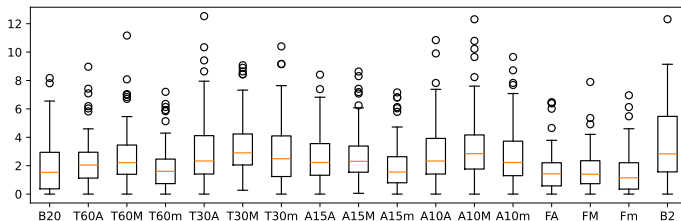


Figure – Optimality gaps (%) for each algorithm variant and baselines for every instance

Baselines are beaten and the best approximation function seems to be min, but the finer the discretization, the worse the final reward

# Conclusion and perspectives

---

## Conclusion

- Improved performances thanks to the algorithm workflow

# Conclusion and perspectives

---

## Conclusion

- Improved performances thanks to the algorithm workflow
- The sub-window generation does not present significant interest for over-constrained instances

# Conclusion and perspectives

## Conclusion

- Improved performances thanks to the algorithm workflow
- The sub-window generation does not present significant interest for over-constrained instances
- The CP model's return does not rival with the baseline with equal computation time but it is a good warm-start for the *LNS*

# Conclusion and perspectives

## Conclusion

- Improved performances thanks to the algorithm workflow
- The sub-window generation does not present significant interest for over-constrained instances
- The CP model's return does not rival with the baseline with equal computation time but it is a good warm-start for the *LNS*

## Perspectives

Dynamic discretization to refine the approximation

**Thank you for your attention !**  
**Any question ?**

[www.onera.fr](http://www.onera.fr)

# Appendix

# Performances

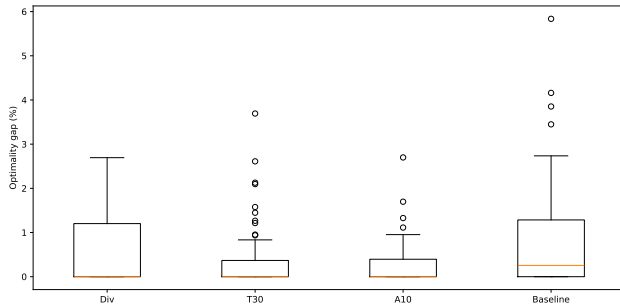


Figure – Optimality gaps (%) for each algorithm variant and baselines for every instance (100 POIs)

# Reward along the workflow

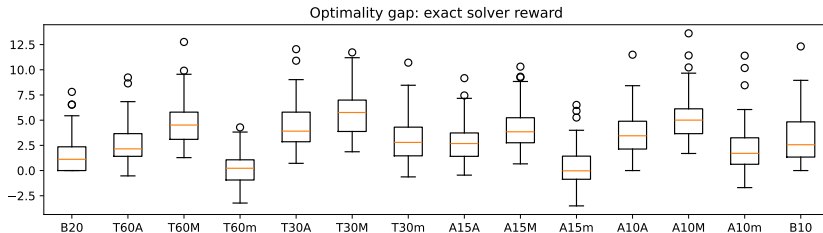


Figure – Optimality gaps (used value : CP model's final reward)

# Reward along the workflow

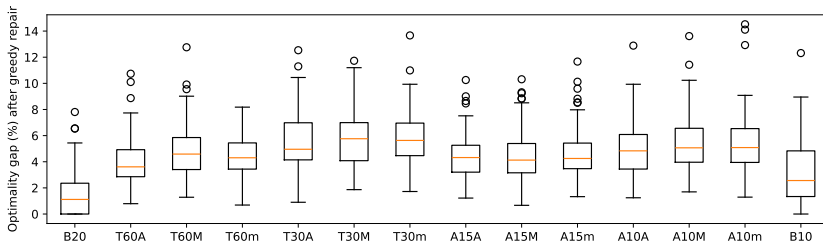


Figure – Optimality gaps (used value : algorithm's reward after greedy repair of the CP solution)

# Reward along the workflow

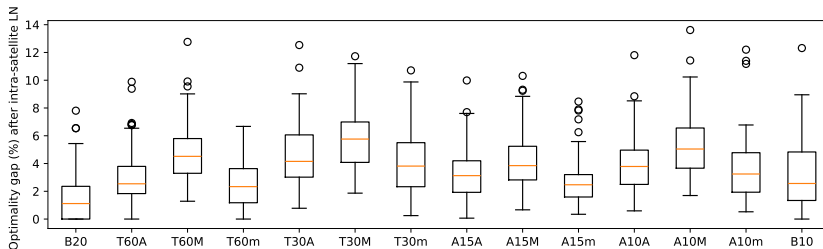


Figure – Optimality gaps (used value : algorithm's reward after intra-satellite LNS)