Planning and Scheduling Activities for Earth Surveillance and Observation Satellites:

A Constraint-based Perspective

Gérard Verfaillie
ONERA, Toulouse, France

Gerard.Verfaillie@onera.fr

ONERA
THE FRENCH AEROSPACE LAB

ICAPS 2013 Summer School, Perugia, Italy, June 4-7, 2013

My research activity and technical background

ONERA - The French Aerospace Lab: the French **research center** in **aeronautics** and **space**.

DCSD department: **automatic control** of dynamic systems.

CD research unit: **high level** control of dynamic systems, control of **discrete event dynamic systems**.

Main application domain: control of **spacecraft** for **Earth surveillance** and **observation**.

Research topics: **planning** and **scheduling** problems, decision-making under **uncertainty**, **constrained optimization** problems, mainly over **discrete domains**.

Objective of the tutorial

To present a rich **real-world application** domain and the associated **planning problems**.

To discuss various **candidate technical approaches**.

To present what **we currently use** to deal with these problems.

$\rightarrow$ To show you that there is never a **unique technical approach** and that enough time must be dedicated to the **problem analysis** and to the choice of the **most suitable modeling** and **solving approach**.

Tutorial outline

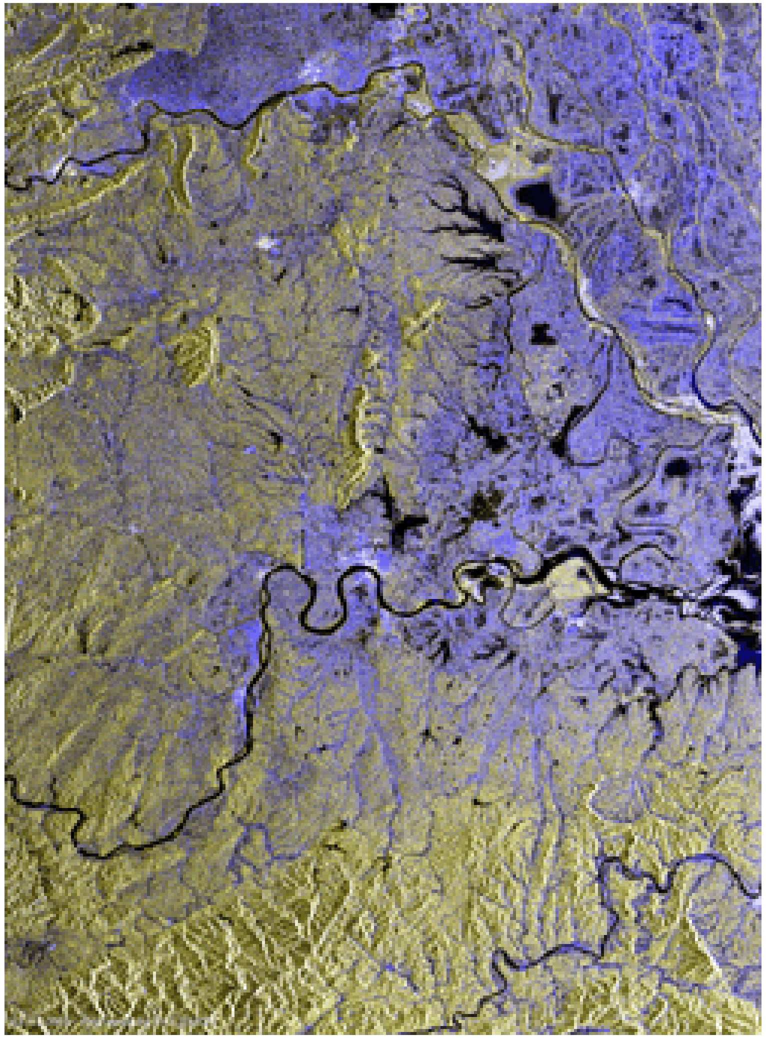
- Space **missions**;
- Space/Ground **systems**;
- Management **modes**;
- Planning **problems**;
- Problem **models**;
- Search **algorithms**;
- Our **current approach**.

Space Missions

Imaging of ground areas

Using either **optical** or **radar** instruments.

Canal Grande in Venice (US Ikonos satellite).



Floods in China (ESA Envisat satellite).

Surveillance and observation of ground phenomena

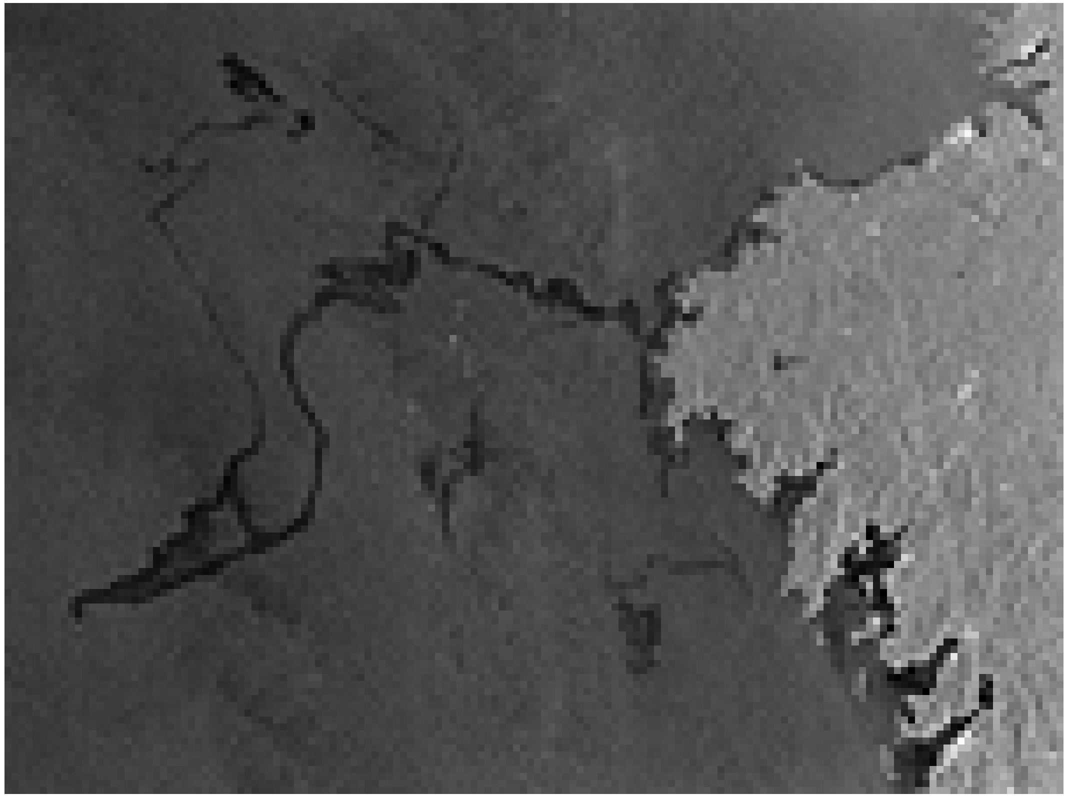
Onboard **detection**, **image analysis**, and **observation decision**.

Forest fires in Amazonia
(ESA Envisat satellite).

Eruption of Etna in Sicily
(ESA Envisat satellite).

Global ocean surveillance

Surveillance of **ship movements**.



Wreck of the Prestige tanker in front of Galicia in Spain in 2002
(ESA Envisat satellite).

Electromagnetic surveillance

Detection, localization, and identification of **ground electromagnetic sources**.

Formation of the Elisa satellites (French Ministry of Defence).

Space/Ground Systems

Space systems

- individual **satellites**;
- **constellations** of satellites, to better cover the Earth surface;
  example: **Pleiades** constellation;
- **formations** of satellites, to perform a joint task that requires inter-satellite distance constraints to be satisfied;
  example: **electromagnetic surveillance** (localization).

For the sequel of this tutorial, focus on **individual** observation satellites.

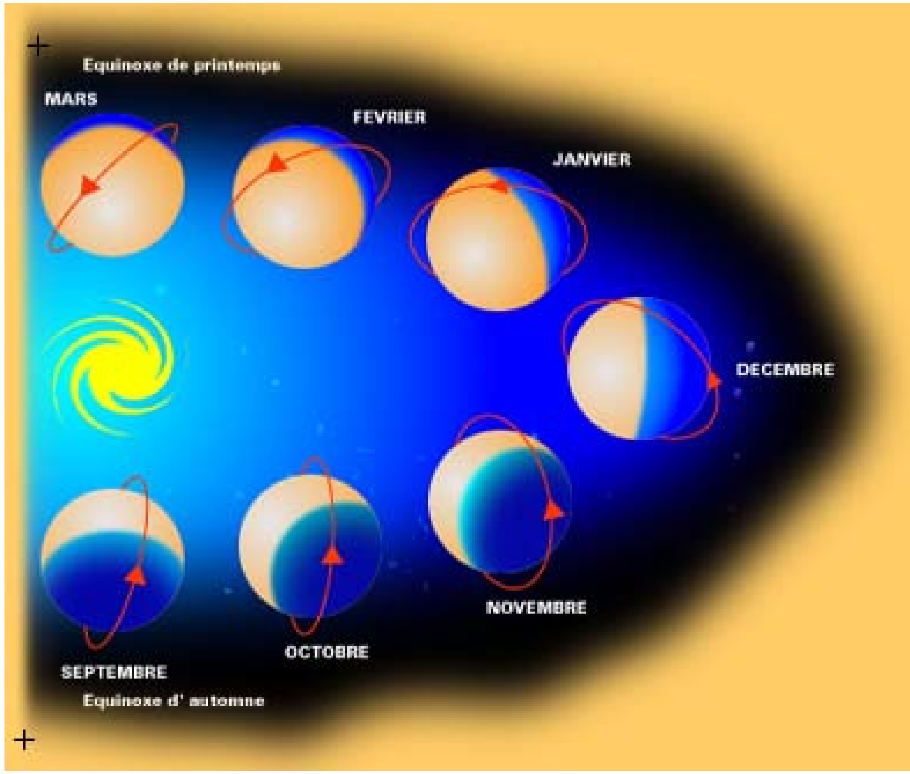
Orbits (1)

Usually, **circular**, **quasi-polar**, **heliosynchronous** orbit.

Low altitude: some hundreds of kilometers ($\approx$ 700 kms).

Alternation of **day** and **night periods**: a dozen of revolutions per day ($\approx$ 14).

Always the **same local hour** when flying over the equator ($\approx$ 10:30).

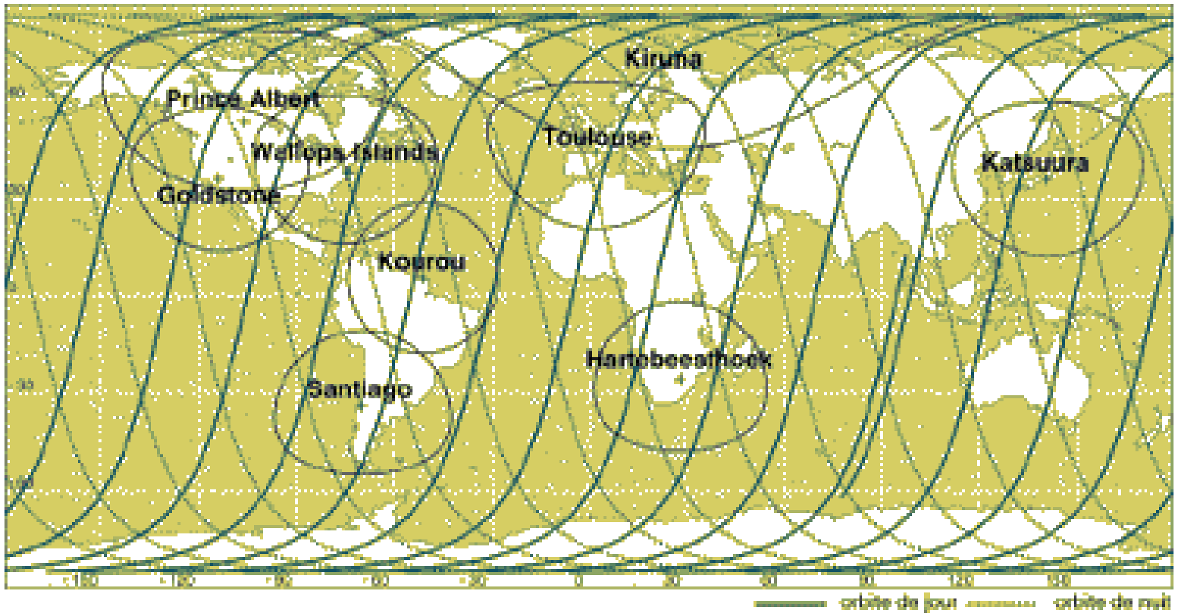


Orbits (2)

Track of the satellite **over one day**, because of the **rotation** of the Earth.

Distance between **two successive tracks** at the equator: some thousands of kilometers ($\approx$ 2500 kms).

Phased orbit with a cycle of some tens of days ($\approx$ 26 days).



Platforms

Orbit control: satellite kept on a **reference orbit**; regular **orbital manoeuvres** using **ergol thrusters**.

Attitude control: see further.

Energy: **production** using **solar panels** only during **day periods** and **storage** using **batteries**.

Space-ground communication: using **low-rate antennas** only when the satellite is in **visibility** of a **control ground station**: around 10% of time.

Payloads

Observation: one or several instruments.

Optical, **infra-red**, or **radar**, possibly **multi-spectral**.

Various **resolutions**, up to sub-metric ones.

Various **swaths** on the ground, from some hundreds of kilometers until some kilometers.

Data recording: using a **mass memory**.

Data download: using **high-rate antennas** when the satellite is in **visibility** of a **reception ground station**.

Non agile (optical) satellites (1)

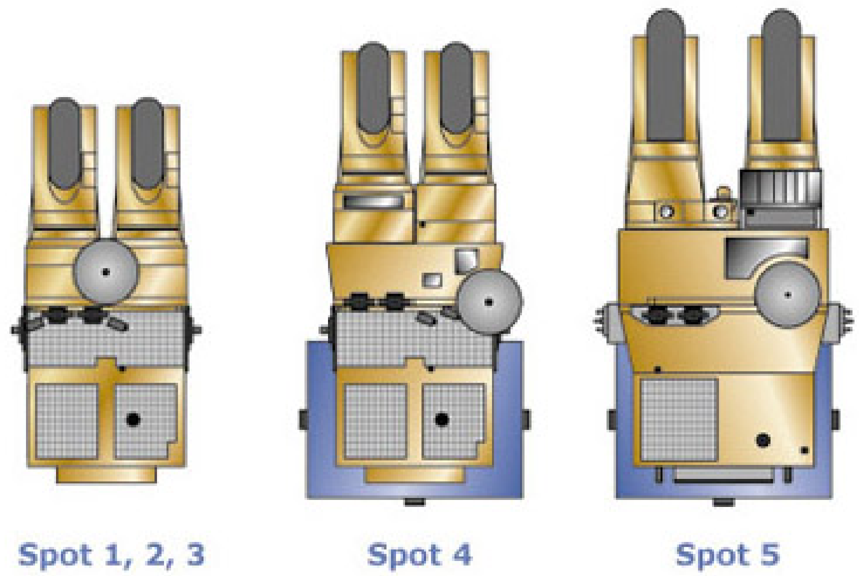
Example: the French **SPOT** family.

Non agile (optical) satellites (2)

Satellite kept in a **geocentric attitude**.

One degree of freedom in terms of observation via a **mobile mirror** in front of each instrument.

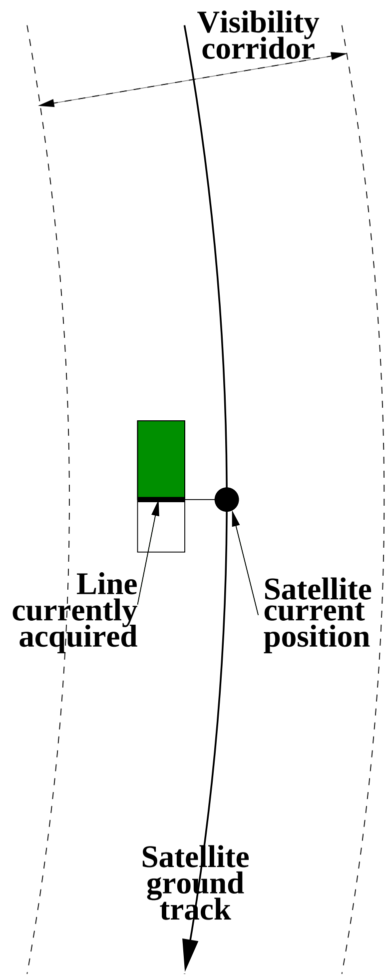
The Spot satellites seen from below:



Non agile (optical) satellites (3)

Observation process:

- instantaneous acquisition of a **line** on the ground via an array of aligned photo-diodes;
- acquisition of a **rectangle** on the ground via the movement of the satellite.

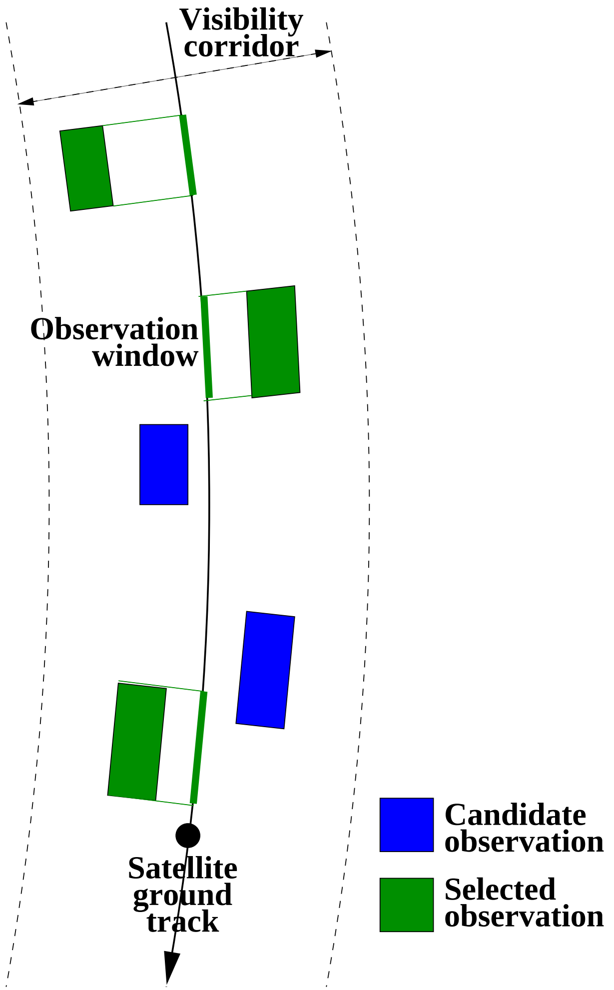


Non agile (optical) satellites (4)

Possible **sequence of observations** →

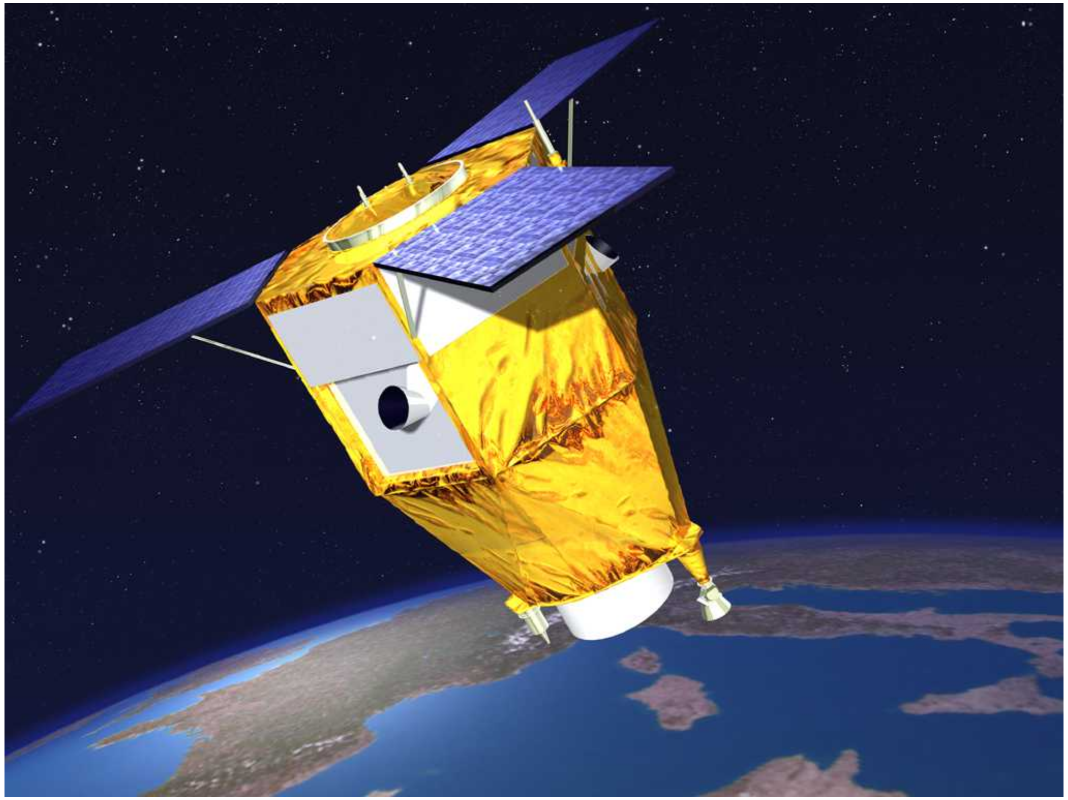
Fixed observation window
associated with each candidate observation.

Incompatibilities between observations
due to an **overlapping** between observation windows
or to an **insufficient transition time**
to allow the required mirror movement.



Agile (optical) satellites (1)

Examples: the US **Ikonos** satellite, the French **Pleiades** satellites.



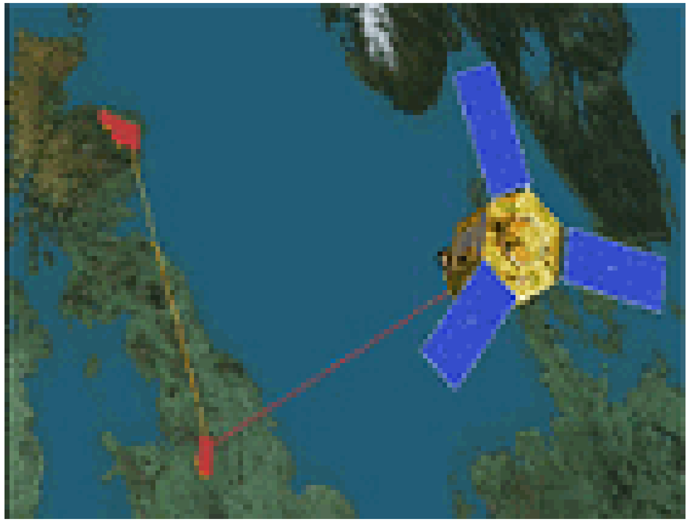
Agile (optical) satellites (2)

Permanent control of the satellite **attitude** along the **three axes** (roll, pitch, and yaw) via **gyroscopic actuators**.

Movement of the **whole satellite**. **No mobile part**.

Allows the satellite to perform **observations** and **transitions** between observations.

See **video simulation** →



Agile (optical) satellites (3)

Possible **sequence of observations** →

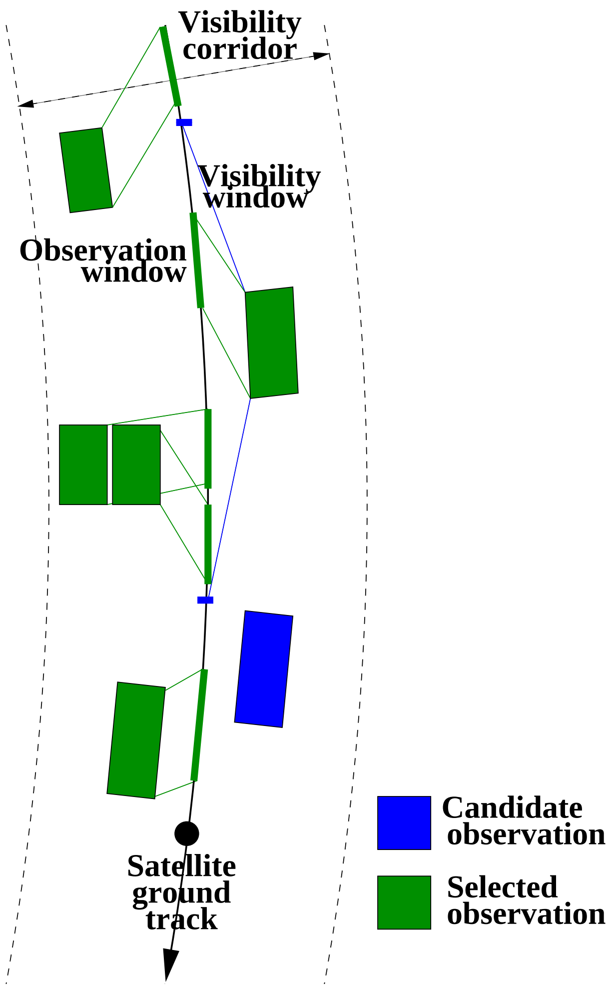
Visibility window
associated with each candidate observation.

Variable observation window
within this visibility window.

More **degrees of freedom** in terms of observation.

More observation sequence **opportunities**.

Allows the satellite to perform
stereo or **multi-stereo** observations.



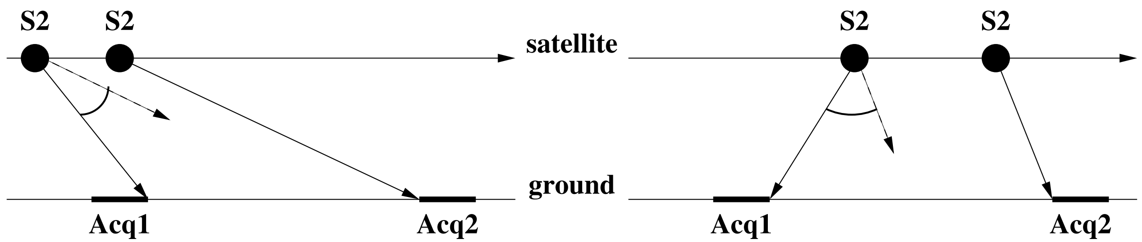
Agile (optical) satellites (4)

Several **difficulties** to be managed:

- the **transition time** between two successive observations depends on the time at which the transition is triggered; its computation requires a complex **continuous optimization** problem to be solved;

- because the antennas are body-mounted on the satellite, windows over which **downloading** is **effectively possible** depend on the **satellite attitude trajectory**;

- because the solar panels are also body-mounted on the satellite, **energy production** depends on the **satellite attitude trajectory**.



Ground components

- **control** stations, to send orders and receive health monitoring information);
- **reception** stations, to receive observation data;
- **processing** centers, to process data and deliver images.

Management Modes

Several possible management modes

- activity plans built **offline on the ground**;
- activities decided **online on board**;
- possible **combinations**; example: observation plans built **on the ground** and download decisions made **on board**.

Offline on the ground

The **most common** management mode.

Observation and download plans built **each day** for the next day, as a function of the **satellite state** and the current set of **observation requests**.

Fully instantiated plans (without any temporal flexibility) sent to the satellites and executed as they are.

$\rightarrow$ **Deterministic planning problems** over a given temporal horizon.

Online on board (1)

Still **prospective**.

Some information may be only **available online on board**. Examples:

- **areas of interest** (forest fires, volcanic eruptions, presence of electromagnetic sources);
- **cloud cover**, important in case of optical observation;
- **data size** in memory after compression.

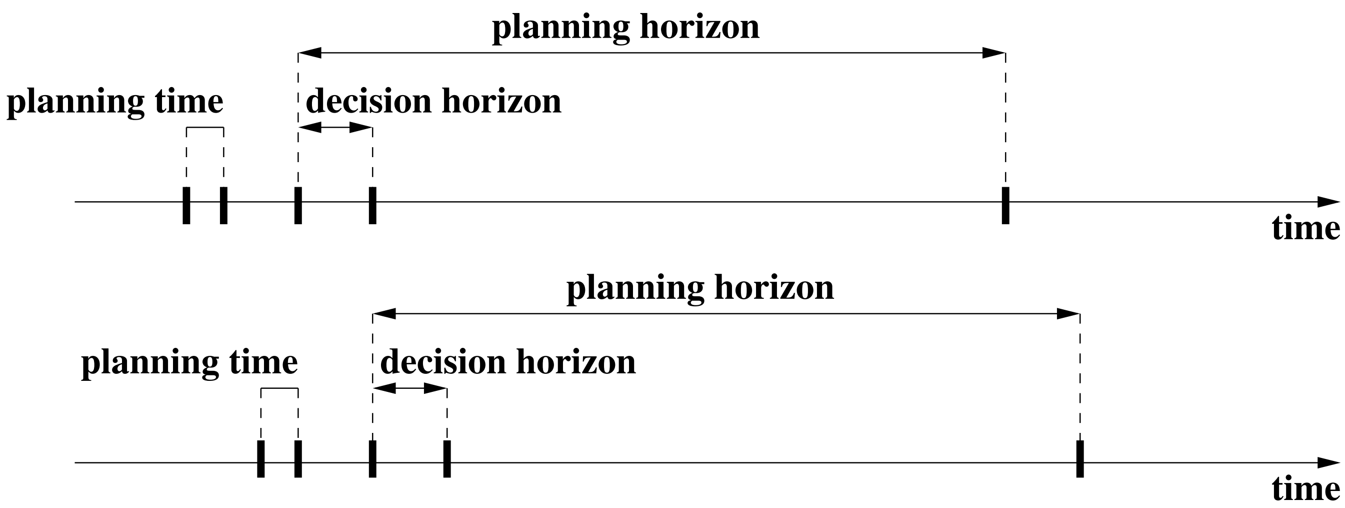
The satellite is **not permanently** within the visibility of a ground control station.

$\rightarrow$ **Online onboard decision-making** is necessary.

Online on board (2)

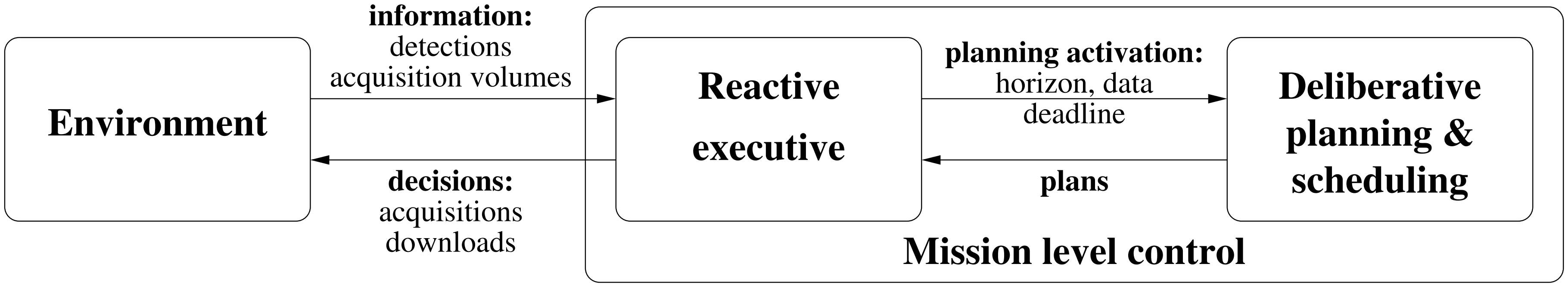
Online planning approach:

- **deterministic planning** over a given **temporal horizon ahead**;
- **execution** of the first activity(ies) in the plan;
- **(re)planning** over a **sliding temporal horizon**.



Online on board (3)

Need for a **reactive/deliberative** architecture for decision-making.



Need for:

- **anytime** planning algorithms, taking into account **embarquability** constraints: **limited CPU** and **RAM**, **static memory allocation** . . .
- **default** decision rules, when no plan has been produced.

Interaction between reactive and deliberative tasks for on-line decision-making, M. Lemaître and G. Verfaillie, ICAPS-07 Workshop on "Planning and Plan Execution for Real-world Systems"

Other possible management modes

Not considered yet.

Activity **policies** built **offline on the ground**.

MDP-like approach (Markov Decision Processes).

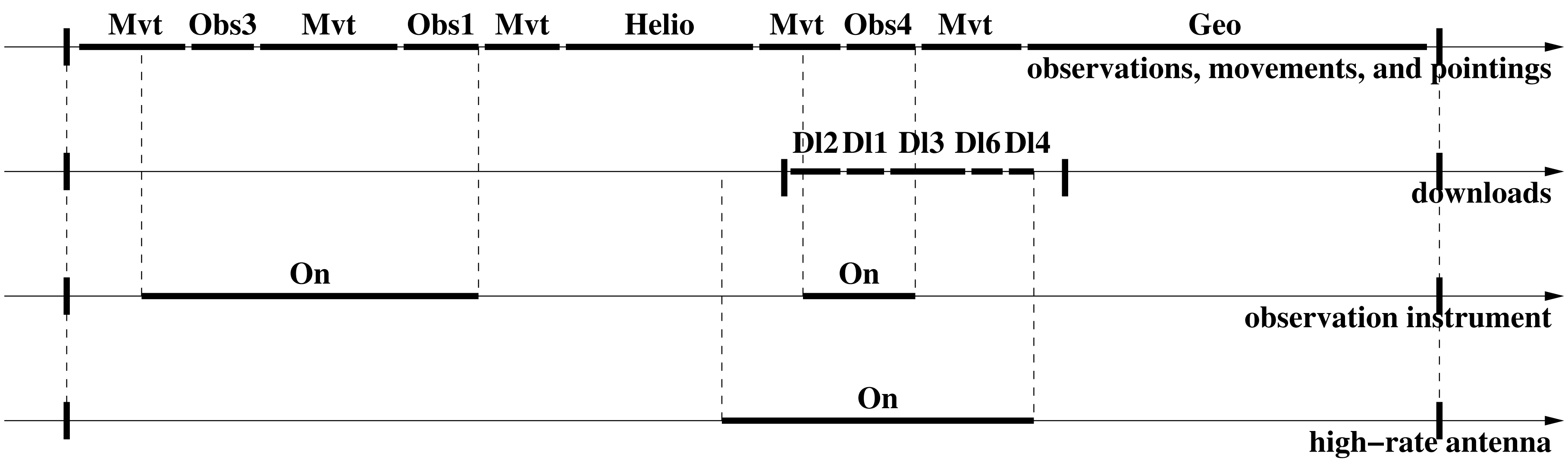
Main obstacle: **high-dimensional hybrid** (discrete/continuous) **state space** to be considered.

Planning Problems

Decisions to be made

Observation and **download** activities: which ones, on which satellite, and when?

Support activities: mirror or satellite attitude **movements**, geocentric or heliocentric satellite **pointings**, instrument **On/Offs** . . .



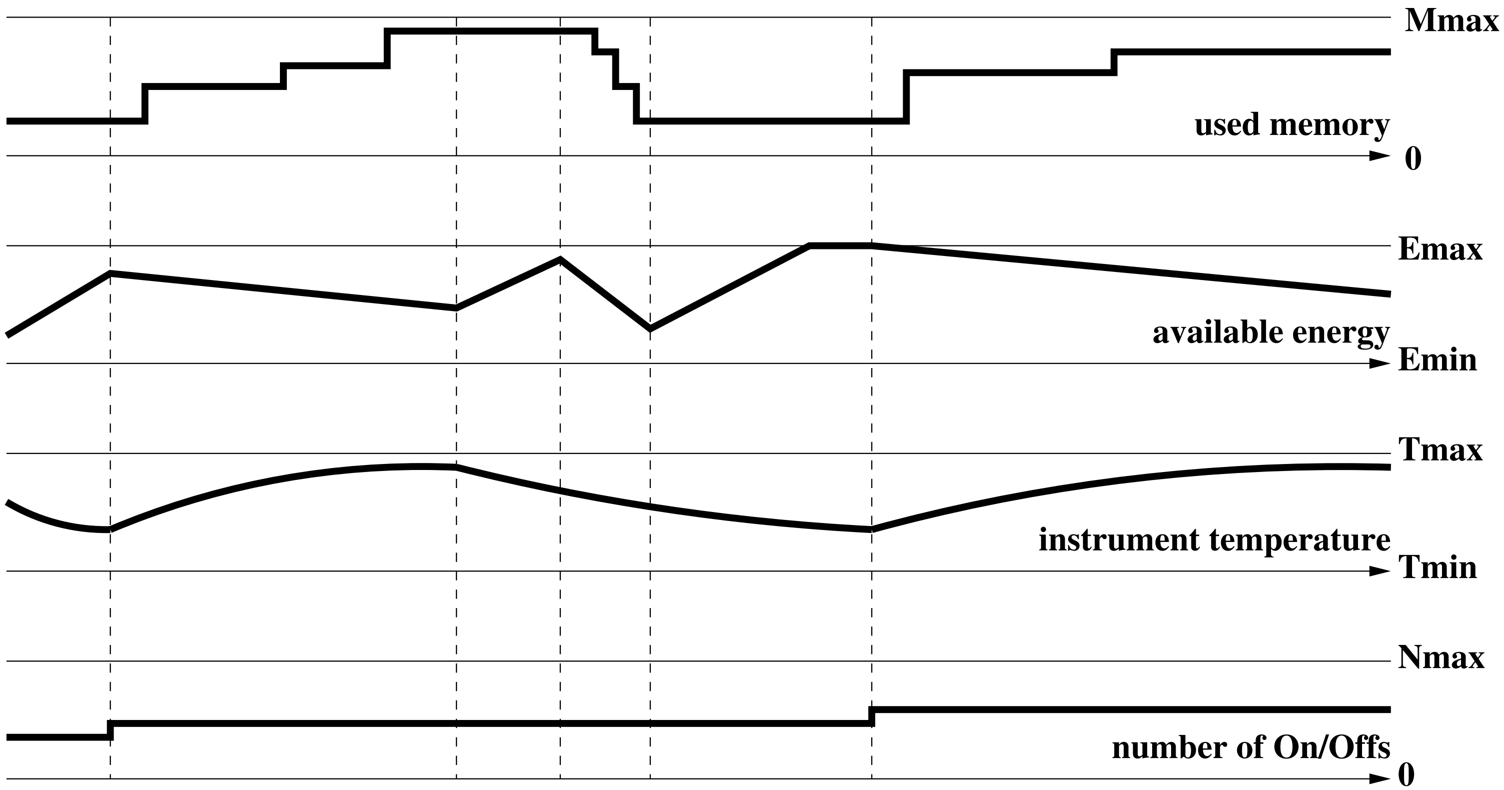
Constraints to be satisfied (1)

Temporal constraints:

- **visibility windows** for observations and downloads;
- **durations** of observations and downloads;
- **no overlapping** between observations and between downloads;
- (time-dependent) **transition times** between observations;
- **precedences**: observation precedes download.

Constraints to be satisfied (2)

Resource constraints: **memory**, **energy**, instrument **temperature**, total **On time**, total **number of On/Offs** . . .



Constraints to be satisfied (3)

Logical activation constraints:

- **download** requires **observation** (before);
- **observation** requires **instrument ON** (during);
- **stereo** observations (either both or none).

Criterion to be optimized

Plan features to be taken into account and combined:

- **priority/weight** of the selected observations and downloads;
- observation **probability** of **success** (possible presence of clouds, weather forecast);
- observation **quality** (observation angle);
- information **age** (temporal distance between observation and download);
- **fairness** between entities, owners of the space system;
- . . .

Instance size

Each day:

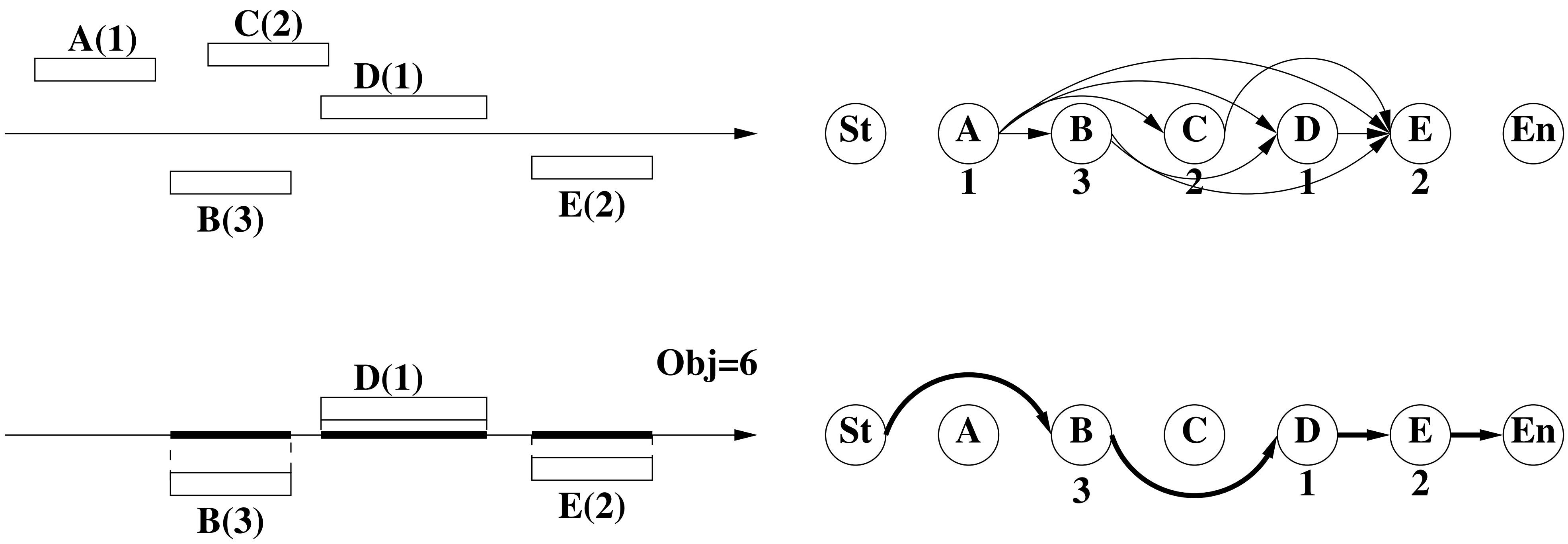
- some **hundreds** of **pending** user **requests**;
- some **thousands** of **candidate observations**;
- some **hundreds** of **observations performed** and **downloaded**.

Problem Models

Longest Path

Selection of **observations** for a **non agile** satellite

→ **Longest path** in a **weighted oriented graph** (from St to En).

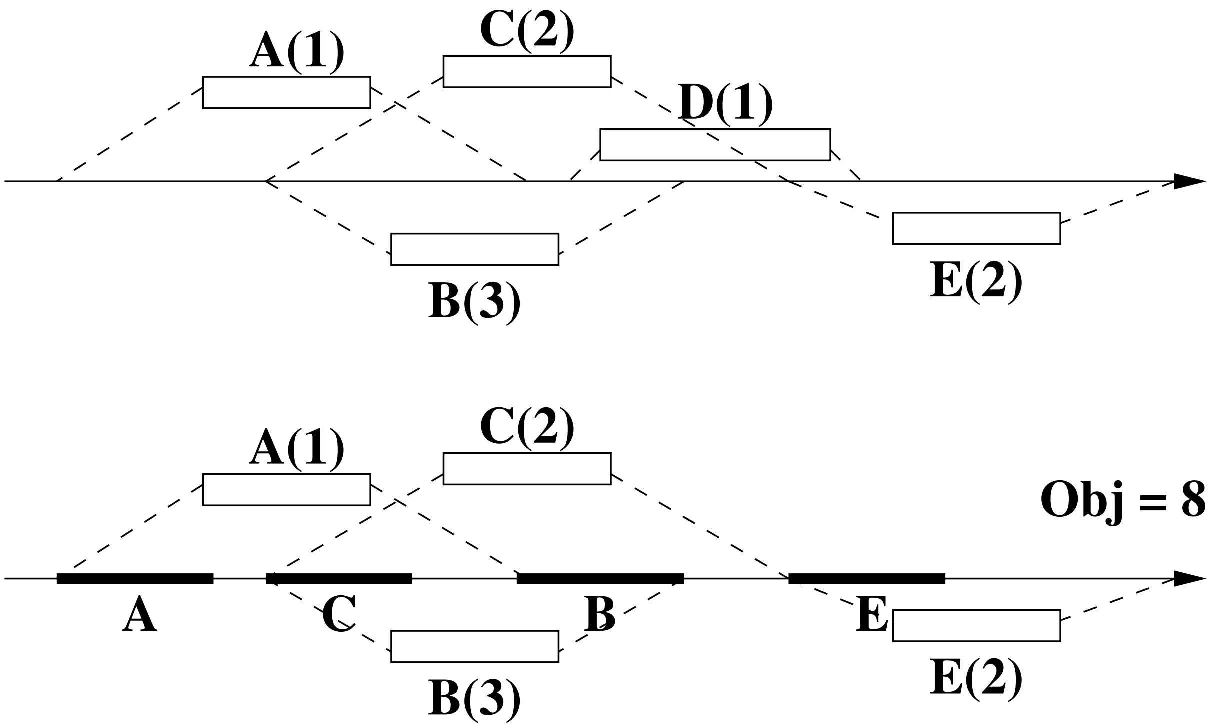


Polynomial problem.

Jobshop Scheduling

Selection and **ordering** of **observations** for an **agile** satellite

→ **Over-constrained jobshop scheduling** problem with **sequence** and **time-dependent transition times**.

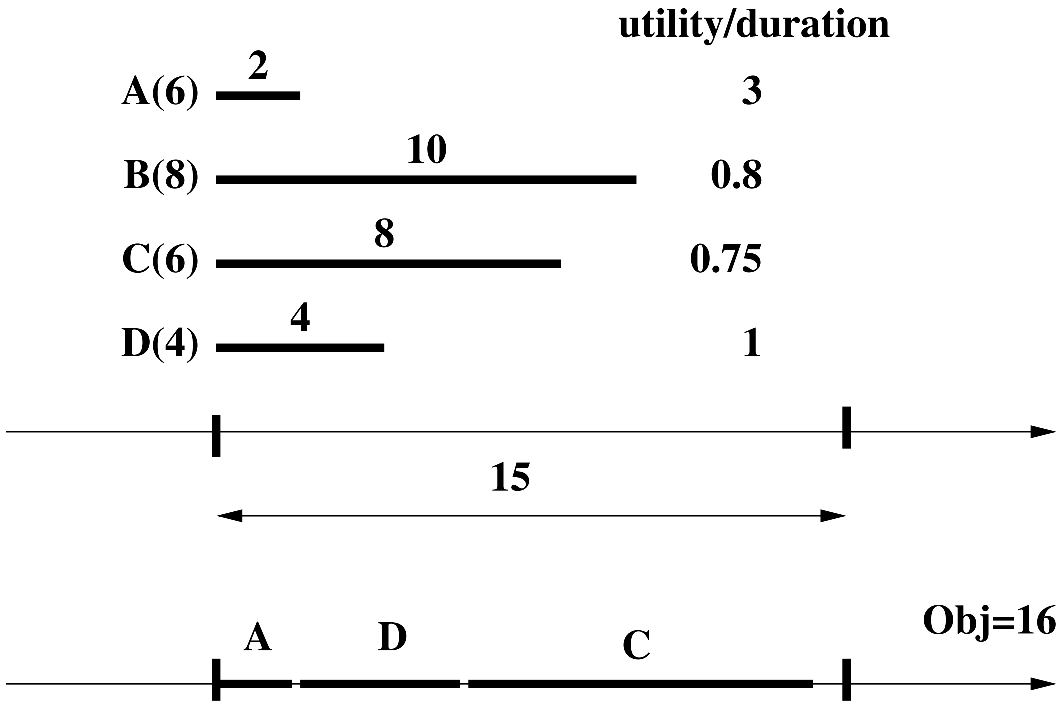


NP-hard problem.

Knapsack Problem

Selection of the **data** to be **downloaded** within a ground reception station **visibility window**

→ **Knapsack** problem.



NP-hard problem, but very **powerful heuristics**: data inserted according to a decreasing ratio **utility/duration**.

Integer Linear Programming

Selection of **observations** for a **non agile** satellite,
with **energy** and **memory** limitations

$\rightarrow$ **Integer linear programming** model.

$$\textit{maximize} \sum_{i=1}^{n} V_i \cdot x_i$$

$$\textit{subject to} \quad \sum_{i=1}^{n} M_i \cdot x_i \leq Mmax$$

$$\sum_{i=1}^{n} E_i \cdot x_i \leq Emax$$

$$\forall i, j \in [1..n] \,|\, (i < j) \wedge (I[i,j] = 1),\ x_i + x_j \leq 1$$

NP-hard problem, but very powerful **generic tools**.

Constraint Programming

Selection and **ordering** of **observations** for an **agile** satellite, with **energy** and **memory** limitations

$\rightarrow$ **Constraint programming** model.

maximize $\sum_{i=1}^{n} V_i \cdot x_i$

subject to $\sum_{i=1}^{n} M_i \cdot x_i \leq Mmax$

$$\sum_{i=1}^{n} E_i \cdot x_i \leq Emax$$

$$\forall i \in [1..n],\ (x_i = 1) \rightarrow (SD_i \leq d_i \leq (ED_i - D_i))$$

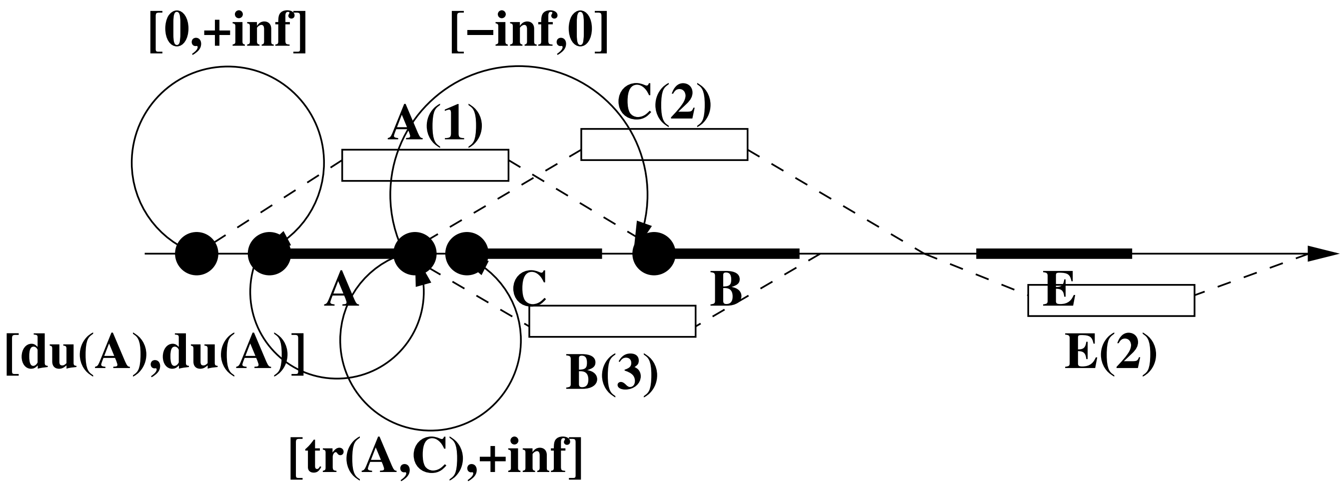
$$\forall i, j \in [1..n] \,|\, i < j,\ (x_i = x_j = 1) \rightarrow ((d_i + D_i + T_{ij} \leq d_j) \vee (d_j + D_j + T_{ji} \leq d_i))$$

NP-hard problem + time **discretization**, but powerful **generic tools**.

(Simple) Temporal Networks

Temporal constraints resulting from a **selection** and an **ordering** of **observations** for an **agile satellite**

$\rightarrow$ **Simple temporal networks** ($x - y \in [a; b]$ constraints).



Temporal **consistency/inconsistency** and computation of the **earliest** and **latest** times for starting and ending times of observations: **polynomial** problems.

$\rightarrow$ Temporal **flexibility**, without time discretization.

Artificial Intelligence Planning

Selection, **assignment**, and **ordering** of the **data** to be **downloaded** $\rightarrow$ **AI planning**.

State variables:

dl: **download in progress**;
$dlable_i$: **data** i **downloadable**;
vis_w: **whithin window** w;
mm: **available memory**.

Action models:

stDl$(i \in \mathbf{Im}, w \in \mathbf{Wd})$:
pre: $\neg dl, vis_w, dlable_i$
post: $dl', \neg dlable'_i$

enDl$(i \in \mathbf{Im}, w \in \mathbf{Wd})$:
pre: dl, vis_w
post: $\neg dl', mm' = mm + M_i$

Longest simple path in an **implicit weighted oriented graph** whose size is an **exponential** function of the number of **state variables**.

Markov Decision Processes

Selection and **ordering** of the **observations** to be performed, with uncertainty about the data volumes

$\rightarrow$ **Markov decision processes**.

State variables:

ob: **observation in progress**;
$dlable_i$: **data i downloadable**;
mm: **available memory**;

Action models:

stOb$(i \in \mathbf{Im})$:
pre: $\neg ob$
post: ob'

enOb$(i \in \mathbf{Im})$:
pre: ob
post: $\neg ob', dlable'_i, mm' = mm - RM_i$

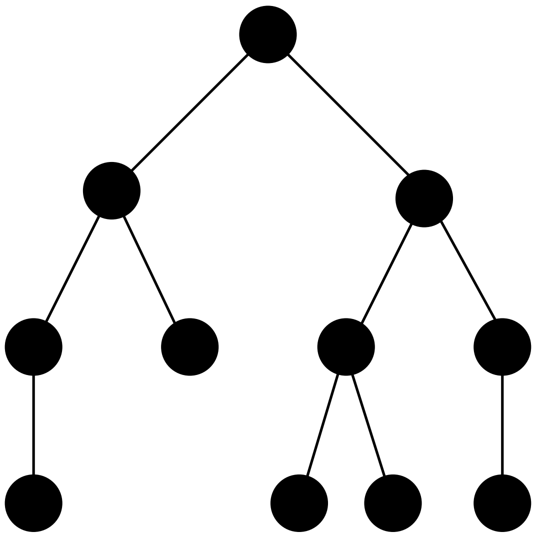
Longest stochastic path, *i.e.* **policy** that maximizes the **expected gain**, in an **implicit weighted oriented graph** whose size is an **exponential** function of the number of **state variables**.

Search Algorithms

Complete optimal search algorithms

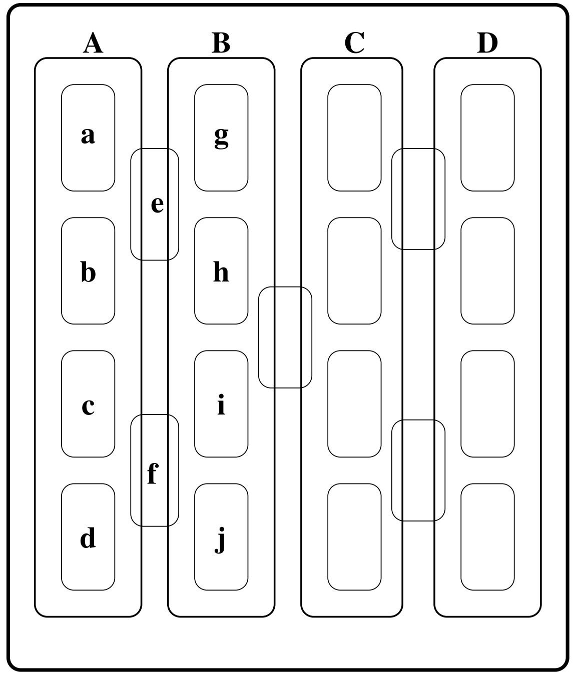
Tree search:
Best First Branch and Bound,
Depth First Backtrack

Recursively **branching** (making choices)
and **reasoning** on subproblems
(bound computing, inconsistency detection).



Dynamic programming:

Recursively solving **problems**
by solving **subproblems**
and aggregating results.

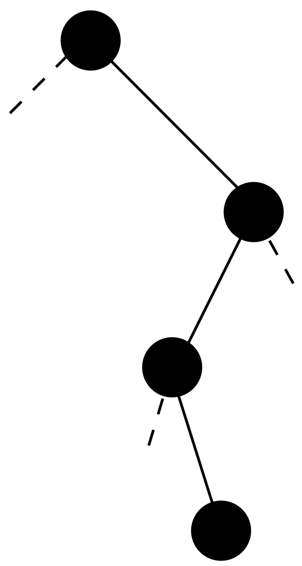


Incomplete approximate search algorithms

Greedy search:

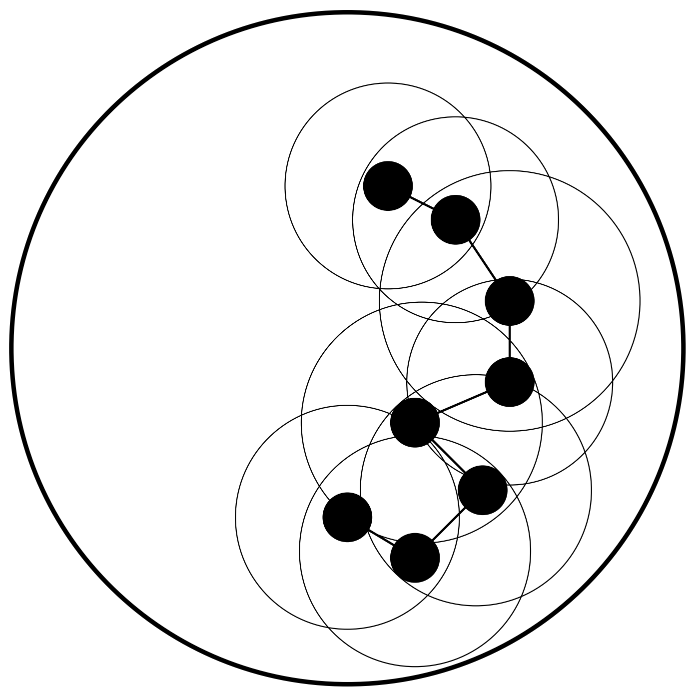
Making successive **heuristic choices**.

Tree search **without backtrack**.



Local search:
Tabu search,
Simulated annealing

Iteratively **locally modifying** solutions.



Our Current Approach:
TECK modeling framework
InCELL solving library

The TECK modeling framework (1)

TECK for **Timeline**, **Event**, and **Constraint**-based **Knowledge**.

Timelines: **evolutions** of **state** variables and **resource** levels.

Events: **impact** timelines.

Constraints: **on** and **between** events.

Target: **deterministic** planning and scheduling problems.

The TECK modeling framework (2)

Motivation: observation that:

- most of the real-world planning and scheduling problems involve a **predefined finite set** (bounded set) of **events**;

- main question: to **assign values** to **event parameters**:
  - **presence** in the plan;
  - **position** in the event sequence;
  - precise **date**;
  - **specific** type-dependent **parameters**;

  by **satisfying** a finite set of **constraints** on these parameters and **optimizing** some **function** of some of these parameters.

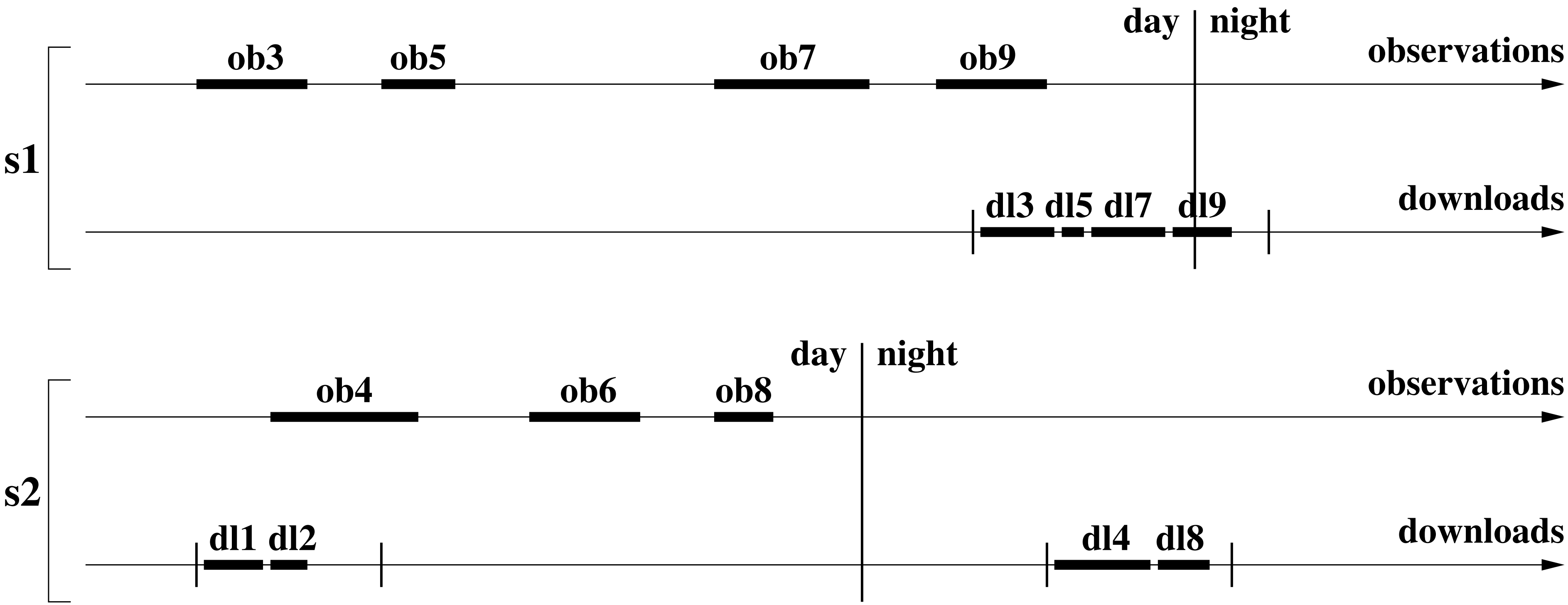
The TECK modeling framework (3)

Four kinds of **knowledge** to be expressed and managed:

- about **events**: **presences** and **parameters** of events;
- about **time**: **positions** and **dates** of events;
- about **state**: discrete **evolutions** of **state** variables;
- about **resources**: discrete or continuous **evolutions** of **resources**.

A timeline, event, and constraint-based modeling framework for planning and scheduling problems,
G. Verfaillie and C. Pralet, ICAPS-13 KEPS Workshop

Planning observations and downloads for a constellation of non agile satellites



TECK planning domain definition

- a finite set **Vs** of **static variables**;
- a finite set **Vd** of **dynamic variables**;
- a finite set **De** of **dependencies** between dynamic variables;
- a finite set **Et** of **event types**.

Static variables

Planning domain parameters that **do not evolve** due to successive **events**.

For each candidate **image** $i \in \mathbf{Im}$:

- the **satellite** $os_i \in \mathbf{Sa} \cup \{0\}$ that performs i (0 when i is not performed);
- the **window** $dw_i \in (\cup_{s \in \mathbf{Sa}} \mathbf{Wd}_s) \cup \{0\}$ used to download i (0 when i is not downloaded).

Really **variables** that must be assigned by the planner.

Dynamic variables (1)

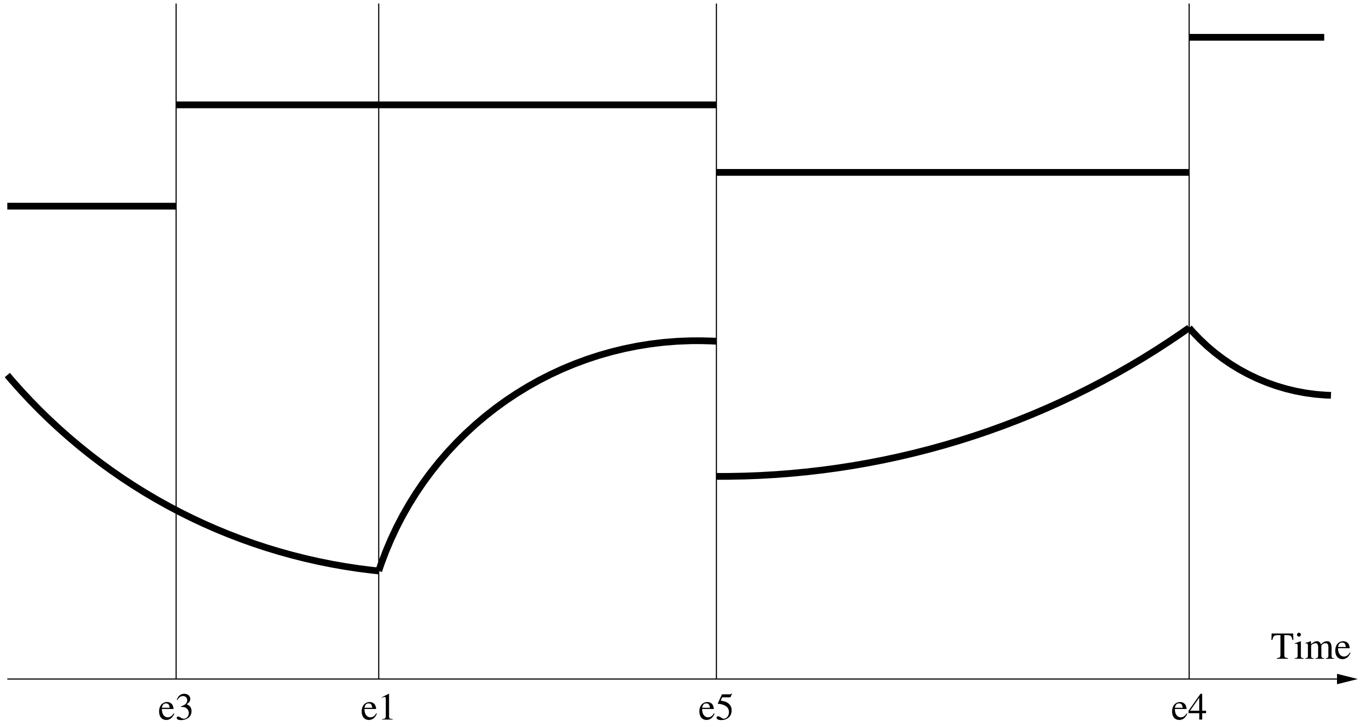
Planning domain parameters that **evolve** due to successive **events** (piecewise constant evolution or not; see the next slide).

For each **satellite** $s \in$ **Sa**:

- the fact $ob_s \in \mathbb{B}$ that s is currently **observing** or not (piecewise constant);
- the fact $dl_s \in \mathbb{B}$ that s is currently **downloading** or not (piecewise constant);
- the fact $su_s \in \mathbb{B}$ that s is currently in a **day** or **night** period (piecewise constant);
- the **memory** $mm_s \in [0; \mathbf{Mmax}]$ currently available on board s (piecewise constant);
- the **power** $pu_s \in [\mathbf{Pmin}; \mathbf{Pmax}]$ currently produced/consumed on board s (piecewise constant);
- the **energy** $en_s \in [\mathbf{Emin}; \mathbf{Emax}]$ currently available on board s (piecewise linear, with a saturation at the maximum level **Emax**).

Dynamic variables (2)

Piecewise constant and **non piecewise constant** evolutions.



Dependencies

The fact that some dynamic variables **functionally depend** on other static or dynamic variables.

For each **satellite** $s \in \mathbf{Sa}$, the **power** pu_s currently produced/consumed on board s depends on the status of s in terms of **observation**, **download**, and **day** or **night** (ob_s, dl_s, and su_s):

$$pu_s = \mathbf{Psu} \cdot su_s - \mathbf{Pob} \cdot ob_s - \mathbf{Pdl} \cdot dl_s - \mathbf{Psa}$$

Event types (1)

Six **event types** **StOb**, **EnOb**, **StDl**, **EnDl**, **StSu**, and **EnSu**.

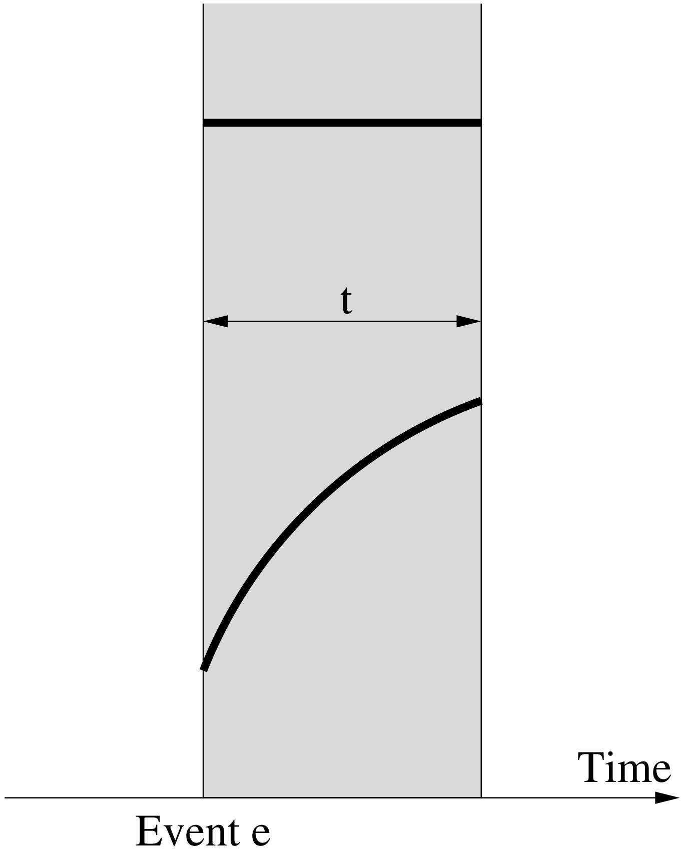
Example: event type **StOb** (**start** of **observation**):

- **parameters**:
  - **image** **im** $\in$ **Im**;
  - **satellite** **sa** $\in$ **Sa**;
- **preconditions**:
  - $ob_{\mathbf{sa}} = 0$
- **effects**:
  - $ob'_{\mathbf{sa}} = 1$
  - $mm'_{\mathbf{sa}} = mm_{\mathbf{sa}} - \mathbf{Mm_{im}}$
  - $en'_{\mathbf{sa}}(t) = min(\mathbf{Emax}, en_{\mathbf{sa}} + t \cdot pu'_{\mathbf{sa}})$ (see next slide)

Event types (2)

Piecewise constant and **non piecewise constant** evolutions.

Evolution after an **event**:



TECK planning problem definition

- a planning **domain** **Pd** $= \langle \mathbf{Vs}, \mathbf{Vd}, \mathbf{De}, \mathbf{Et} \rangle$;
- an **initial state** **I**;
- a **temporal horizon** **H**;
- a finite set **E** of **events**;
- a finite set **Cs** of **constraints** on **static variables**;
- a finite set **Ce** of **constraints** on **events**;
- a finite set **Cd** of **constraints** on **states**;
- a **criterion** **Cr** to be optimized.

Events

- **type** (constant);
- **presence** (variable);
- **position** in the event sequence (variable);
- **date** (variable);
- type-dependent **parameters** (variables).

- for each **satellite** $s \in$ **Sa** and each **day period** $w \in$ **Ws**$_s$, two events $stSu_w$ and $enSu_w$ of respective type **StSu** and **EnSu** (start and end of **day period**);
- for each candidate **image** $i \in$ **Im**, two events $stOb_i$ and $enOb_i$ of respective type **StOb** and **EnOb** (start and end of **observation**) two events $stDl_i$ and $enDl_i$ of respective type **StDl** and **EnDl** (start and end of **download**).

Static constraints

Constraints on **static variables** only.

Examples:

Download implies **observation**:

$$\forall i \in \mathbf{Im} : \ (dw_i \neq 0) \rightarrow (os_i \neq 0)$$

Observation and **download** on the same satellite:

$$\forall i \in \mathbf{Im} : \ (dw_i \neq 0) \rightarrow (dw_i \in \mathbf{Wd}_{os_i})$$

Capacity of download windows:

$$\forall s \in \mathbf{Sa}, \forall w \in \mathbf{Wd}_s : \ (\sum_{i \in \mathbf{Im}} \mathbf{Mm}_i \cdot (dw_i = w)) \leq (\mathbf{Et}_w - \mathbf{St}_w) \cdot \mathbf{Dr}$$

Constraints on events

Constraints on **event presences**, **positions**, **dates**, and **parameters**.

Examples:

End of observation iff **start** of observation:

$$\forall i \in \mathbf{Im} : \ \mathbf{pres}(stOb_i) = \mathbf{pres}(enOb_i)$$

Download within the **chosen window**:

$$\forall i \in \mathbf{Im} : (\mathbf{pres}(stDl_i) = 1) \rightarrow ((\mathbf{dat}(stDl_i) \geq \mathbf{St}_{dw_i}) \wedge (\mathbf{dat}(enDl_i) \leq \mathbf{Et}_{dw_i}))$$

Observation before **download**:

$$\forall i \in \mathbf{Im} : \ (\mathbf{pres}(stDl_i) = 1) \rightarrow (\mathbf{dat}(enOb_i) \leq \mathbf{dat}(stDl_i))$$

Parameters of events of **start** and **end** of observation:

$$\forall i \in \mathbf{Im} : (\mathbf{pres}(stOb_i) = 1) \rightarrow ((\mathbf{im}(stOb_i) = i) \wedge (\mathbf{sa}(stOb_i) = \mathbf{sa}(enOb_i) = os_i)$$

Constraints on states

No such constraint except the **domains** of the **state variables** which enforce for example that, on each satellite, the level of energy never goes below **Emin**.

Optimization criterion

Sum of the **weights** of the **downloaded images**:

$$\sum_{i \in \textbf{Im}} (dw_i \neq 0) \cdot \textbf{Wg}_i$$

The InCELL solving library (1)

Power of **local search** algorithms:

- good **anytime** behavior: **quick production** of good quality solutions and **opportunistic search** for improvements as long as computing time is available;
- in **dynamic** contexts (changes in the planning problem, as with **online planning**), ability to **restart** from the **previous solution** with the same algorithm ;
- widely used in the **space domain**; see for example the JPL **ASPEN** planner.

The InCELL solving library (2)

One of the **keys** to the success of local search:

- **number** of **local moves** performed within a limited time;
- **quick local moves**: to **choose** the move, to **evaluate** constraints and criterion, and to **execute** it in case of positive evaluation result.

→ use of so-called **invariants**, inspired from **Constraint-based Local Search** (Van Hentenryck and Michel, 2005), to quickly reevaluate constraints and criterion after any local move.

→ **InCELL** for **Invariant-based Constraint EvaLuation Library**.

Dynamic Online Planning and Scheduling Using a Static Invariant-based Evaluation Model, C. Pralet and G. Verfaillie, ICAPS-13

Invariants

To represent **expressions**.

Example: $x \leftarrow \sum_{i=1}^{N} y_i$

In case of **change** in the value of some y_j: $x' = x - y_j + y'_j$

In InCELL, **multiple-input multiple-output invariants** to represent **expressions** and **constraints**.

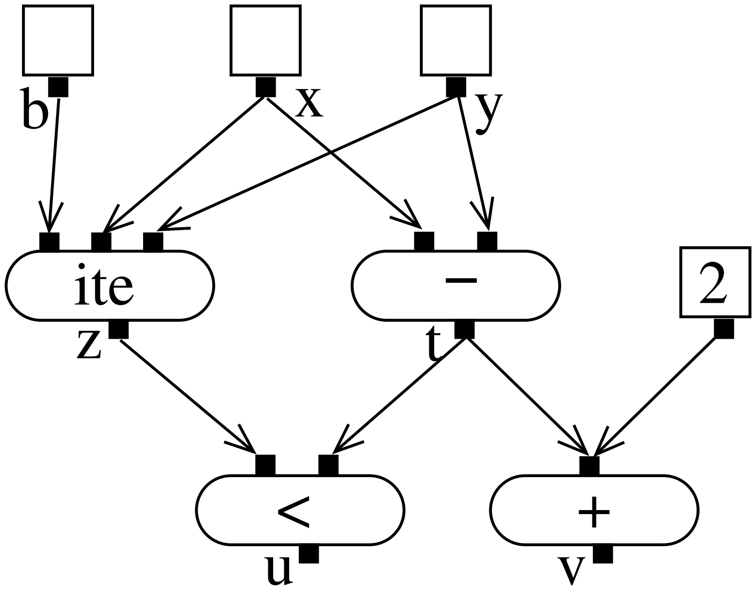
Example: $\sum_{i=1}^{N} y_i \leq K$

For **constraints**, evaluation stops in case of **inconsistency** detection.

DAG of Invariants

In InCELL, **DAG** (Directed Acyclic Graph) of **invariants** (no cycle) to represent constraint optimization problems.

$z \leftarrow (\textit{if } b \textit{ then } x \textit{ else } y)$
$t \leftarrow (x - y)$
$u \leftarrow (z < t)$
$v \leftarrow (t + 2)$



In case of **change** in the value of any **root** variable (no parent in the DAG), due to algorithmic **choices** or to **changes** in the planning problem, the DAG is **lazily** reevaluated: any invariant is reevaluated only **when necessary** and only **once**.

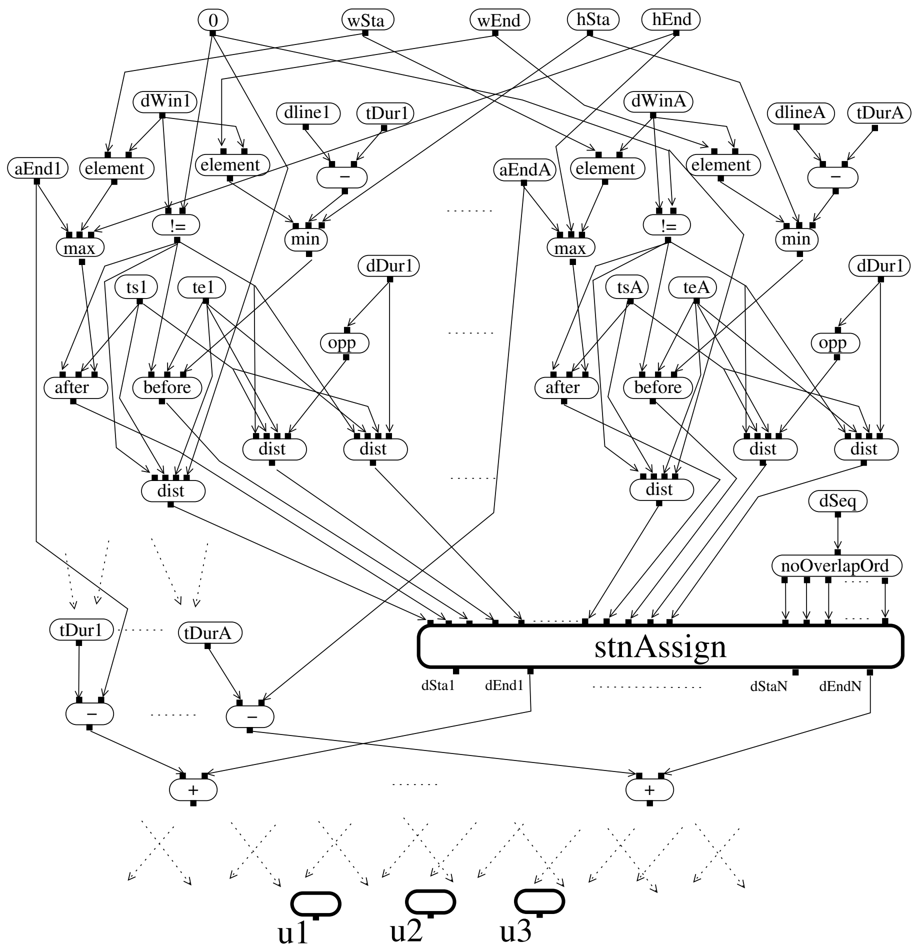
InCELL scheduling features

Special features dedicated to **scheduling** problems:

- time **points**;
- time **intervals**;
- unary and binary **temporal constraints**;
- **simple temporal network**: unique invariant which gathers all the temporal constraints, detects inconsistency, and manages temporal flexibility;
- **resource constraints**, with piecewise **constant** or **linear** evolutions.

Example of DAG

Small part of the DAG resulting from the modeling of the problem of planning **acquisitions** and **downloads** for an **agile** Earth observing satellite.



Possible future

Possible future

More and more **autonomous "intelligent"** spacecraft:

- onboard **data analysis**;
- onboard autonomous **decision-making**;
- onboard **fault detection**, **diagnosis**, and **reconfiguration**;
- long-term **learning**.

Fleets of connected spacecraft, aircraft, and ground systems (mobile or not) for **Earth surveillance** $\rightarrow$ **distributed "intelligence"**.