From Scheduling to Planning with Timelines: An history of successful Applications in Space

Part-1

Nicola Policella
ESA/ESOC
Germany

ICAPS 2013 Summer School
Special track on P&S Space Applications
Perugia, 4 June 2013

From Scheduling to Planning with Timelines:
An history of successful Applications in Space

- Basic AI Planning and Scheduling concepts
  - successful applications in the space domain
- Project scheduling
  - Precedence Constraint Posting
- A common knowledge representation core based on timelines
- Timeline-based modeling and planning
- Special Track on Space Applications

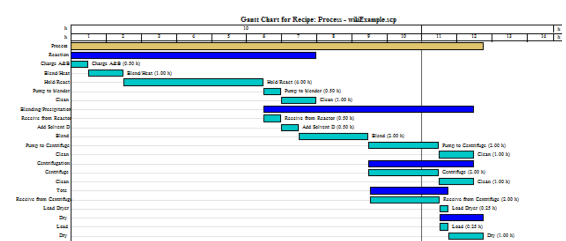
This Talk

- What is Scheduling ?
- Project Scheduling Problem
  - Precedence Constraint Posting
  - Scheduling Under uncertainty
- Final conclusions

What is Scheduling?

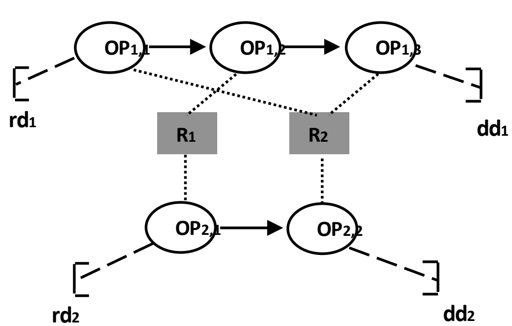
- To schedule (from Merriam-Webster):
  1. to appoint, assign, or designate for a fixed time
  2. to place in a schedule
  3. to make a schedule of

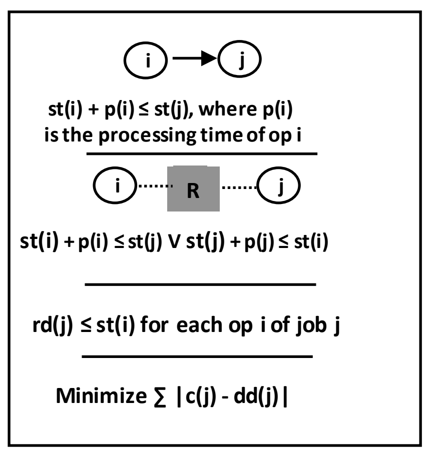
- Scheduling can generally be described as allocating (and optimizing) scarce resources over time (to perform a set of tasks).

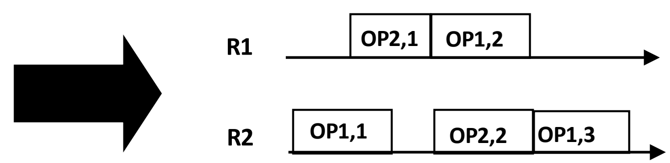


What is Scheduling?

- Classic view:
  - Scheduling is a puzzle solving activity-
    - Given problem constraints and objective criterion, figure out how to best tile the *capacity over time* surface with operations
  - **Research agenda** - specify new problems and/or provide new best solutions







Scheduling vs. Planning

- In classical scheduling, all the required operations (e.g., flights, production jobs) are known at the beginning of the solving process

- In some types of problem, we may choose not to schedule all operations but typically assume that we never add to the set of operations during search

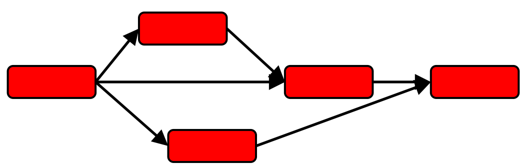
Scheduling vs. Planning

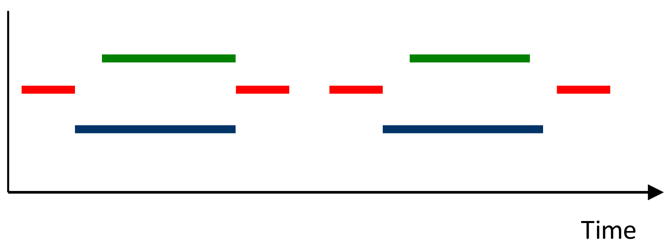
Space operations vs. A.I. : different meanings

Planning

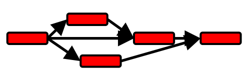
Scheduling

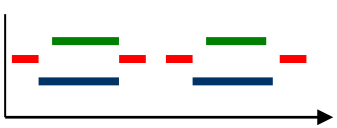
A.I.





Space Operations





FACD0013|0|5|2|1|A|0|
FACD0014|0|5|2|1|A|0|
FACD0015|0|5|2|1|A|0|
FACD0016|0|5|2|1|A|0|
FACD0017|0|5|2|1|B|0|
FACD0018|0|5|2|1|B|0|

Different problems

- Scheduling emerges in various domains, such as
  - nurse scheduling
    - Allocate people (with specific skills) on each shift
    - Regulations and working hours
  - airplane landing scheduling,
    - Allocate services to the airplane
  - production scheduling.
    - Series of operations
    - Each operation requires resources

- Different types of problems
  - Dynamic scheduling
    - Scheduling tasks in a CPU
  - Project scheduling
    - Organizing all the tasks for building a bridge
  - Oversubscribed scheduling
    - Allocating satellite activities over a time interval

Complexity ..

- Complexity of problem scheduling varies from polynomial to NP-Hard
  - Depends on:
    - Type of constraints
    - Presence of resources (binary/multi-capacitive)
    - Optimization

- Complexity results for different classes of scheduling problems can be found under
  - http://www.mathematik.uni-osnabrueck.de/research/OR/class/

How to Live with *NP*-hard Scheduling Problems?

- Small sized problems can be solved by
  - Constraint Programming (CP)
    - Complete search and inference
  - Mixed Integer Programming (MIP)
    - Complete search and relaxation

- To solve problems of larger size one has to apply
  - Approximation algorithms
  - Meta-heuristics / Local Search
    - Incomplete search

Project Scheduling Problem

Project Scheduling (1)

“.. the project scheduling problem involves the scheduling of project activities subject to precedence and/or resource constraints .. ”

- Example:
  - Production scheduling
  - Air traffic control
  - Timetabling
  - Personnel scheduling
  - Railway scheduling

- A quite general scheduling problem is the *Resource Constrained Project Scheduling Problem* (RCPSP)

Project Scheduling (2)

- Scheduling process results in a (*baseline) schedule*

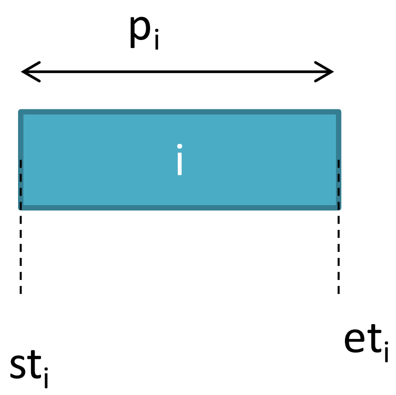
  – to allocate resources to the different project activities to optimize some measure of performance.

  – basis for planning external activities,
    - material procurement
    - shipping due dates to customers

Project Scheduling: Ingredients

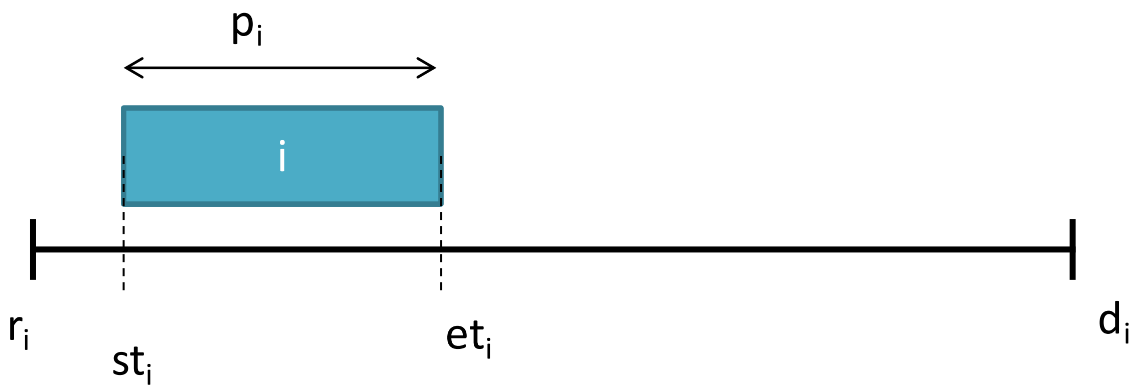
- Activities
- Time (constraint)
- Resources (constraint)

Activity



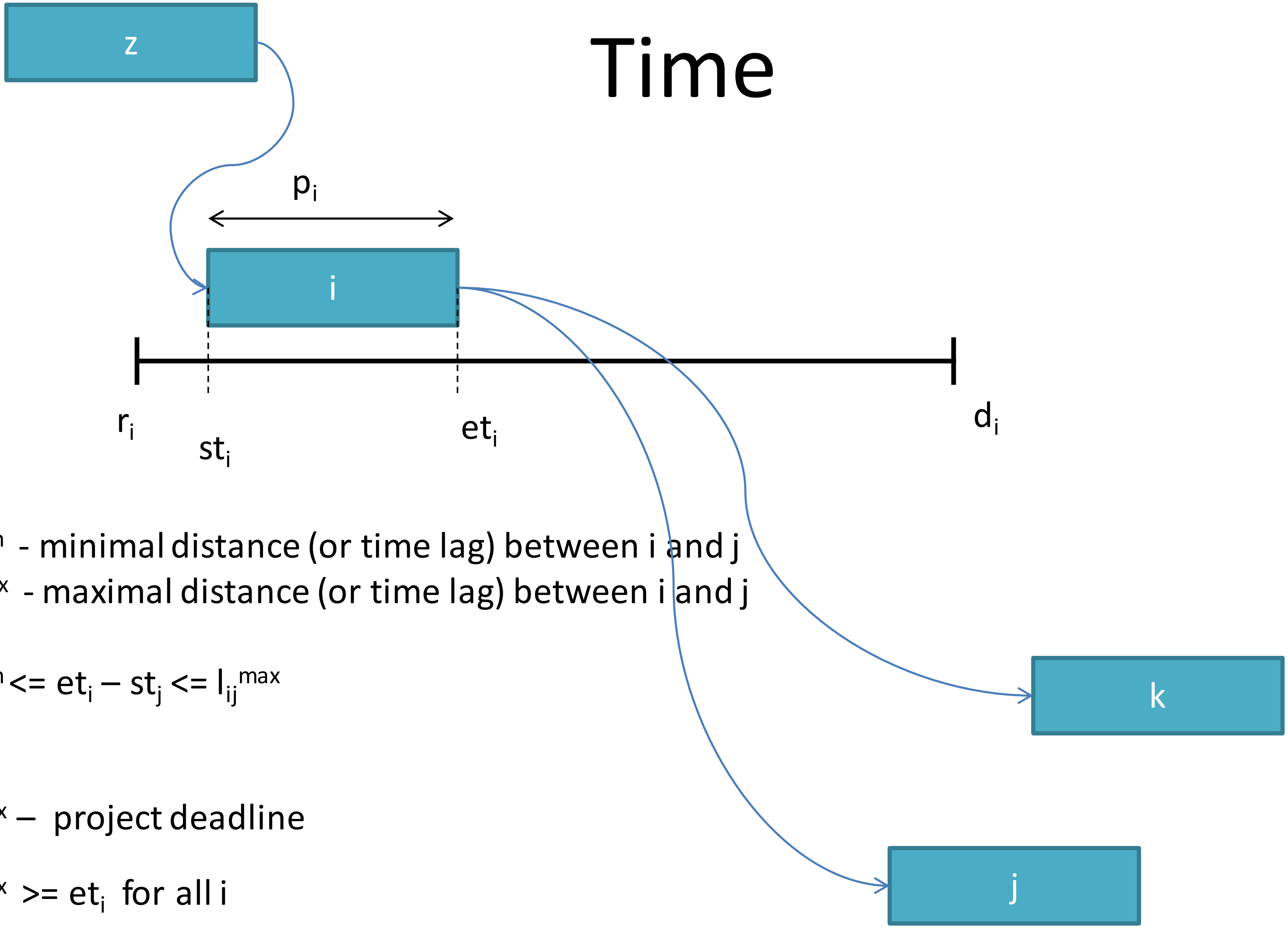
st_i – start time
et_i – end time
p_i – processing time or duration … $et_i >= st_i + p_i$
w_i – weight or priority

Time



r_i - release time … $st_i >= r_i$
d_i - due date … $et_i <= d_i$

Time



l_{ij}^{min} - minimal distance (or time lag) between i and j

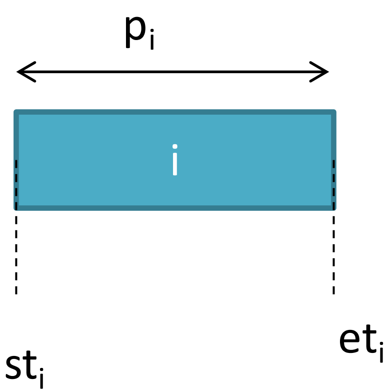
l_{ij}^{max} - maximal distance (or time lag) between i and j

$l_{ij}^{min} <= et_i - st_j <= l_{ij}^{max}$

d^{max} – project deadline

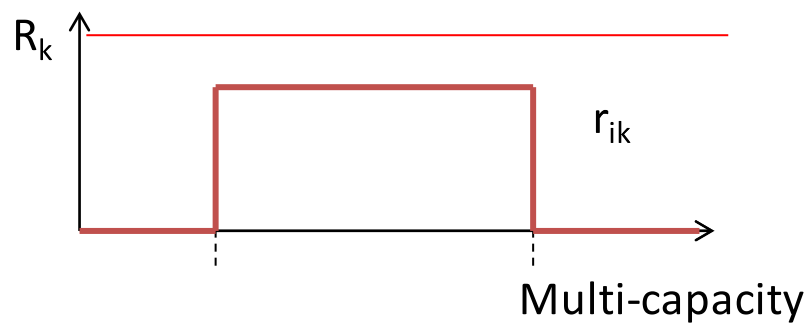
d^{max} >= et_i for all i

Resource

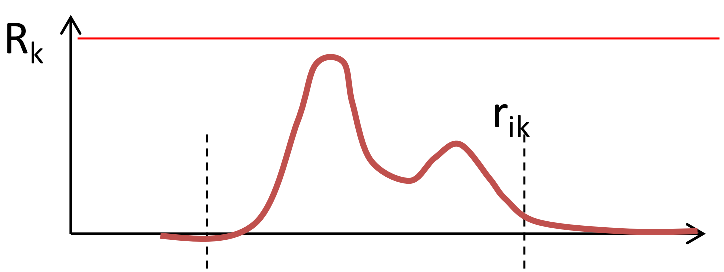


p_i
i
st_i
et_i

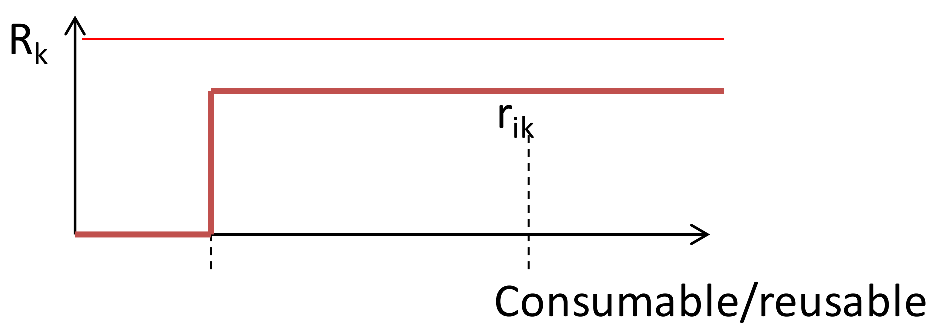
1
Binary



R_k
r_{ik}
Multi-capacity

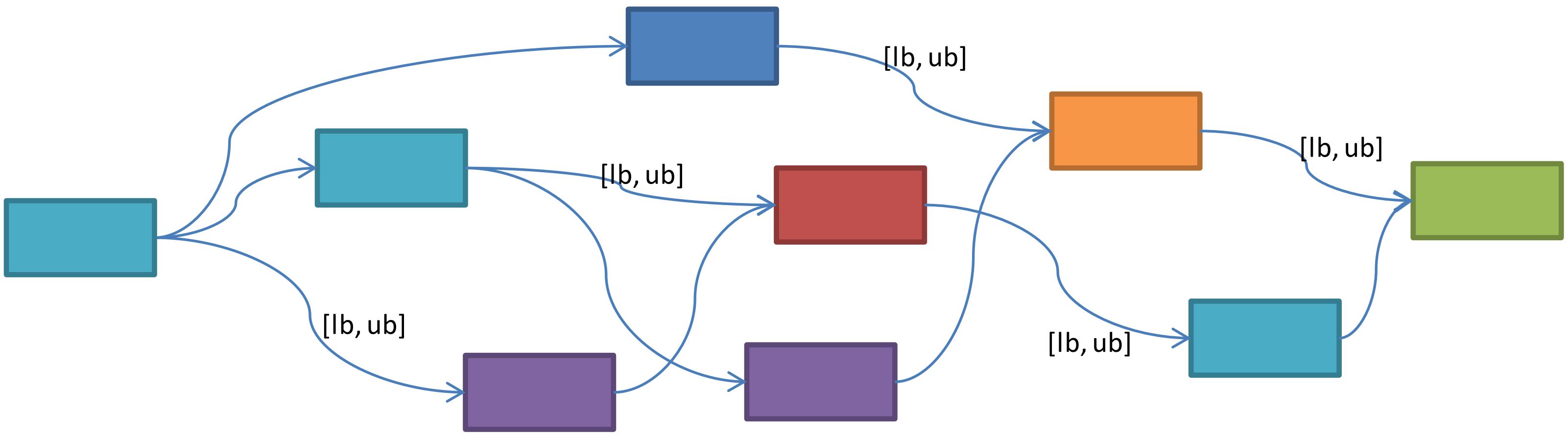


R_k
r_{ik}



R_k
r_{ik}
Consumable/reusable

Project scheduling: an activity-on-the-node (AON) format

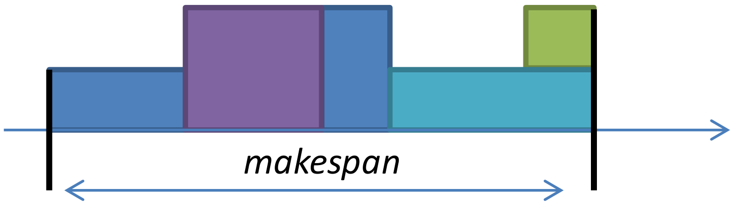


Schedules

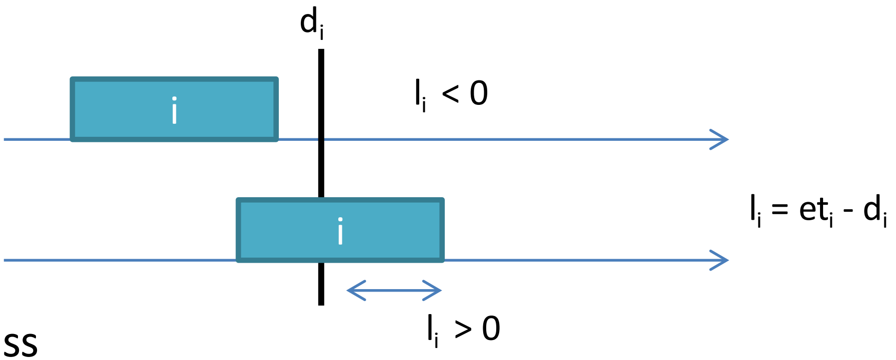
- A solution to a scheduling problem or schedule consists in deciding the start time (and end-time) of each activity i, such as:
  - All the time constraints are satisfied (Time Feasibility)
  - All the resource constraints are satisfied (Resource Feasibility)

Evaluate schedules

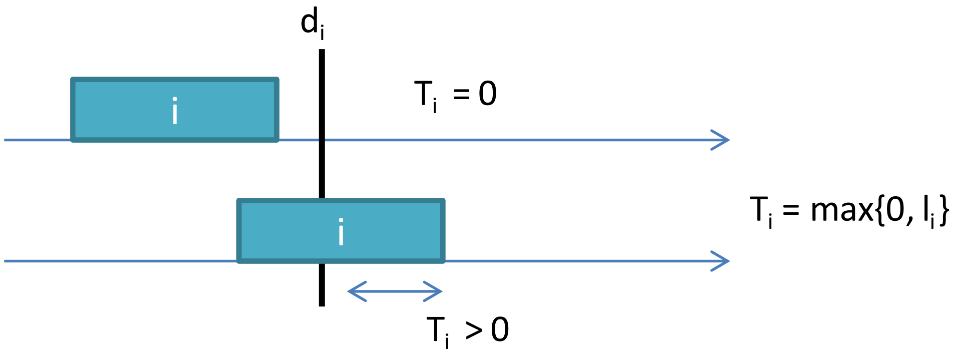
- Different metrics, e.g.:
  - Minimize Makespan (maximum completion time)



  - Minimize Maximum Lateness



  - Minimize Total (Weighted) Tardiness



But …

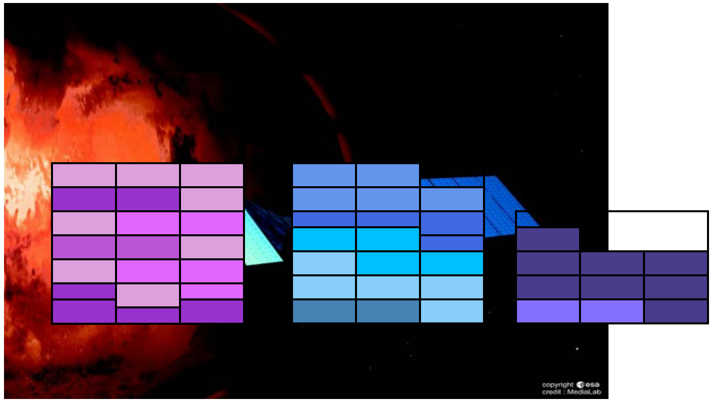
- Is it sufficient to allocate a start (and end time) for each activity ?
  - Is the concept of scheduling solution sufficient ?

- Are the previous metrics the right means to measure schedule's quality ?

- See later …

An example …

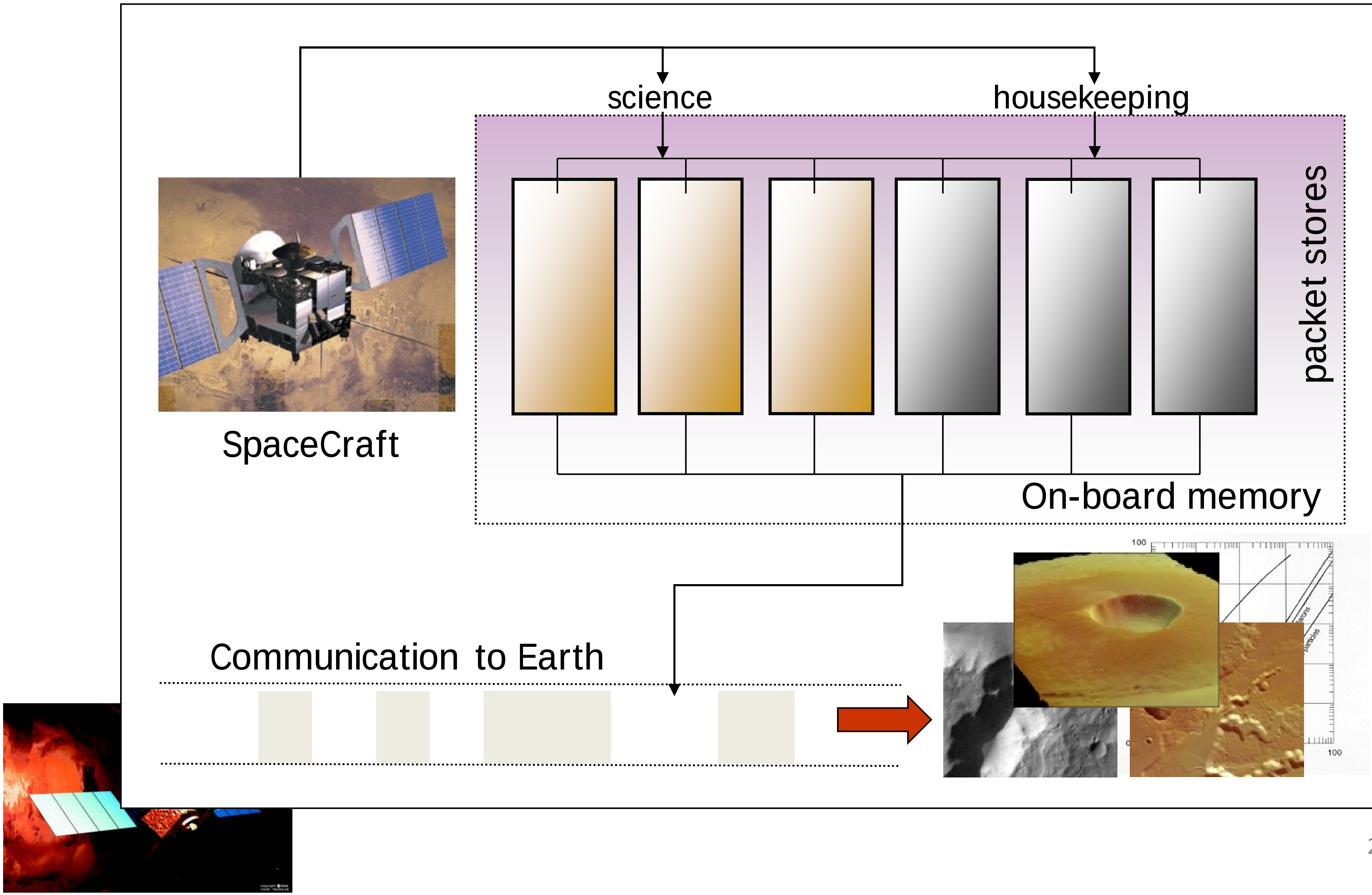
The Mars Express Memory Dumping Problem



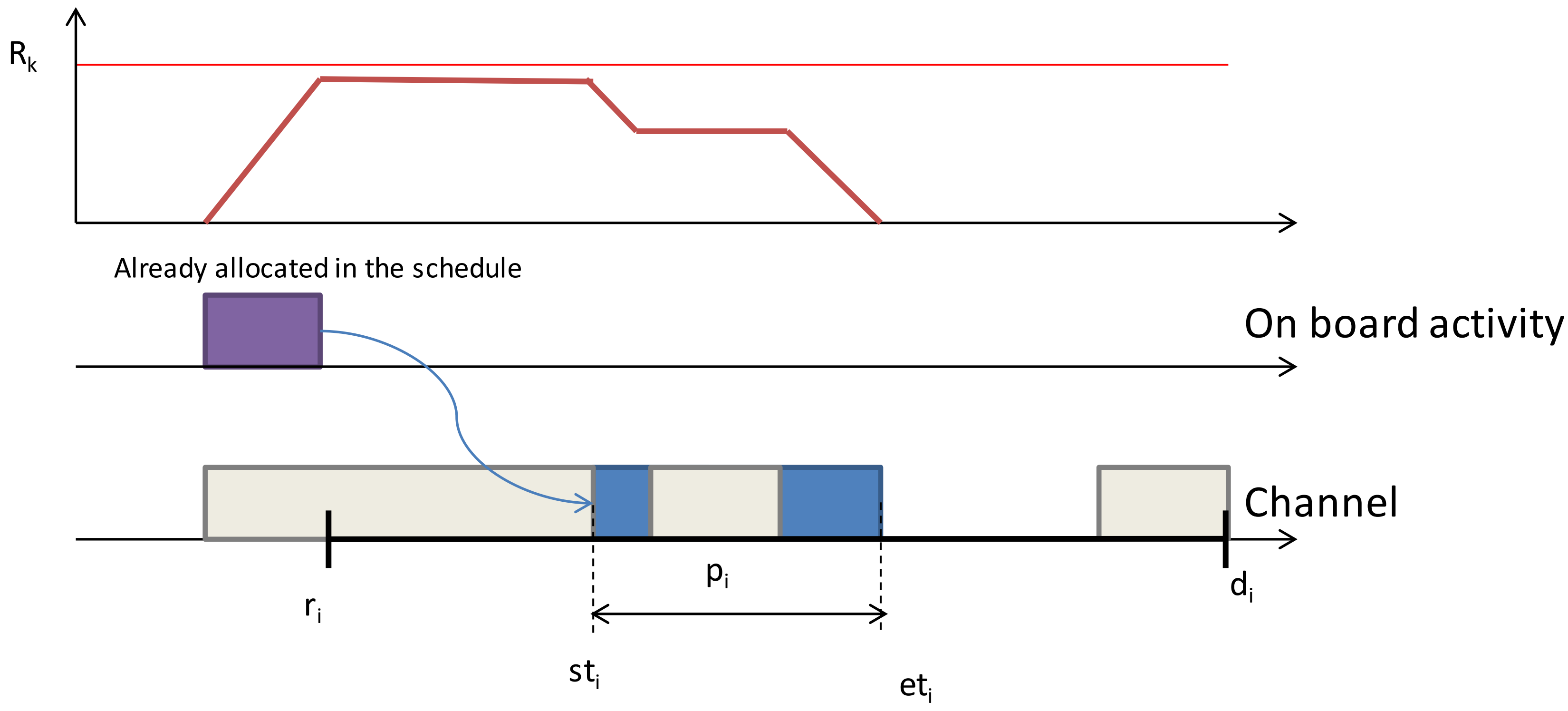
Overwrite!

No visibility

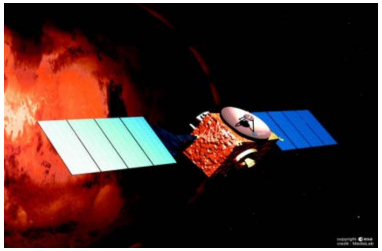
Relevant Mars Express parts



Example



R_k
Already allocated in the schedule
On board activity
Channel
r_i
p_i
d_i
st_i
et_i



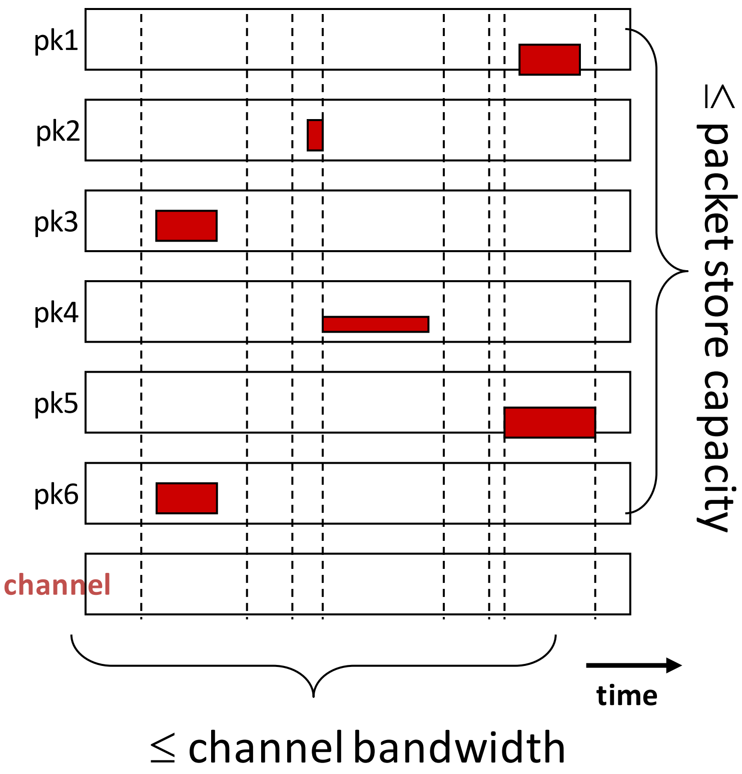
A Timeline Model

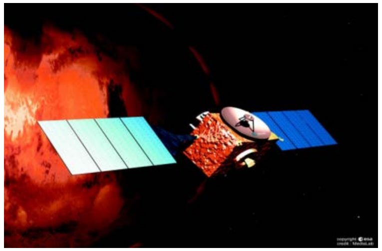
- For each packet store (and the communication channel) we define a different timeline

- The temporal horizon is subdivided in contiguous time intervals such that instantaneous memory operations can happen only at the edges

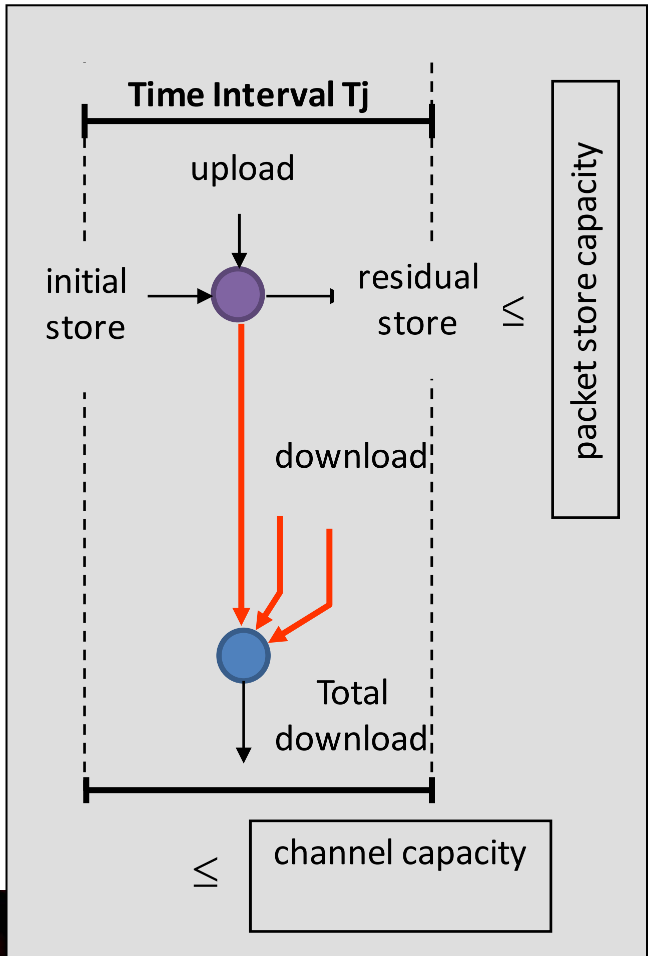
- Decision variables (*flow values*) represent the volume of data dumped within each interval

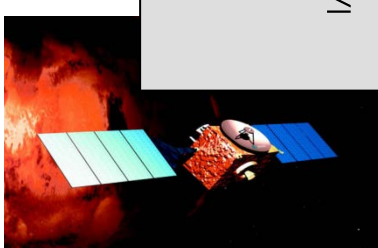
- Two types of constraints:
  - the ones imposed by the channel bandwidth and
  - the ones by the packet stores capacity

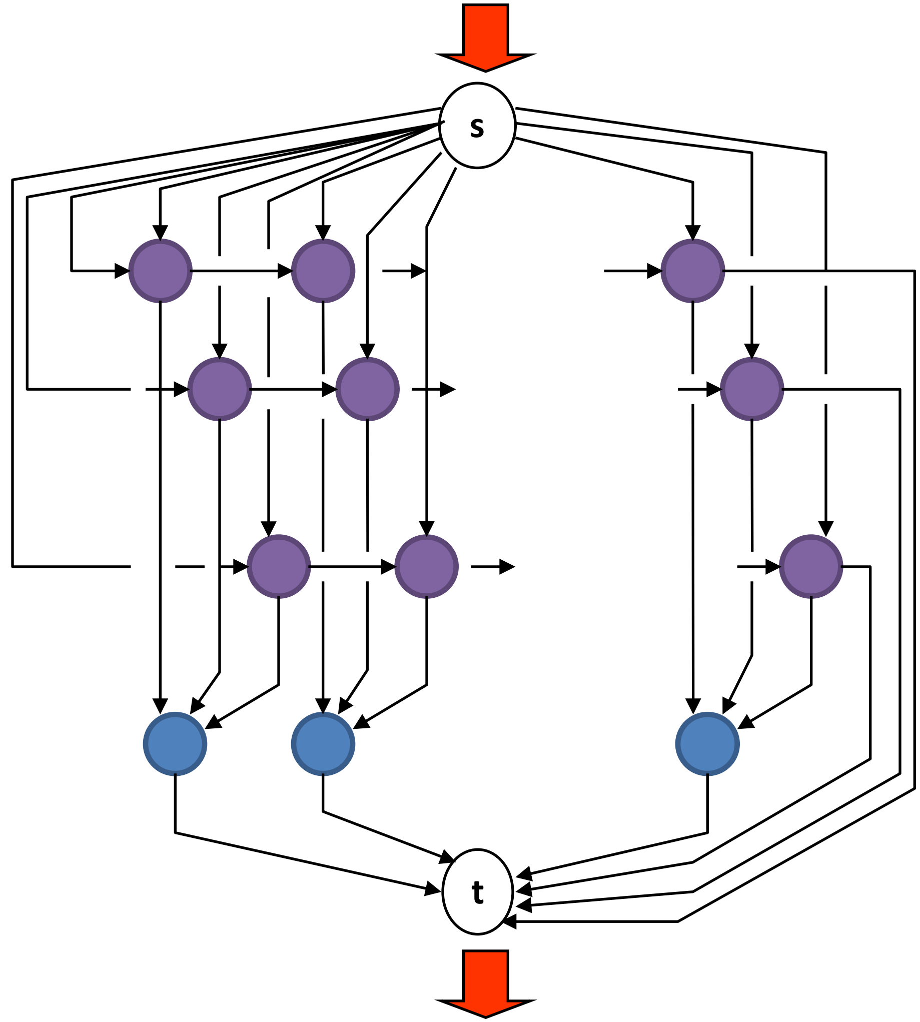




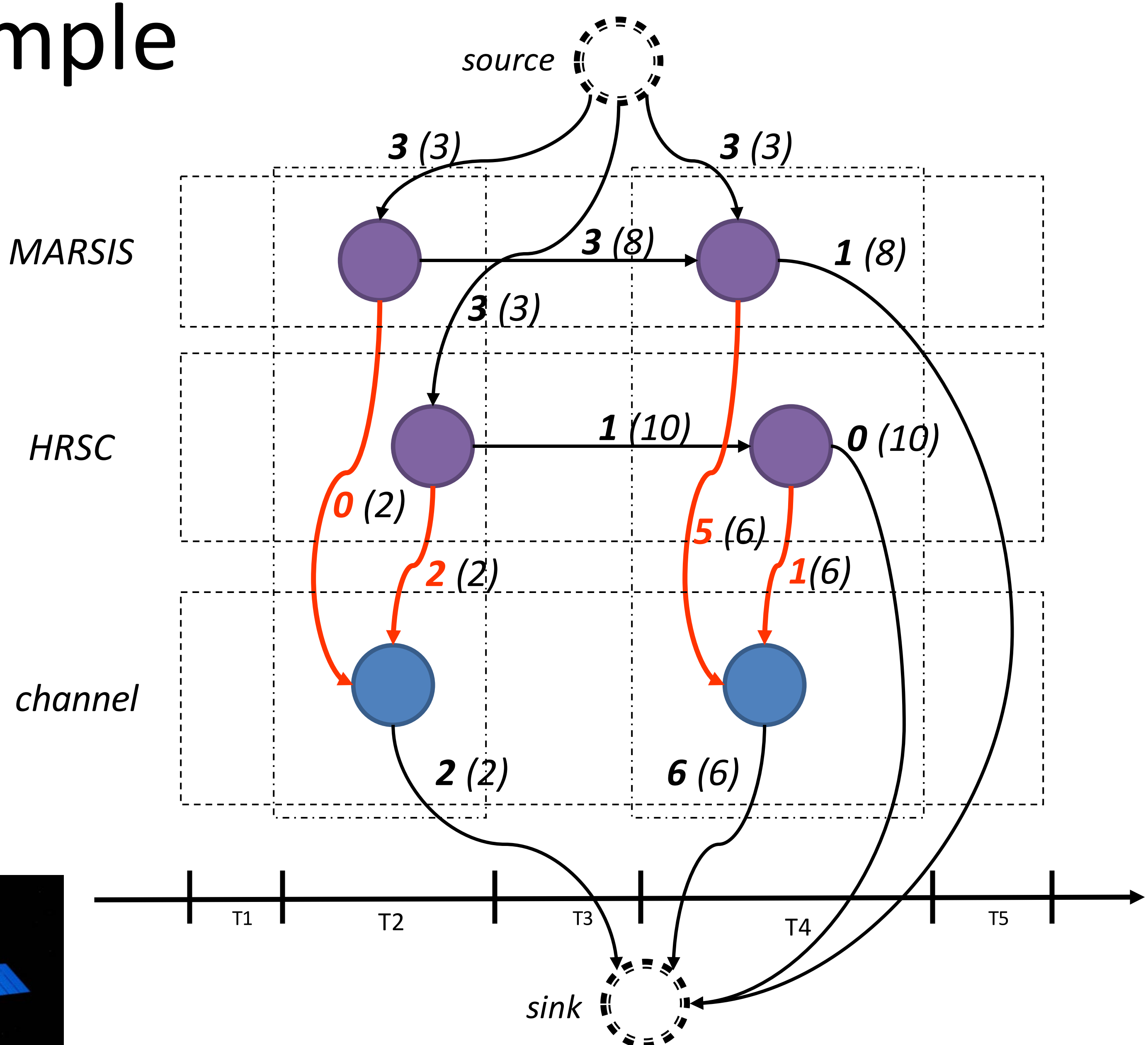
Using a flow-network on the model

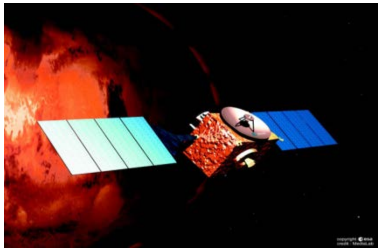






Example


source
3 (3)
3 (3)
MARSIS
3 (8)
1 (8)
3 (3)
HRSC
1 (10)
0 (10)
0 (2)
2 (2)
5 (6)
1(6)
channel
2 (2)
6 (6)
T1
T2
T3
T4
T5
sink



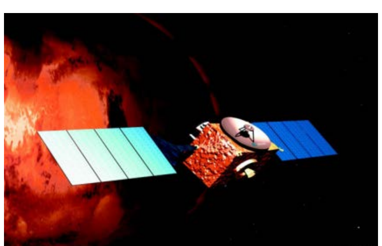
Core Algorithm

- Based on max-flow
  - Complete algorithm: the dumping problem has a solution *iff* the maximal flow over the problem horizon is equal to total data stored
  - Low polynomial cost: best known $O(n^{2.5})$

- Housekeeping needs daily download
  - Backtracking search to accommodate it before planning for science data

Algorithm 1: Generation of down-link commands

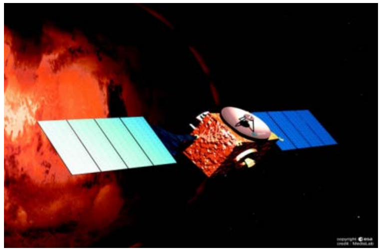
```
Input: problem P
Output: Down-link commands Cmd

// Data Dump Level
// Produce a data allocation
// over the communication channel

while S incomplete do
    S ← HousekeepingAllocation(P)
    S ← MaxFlowAllocation(P, S)
    if No more HK allocation then
        break

// Packet Level
// Generate data commands

Cmd ← GenerateDownLink(S)

return Cmd
```



Classic approach vs. real needs

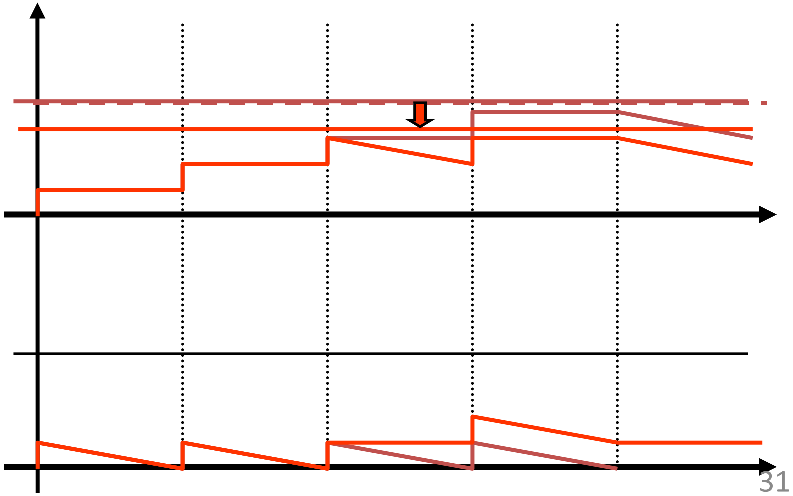
Initial solution

- Solutions which optimize the
  - Makespan
  - Returning time: $\min(et_i - r_i)$
- Very fractioned solutions
  - Fractioned download

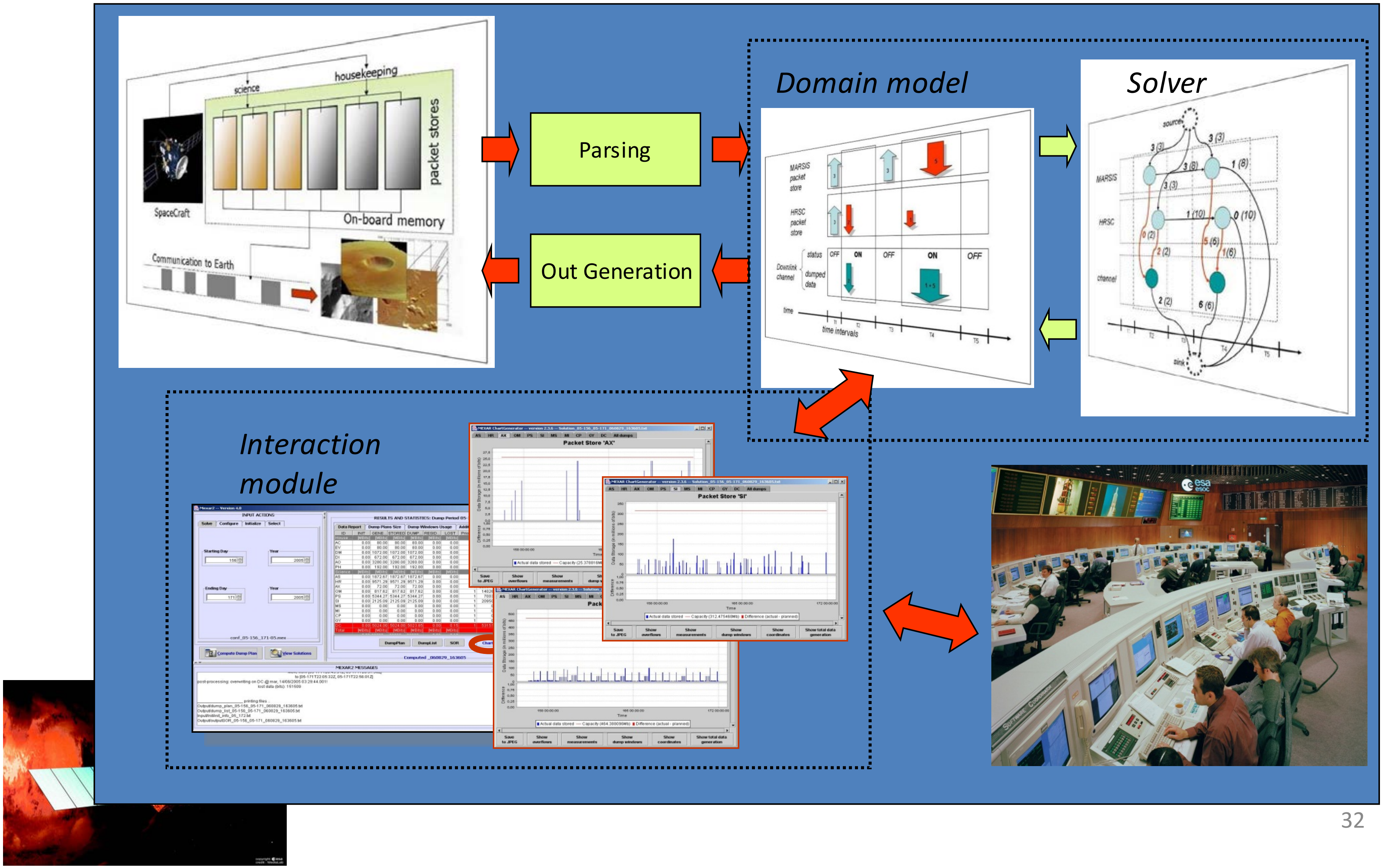


What the users wanted

- Always generate a solution (relaxation)
- Control solution length
- Optimize the robustness instead ..

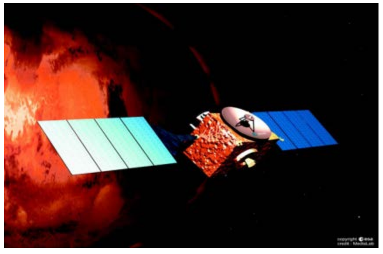


Final architecture: end-to-end cycle



Results: users perspective

- It solves the problem …
  - very important precondition to be taken seriously
- Reduction of working time
  - Previous practice was a nightmare (every single day)
  - Internal evaluation estimate to 50% the reduction of workload
- Increase of science return
  - Running MEXAR2 in advance identifies bottlenecks
  - Enable feedback to science payload PIs
- Reduction of costs
  - Save commitment (i.e., money) on ground stations



Questions ??

Precedence Constraint Posting

RCPSP/max

- A set of activities $V = \{a_1, a_2, \dots, a_n\}$
  - Each activity has a fixed *processing time*, or *duration*, p_i.
  - Any given activity must be scheduled without preemption.

- A set of Temporal Constraints designate minimum and maximum time lags between the start times of two activities
  - $l_{ij}^{min} \le s_j - s_i \le l_{ij}^{max}$

- A set of Reusable Resources. During their processing, activities require specific resource units from a set $R = \{r_1, \dots, r_m\}$
  - Resources are *reusable*, i.e. they are released when no longer required by an activity and are then available for use by another activity
  - Each activity a_i requires of the use of req_{ik} units of the resource r_k during its processing time p_i.
  - Each resource r_k has a limited capacity of c_k units.
  - For each resource:
    - $\sum_{c_i \le t < e_i} req_{ik} \le c_{ik}$

- A *schedule* is an assignment of start times to activities $a_1, a_2, \dots, a_n$, i.e., $S = (s_1, s_2, \dots, s_n)$

Precedence Constraint Posting

- A solving paradigm based
  - Constraint programming
    - CSP
  - The presence of Simple Temporal Problem
    - Used to identify and analyze the current solution
  - Iterative Repair
    - Precedence Constraints are posted to repair the current solution

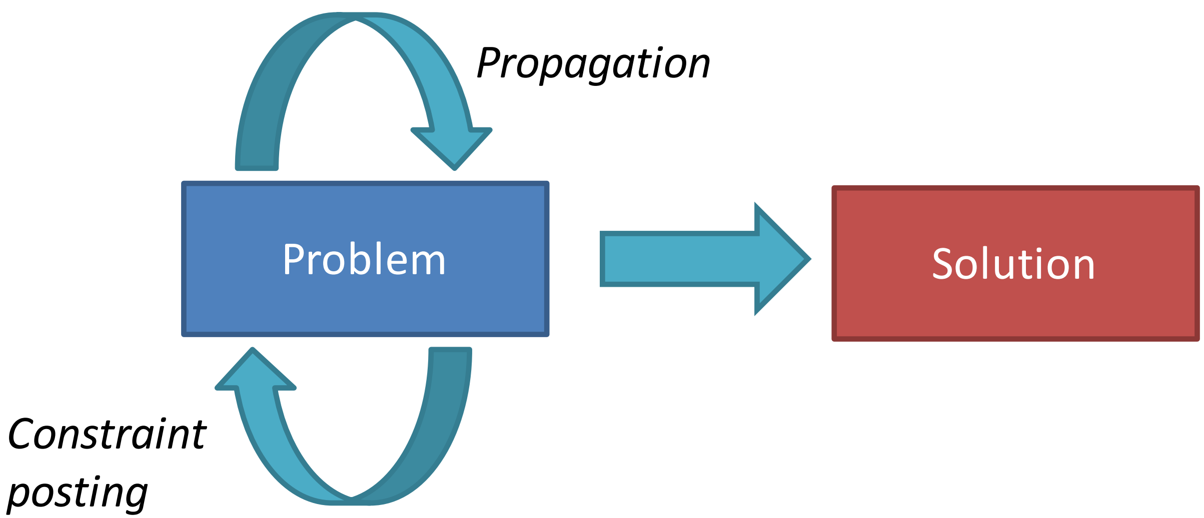
Constraint Satisfaction Problem (CSP)

- An instance of CSP involves
  - a set of Decision Variables $X = \{X_1, X_2, \ldots., X_n\}$
  - a Domain of possible values D_i for each variable
  - a set of Constraints $C = \{C_1, C_2, \ldots, C_q\}$, such that $C_j \subseteq D_1 \times D_2 \times \ldots \times D_n$

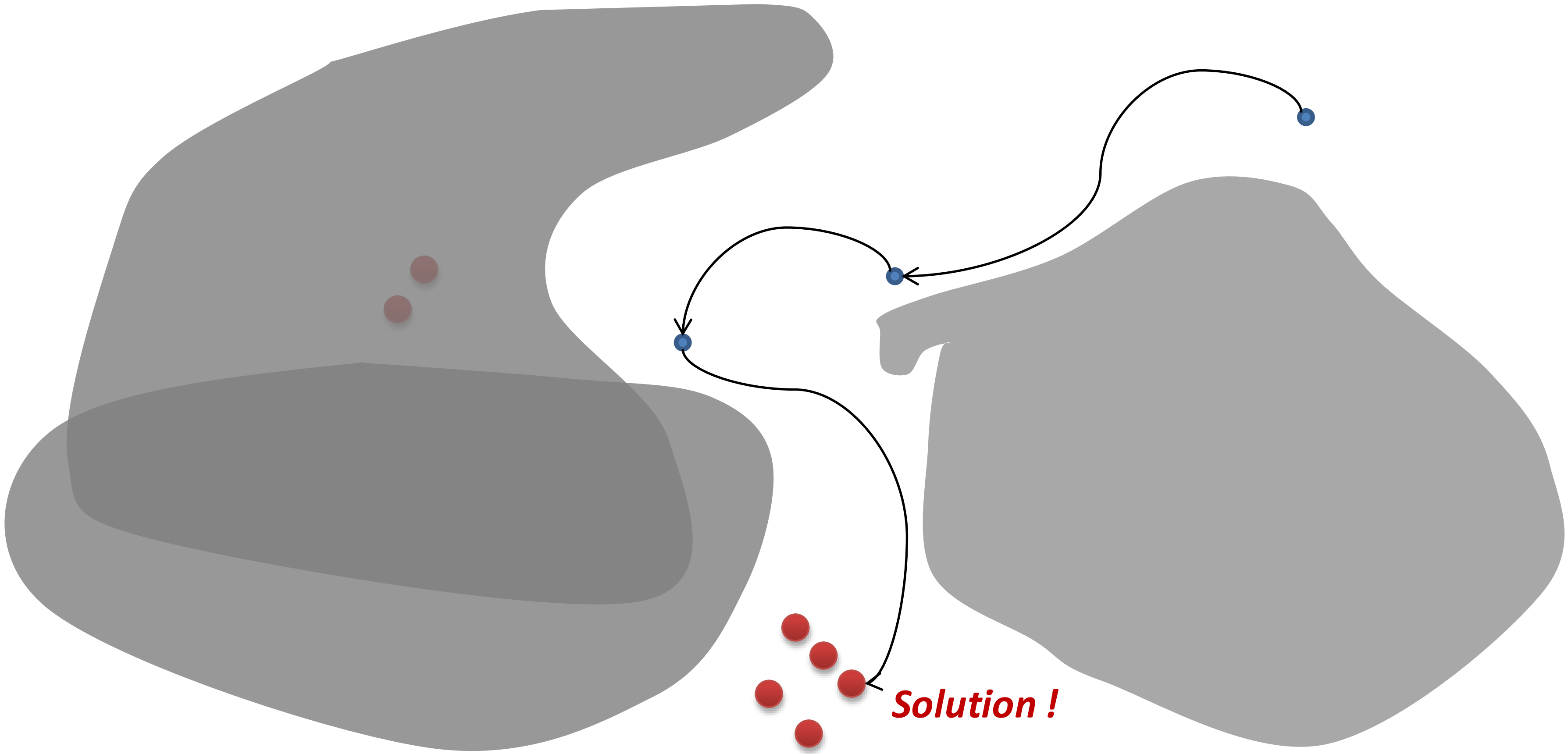
- A Solution is an assignment of domain values to all variables consistent with all the constraints C_j

Solving Approach

- Based on two aspects:
  - Propagation of constraints
  - Decision of new constraints to solve current conflicts (next slides)

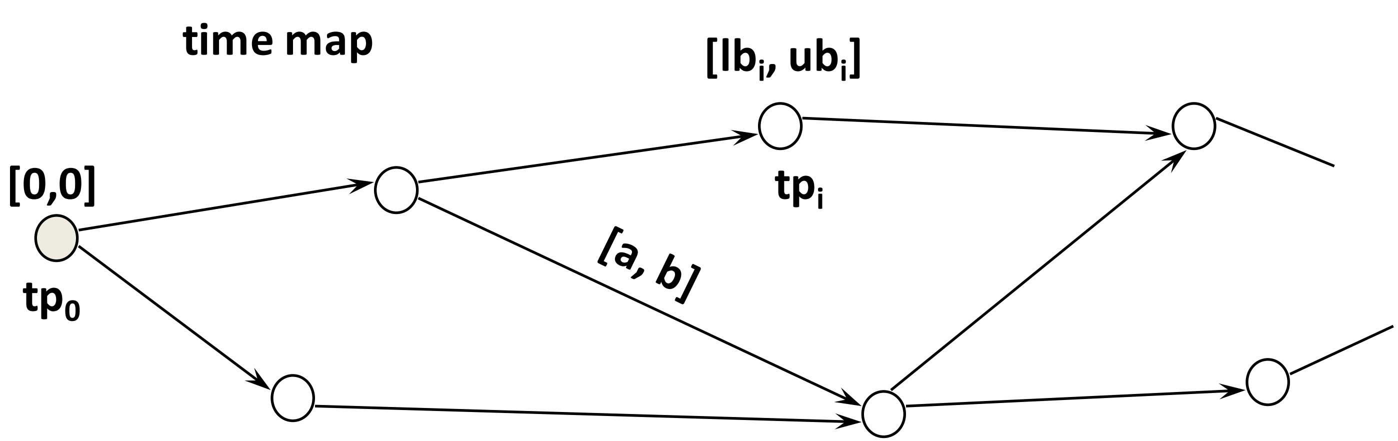


Solving Approach

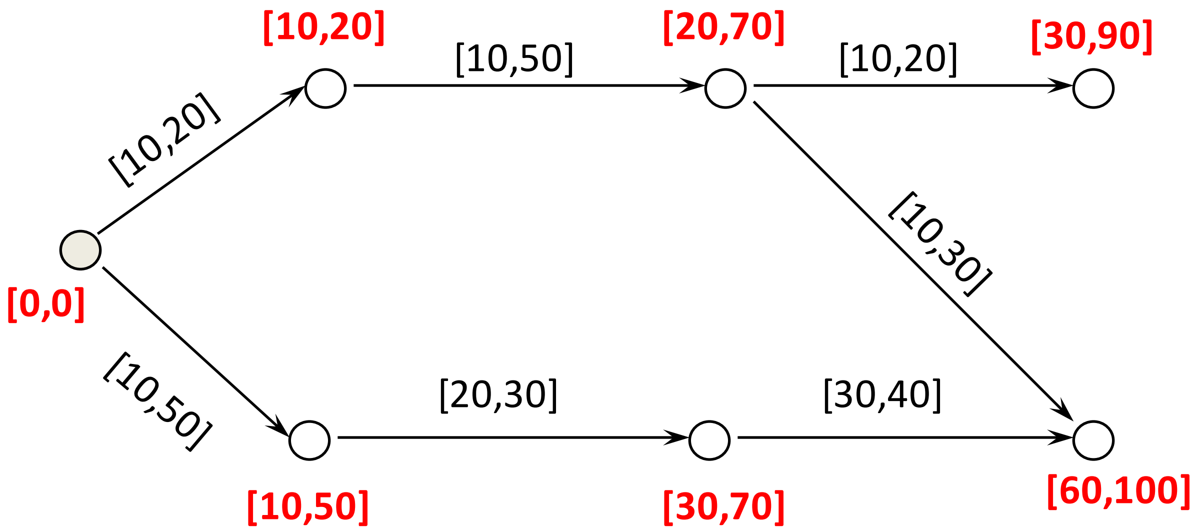


Simple Temporal Problem (STP)

- *A Simple Temporal Problem* is a set of n **variables** (time points) $\{tp_i\}$ with domain $[lb_i, ub_i]$ and a set of **constraints** $\{a \leq tp_j - tp_i \leq b\}$.
- A particular time point $\mathbf{tp_0}$ called time origin with domain *[0,0]*
- The problem is *consistent* when an instantiation of the variables $\{tp_i\}$ exists such that satisfies all the constraints
- A *time-map* represents a *Simple Temporal Problem*.



An example of time-map



STP Properties

- STP is a special case of Constraint Satisfaction Problem (CSP)

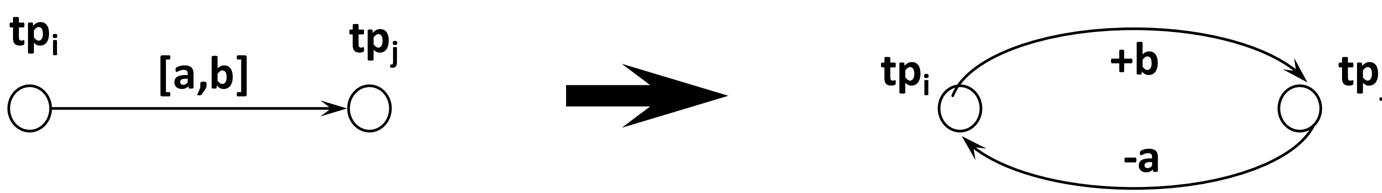
- Can be reduced to a Shortest Path Problem on a distance graph $\boldsymbol{G_d}$
  - Inconsistency **<=>** negative cycles in G_d
  - Variables (tp_i) domains **<=>** Shortest Path Trees

- Supports incremental modification
  - Insertion of new constraints
  - removal of constraints

- Has an equivalent set of constraints called Minimal Network

STP as a Shortest Paths Problem

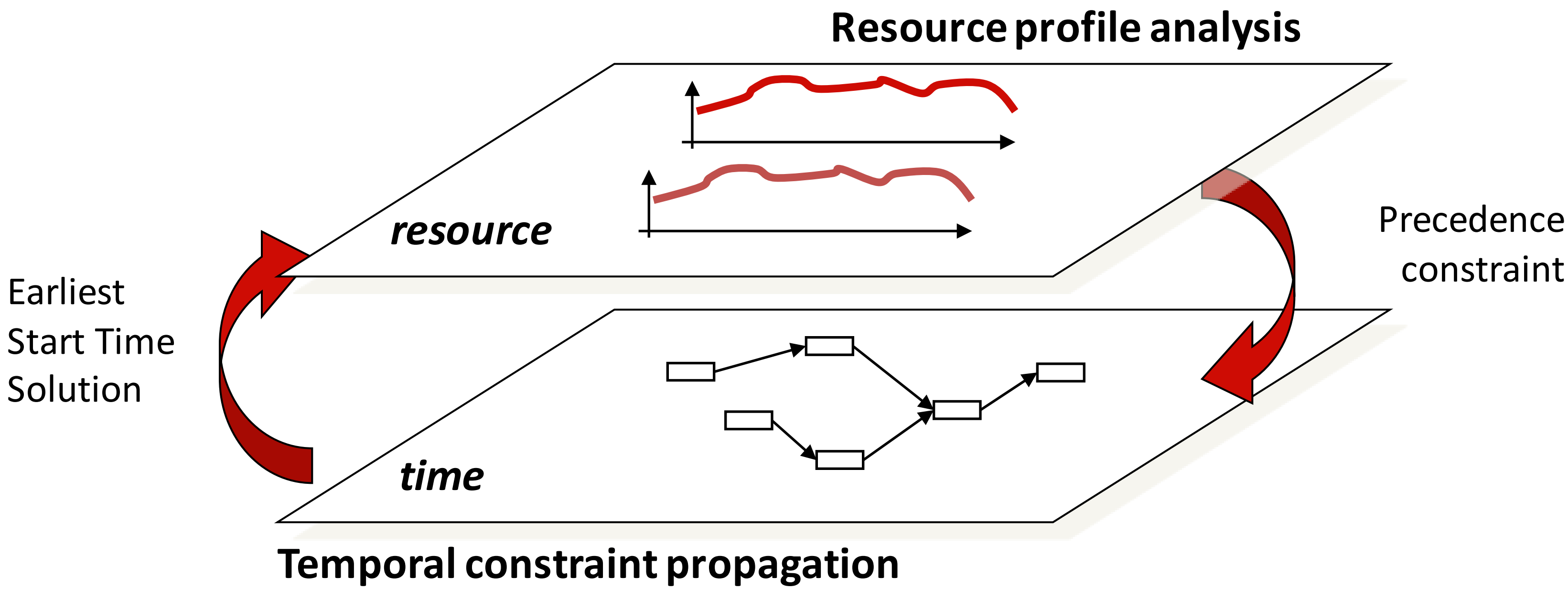
- STP problem can be reduced to a shortest paths problem on a graph $G_d(V,E)$, where V is the set of time points and E is the set of labelled edges such that:



- A STP is <u>consistent</u> **iff** G_d has no negative cycles [Decter et al, 1991].

- An STP has at least two solutions:
  - EST, in which each variable is assigned to its earliest possible time
  - LST, in which each variable is assigned to its latest possible time

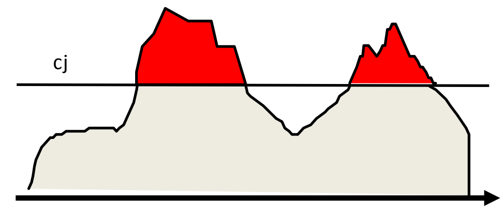
Precedence Constraint Posting (PCP)



- Variables:
  - start and end times of each activity
- Domains
  - A schedule horizon [0, H]
- Constraints
  - Temporal constraints (e.g., duration of activities, min-max separation between activities, simple precedence)
  - Resource constraints (e.g., bounds on capacities)

Precedence Constraint Posting (PCP)

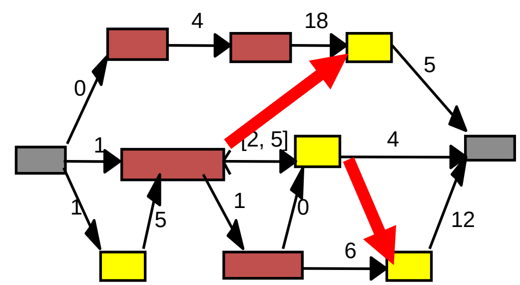
```
loop
  propagate(CSP)
  compute-conflicts(CSP)
  if no-conflict then
    return-solution
  else
    if unresolvable-conflicts then
      return-fail
    else
      select-conflict
      select-precedence
      post(precedence)
end-loop
```



Resource profile

Remove resource violations *posting* further precedence constraint in the temporal network

SOLUTION FOUND!



Temporal Net

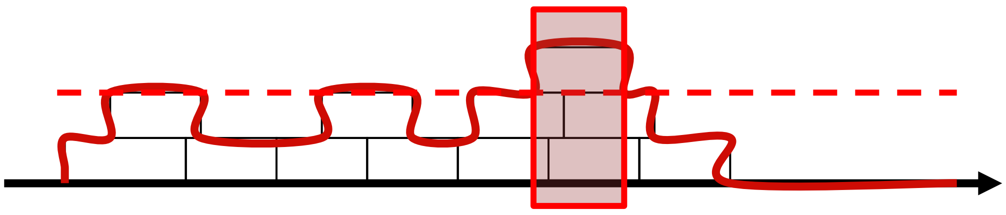
Inconsistent values can be pruned in polynomial time (Dechter, 91)

Solving Approach

- Three steps:
  - Identify conflicts
  - Conflict ordering
  - Conflict resolution

- Based on the works from
    - Chien & Smith,
    - Cesta, Oddi & Smith
    - Policella, Cesta, Oddi, & Smith

Identify conflicts



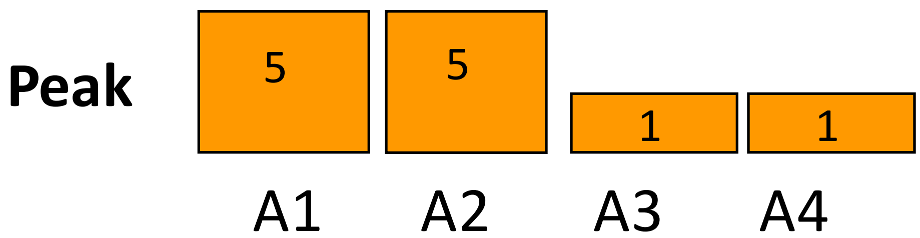
- Analyze the Earliest Start Solution profile
  – Earliest start time allocation assures to have time feasible solution

- Compute Conflict Peaks
  – there is a conflict peak on resource rk at time t if the resource requirement of the activities scheduled in t exceeds the resource capacity capk of rk

- Compute MCSs on Peaks
  – a Minimal Critical Set (MCS) is a conflict such that each of its proper subsets is not a conflict

MCSs? (Example)

Resource capacity = 8



Pairs of activities:
{A1 A2}{A1 A3}{A1 A4} {A2 A3} {A2 A4}{A3 A4}

MCSs:
{A1 A2}

[Erschler et al., 1990]

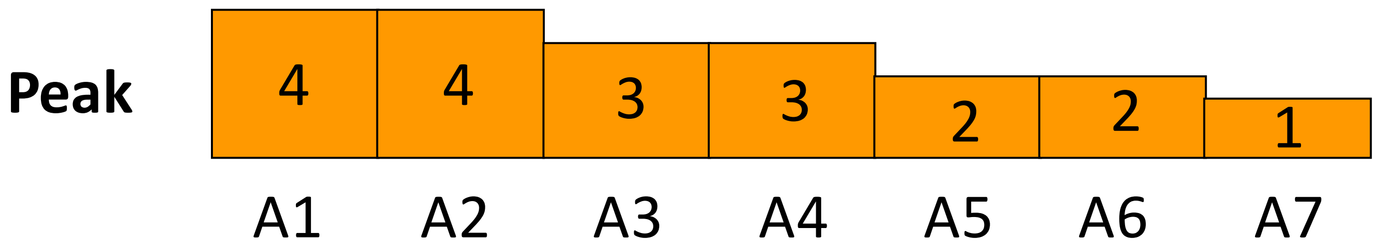
Approximate Computation of MCSs

- The number of minimal critical sets is exponential in the general case

- Proposal:
  Sampling them with an approximate analysis on peaks
  - Linear sampling
  - Quadratic sampling

Two Peak Sampling Methods (Example)

Resource capacity = 6



MCSs

Linear Sampling:

{A1 A2}

{A2 A3}

{A3 A4 A5}

{A4 A5 A6}

Quadratic Sampling:

{A1 A2} {A1 A3} {A1 A4}

{A2 A3} {A2 A4}

{A3 A4 A5}{A3 A4 A6}{A3 A4 A7}

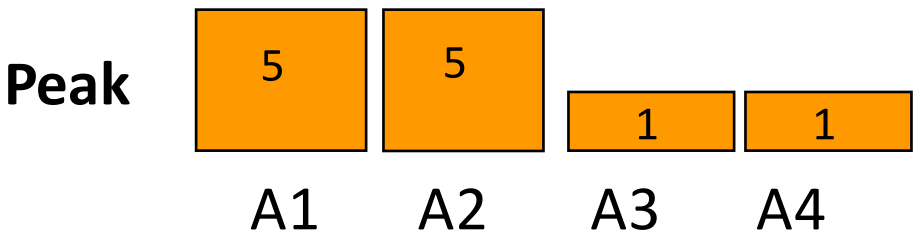
{A4 A5 A6}

Conflict ordering

- Which MCS to resolve first
  - Use estimator K [Laborie&Ghallab,IJCAI'95] to order MCSs
    - Least commitment strategy [Weld, AIM 94]
  - "Select the MCS that is temporally closest to an unsolvable state"
  - Constraint: $\rho = A\ before\ B$
    - Where A – B in [dmin, dmax]
    - $commit(\rho) = \frac{\min(dmax,0) - \min(dmin,0)}{dmax - dmin}$
  - Set of constraints that can be posted $\Phi = \{\rho 1, \ldots, \rho k\}$
    - $\frac{1}{K(\Phi)} = \sum_{i=1}^{k} \frac{1}{1 + commit(\rho i) - commit(\rho min)}$

Conflict resolution

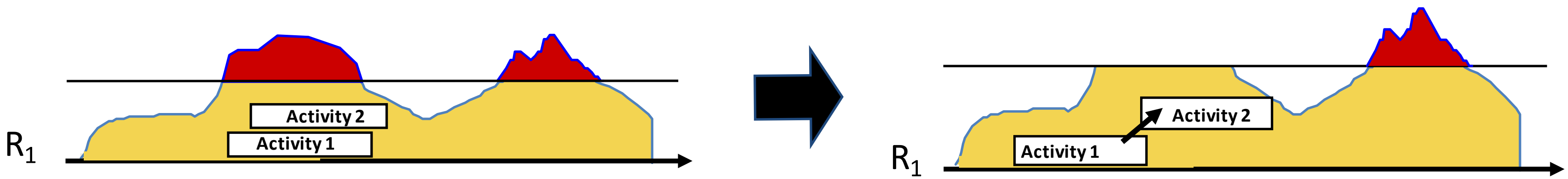
- how to choose the precedence constraint
  - Use slack-based heuristics [Smith&Cheng,AAAI’93]



MCSs:
{A1 A2}

A1 before A2 OR A2 before A1 ??

Why Precedence Constraint Posting ?



Identify a contention peak and post a leveling constraint

Advantages

- Retain flexibility implied by problem constraints (time and capacity)
- Can establish conditions for guaranteed executability
- Mixed initiative
- Be exploited in planning and scheduling integration

Questions ??

Project Scheduling under Uncertainty

What's Missing from the Classical View of Scheduling

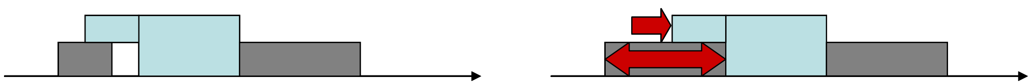
- Practical problems can rarely be formulated as static optimization tasks

  – Ongoing iterative process

  – Situated in a larger problem-solving context

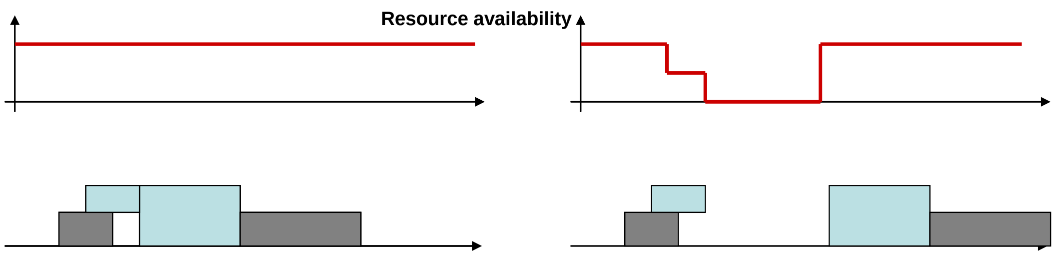
  – Dynamic, unpredictable environment

Scheduling with Uncertainty



temporal uncertainty

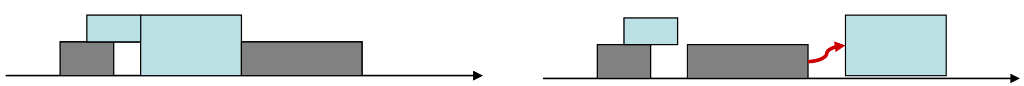
Activities last longer than expected or they can be postponed



resource uncertainty

Difference between nominal (left) and actual (right) resource availability.

Reduction of resource availability blocks the execution of some activities
an their consequent delay

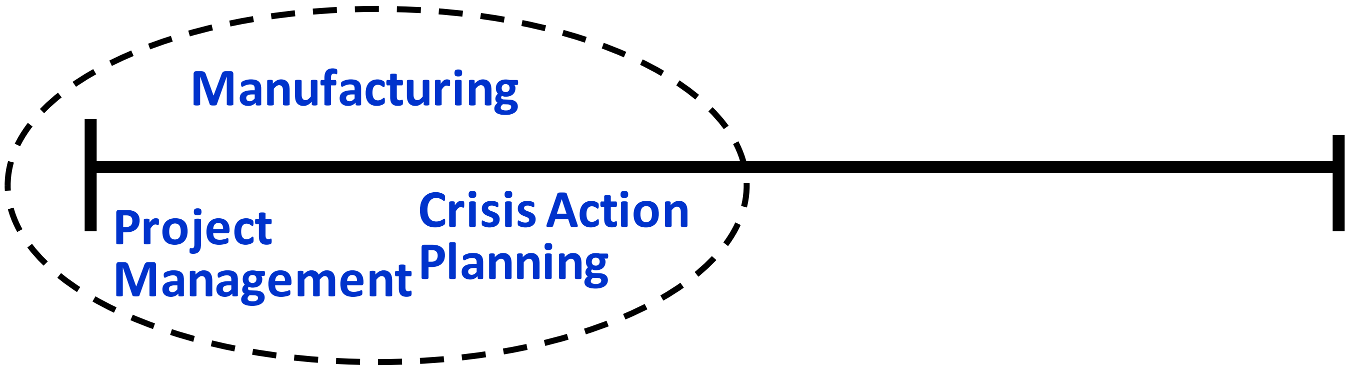


causal uncertainty

A new precedence relation between a pair of activities requires a revision of previous choices

Managing Change

- *"Scheduling" is really an ongoing process of responding to change*



- Predictable, Stable Environment
  - Optimized plans
- Unpredictable, Dynamic Environment
  - Robust response

Approaches to Managing Change

– Build schedules that retain flexibility

– Produce schedules that promote localized recovery

– Incremental re-scheduling techniques (e.g., that consider “continuity” as an objective criteria)

– Self-scheduling control systems

Approaches to Managing Change

- Approaching a scheduling problem requires the coupling of:

1. a **predictive** scheduling engine, to propose a possible solution

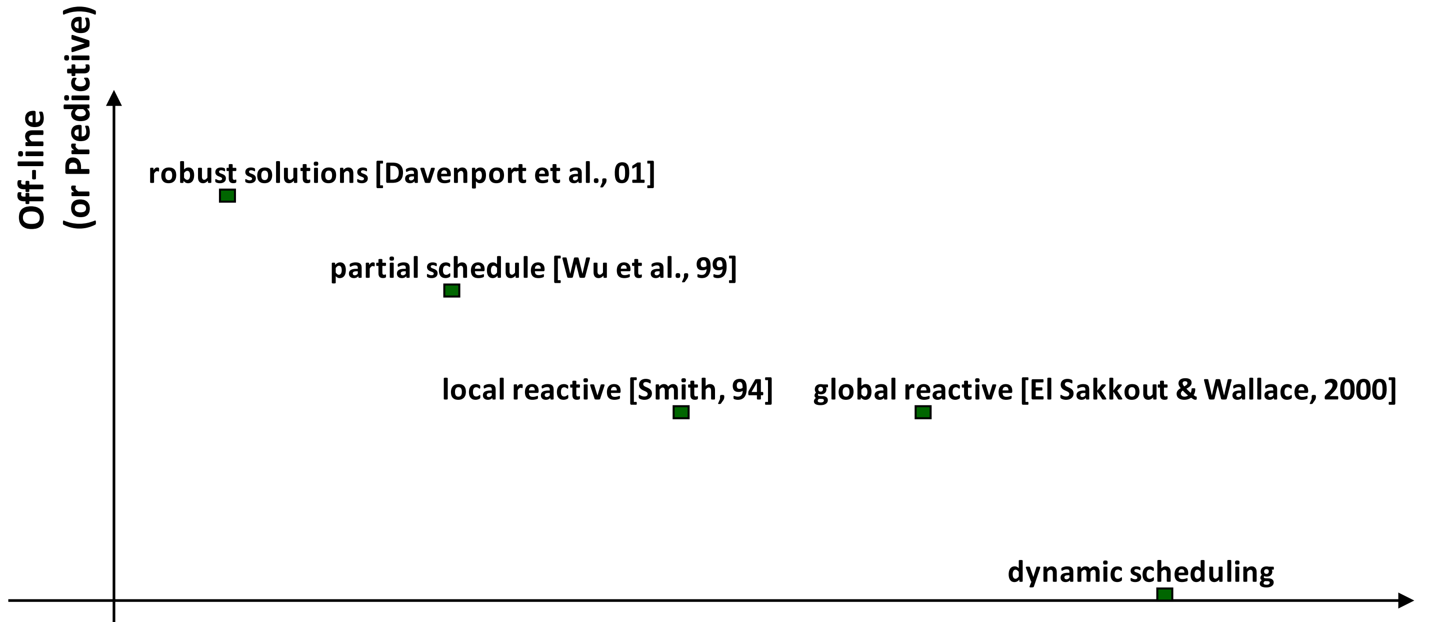
2. a **reactive** scheduling engine, to manage the current solution and to make “repairs” during the execution

Incremental Schedule Repair

- Several competing approaches to maintaining solution stability
  - Minimally disruptive schedule revision (temporal delay, resource area, etc.)
  - Priority-based change
  - Regeneration with preference for same decisions

- Even less understanding of how to trade stability concerns off against (re)optimization needs

Current Approaches



predictive

- Producing solutions based on uncertainty knowledge
  - Synthesis of robust solutions [Davenport et al., 01]
- Using flexibility to cope with uncertainty
  - Partially defined solutions [Wu et al., 99]

reactive

- Dealing with uncertainty during schedules execution
  - Local versus Global approach [Smith, 94; El Sakkout & Wallace, 2000]

Current Approaches (2)

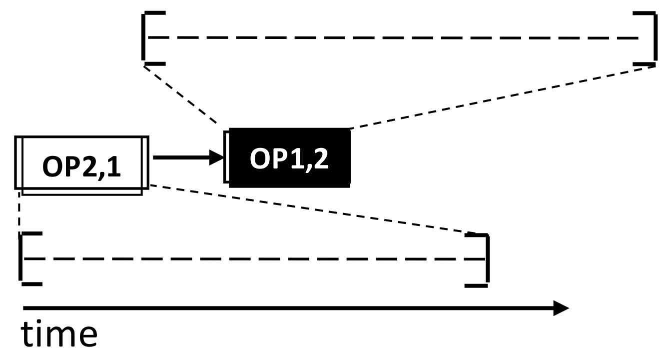
ROBUSTNESS

- different views:
  - Execution based.
    - small & fast repairs [Ginsberg et al., 98]. Not dependent of the rescheduling algorithm used.
  - Quality based.
    - maintaining the makespan (quality) stable [Leon et al., 94]
  - Solution based
    - maintaining the solution close to the original one [Leus et al., 04]

Partial order schedule

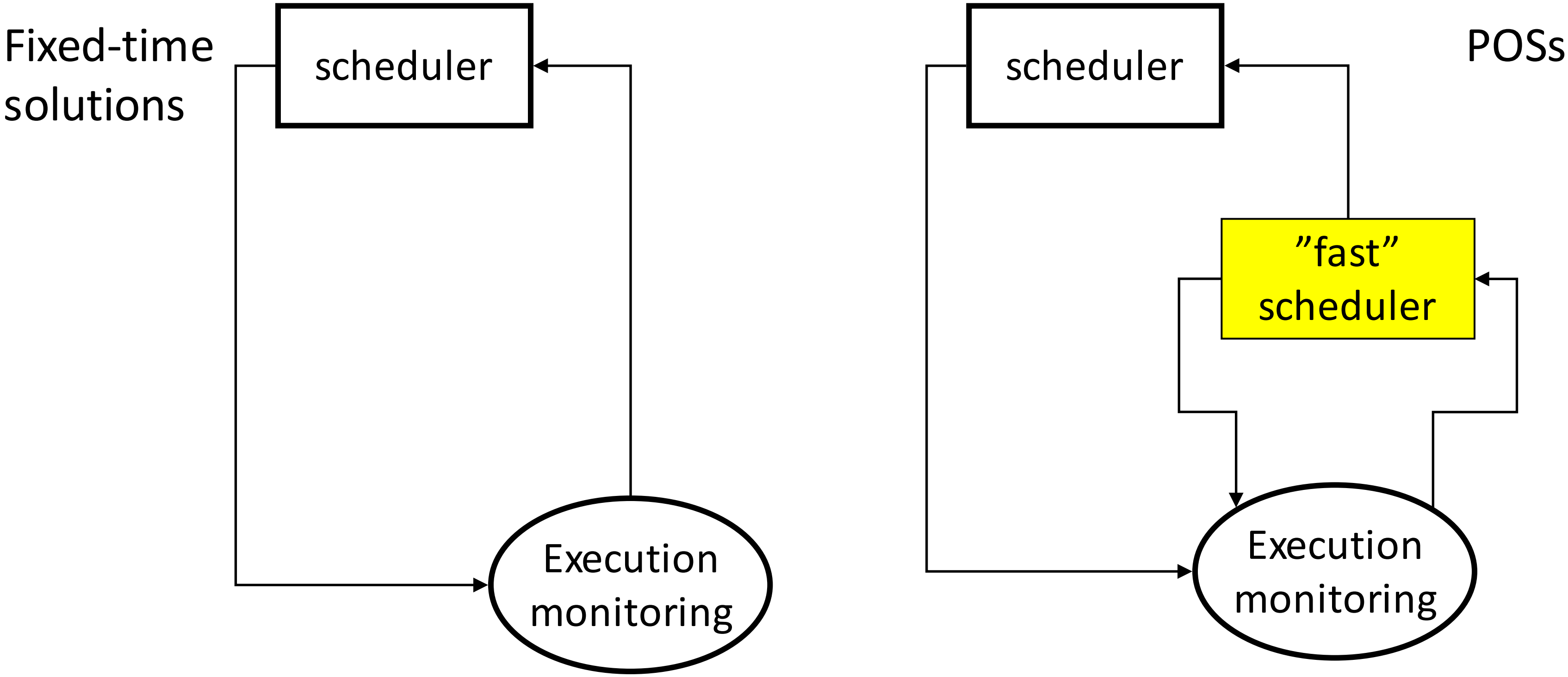
- Given a scheduling problem a Partial Order Schedule (POS) is a set of solutions that can be represented as a temporal graph [any time feasible schedule defined in the graph is also a resource feasible schedule]

- An interesting property of POSs is “fast re-scheduling”:
  - many external changes may have a reactive response accomplished via simple propagation in the underlying temporal network (a polynomial time calculation).

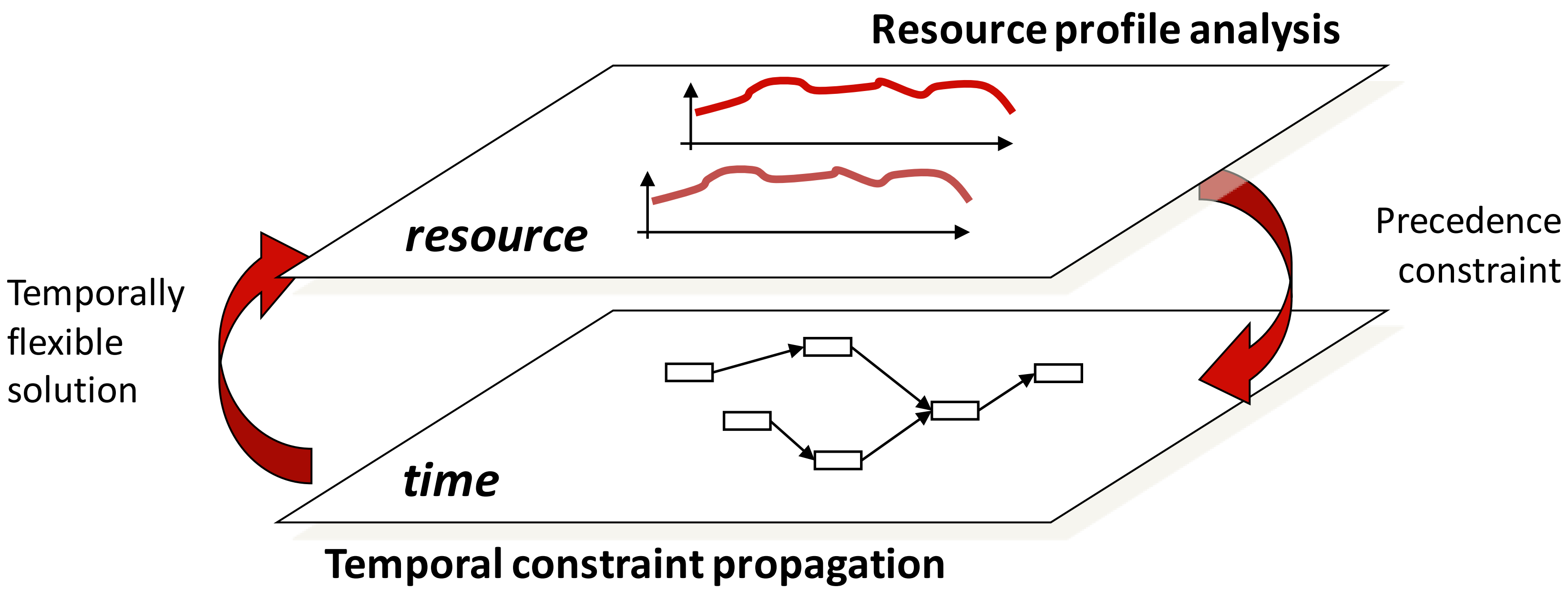


- Sequence activities that compete for resources letting start and end times float

Partial order schedule



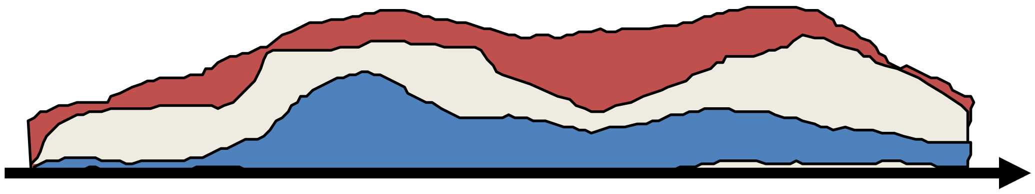
How to compute POSs



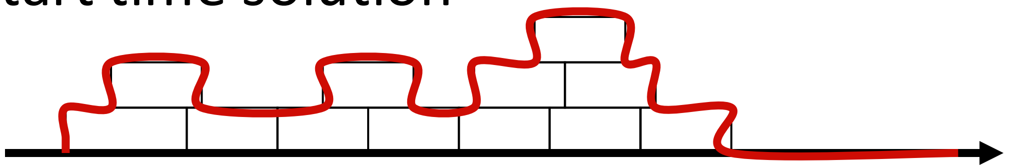
- Variables:
  - start and end times of each activity
- Domains
  - A schedule horizon [0, H]
- Constraints
  - Temporal constraints (e.g., duration of activities, min-max separation between activities, simple precedences)
  - Resource constraints (e.g., bounds on capacities)

Two different resource profiles

- Because of the flexibility of the temporal network an exact computation of demand profile is not possible
- An intuitive compromise consists of computing its upper-bound and lower-bound projections

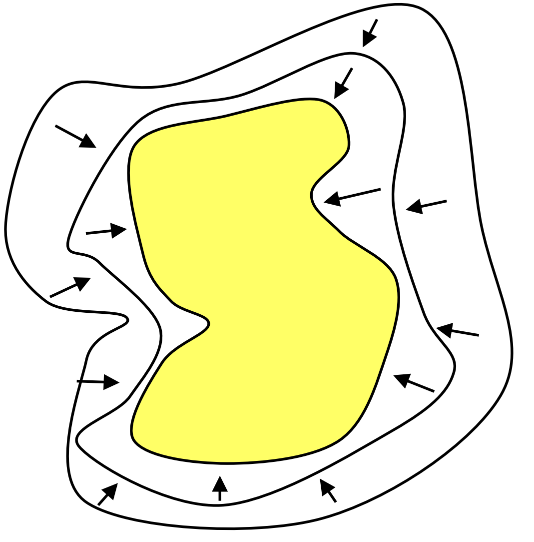


- A different way is to compute the resource profile for a specific point in the search space:
  - Earliest start time solution



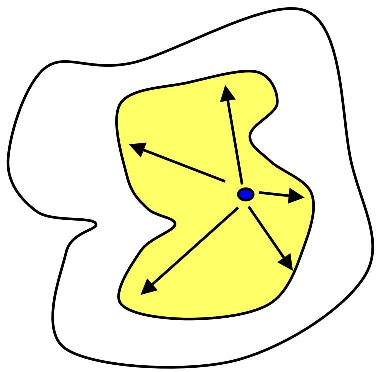
EBA: Envelope Based Analysis

- A least commitment POS generation
- Maintain the flexibility of temporal network
- Compute the tightest possible resource bounds of the “flexible” profile using Resource Envelope computation [Muscettola, 02]

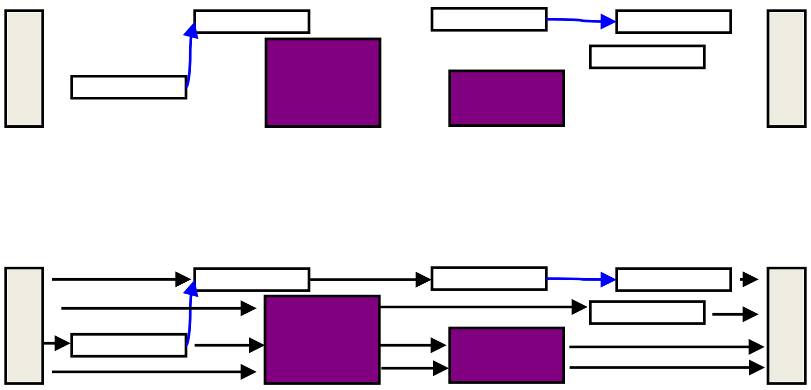


ESTA^C: Earliest Start Time Analysis

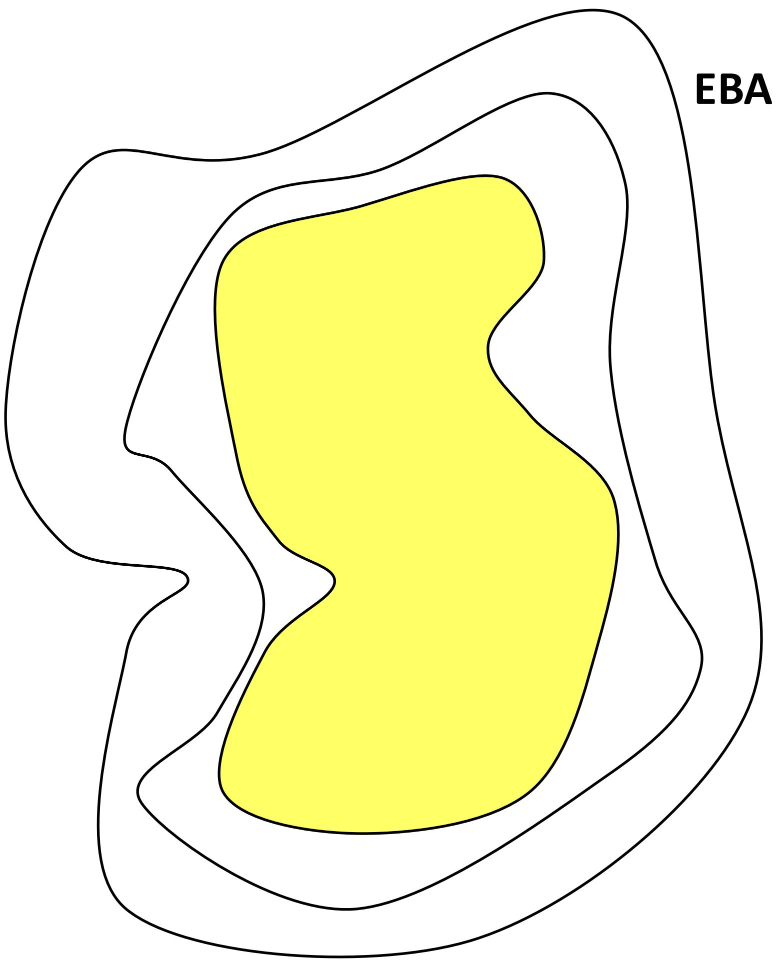
- Computing a Resource Profile in the Earliest Start Time of the temporal network

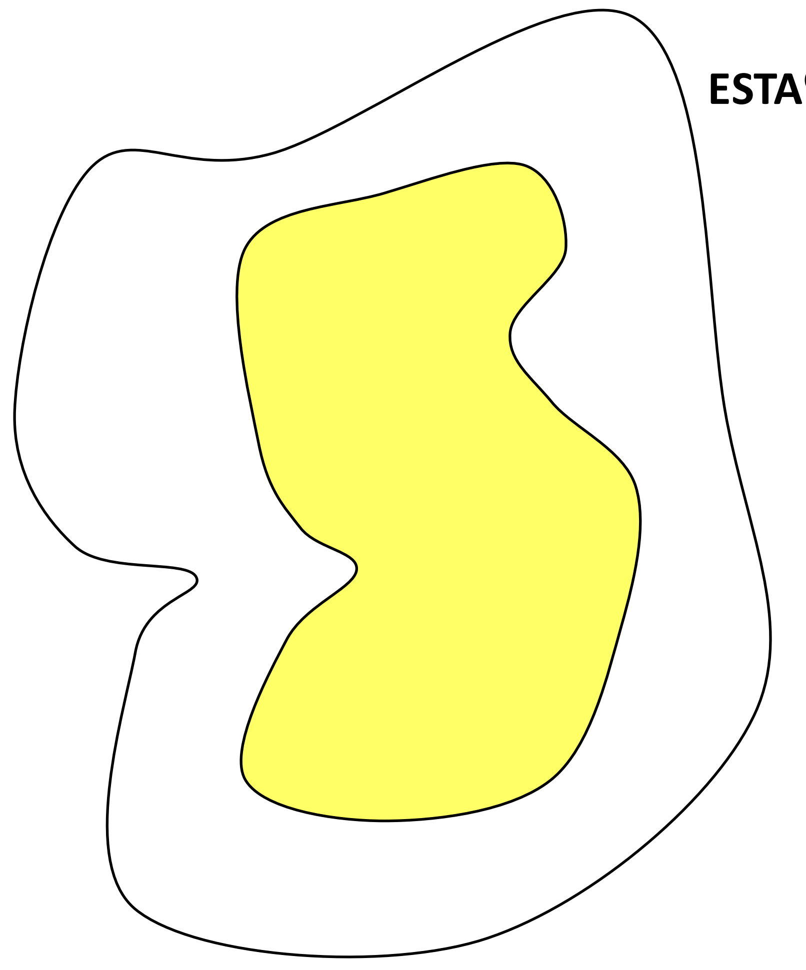


- Two Step process
  - Generate a single solution reasoning on the earliest start time profile
  - Post process the solution transforming it into a POS through chaining



in a nutshell ..





Scheduling with Uncertainty

- Still an open problem
  - Type of scheduling problem has to be considered
  - Type of uncertainty is crucial
    - Execution environment

- For types of uncertainty [De Meyer et al., 2002]:
  - Variation
    - combined small influences (worker sickness, weather, delayed deliveries, unanticipated difficulties of activities, resource unavailability, etc.)
  - Foreseen uncertainty
    - influences that are well-understood but that the project management team cannot be sure will occur (for example, side effects of new drugs)
  - Unforeseen uncertainty,
    - Cannot be identified during project planning. Typically occur in projects that push a technology envelope or enter a new or partially known market
  - Chaos

Scheduling:
Final considerations and Challenges

Building scheduling applications

- Four reasons are given for the lack of success:

1. complexity of knowledge elicitation;
2. uncertainty;
3. difficulties in human–computer interaction;
4. oversimplification of the problem.

Methodological Issues

- How relevant are the problems that are solved?
  - Idealized constraint models
  - Emphasis on easier objective criteria
  - Real problems do not have random structure

- Overcoming these objections is clearly one continuing direction for scheduling research

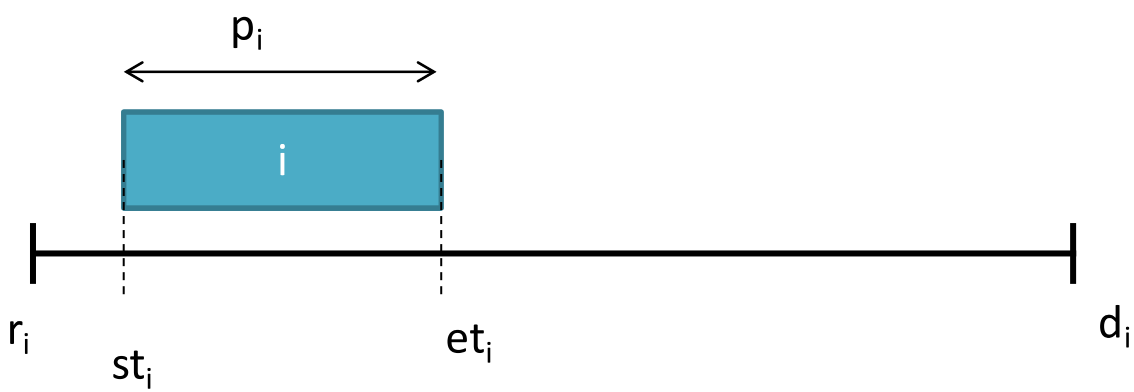
Challenges

- Scheduling models that incorporate richer activity models
- Can integrated P & S problems really be solved as one big optimization task?
- How to achieve tighter interleaving of action selection and resource allocation processes
- Mixed-Initiative Scheduling
- Requirement Analysis

Challenges

- Scheduling models that incorporate richer activity models
- Can integrated P & S problems really be solved as one big optimization task?
- How to achieve tighter interleaving of action selection and resource allocation processes
- Mixed-Initiative Scheduling
- Requirement Analysis

Activity



Inside the activity. Logic ?

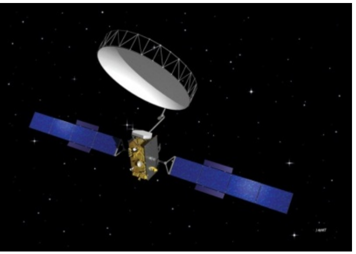
- Part of a process
- A process itself

Another example …

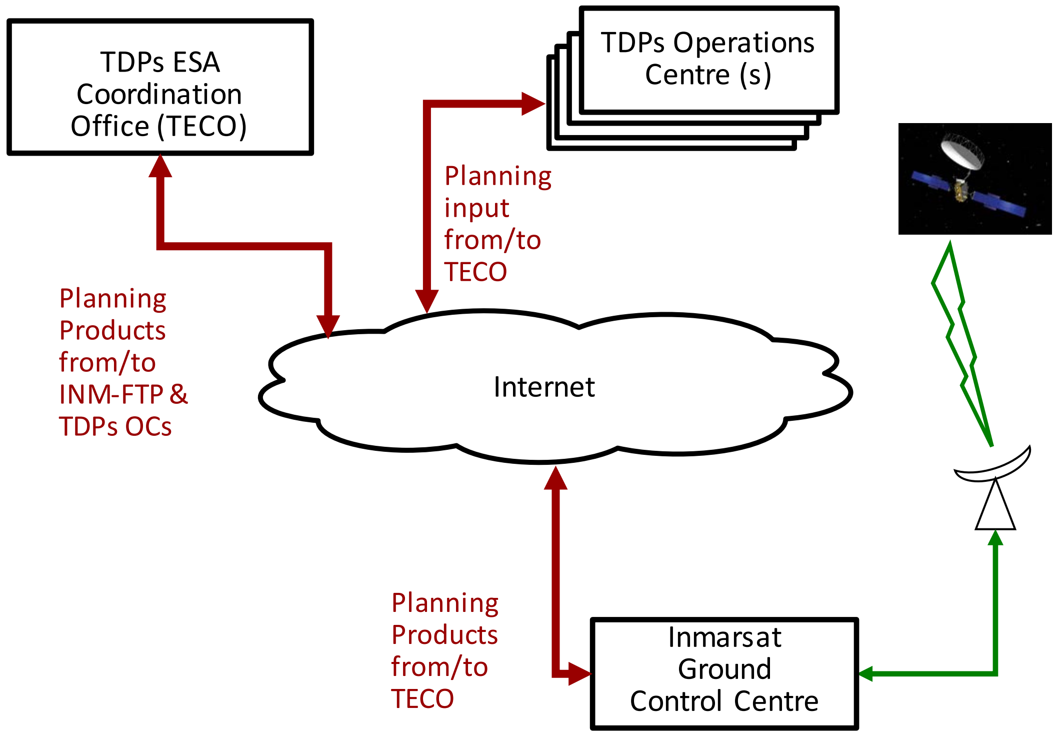
Mission Description

- Alphasat, based on the new Alphabus platform, will be delivered to orbit to be operated by Inmarsat in 2013.
  - It will carry an Inmarsat commercial communication payload
  - 4 Technology Demonstration Payloads (operated as secondary payloads)

- TDPs:
  - An advanced Laser Communication Terminal;
  - A Q-V Band communications experiment;
  - An advanced Star Tracker with active pixel detector;
  - An environment effects facility to monitor the GEO radiation environment and its effects on electronic components and sensors.



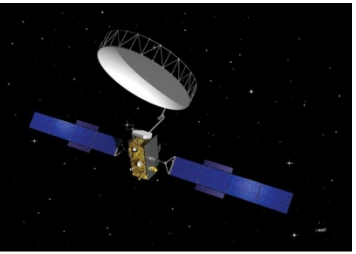
Workflow and planning interfaces



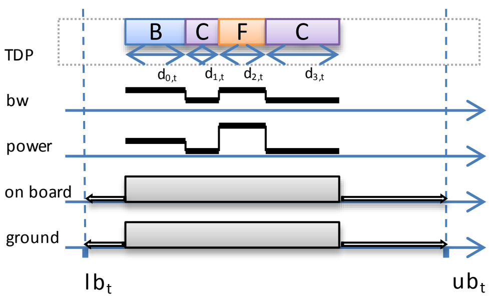
Main goals:

- TDPs activity coordination + planning support
  - Collection of activity/task requests
  - Conflict identification and resolution
  - Generation of the final activity plan

- The planning process is completely automated



Tasks & Constraints



- Task Requests
  - TDP
  - Feasible Time interval
  - Sequence of subtasks <dur, submode, bandwidth, power>
  - “On-board only”/ “on-board + on-ground”
  - A weight value w

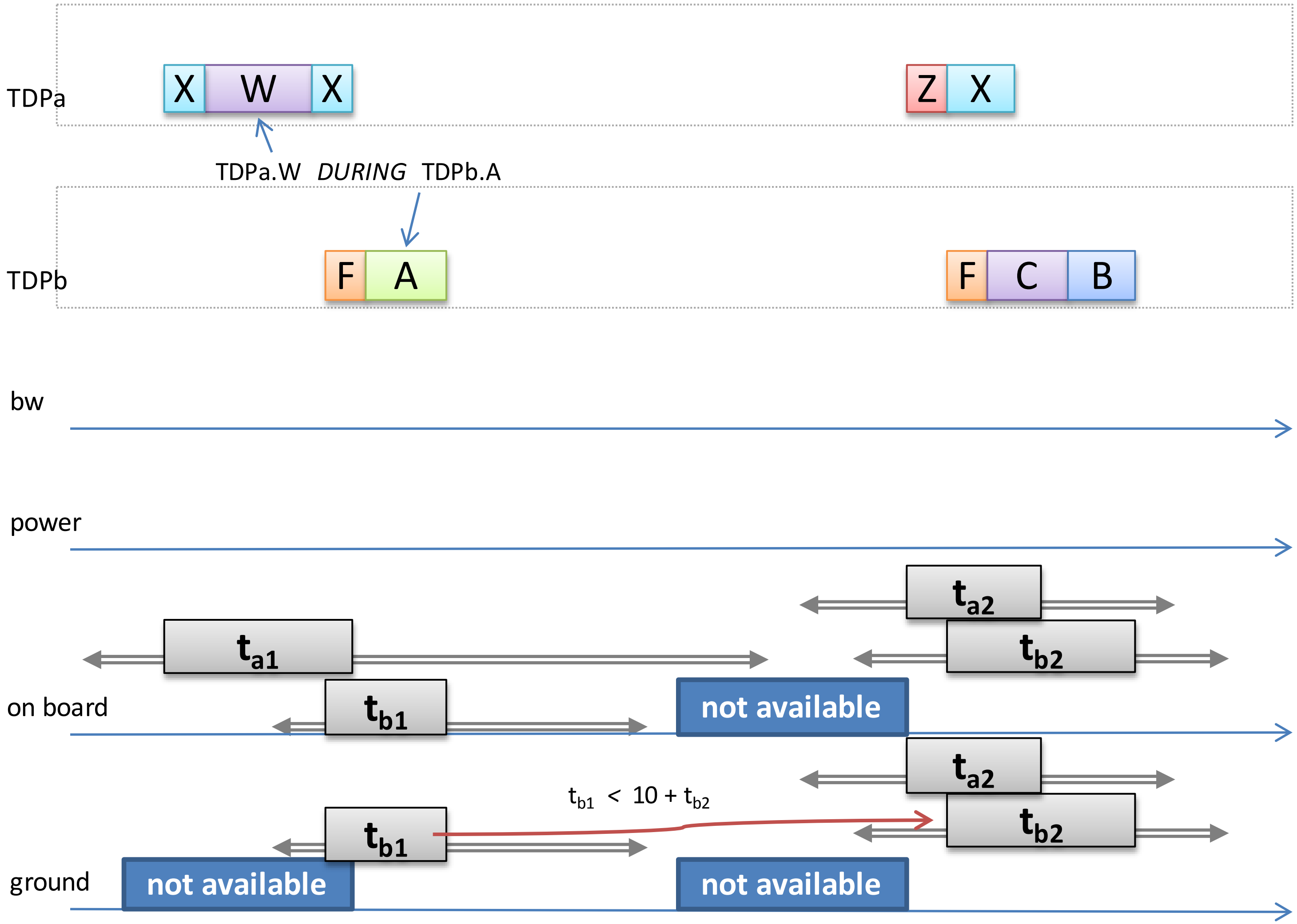
- Constraints
  - Among TDPs – e.g., *TDP1.B during TDP2.Z*
  - TDP vs S/C – e.g., *TDP1.F not-during S/C.manoeuvres*
  - Among tasks – e.g., *tr1<tr2 and allocate(tr1) iff allocate(tr2)*
  - Resource constraints – e.g., bandwidth and power limits

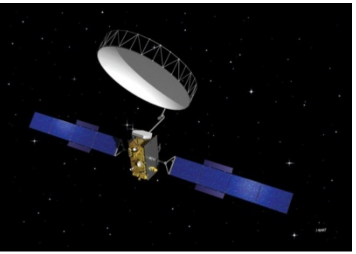
Problem and solution

- Problem:
  - A set of task requests
  - A set if initial submode for each TDP
  - A set of constraints
  - S/C status and availability

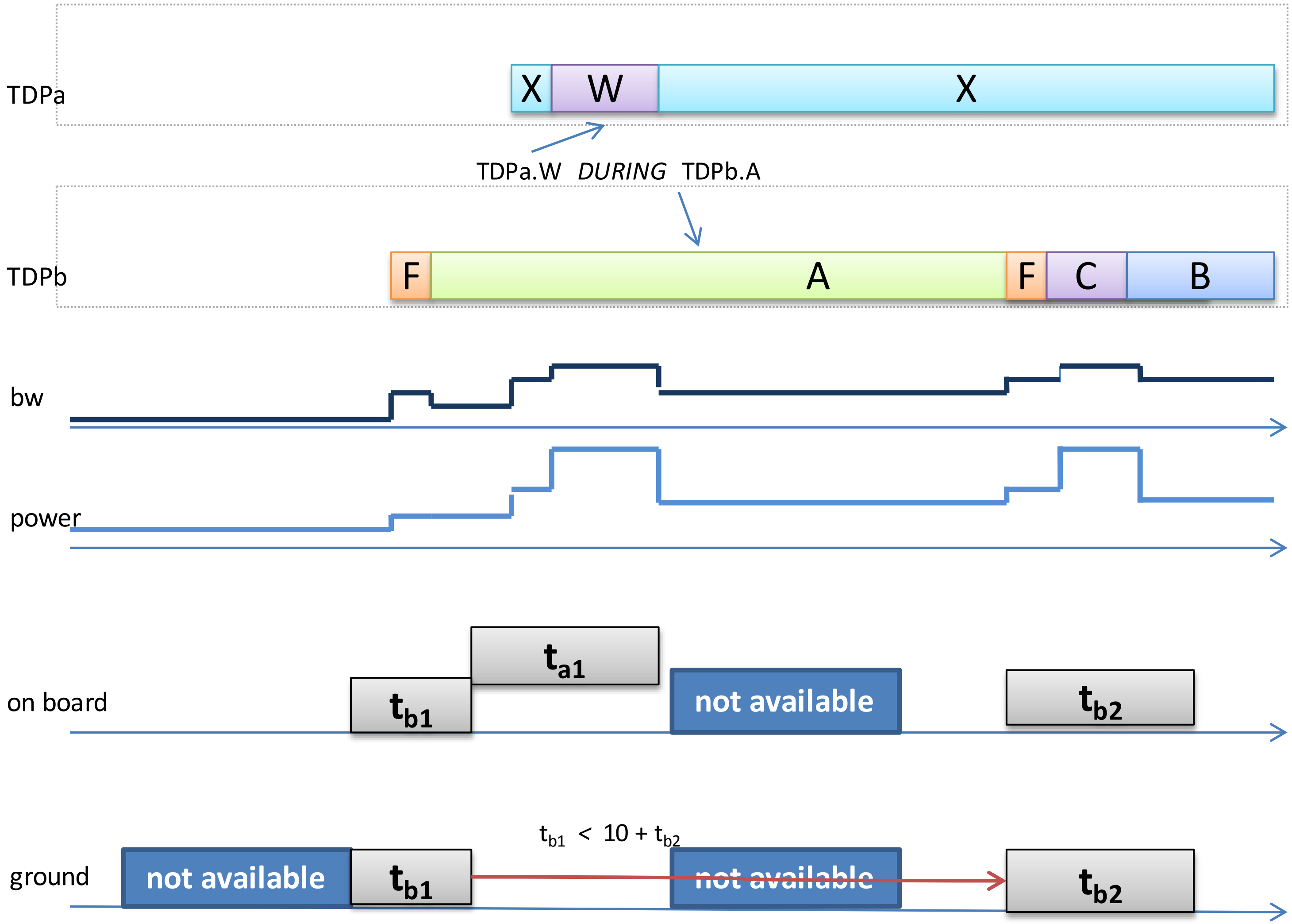
- Solution
  - A set S of allocated tasks
  - Maximize $Value(s) = \sum_{tr \in S} w(tr)$

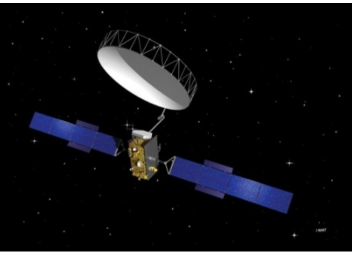
Example – problem





Example – solution



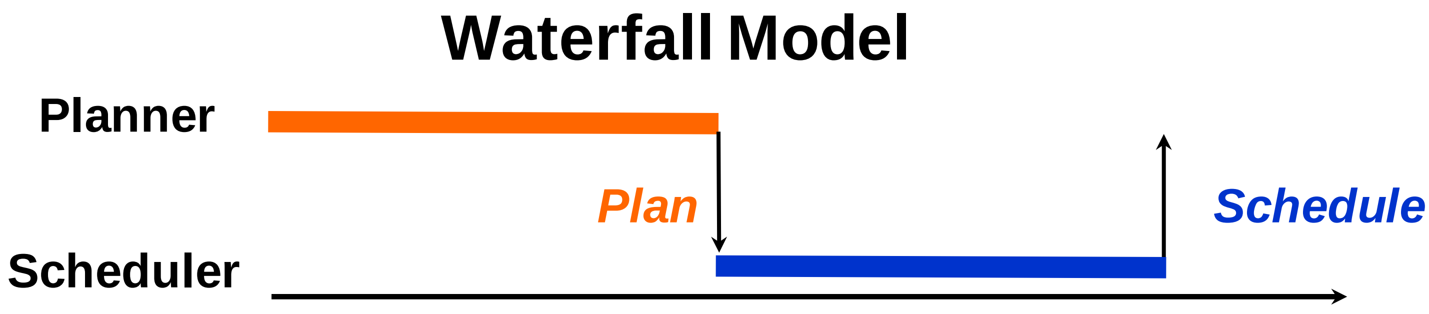


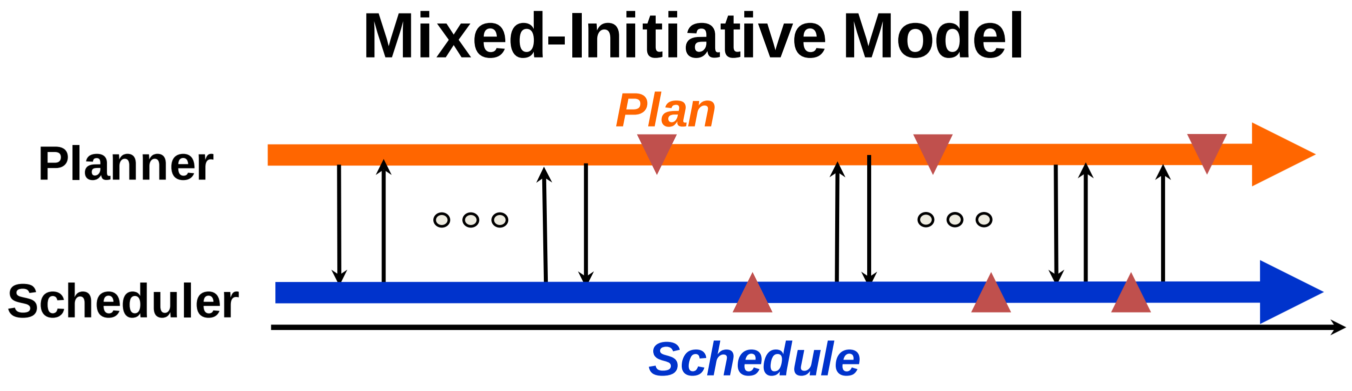
Challenges

- Scheduling models that incorporate richer activity models
- Can integrated P & S problems really be solved as one big optimization task?
- How to achieve tighter interleaving of action selection and resource allocation processes
- Mixed-Initiative Scheduling
- Requirement Analysis

Integrating Planning & Scheduling

"Planning & scheduling are rarely separable"





Challenges

- Scheduling models that incorporate richer activity models
- Can integrated P & S problems really be solved as one big optimization task?
- How to achieve tighter interleaving of action selection and resource allocation processes
- Mixed-Initiative Scheduling
- Requirement Analysis

Mixed-Initiative Scheduling Challenges

- Management of user context across decision cycles
- Explanation of scheduling decisions
  - Why did you do this?
  - Why didn’t you do that?
- Adjustable autonomy

The same example …

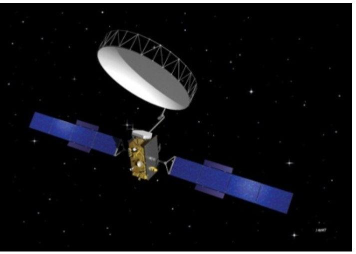
Explanation

Need of generating explanation

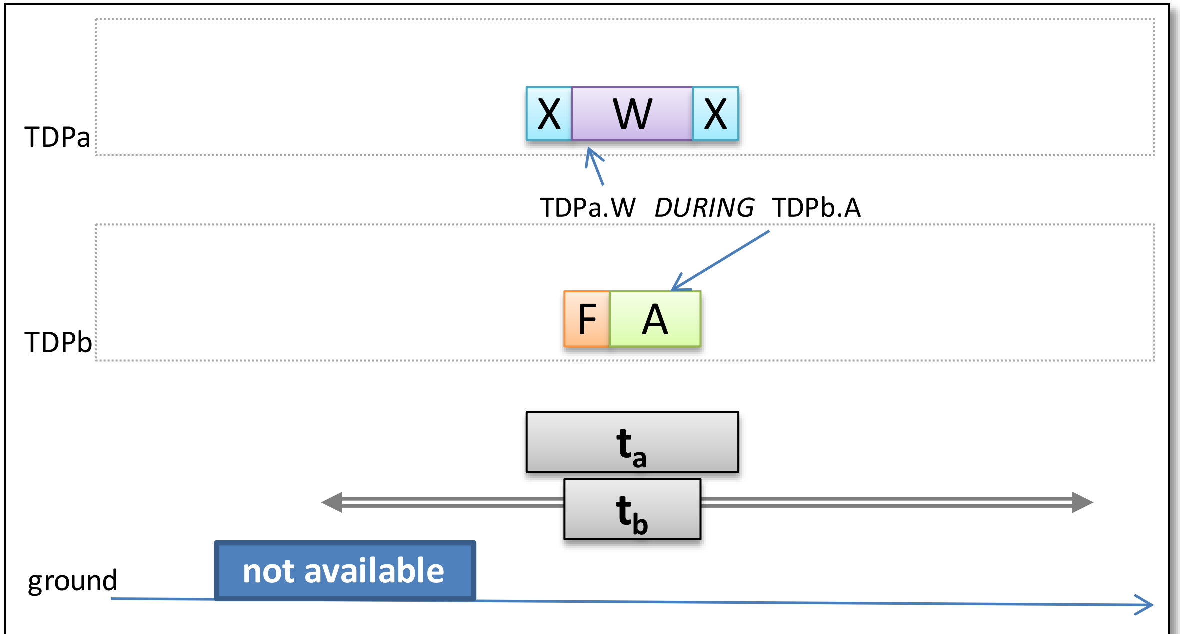
- TECO as the system has been designed to be completely automated
- A proper explanation is also needed to have effective iterations between TDPOCs and TECO.

The approach is based on the following points:

- A “protocol” to exchange information between TDP-OCs and TECO
- “Labeled decisions” with information about the solver and the motivation of the decision.
- An “Explanation Generator” module to generate the information for the system users by applying the given protocol.

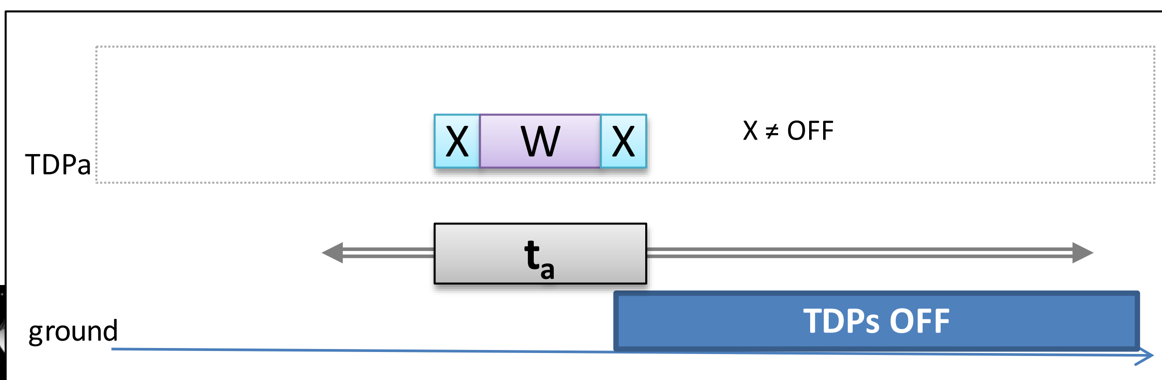


Explanation – examples



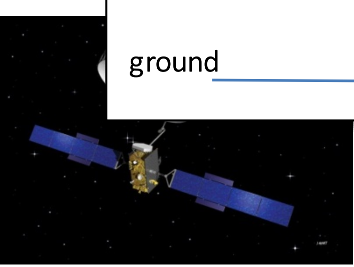
planner: B *before* A
scheduler: *no solution*

Explanation should consider
1) scheduling conflict
2) planner decision
3) domain theory



scheduler: A *before* TDPs OFF
planner: *no solution*

Explanation should consider
1) planning conflict
2) domain theory

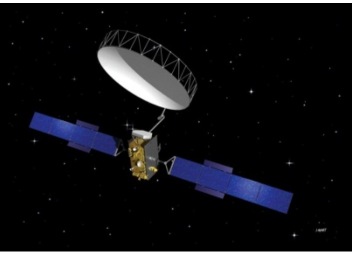


Explanation

- The final explanation is not generated directly by the different solvers
  - the solvers can have only a limited view of the current situation
  - to allow decoupling the set of used solvers from the final explanation generation process (and the associated protocol)

- The approach consists in “tracing back” all the decisions
  - collecting the different annotations added during the solving process.
  - identifying the specific case based both on the solvers and on the content of the different annotations

- Modular approach to facilitate future re-usability and evolution
  - a different type of user accessing the TECO system (e.g., web-client) which can require a different protocol
  - modifying the set of solvers.



Challenges

- Scheduling models that incorporate richer activity models
- Can integrated P & S problems really be solved as one big optimization task?
- How to achieve tighter interleaving of action selection and resource allocation processes
- Mixed-Initiative Scheduling
- Requirement Analysis

Requirements Analysis

- *“Scheduling is really a process of getting the constraints right”*
- Current tools designed around a “Specify and Solve” model of user/system interaction
  - Inefficient problem solving cycle
- Optimization techniques try to solve the problem WHILE a human changes the problem to make it solvable!
- Human scheduler decisions are often based on knowledge not represented in the scheduling problem!
- Mixed-Initiative solution models
  - Incremental solution of relaxed problems
  - Iterative adjustment of problem constraints, preferences, priorities

Research Directions for the Next 10 Years

- Deeper integration of AI and OR techniques
- Robust schedules and scheduling
- Global coherence through local interaction
- Extension to larger-scoped problem-solving processes
- Rapid construction of high performance scheduling services

Thank you !

References

- P. Brucker, A. Drexl, R. Mohring, K. Neumann, and E. Pesch, "Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods," *European Journal of Operations Research*. 1998
- A. Cesta, A. Oddi, and S.F. Smith, "A Constraint-Based Method for Project Scheduling with Time Windows," Journal of Heuristics, n. 8, pp. 109–136, 2002
- A. Cesta, G. Cortellessa, M. Denis, A. Donati, S. Fratini, A. Oddi, N. Policella, E. Rabenau, and J. Schulster, "MEXAR2: AI Solves Mission Planner Problems", in *IEEE Intelligent Systems*, pp. 12-19, issue July/August 2007.
- C. Cheng, and S.F. Smith. "Generating Feasible Schedules under Complex Metric Constraints." In *Proceedings 12th National Conference on AI (AAAI-94)*, 1994
- A. J. Davenport, C. Gefflot, and J. C. Beck. Slack-based Techniques for Robust Schedules. In Proceedings of 6th European Conference on Planning, ECP-01, 2001.
- A. De Meyer, C. H. Loch, and M. Tisch, "Managing project uncertainty: From variation to chaos," *MIT Sloan Management Review*, pp. 60–67, Winter 2002.
- R. Dechter, I. Meiri, and J. Pearl. "Temporal Constraint Networks." *Artificial Intelligence* 49, 61–95, 1991
- H. H. El Sakkout and M. G. Wallace. Probe Backtrack Search for Minimal Perturbation in Dynamic Scheduling. Constraints, 5(4):359–388, 2000.
- J. Erschler, P. Lopez, and C. Thuriot. Temporal reasoning under resource constraints: Application to task scheduling. In Advances in Support Systems Research, 1990
- M. L. Ginsberg, A. J. Parkes, and A. Roy. Supermodels and Robustness. In Proceedings of the 15th National Conference on Artificial Intelligence, AAAI-98, pages 334–339. AAAI Press, 1998.
- P. Laborie and M. Ghallab, "Planning with Sharable Resource Constraints." IJCAI 1995
- R. Leus. The generation of stable project plans. PhD thesis, Katholieke Universiteit Leuven, 2003.
- V. Leon, S. D. Wu, and R. H. Storer. Robustness measures and robust scheduling for job shops. IIE Transactions, 26(5):32–43, September 1994.
- N. Muscettola. "Computing the Envelope for Stepwise-Constant Resource Allocations". In Principles and Practice of Constraint Programming, 8th International
- Conference, CP 2002, 2002.
- A. Oddi and N. Policella, "Improving Robustness of Spacecraft Downlink Schedules", in *IEEE Transactions on Systems, Man, and Cybernetics (Part C)*, vol. 37, no. 5, pp. 887-896, 2007.
- N. Policella, "Scheduling with uncertainty — A proactive approach using partial order schedules," Ph.D. Thesis, Italy: Università degli Studi di Roma "La Sapienza", 2005.
- N. Policella, A. Cesta, A. Oddi, and S.F. Smith, "From Precedence Constraint Posting to Partial Order Schedules: A CSP approach to Robust Scheduling", in *AI Communications*, vol. 20, no. 3, pp. 163-180, 2007.
- N. Policella, A. Cesta, A. Oddi, and S.F. Smith, "Solve-and-robustify: Synthesizing partial order schedules by chaining", in Journal of Scheduling, 2008
- N. Policella, H. Oliveira, and E. Benzi, "Planning Spacecraft Activities: An automated approach", in ICAPS 2013
- S.F. Smith and C. Cheng, "Slack-Based Heuristics for Constraint Satisfaction Scheduling." In *Proceedings 11th National Conference on AI (AAA1-93)*, 1993
- S. F. Smith. Reactive Scheduling Systems. In D. Brown and W. Scherer, editors, Intelligent Scheduling Systems. Kluwer Academic Publishers, 1994.
- D. Weld, An Introduction to Least Commitment Planning. In AI Magazine, 1994
- S. D. Wu, E. S. Beyon, and R. H. Storer. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. Operations Research, 47(1):113–124, January – February 1999.