Robust Observation Planning via Facet Reasoning

Jennifer Santos[1] Damien Van Meerbeeck[1] Mika Skjelnes[1]
Arnaud Lequen[1] Daniel Gnad[1,2]

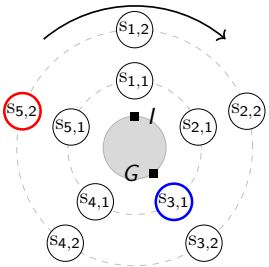
[1]Linköping University [2]Heidelberg University

Outline

Earth observation with satellite constellations

Observation planning: schedule the operations of a constellation to

- take pictures of a set of *regions of interest* (ROIs),
- send every picture down to a ground station.



Acquisition cannot be planned independently from delivery

- Downlink opportunities are scarce: few satellites see a ground station.
- On-board memory is limited.
- An image may have to be **forwarded through the constellation** before it can reach the ground.

Replanning is the norm, not the exception

Foreseen constraints (orbital geometry)

- A satellite can image a ROI only while flying over it.
- It can downlink only in line of sight of a ground station.

$\Rightarrow$ Opportunities come in **narrow, periodic windows**.

Replanning is the norm, not the exception

Foreseen constraints (orbital geometry)

- A satellite can image a ROI only while flying over it.
- It can downlink only in line of sight of a ground station.

$\Rightarrow$ Opportunities come in **narrow, periodic windows**.

Unforeseen constraints (dynamic environment)

- Cloud cover invalidates an observation window.
- A hardware fault disables a satellite.
- A communication restriction closes a link.

Replanning is the norm, not the exception

Foreseen constraints (orbital geometry)

- A satellite can image a ROI only while flying over it.
- It can downlink only in line of sight of a ground station.

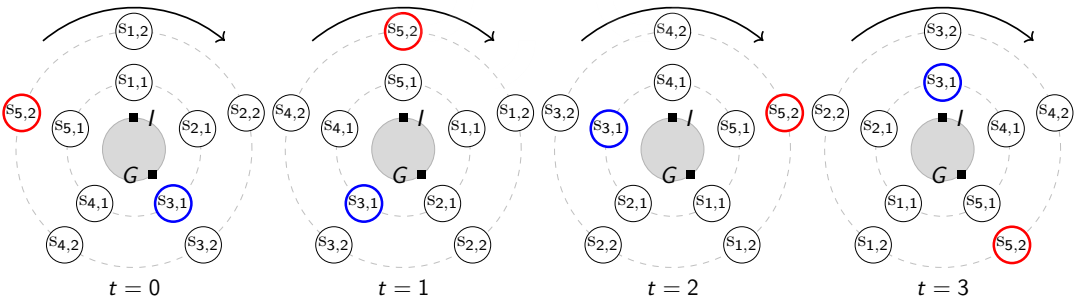
$\Rightarrow$ Opportunities come in **narrow, periodic windows**.

Unforeseen constraints (dynamic environment)

- Cloud cover invalidates an observation window.
- A hardware fault disables a satellite.
- A communication restriction closes a link.

Replanning *from scratch* is wasteful when most of the solution remains valid, and may discard manual adjustments made by the operator.

Orbital motion



- The satellite constellation rotates periodically: after P steps every satellite is back at its starting position.
- Cameras and comms are attached to **positions in the grid**, so which physical satellite can see the ROI I or the ground station G changes with t.

Planning task (SAS^+)

$\Pi = \langle \mathcal{V}, \mathcal{O}, I, G \rangle$: classical planning task with full observability, deterministic actions

- $\mathcal{V}$: finite-domain state variables
- $\mathcal{O}$: actions with preconditions and effects
- I: initial state
- G: goal

- A **plan** is a sequence of actions leading from I to a state satisfying G.
- Given a horizon h, $Plans_h(\Pi)$ is the set of plans of length at most h.

Bounded-horizon classical planning

Planning task (SAS$^+$)

$\Pi = \langle \mathcal{V}, \mathcal{O}, I, G \rangle$: classical planning task with full observability, deterministic actions

- $\mathcal{V}$: finite-domain state variables
- $\mathcal{O}$: actions with preconditions and effects
- I: initial state
- G: goal

- A **plan** is a sequence of actions leading from I to a state satisfying G.
- Given a horizon h, $Plans_h(\Pi)$ is the set of plans of length at most h.

A traditional planner returns **one** plan out of $Plans_h(\Pi)$.

What if we could *navigate* the whole plan space instead of committing to a single plan?

Facet [Gnad et al., AAAI 2025]

A pair (a, t) is a **facet** if some plan in $Plans_h(\Pi)$ applies action a at step t, and some plan does not.

- A facet is a meaningful **unit of variability** of the plan space
- **Enforcing** (a, t): keep only plans where a occurs at step t
- **Forbidding** (a, t): keep only plans where it does not
- Either way, at least one plan always remains
- A **route** records the user's navigation history; it can be undone or extended at any point

Facet reasoning

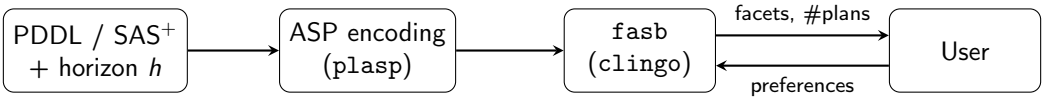
Facet [Gnad et al., AAAI 2025]

A pair (a, t) is a **facet** if some plan in $Plans_h(\Pi)$ applies action a at step t, and some plan does not.

- A facet is a meaningful **unit of variability** of the plan space
- **Enforcing** (a, t): keep only plans where a occurs at step t
- **Forbidding** (a, t): keep only plans where it does not
- Either way, at least one plan always remains
- A **route** records the user's navigation history; it can be undone or extended at any point

Reasoning over facets is significantly **cheaper** than reasoning over plans [Speck et al., AAAI 2025] $\Rightarrow$ responsive interaction even in huge plan spaces.

PlanPilot [Gnad et al., KR 2025]: interactive plan-space navigation.



At each interaction step, the user can:

- query the set of **available facets**,
- count or enumerate the **remaining plans**,
- **activate** a facet (restricts the plan space) or **deactivate** one (rolls back).

PDDL model

Two types (*satellite*, *image*), six predicates, three actions.

Dynamic predicates

- *has-image*(s, i)
- *is-full*(s)
- *transmitted*(i)

Static predicates

- *connected*(s_1, s_2)
- *can-take-image*(s, i)
- *can-earth-transmit*(s)

- *take-picture*(s, i): requires the ability + empty memory.
- *transmit-satellite*(s_1, s_2, i): forward to a connected neighbour with free memory.
- *transmit-earth*(s, i): downlink when in sight of a ground station.

Goal: *transmitted*(i) for every image i.

Where did time go?

The PDDL domain contains **no durative actions and no time literals**.

- The instance file specifies, for each image, the (satellite, offset) pairs at which a picture may be taken, and for each satellite the offsets at which it can downlink.
- Offsets are interpreted **modulo the orbital period** P: windows recur every P steps.
- The predicates *can-take-image* and *can-earth-transmit* are **scheduled outside PDDL**, directly in the ASP encoding.

Where did time go?

The PDDL domain contains **no durative actions and no time literals**.

- The instance file specifies, for each image, the (satellite, offset) pairs at which a picture may be taken, and for each satellite the offsets at which it can downlink.
- Offsets are interpreted **modulo the orbital period** P: windows recur every P steps.
- The predicates *can-take-image* and *can-earth-transmit* are **scheduled outside PDDL**, directly in the ASP encoding.

Why this separation?

- The PDDL domain stays compact and reusable across instances.
- All temporal reasoning lives on the layer where PlanPilot operates
  $\Rightarrow$ temporal constraints become **facets** the user can manipulate.

Extending PlanPilot's encoding

Stock encoding: exactly **one action per time step**

1. Parallel actions ($\forall$-step semantics)

- Distinct satellites may act **in the same orbital step** (one takes a picture while another forwards an image).
- A set of actions may fire jointly at step t if every linearisation is applicable and yields the same state.
- Keeps the horizon in *orbital* steps, not action steps.

2. Facets over the schedule

- Fluent predicates are driven by the periodic schedule inside the ASP program.
- Facets range over parallel occurrences **and over the predicate schedule** $\Rightarrow$ users can enforce or forbid temporal constraints on observation windows.

Benchmarks

Three benchmark sets, six instances each (`prob-`m`-`L`-`n):

Set 1: Expanding images

Fixed constellation ($L = 3$, $m = 8$, $P = 8$), number of images $n \in \{2, \ldots, 7\}$.

Set 2: Infrared images

Same constellation; images split into **optical** and **infrared** categories, each acquirable only by a satellite with the matching sensor.

Set 3: Expanding satellites

Fixed $n = 2$ images, $L = 3$; satellites per layer $m \in \{9, \ldots, 14\}$, period $P = m$.

Setup: horizon fixed to the optimal h^*; 30 min / 4 GiB per run; first solution returned.

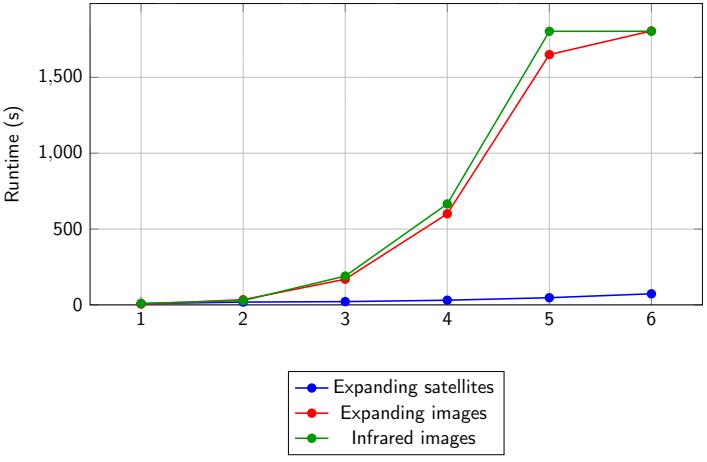
Results

Benchmark set	Instance	$\lvert\pi^*\rvert$	Solved	Time (s)
Expanding image	prob-8-3-2	17	✓	5.21
	prob-8-3-3	25	✓	33.88
	prob-8-3-4	33	✓	169.06
	prob-8-3-5	41	✓	600.44
	prob-8-3-6	49	✓	1649.99
	prob-8-3-7	57	×	–
Infrared image	prob-8-3-2-inf	17	✓	9.09
	prob-8-3-3-inf	25	✓	28.56
	prob-8-3-4-inf	33	✓	189.73
	prob-8-3-5-inf	41	✓	665.66
	prob-8-3-6-inf	49	×	–
	prob-8-3-7-inf	57	×	–
Expanding satellites	prob-9-3-2	19	✓	7.95
	prob-10-3-2	21	✓	18.31
	prob-11-3-2	23	✓	21.41
	prob-12-3-2	25	✓	30.66
	prob-13-3-2	27	✓	47.19
	prob-14-3-2	29	✓	72.98

- **Expanding images:** solved up to $n = 6$; runtime grows steeply with n.
- **Infrared:** sensor constraint makes it harder (first failure at $n = 5$), but non-trivial instances remain in reach.
- **Expanding satellites:** *all* instances solved; runtime grows only mildly with m.

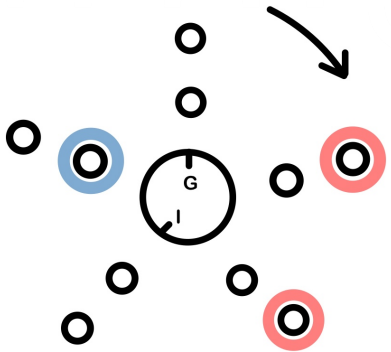
The number of **images** drives the difficulty, not the size of the constellation.

Runtime scaling



Runtime for each solved instance, ordered by increasing difficulty within each set.

Case study: instance prob-5-2-1



- $m = 5$, $L = 2$, $n = 1$, $P = 5$, $h^* = 7$
- *s2-l2*, *s3-l2* can acquire the image
- *s4-l1* is the only downlink

Ask for one plan (`! 1`):

```
! 1
solution 1:
occurs([take-pic,s3-l2,i1],2)
occurs([take-pic,s2-l2,i1],3)
occurs([transmit-sat,s2-l2,s1-l2,i1],4)
occurs([transmit-sat,s3-l2,s3-l1,i1],4)
occurs([transmit-sat,s1-l2,s5-l2,i1],5)
occurs([transmit-sat,s3-l1,s4-l1,i1],5)
occurs([transmit-sat,s5-l2,s5-l1,i1],6)
occurs([transmit-earth,s4-l1,i1],7)
occurs([transmit-sat,s5-l1,s1-l1,i1],7)
```

Both eligible satellites acquire the image; the copy taken by *s3-l2* reaches *s4-l1* and is downlinked at step 7.

Unexpected event 1: cloud cover

A cloud is predicted over the ROI at time step 2
$\Rightarrow$ forbid the corresponding facet and ask for a new solution:

```
+ ~occurs([take-pic,s3-l2,i1],2)
! 1
solution 1:
occurs([take-pic,s2-l2,i1],3)
occurs([transmit-sat,s2-l2,s3-l2,i1],4)
occurs([transmit-sat,s3-l2,s3-l1,i1],5)
occurs([transmit-sat,s3-l1,s4-l1,i1],6)
occurs([take-pic,s3-l2,i1],7)
occurs([transmit-earth,s4-l1,i1],7)
```

- *s2-l2* now acquires the image at step 3 instead.
- The rest of the plan is repaired around it — **no replanning from scratch**.
- The constraint only blocks step 2: *s3-l2* is still free to acquire later.

Unexpected event 2: maintenance

Satellite *s5-l2* is unavailable for communication at time step 5
$\Rightarrow$ forbid all communication facets involving it at that step:

```
+ ~occurs([transmit-sat,s4-l2,s5-l2,i1],5)
+ ~occurs([transmit-sat,s5-l2,s4-l2,i1],5)
+ ~occurs([transmit-sat,s5-l2,s1-l2,i1],5)
+ ~occurs([transmit-sat,s1-l2,s5-l2,i1],5)
+ ~occurs([transmit-sat,s5-l2,s5-l1,i1],5)
! 1
solution 1:
occurs([take-pic,s3-l2,i1],2)
occurs([take-pic,s2-l2,i1],3)
occurs([transmit-sat,s3-l2,s4-l2,i1],3)
...
occurs([transmit-earth,s4-l1,i1],7)
```

- The new plan routes the image around *s5-l2* at step 5.
- Goal still achieved within the horizon $h^* = 7$.

Weather, faults, maintenance: all expressed with the **same facet mechanism**.

Take-aways

Robust observation planning via facet reasoning

- Constellation observation planning encoded in PlanPilot: compact PDDL domain, temporal windows scheduled in the ASP layer.
- Extended encoding: **parallel actions** per orbital step, facets over the **schedule**.
- Scales with constellation size; the number of images is the bottleneck.
- Unexpected events (clouds, faults, maintenance) handled by **enforcing/forbidding facets**.

Future work

- Explicit constraints: observations at specific times, downlink deadlines.
- Richer satellite models: memory management, more sensor types.
- Dynamic constellation topologies via the same facet mechanism.
- Scaling to larger constellations.