

---

# **MEXAR**

## **Integrated AI Technologies to Support MARS EXPRESS Mission Planning**

**Amedeo Cesta, Angelo Oddi, Gabriella Cortellessa and Nicola Policella**

### **PST**

Planning and Scheduling Team

**ISTC-CNR, National Research Council of Italy  
Institute for Cognitive Science and Technology**  
Viale Marx 15, I-00137 Rome, Italy  
{cesta, oddi, corte, policella}@ip.rm.cnr.it  
<http://ip.rm.cnr.it>



---

## **Outline**

- ✍ Generalities
- ✍ MEX-MDP: Problem Formulation
- ✍ A CSP Approach to Problem Solving
- ✍ MEXAR the Interactive Problem Solver
- ✍ Conclusions



---

## Outline

- ▶ **Generalities**
- ▶ MEX-MDP: Problem Formulation
- ▶ A CSP Approach to Problem Solving
- ▶ MEXAR the Interactive Problem Solver
- ▶ Conclusions



---

## Goals of the Study

1. Select a mission planning problem in MARS EXPRESS
2. Study algorithms for its solution
  - Applying AI Techniques
3. Develop a software demonstrator that integrates such algorithms



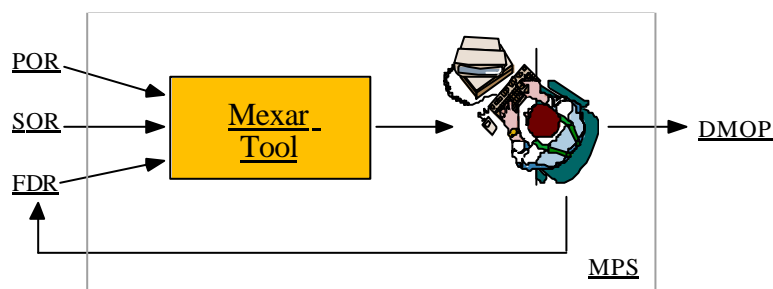
## Integrated Artificial Intelligence Techniques

- Planning and Scheduling
- Constraint Satisfaction Problem Solving (CSP)
- Intelligent Interaction

**Is it possible to solve a realistic problem?**



## Mission Planning System Work Cycle



Mostly a hand made activity  
(with consultation of  
supporting tools)



Initial Idea: An interactive tool  
for incremental refinement  
of decisions

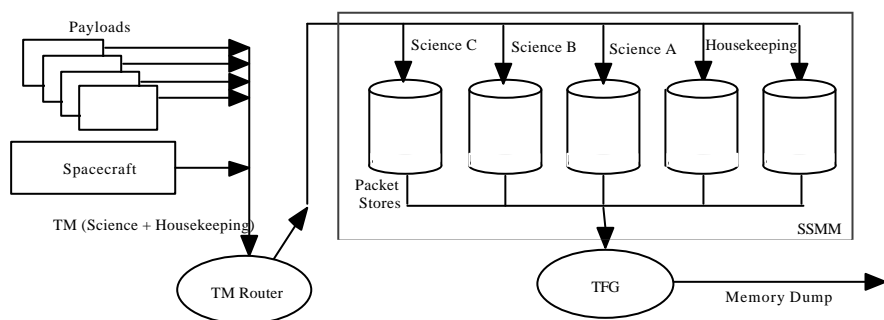


## Outline

- ▶ Generalities
- ▶ **MEX-MDP: Problem Formulation**
- ▶ A CSP Approach to Problem Solving
- ▶ MEXAR the Interactive Problem Solver
- ▶ Conclusions



## The selected problem: Generating memory dump sequences



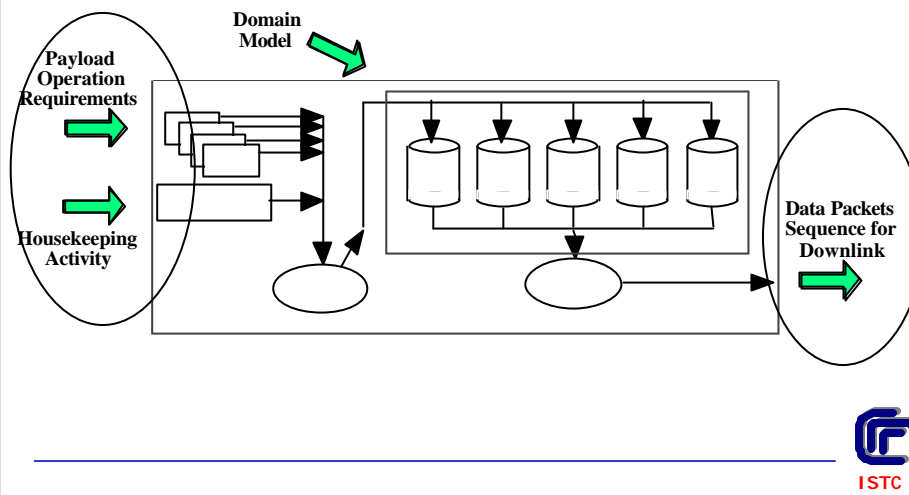
A person is specifically responsible for generating SORs activities to dump science and HK to Earth



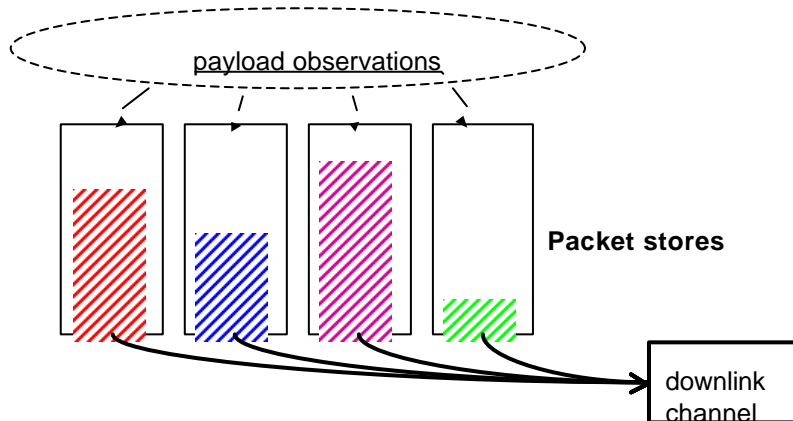
A tool for automated generation of dump SORs?  
(the data packets on the downlink)



## Focusing on Input/Output



## MEX-MDP: MARS EXPRESS Memory Dumping Problem

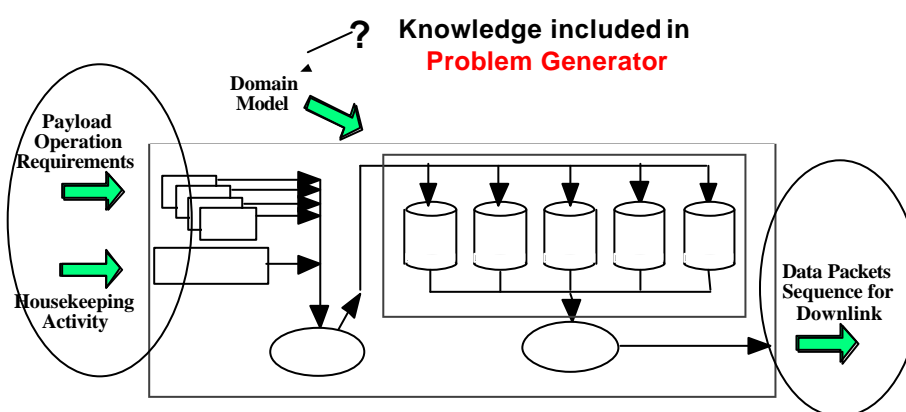


## Domain Model

- Resources
  - Payloads
  - Packet Stores
  - Communication Channel(s)
- Activities
  - Payload Operations
  - Memory Dumps
  - Continuous Data Stream



## Focusing on Input/Output

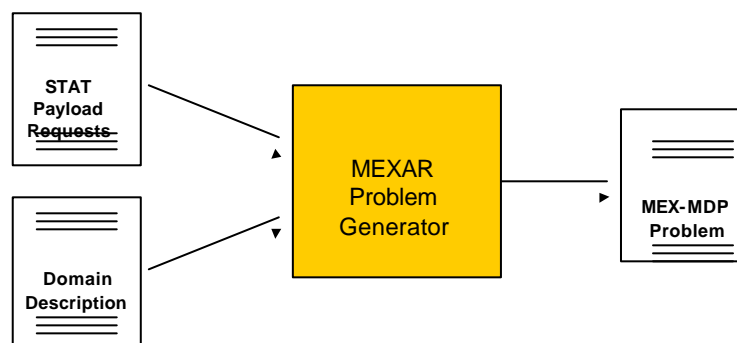


## Problem Generation by “Tuning” Domain Descriptions

- A set of well founded **POR files** from the STAT distribution
- The **Domain Description file** that contains parameters for:
  - Payload description
  - Data decomposition model
  - Packet stores for science
  - Packet stores for spacecraft data

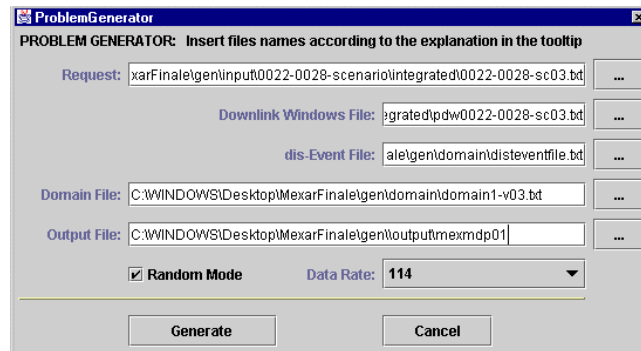


## The Problem Generator (1)



## The Problem Generator (2)

Now Integrated in the Java code



Depends on:

- POR files (now from STAT)
- Domain Specification



## Outline

- ▶ Generalities
- ▶ MEX-MDP: Problem Formulation
- ▶ **A CSP Approach to Problem Solving**
- ▶ MEXAR the Interactive Problem Solver
- ▶ Conclusions





## A CSP (Constraint-Satisfaction Problem) Model for MEX-MDP

- MEX-MDP can be modeled as a special case of the Constraint Satisfaction Problem
- An instance of CSP involves
  - a set of **Decision Variables**  $X = \{X_1, X_2, \dots, X_n\}$ ,
  - a **Domain** of possible values  $D_i$  for each variable, and
  - a set of **Constraints**  $C = \{C_1, C_2, \dots, C_q\}$ , such that  $C_j ? D_1 ? D_2 ? \dots ? D_n$
- A **Solution** is an assignment of domain values to all variables consistent with all the constraints  $C_j$



## Basic CSP Search Procedure

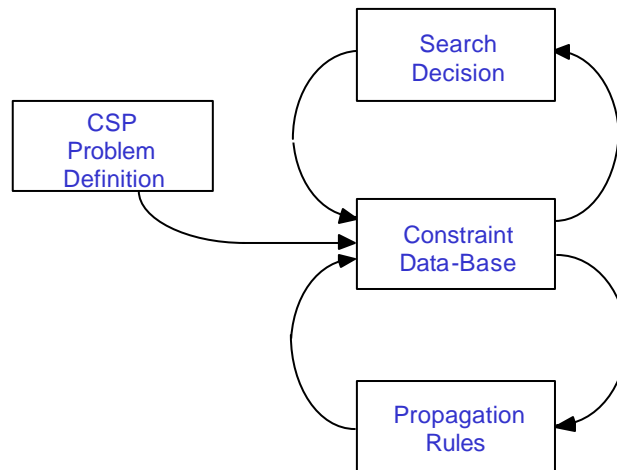
**CSP solver**(problem)

```
while not (problem solved or infeasible) {  
  - Remove inconsistent values from the set of  
    domains  $D_i$  (Propagation)  
  - Select a decision variable  $X_i$   
  - Select a value for the variable  $X_i$  in  $D_i$   
}
```

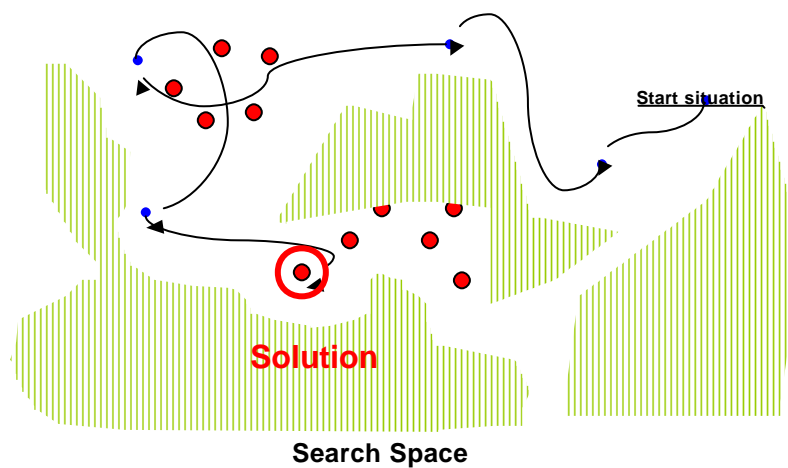
A problem is **solved** when all the decision variables are assigned. It is **infeasible** when at least one domain  $D_i$  becomes empty.



## Problem Solving with CSP (Constraint-Satisfaction Problem)



## CSP Algorithms: theory

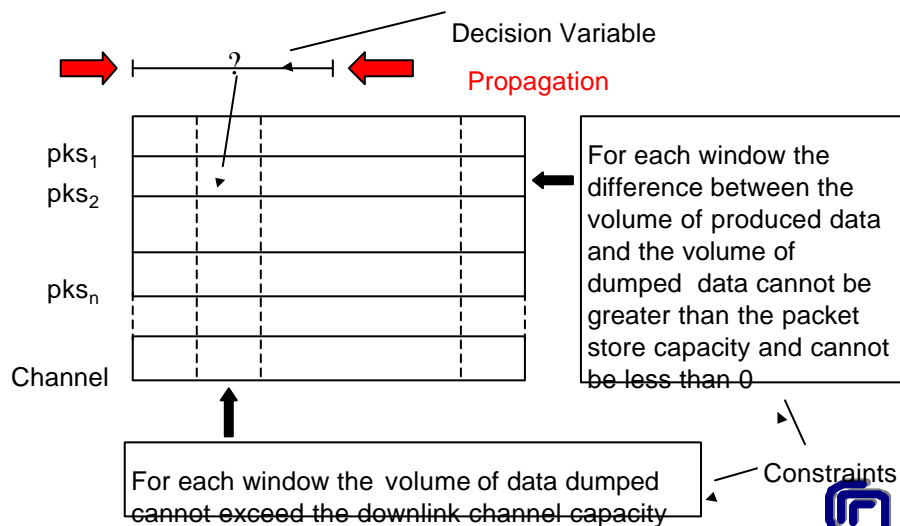


## CSP modelling for MEX-MDP (1/2)

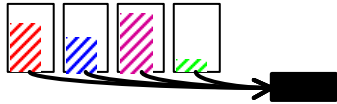
- For each packet store we consider a different timeline
- The temporal horizon is subdivided in contiguous time slots (windows) such that instantaneous memory operations can happen only at the edges
- Decision Variables represent the volume of data dumped within each time slots
- Two types of constraints:
  - the ones imposed by the channel and
  - the ones by the packet stores capacity



## CSP modelling for MEX-MDP (2/2)



## An Example of Representation



The communication channel is divided in temporal intervals

6	6	4		6	6
---	---	---	--	---	---

For each packet store we represent the amount of packet store data to dump in each different interval

2				6	
4					
	6				4
		3			



## A Basic Solver

A Greedy Algorithm



## A Greedy CSP solver for MEX-MDP

- This procedure works at two levels of abstraction:
  - **Planning level:** all the decision variables are assigned according to a set of priority rules designed for MEX-MDP. This level only considers data volume and the related constraints
  - **Scheduling level:** within each time slots a sequence of memory dumps operations is generated according to the constraints imposed by the decision variables and a set of priority rules. This level takes into account the data packetization



## A Greedy CSP solver for MEX-MDP (1/2)

**SmmDump-Planning**(Mex-Mdp)

$i = 0$

*Propagation()*

**for each** *downlink window*( $i$ ) {

**while** (dump capacity is available in *downlink window*( $i$ ))

    {

        - **Select** a packet store  $\mathbf{pks}_j$  with a priority rule

        - **Select** a value for the decision variable  $?_{ij}$

        - *Propagation()*

    }

$i = i + 1$

}

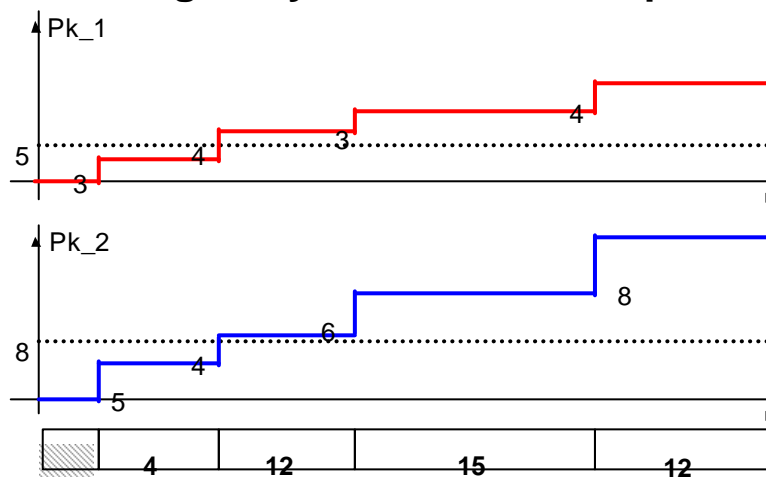


## A Greedy CSP solver for MEX-MDP (2/2)

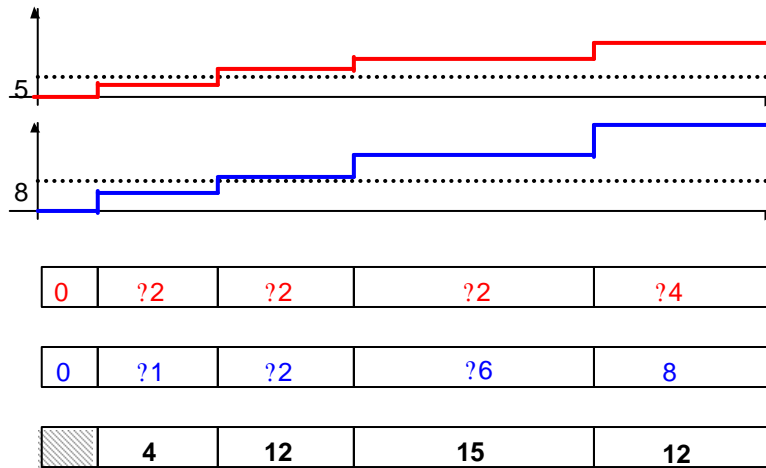
- The procedure ends with failure when the *Propagation* function detects an inconsistency
- A one pass strategy: it assigns decision variables in increasing order of time
- Two planning priority rules:
  - **CFF** (Closest to Fill First): select the packet store with the highest data volume
  - **HPF** (Highest Priority First): selects the packet store with the highest priority



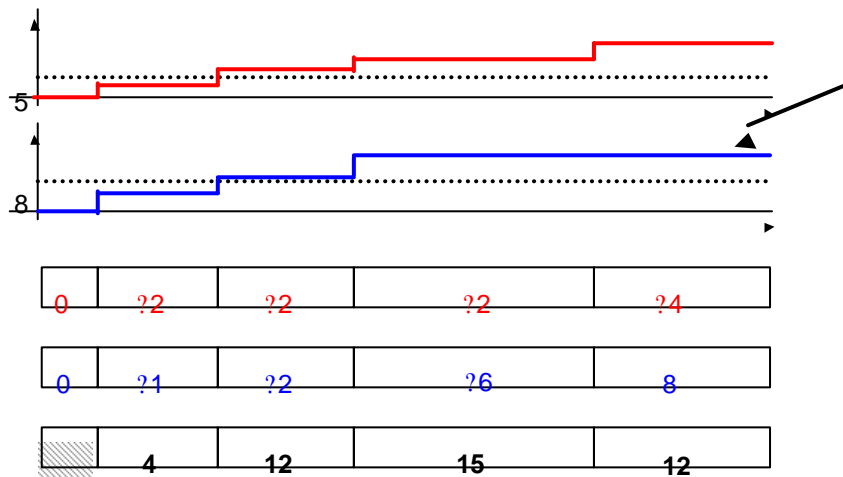
### The greedy solver: an example



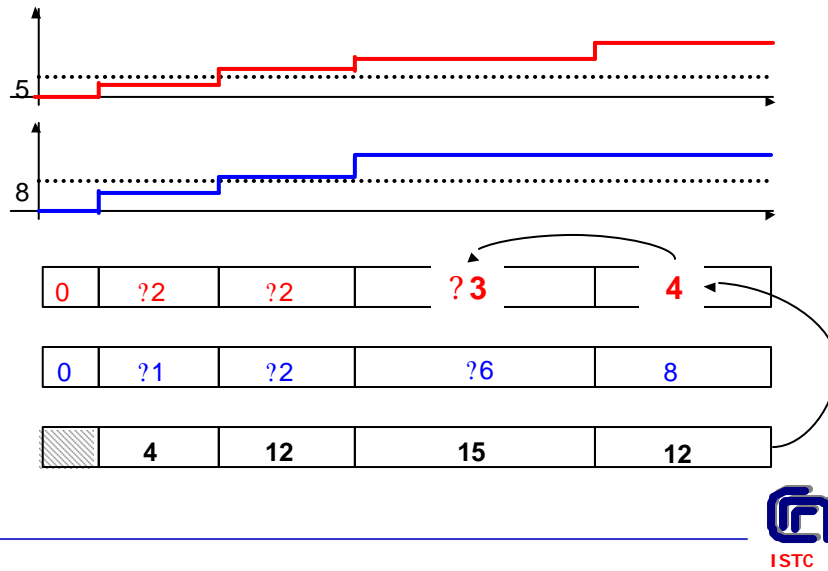
## The greedy solver: an example



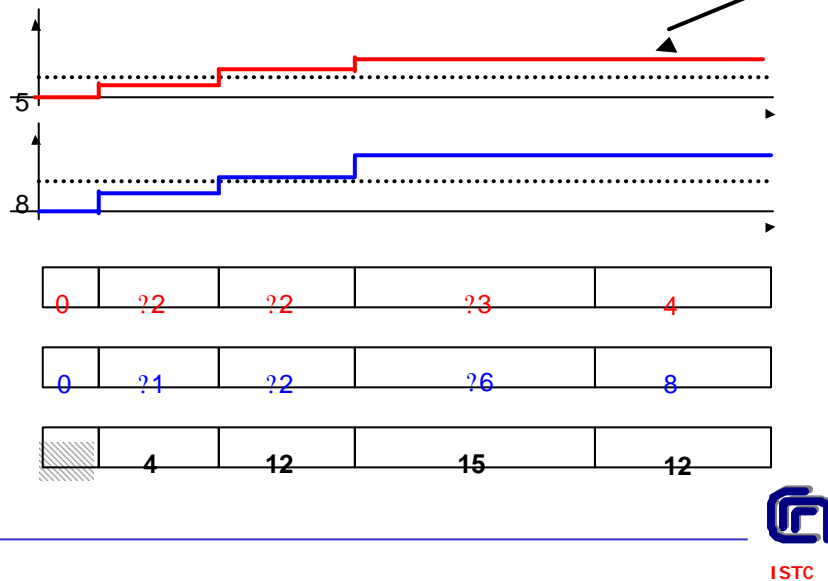
## The greedy solver: an example



## The greedy solver: an example

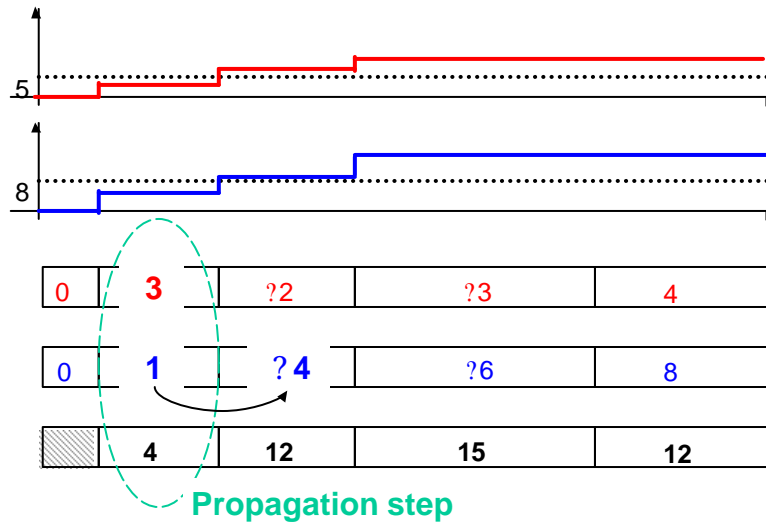


## The greedy solver: an example

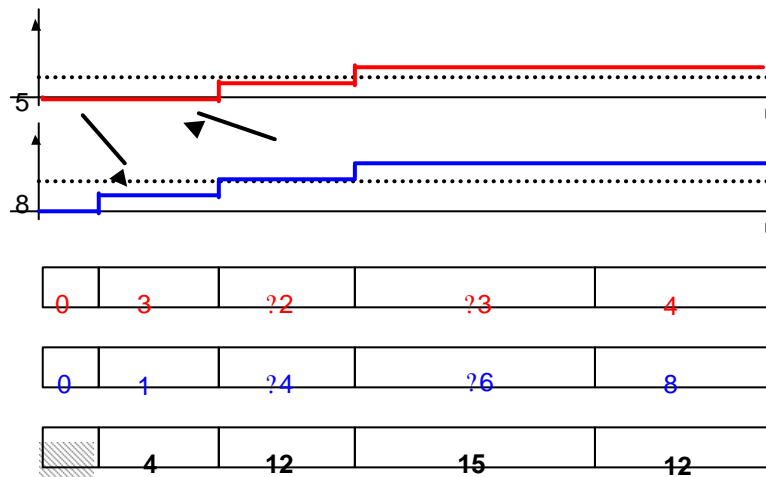




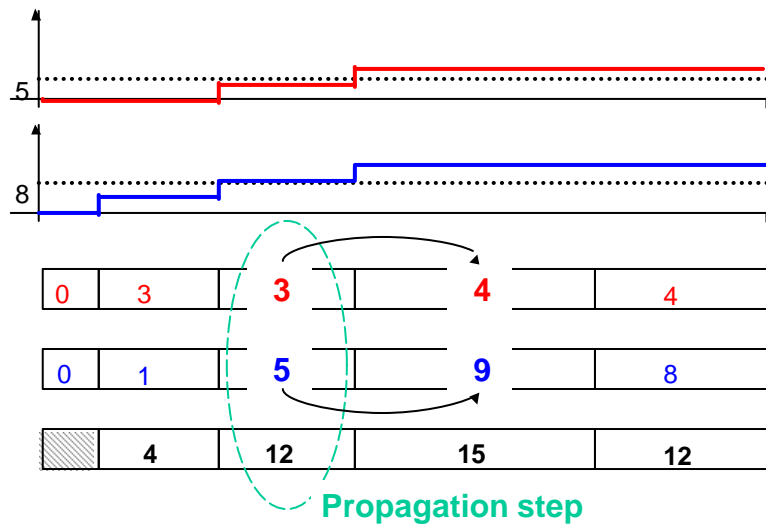
## The greedy solver: an example



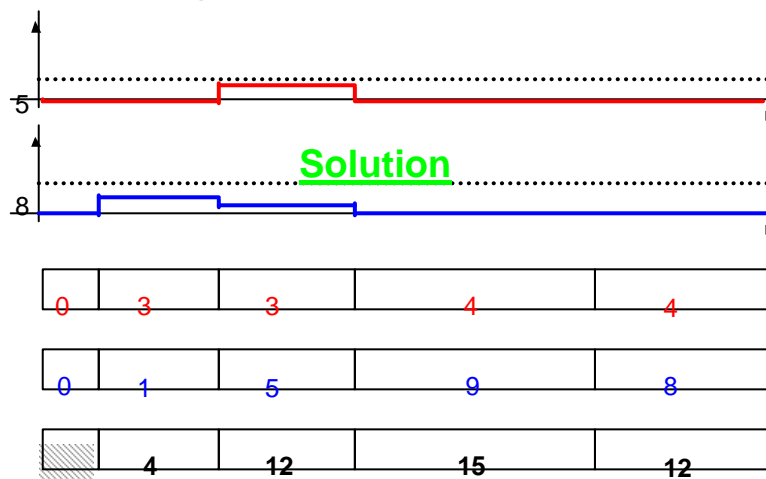
## The greedy solver: an example



## The greedy solver: an example



## The greedy solver: an example



---

## Optimization

### Using Local Search



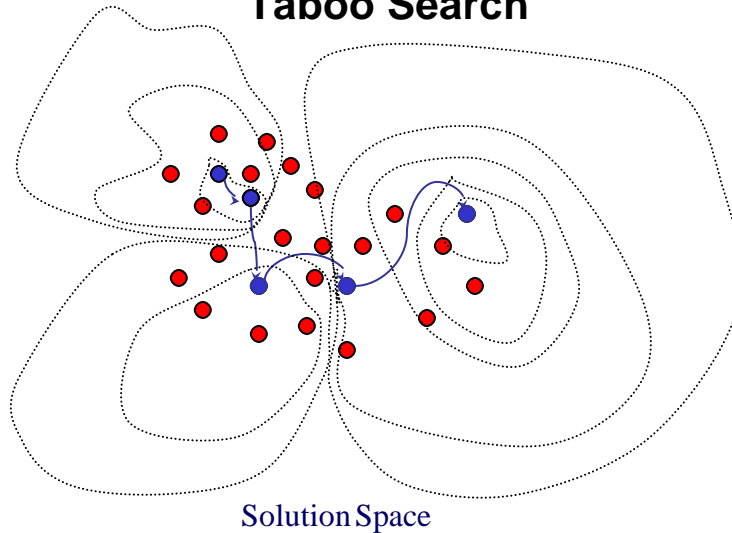
---

## Local Search

- **Taboo search** is a local search approach recently applied to an increasing number of combinatorial optimisation problems
- It is based on the notion of **move**. A move is a function which transforms one solution into another
- For any solution  $S$ , a subset of moves applied to  $S$  is computed such that a subset of solutions called the **neighbourhood** of  $S$  is generated



## Taboo Search



## Taboo Search (2)

- The algorithm starts from an initial solution  $S_0$ .
- At each *step*  $i$  searches the neighbourhood of the current solution  $S_i$ .
- Find the solution  $S_{i+1}$  with the best value of a given objective function.  $S_{i+1}$  **becomes the new current solution** and the process is iterated to find a new best neighbour
- During the search  $S^*$  is the current best solution

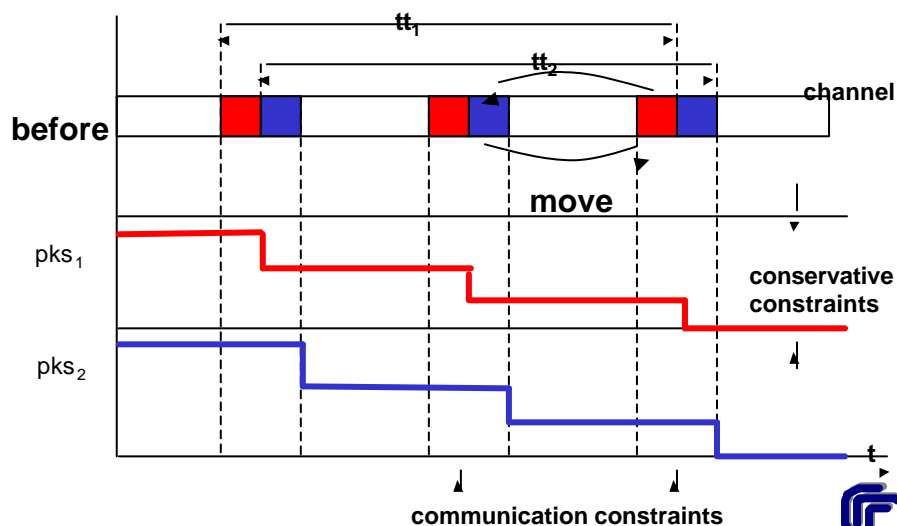


## Taboo Search (3)

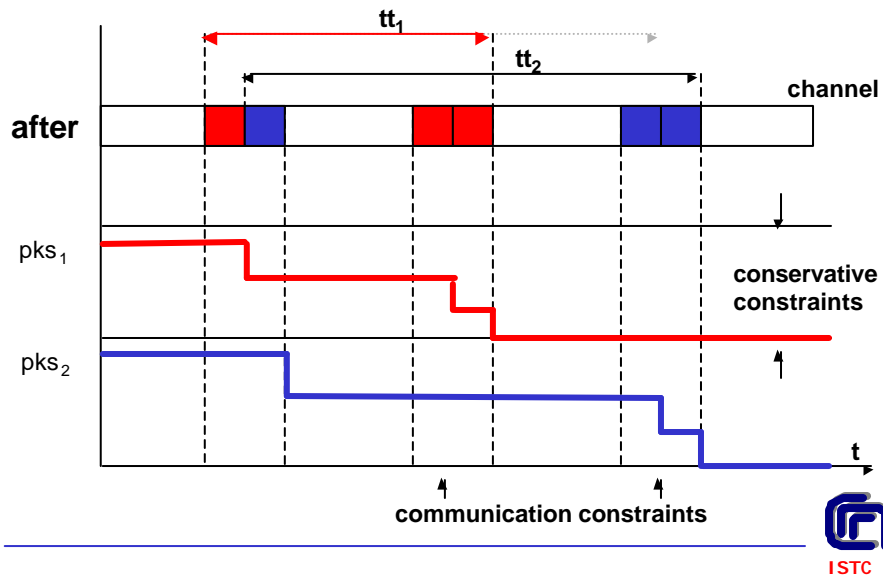
- The search stops when the algorithm performs  $Max\_iter$  moves without improvements on  $S^*$
- To prevent cycling, it is not allowed to turn back to solutions visited in the previous  $MaxSt$  steps. A taboo list is defined with the last  $MaxSt$  moves (managed as a *first-in-first-out* queue)



## Key aspect: the move definition in MEX-MDP



## Key aspect: the move definition in MEX-MDP



## Summary on Problem Solving

- A CSP Model for MEX-MDP
- Propagation Rules
- Greedy Solver
- Taboo Search

**Evaluation?**

---

## Evaluation Functions

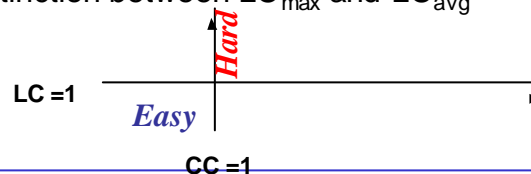
- MTT  
Mean Turnover Time
- DWMTT  
Data Weighted Mean Turnover Time
- PWMTT  
Priority Weighted Mean Turnover Time



---

## Classifying Benchmarks

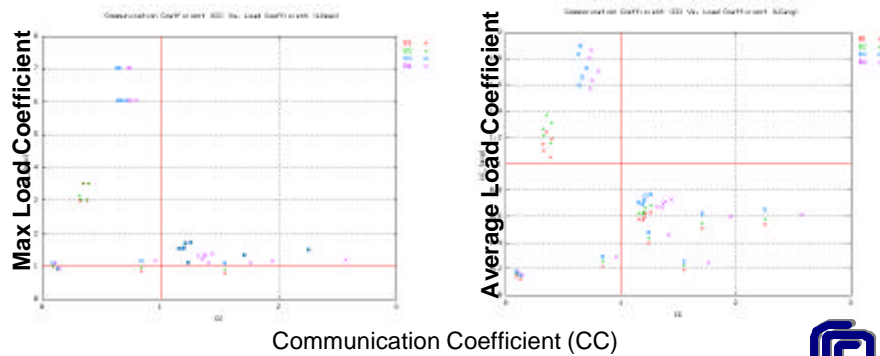
- **CC** - Communication Coefficient
  - Ratio bw. Amount of produced data and Transmission capacity on the same interval of time
- **LC** - Load Coefficient
  - Ratio bw. Amount of produced data and Storing capacity on packet stores
  - Distinction between  $LC_{max}$  and  $LC_{avg}$



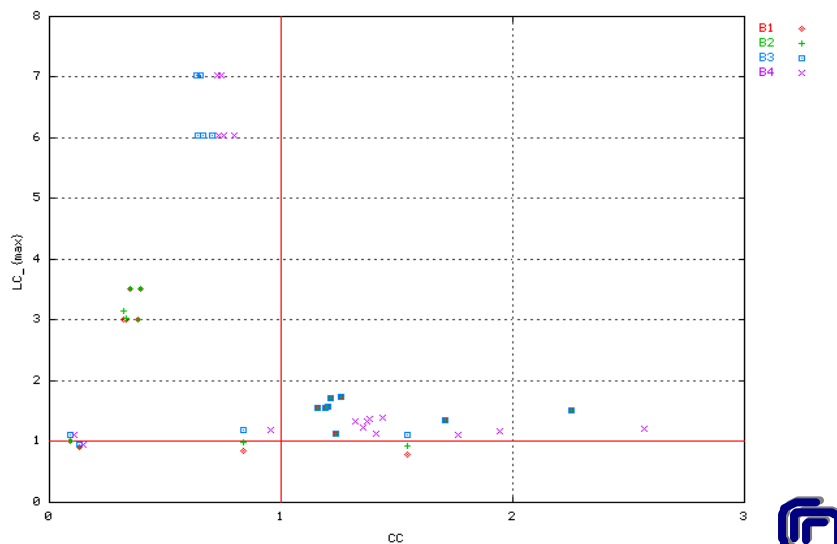
## Different Types of Problems

Four different benchmark sets: B1, B2, B3 and B4

Communication Coefficient (CC) Vs. Load Coefficient

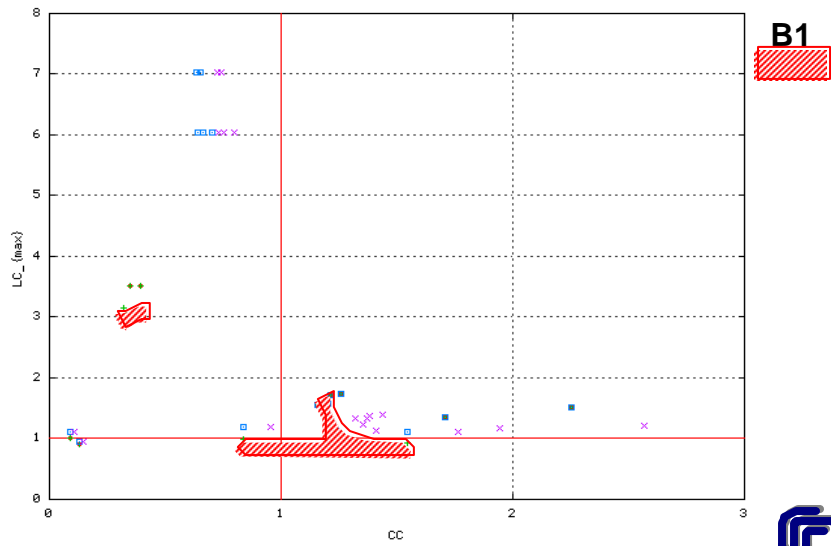


Communication Coefficient (CC) vs. Load Coefficient ( $LC_{max}$ )

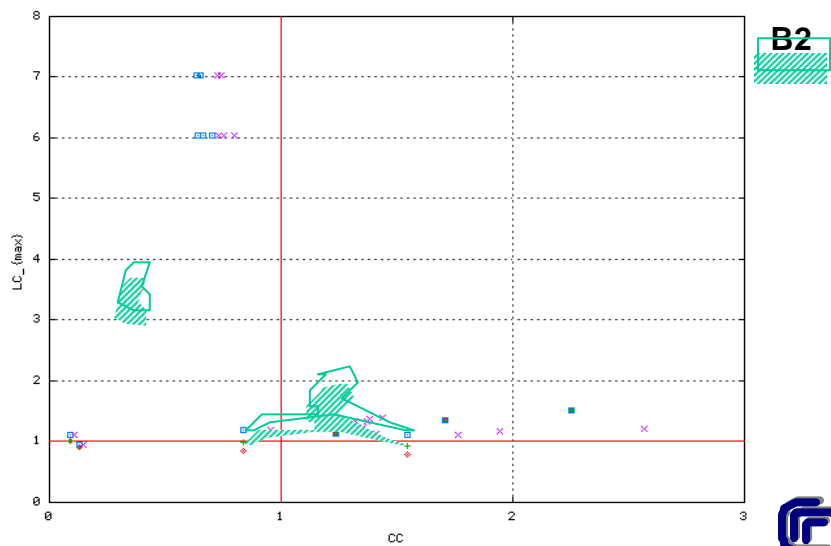




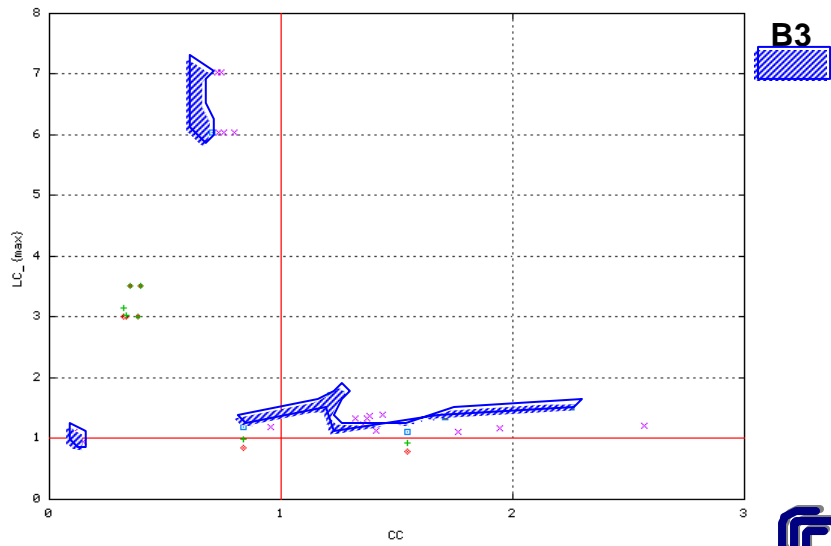
Communication Coefficient (CC) vs. Load Coefficient ( $LC_{max}$ )



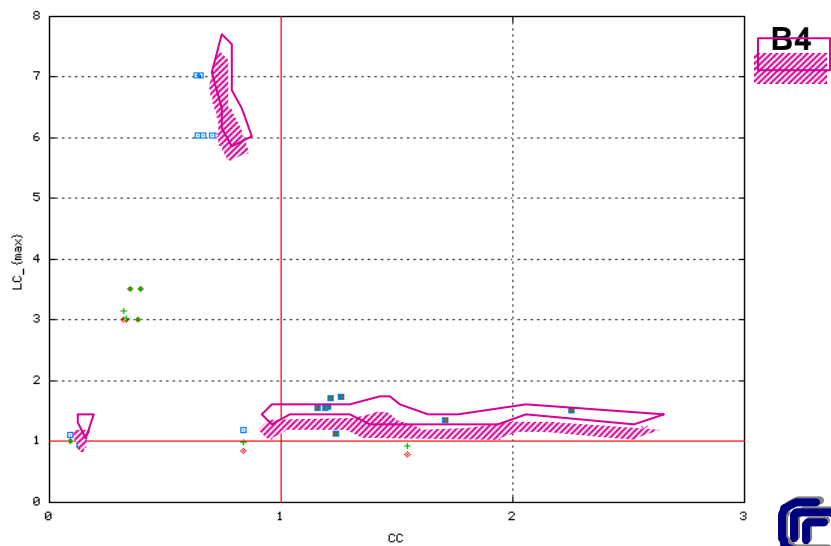
Communication Coefficient (CC) vs. Load Coefficient ( $LC_{max}$ )



Communication Coefficient (CC) vs. Load Coefficient ( $LC_{max}$ )



Communication Coefficient (CC) vs. Load Coefficient ( $LC_{max}$ )



---

## **A Lower Bound for MTT**

- Obtained by relaxation of the capacity constraints in the packet store
- Reduction to a scheduling problem that minimizes the Mean Flow Time (with preemption allowed)
- An optimal solution is obtained in polynomial time and used here
  - To distinguish easy-hard problems
  - As a reference for the solution quality



---

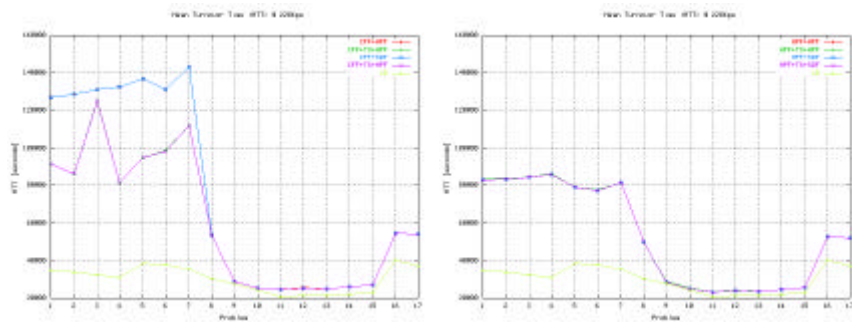
## **Experimentation**

Results



## Experimental Results (1/4)

Benchmark B1: the lowest average load on the packet stores



CFF planning

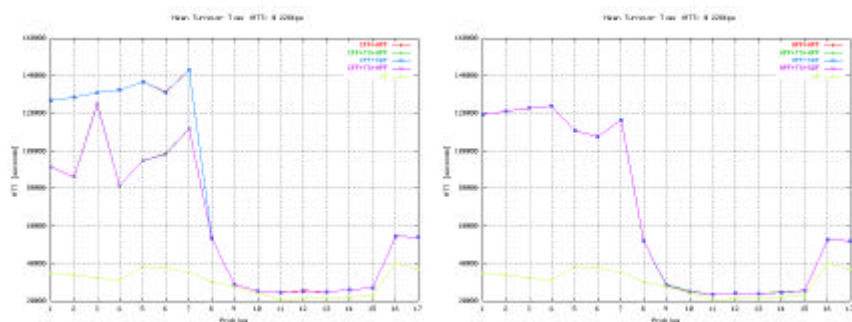
HPF planning

The best results are obtained with the greedy strategy



## Experimental Results (2/4)

Benchmark B2: an higher average load on the packet stores respect to B1



CFF planning

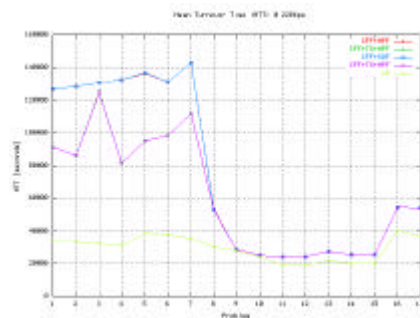
HPF planning

The best results are obtained with the **tabu search** strategy, the improvement on MTT is up to **30%**

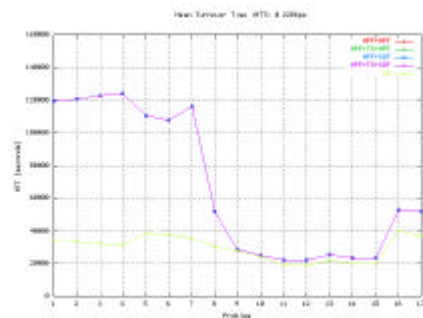


## Experimental Results (3/4)

Benchmark B3: higher load on the packet stores respect to B1 and B2



CFF planning



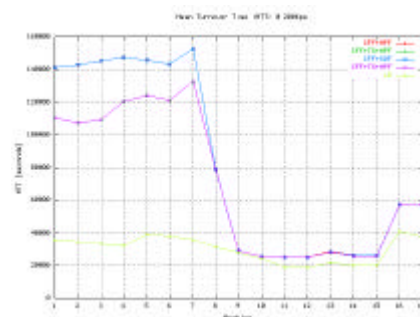
HPF planning

The best results are obtained with the **tabu search** strategy, the improvement on MTT is up to **30%**

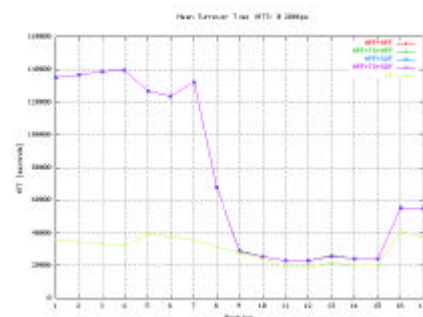


## Experimental Results (4/4)

Benchmark B4: an higher average value of CC respect to B1 an B2



CFF planning



HPF planning

The best results are obtained with the **tabu search** strategy, the improvement on MTT is up to **20%**



---

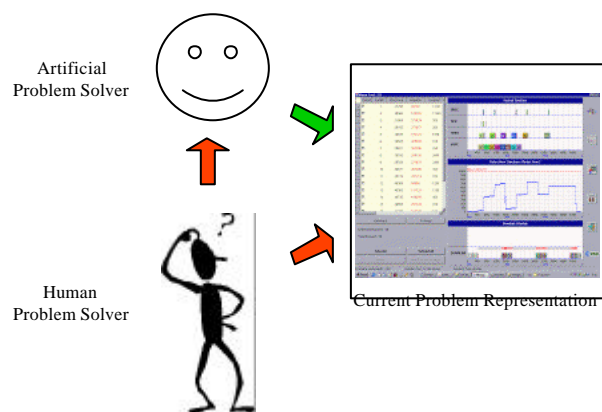
## Outline

- ▶ Generalities
- ▶ MEX-MDP: Problem Formulation
- ▶ A CSP Approach to Problem Solving
- ▶ **MEXAR the Interactive Problem Solver**
- ▶ Conclusions

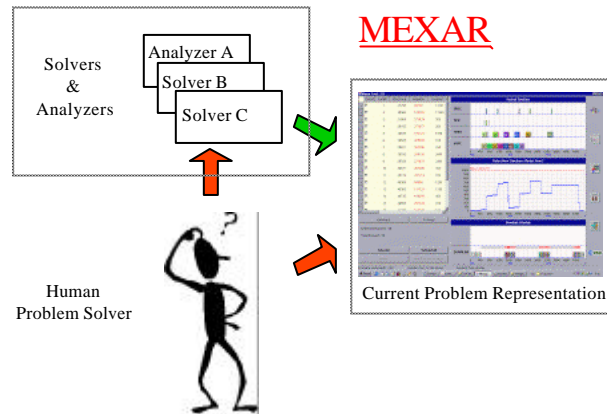


---

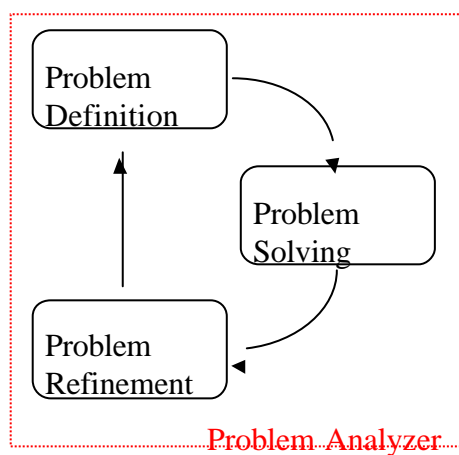
## Mixed-Initiative Problem Solving



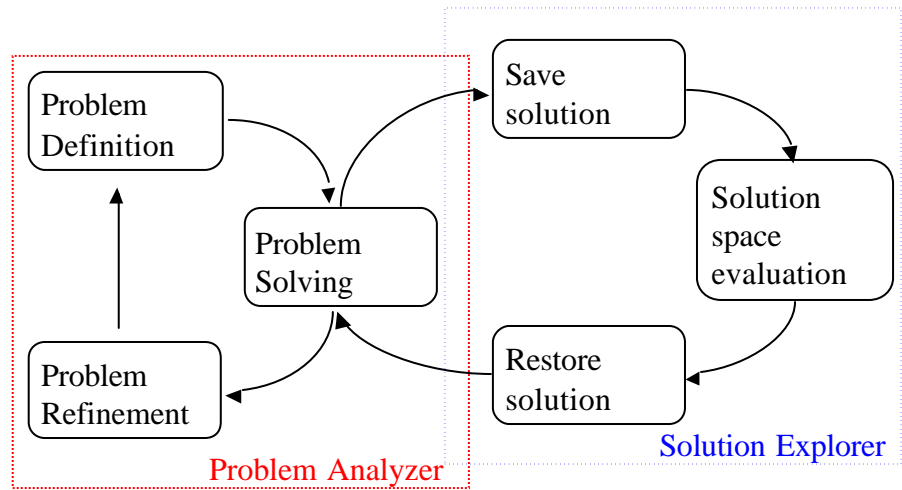
## Mixed-Initiative Problem Solving



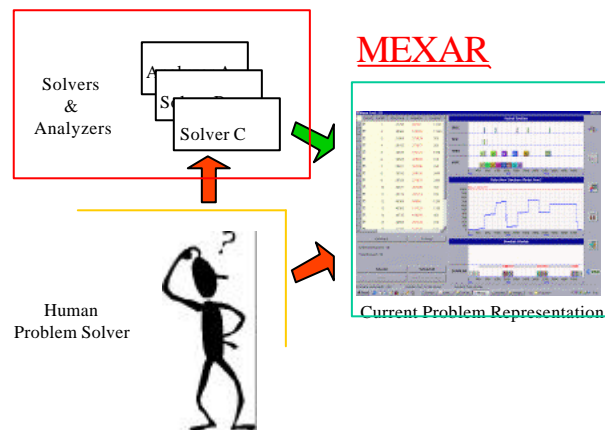
## MEXAR Funtionalities



## MEXAR Funtionalities

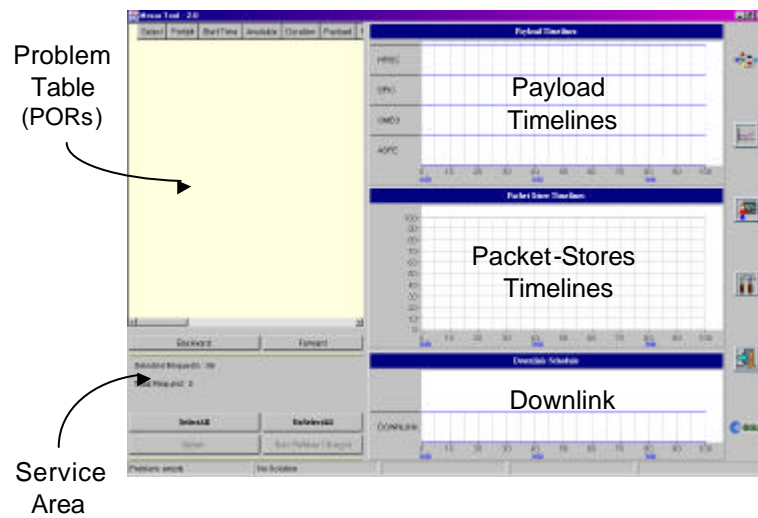


## Different Software Components

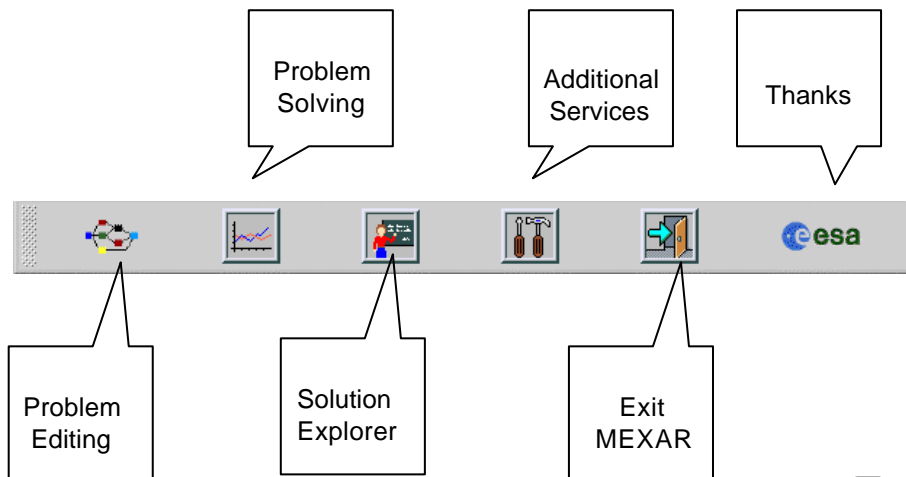




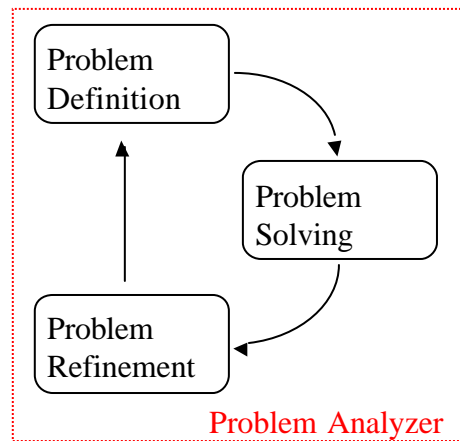
## The Problem Analyzer Layout



## The Available Commands



## MEXAR



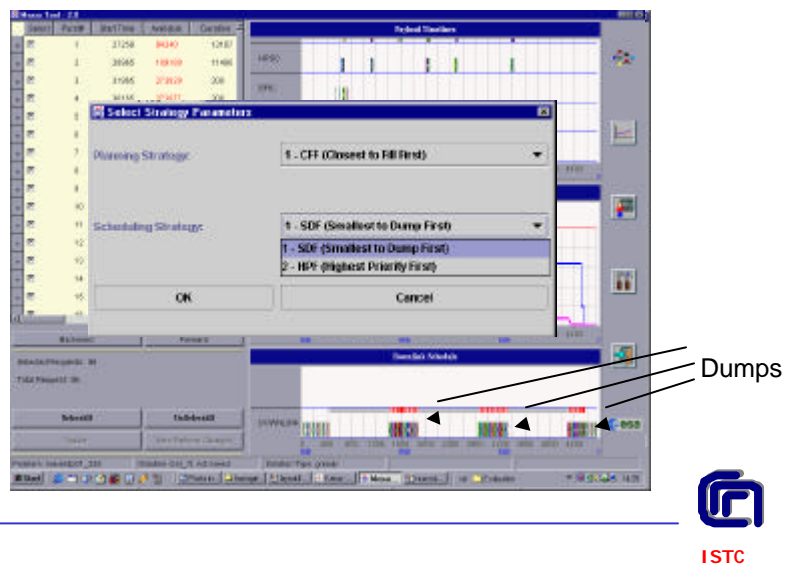
## Loading a problem



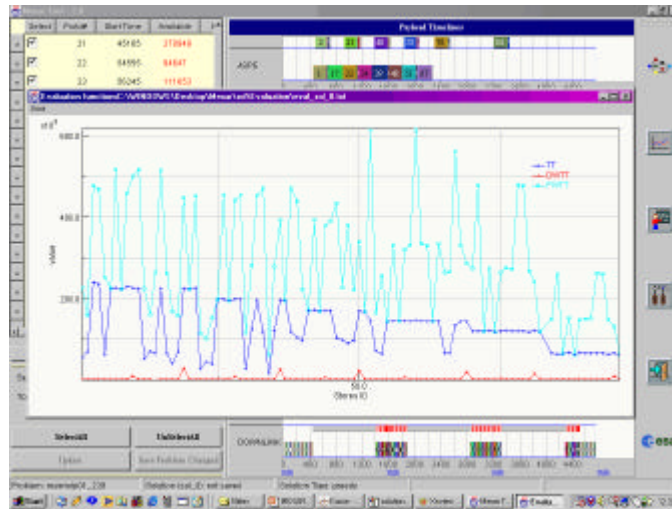
## Inspecting a MEX-MDP problem



## Calling the Greedy Solver



## Evaluating the solution



## Further information on the solution

Solution  
Table

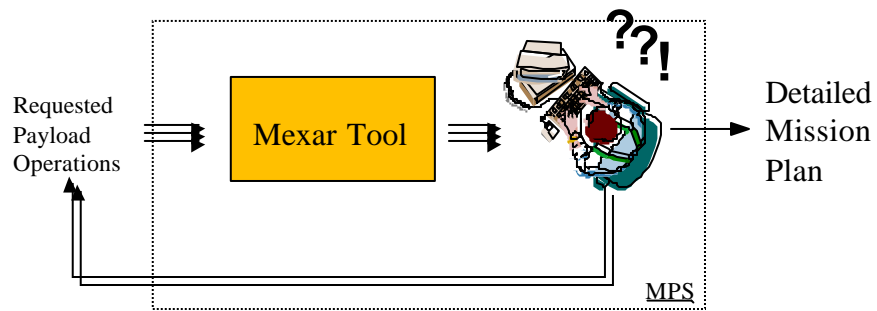
Period	Start Time	End Time	Value	Period	Start Time	End Time	Value
21	210448	210448	0.00	21	210448	210448	0.00
22	210447	210447	1.00	22	210447	210447	1.00
23	210445	210445	1.00	23	210445	210445	1.00
24	210445	210445	1.00	24	210445	210445	1.00

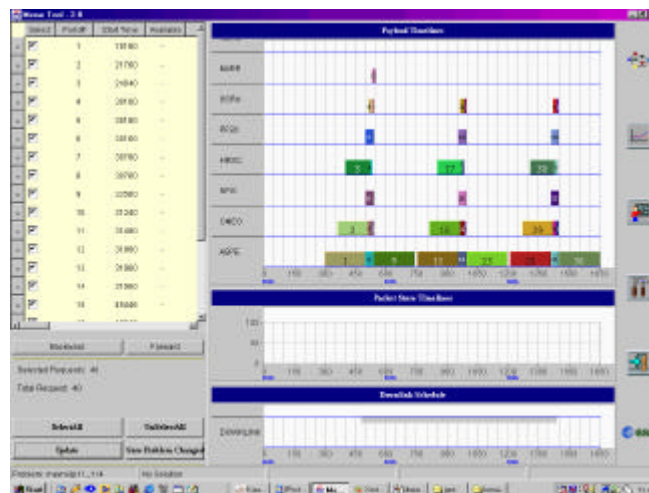
id	Order	Time	Value	Period	Start Time	End Time	Value
126	1	210448	0.00	21	210448	210448	0.00
127	2	210447	1.00	22	210447	210447	1.00
128	3	210445	1.00	23	210445	210445	1.00
129	4	210445	1.00	24	210445	210445	1.00
130	5	210445	1.00	24	210445	210445	1.00
131	6	210445	1.00	24	210445	210445	1.00
132	7	210445	1.00	24	210445	210445	1.00
133	8	210445	1.00	24	210445	210445	1.00
134	9	210445	1.00	24	210445	210445	1.00
135	10	210445	1.00	24	210445	210445	1.00
136	11	210445	1.00	24	210445	210445	1.00
137	12	210445	1.00	24	210445	210445	1.00
138	13	210445	1.00	24	210445	210445	1.00
139	14	210445	1.00	24	210445	210445	1.00
140	15	210445	1.00	24	210445	210445	1.00
141	16	210445	1.00	24	210445	210445	1.00
142	17	210445	1.00	24	210445	210445	1.00
143	18	210445	1.00	24	210445	210445	1.00
144	19	210445	1.00	24	210445	210445	1.00
145	20	210445	1.00	24	210445	210445	1.00
146	21	210445	1.00	24	210445	210445	1.00
147	22	210445	1.00	24	210445	210445	1.00
148	23	210445	1.00	24	210445	210445	1.00
149	24	210445	1.00	24	210445	210445	1.00
150	25	210445	1.00	24	210445	210445	1.00
151	26	210445	1.00	24	210445	210445	1.00
152	27	210445	1.00	24	210445	210445	1.00
153	28	210445	1.00	24	210445	210445	1.00
154	29	210445	1.00	24	210445	210445	1.00
155	30	210445	1.00	24	210445	210445	1.00



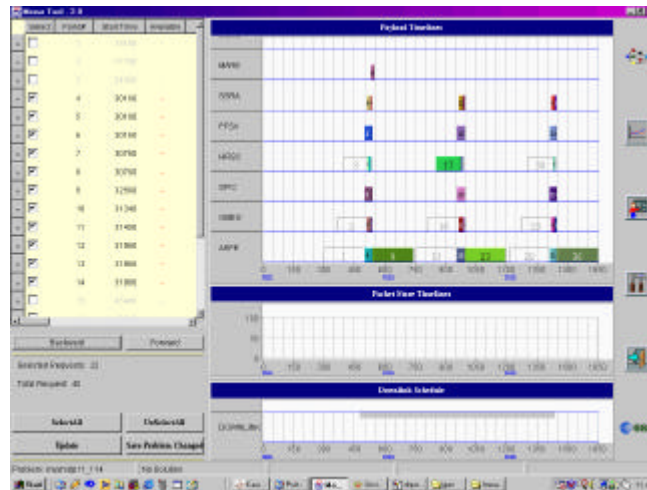
## Refining the Problem



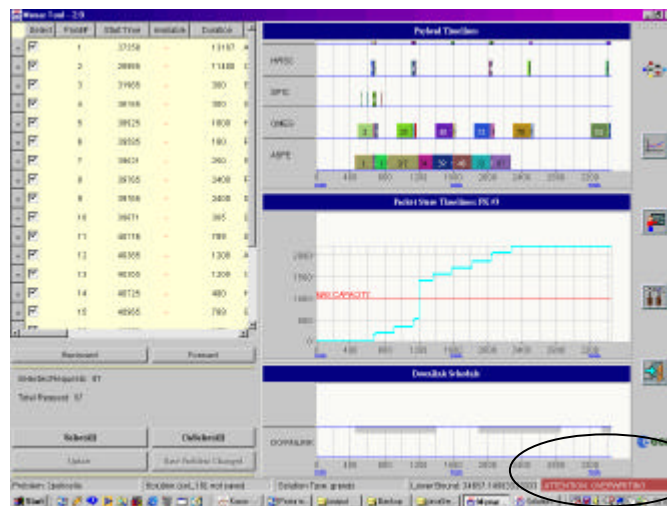
## Refining the Problem

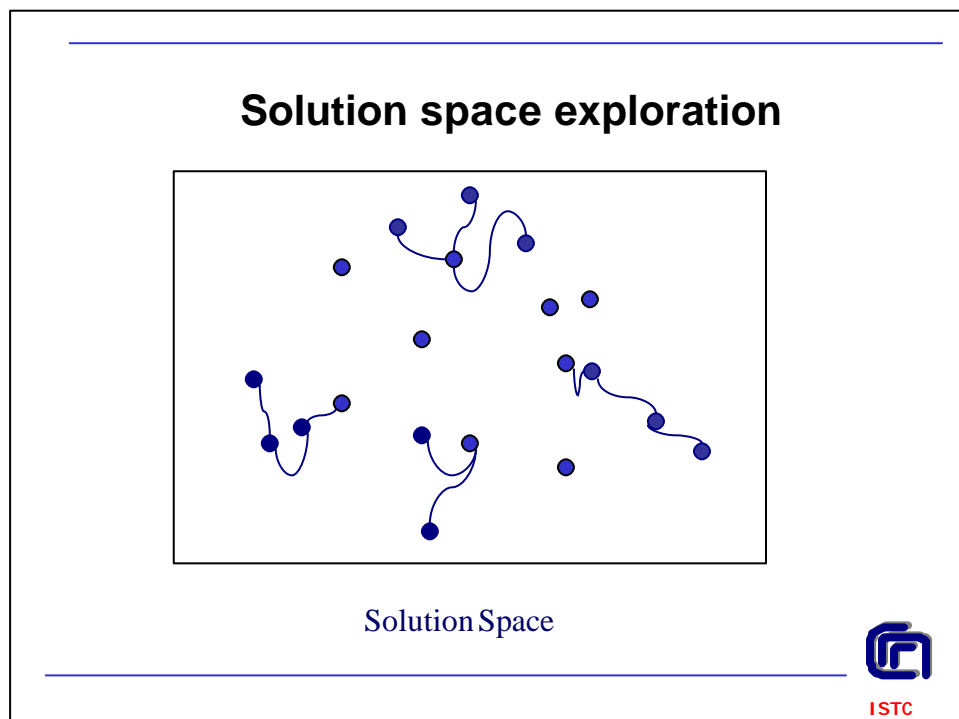
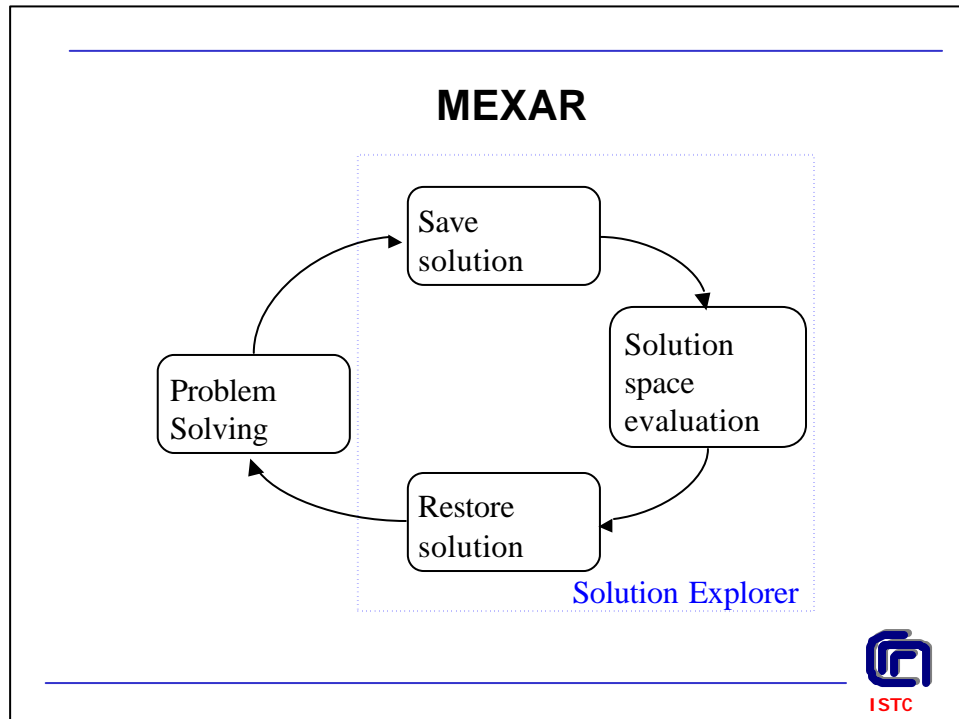


## Refining the Problem

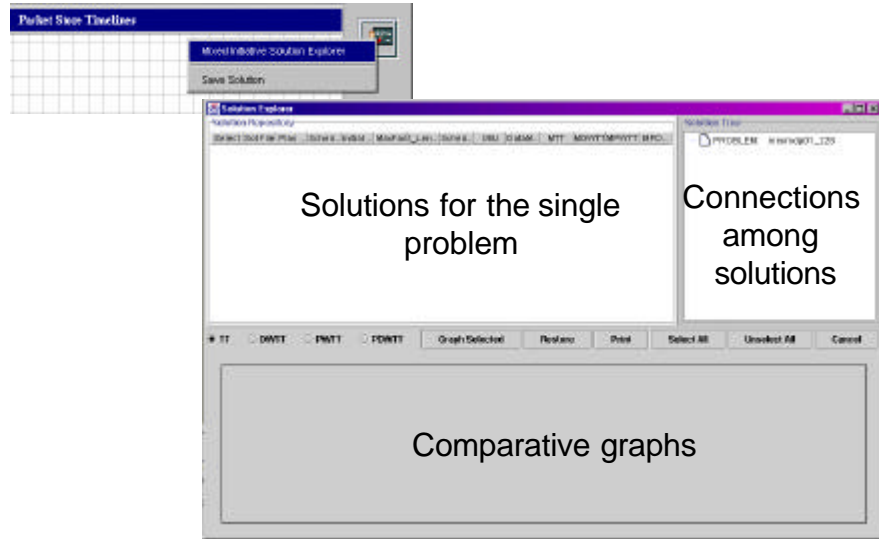


## A possible use





## The Mixed-Initiative Solution Explorer



ISTC

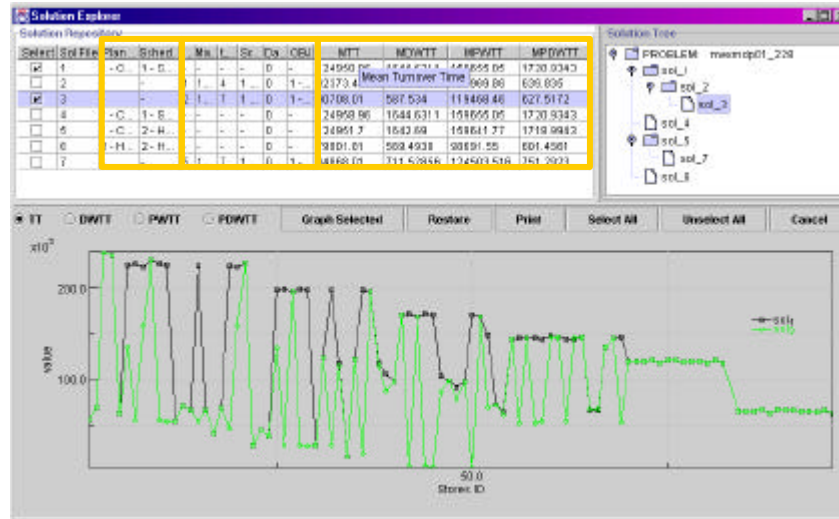
## When a solution is good: **Save it!**



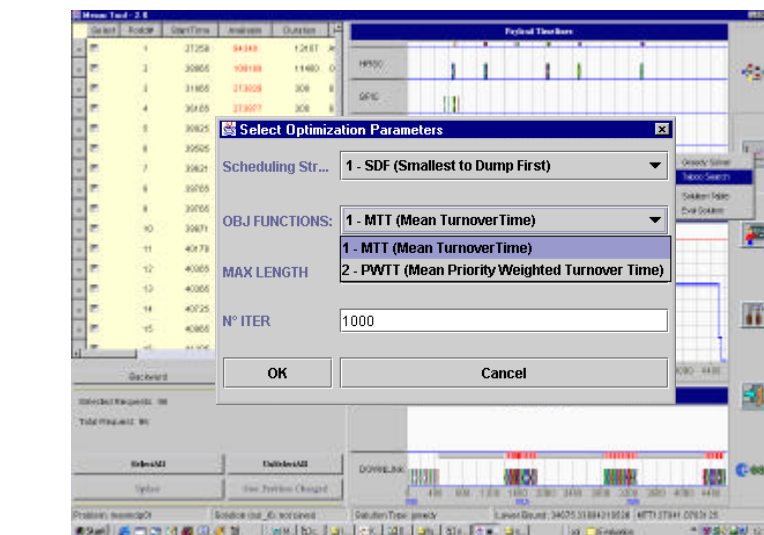
ISTC



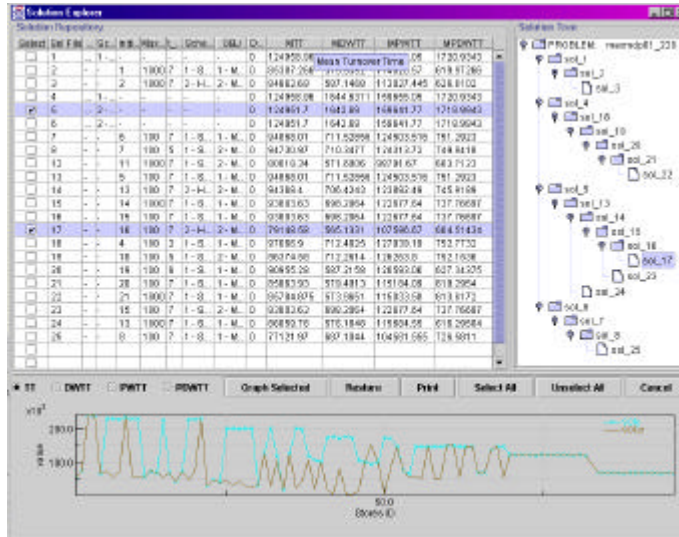
## Exploration



## Restore + Taboo



## Exploration (2)



## Outline

- ▶ Generalities
- ▶ MEX-MDP: Problem Formulation
- ▶ A CSP Approach to Problem Solving
- ▶ MEXAR the Interactive Problem Solver
- ▶ **Conclusions**



---

## Goals vs. Results

### 1. Select a mission planning problem in MARS EXPRESS

- Selected the MEX-MDP
  - It is solved “manually”
  - It requires a person constantly
- Developed a Problem Generator

### 2. Study algorithms for its solution

- Modeled as a CSP problem
- Developed a set of propagation rules
- A solver greedy algorithm
  - Planning phase
  - Scheduling phase
- An optimizer based on taboo search



---

## Goals vs. Results (2)

### 3. Develop a software demonstrator that integrates such algorithms

- Developed MEXAR
- Complete implementation in Java
  - - Problem Solver
  - - Interaction Module
- Example of Mixed Initiative Problem Solving
- Problem Analyzer
  - - “Glass Box” Inspection
  - - Problem Refinement
- Solution Explorer

