# Heuristics for Controlling Temporal Planning and Execution

Subbarao Kambhampati

Yochan Group

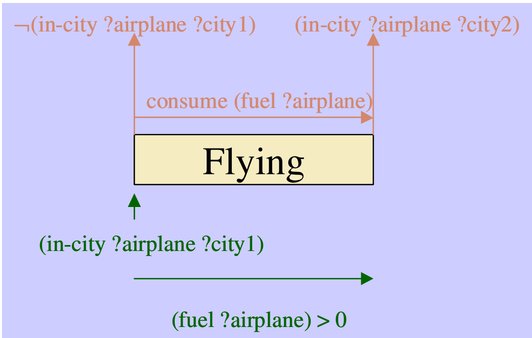
Arizona State University

# Overview

- Summary of the objectives
- Results obtained since the last meeting
- Current/Future Activity
- Updated Quad Chart
- --Put the tradeoff curve
- --read the sapa stuff.

# Motivation & Issues

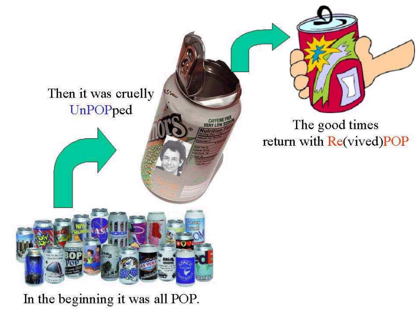
- Metric-Temporal planning is important for Space applications--- BUT it doesn't seem to SCALE ☹
  - RAX planner depended to a large extent on hand-coded heuristic control knowledge that can be laborious and error-prone
    - *NOT for Nicola and Kanna though…*
  - Our objective was to see how far we can scale up with automatically extracted heuristics
- Challenges:
  - Lack of a satisfactory base planner
  - More types of constraints to handle
  - Multi-objective nature of the planning



# Accomplishments

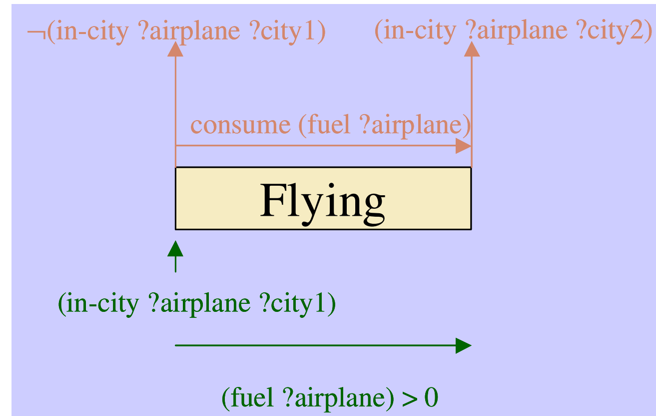
- Sapa: a Metric Temporal Planning System
  - Handles durative actions consuming continuous resources.
  - Heuristics based on Time-sensitive cost functions to support multi-objective search(*)
    - First estimates from "relaxed planning graphs"
    - Adjustments for handling continuous resources; negative interactions
  - Linear time greedy post processing techniques to convert position-constrained plans to order constrained ones.
  - Fully Implemented in JAVA
    - Available as an applet online
  - Achieved very good performance in the metric temporal planning track at IPC-2002
    - Best performer for the NASA-inspired domains (which were also the most expressive of the lot)
      - (Invited to submit a paper to the JAIR special issue on ICP-2002)

# Other Accomplishments/Directions




- RePOP: Heuristic Control for partial order planners
  - RePOP was the first partial order planner that was able to outperform Graphplan
    - VHPOP [CMU], a planner based on RePOP ideas was the "best newcomer" at ICP-2002
  - Currently extending to metric-temporal planning (Re(ze)NO?)
- AltAlt-p: Generation of parallel plans in a heuristic state search framework: AltAlt-p
  - Outperforms Graphplan style planners in generating parallel plans
- Multi-Pegg: Supporting multi-objective search in Graphplan framework
  - Uses a memory-based variant of Graphplan called PEGG to efficiently search for multiple solutions
- Calt-alt?: Heuristics for Conformant/Contingent Planning
  - Generalizes planning graphs to extract heuristics for supporting conformant planning

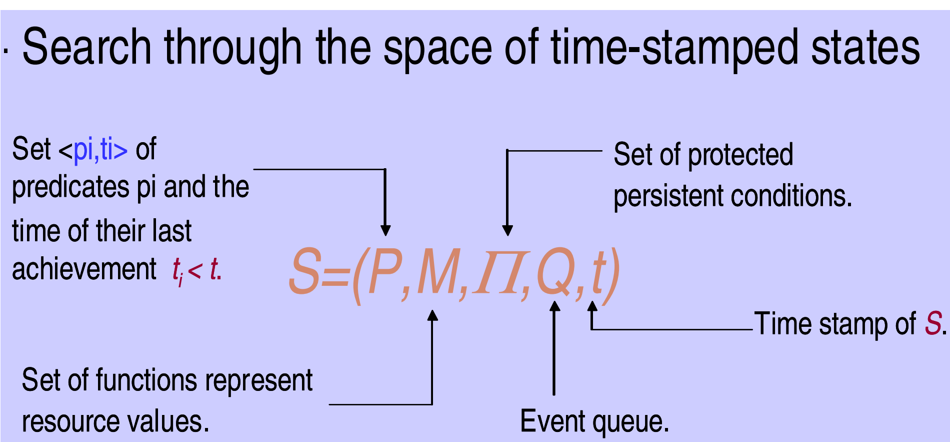
# SAPA: Search Overview




- **Goal Satisfaction**:

  $S=(P,M,\Pi,Q,t) \Rightarrow G$ if $\forall \langle p_i,t_i\rangle \in G$ either:
  - $\exists \langle p_i,t_j\rangle \in P$, $t_j < t_i$ and no event in $Q$ deletes $p_i$.
  - $\exists e \in Q$ that adds $p_i$ at time $t_e < t_i$.
- **Action Application**:

  Action A is applicable in S if:
  - All instantaneous preconditions of A are satisfied by P and M.
  - A's effects do not interfere with Π and Q.
  - No event in Q interferes with persistent preconditions of A.
- When A is applied to S:
  - S is updated according to A's instantaneous effects.
  - Persistent preconditions of A are put in Π
  - Delayed effects of A are put in Q.



# Deriving heuristics for SAPA

We use phased relaxation approach to derive different heuristics

Relax the negative logical and resource effects
to build the Relaxed Temporal Planning Graph

Pruning a bad state while preserving the completeness.

Deriving admissible heuristics:
–To minimize solution's makespan.
–To maximize slack-based objective functions.

Find relaxed solution which is used as distance heuristics

Adjust the heuristic values using the negative interaction **(Future work)**

Adjust the heuristic values using the resource consumption Information.

[AltAlt,AIJ2001]

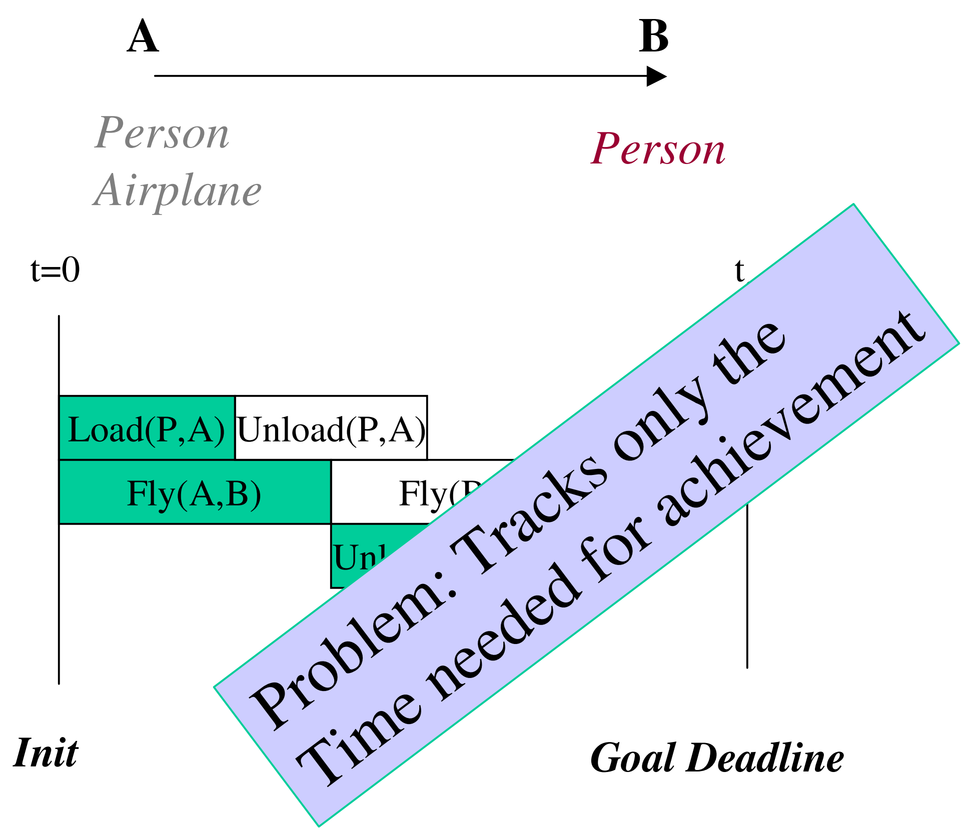
# Relaxed Temporal Planning Graph

*Heuristics in **Sapa** are derived from the Graphplan-style bi-level relaxed temporal planning graph (RTPG)*
*--resource/interaction adjustments follow*

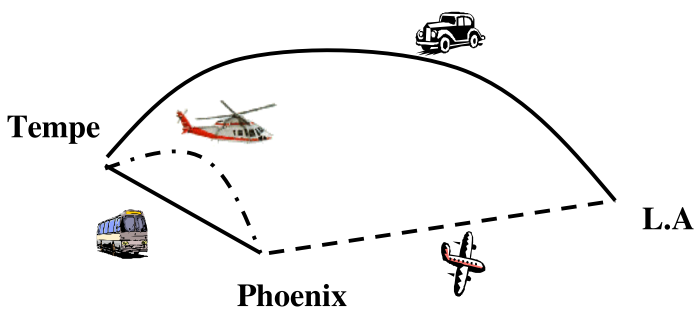
**Relaxed Action:**

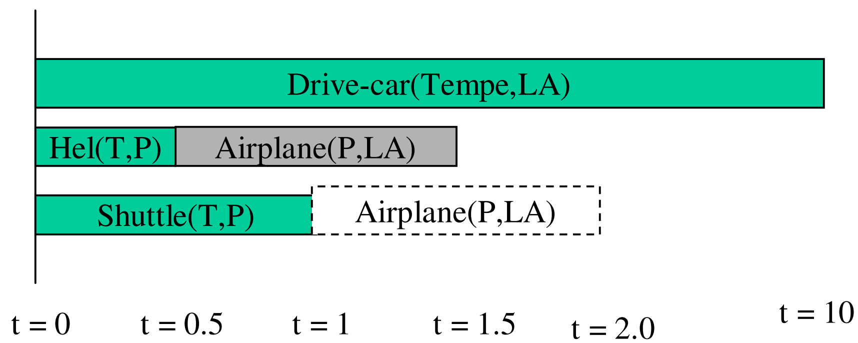
- No delete effects
- No resource consumption

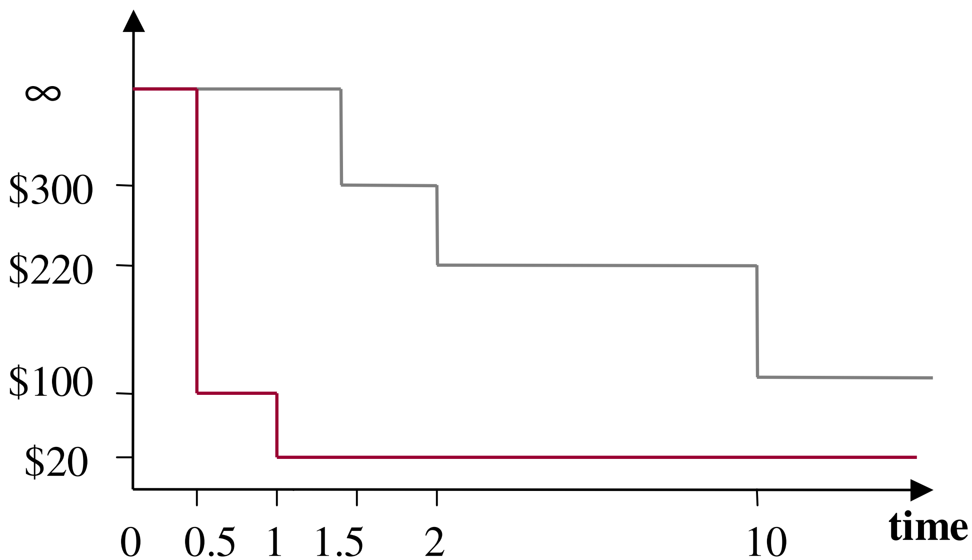
```
while(true)
    forall A≠advance-time applicable in S
        S = Apply(A,S)
        if S⇒G then Terminate{solution}

    S' = Apply(advance-time,S)
    if ∃(p_i,t_i) ∈ G such that
        t_i < Time(S') and p_i ∉ S then
                Terminate{non-solution}
    else S = S'
end while;
```



# Estimating the Cost Function



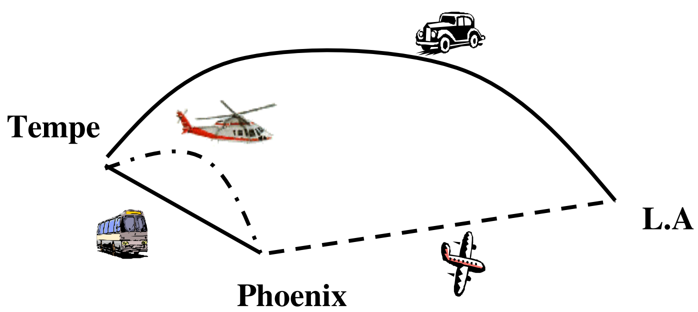



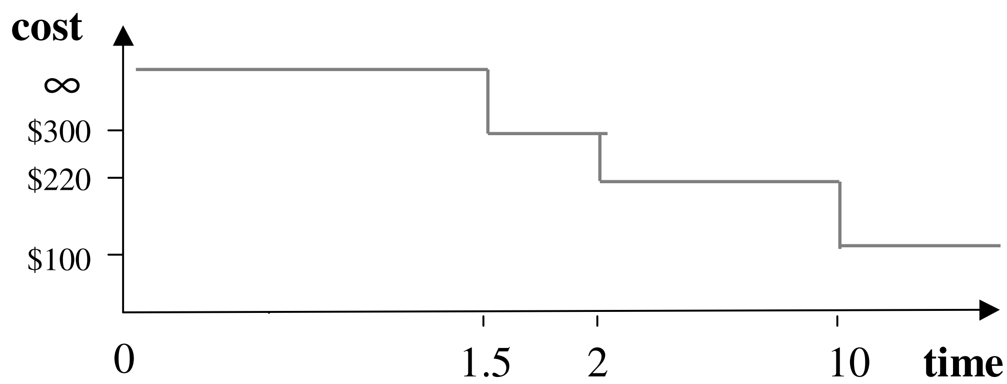



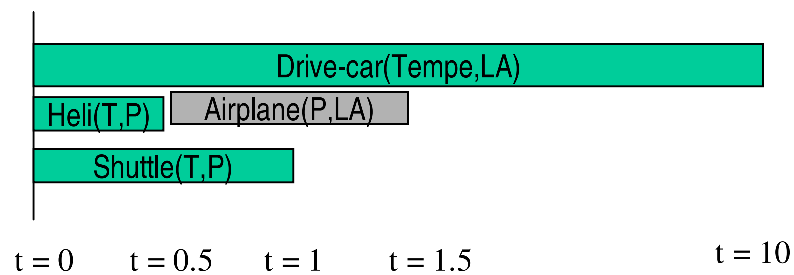

**Cost(At(LA))** — **Cost(At(Phx)) = Cost(Flight(Phx,LA))**

**Shuttle(Tempe,Phx):**
Cost: \$20; Time: 1.0 hour
**Helicopter(Tempe,Phx):**
Cost: \$100; Time: 0.5 hour
**Car(Tempe,LA):**
Cost: \$100; Time: 10 hour
**Airplane(Phx,LA):**
Cost: \$200; Time: 1.0 hour

# Time-sensitive Cost Functions



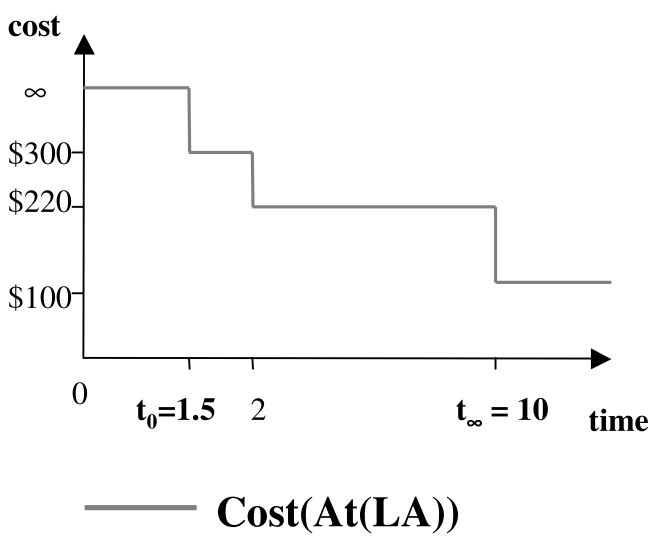




**Shuttle(Tempe,Phx):**
Cost: $20; Time: 1.0 hour
**Helicopter(Tempe,Phx):**
Cost: $100; Time: 0.5 hour
**Car(Tempe,LA):**
Cost: $100; Time: 10 hour
**Airplane(Phx,LA):**
Cost: $200; Time: 1.0 hour

- Standard (Temporal) planning graph (TPG) shows the time-related estimates e.g. earliest time to achieve fact, or to execute action
- TPG **does not show** the cost estimates to achieve facts or execute actions

# Heuristic estimation using the cost functions

**The cost functions have information to track both temporal and cost metric of the plan, and their inter-dependent relations !!!**




Earliest achieve time: $t_0 = 1.5$
Lowest cost time: $t_\infty = 10$

- If the objective function is to minimize time: $h = t_0$
- If the objective function is to minimize cost: $h = CostAggregate(G, t_\infty)$
- If the objective function is the function of both time and cost

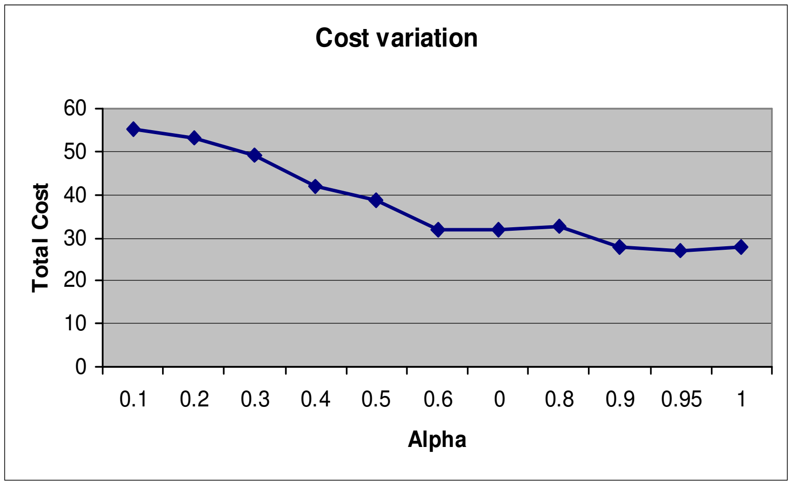
  O = f(time,cost) then:

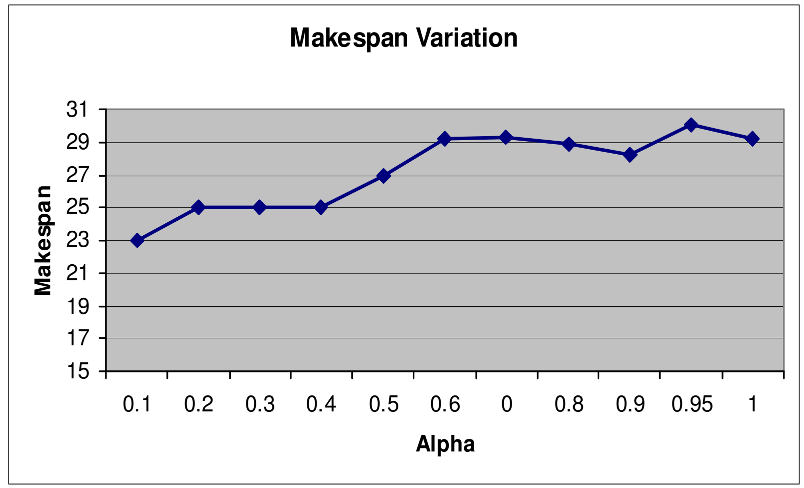
  $h = \min f(t, Cost(G,t)) \;\; s.t. \;\; t_0 \leq t \leq t_\infty$

**Eg:** f(time,cost) = 100.makespan + Cost then

$h = 100x2 + 220$ at $t_0 \leq t = 2 \leq t_\infty$

# Cost-sensitive heuristics help us handle a variety of makespan/cost tradeoffs



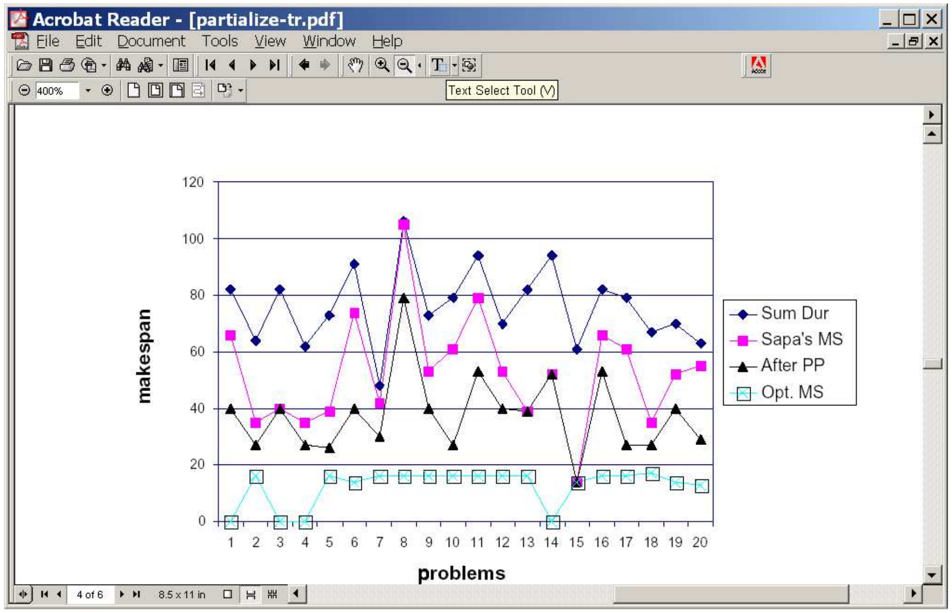




Results over 20 randomly generated temporal logistics problems involve moving 4 packages between different locations in 3 cities:

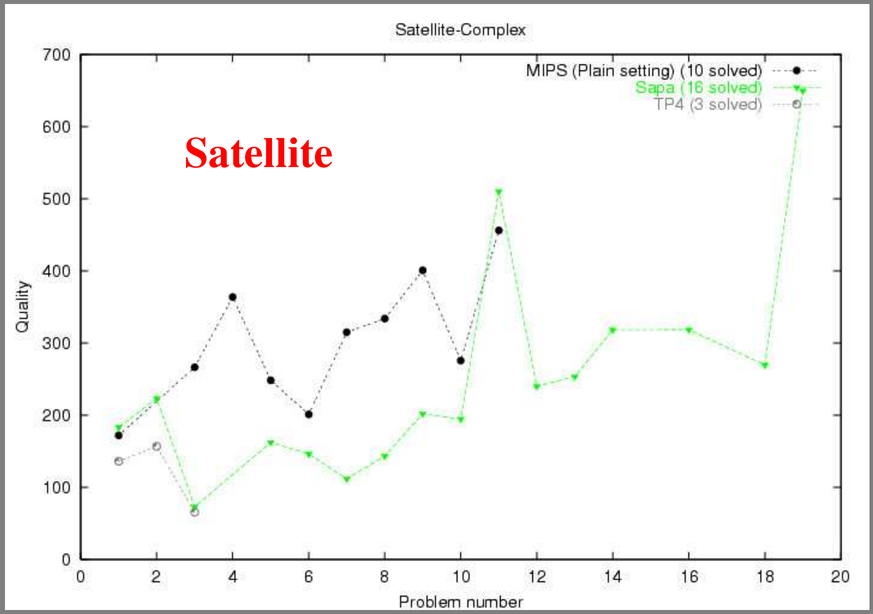
$$O = f(time,cost) = \alpha.Makespan + (1- \alpha).TotalCost$$

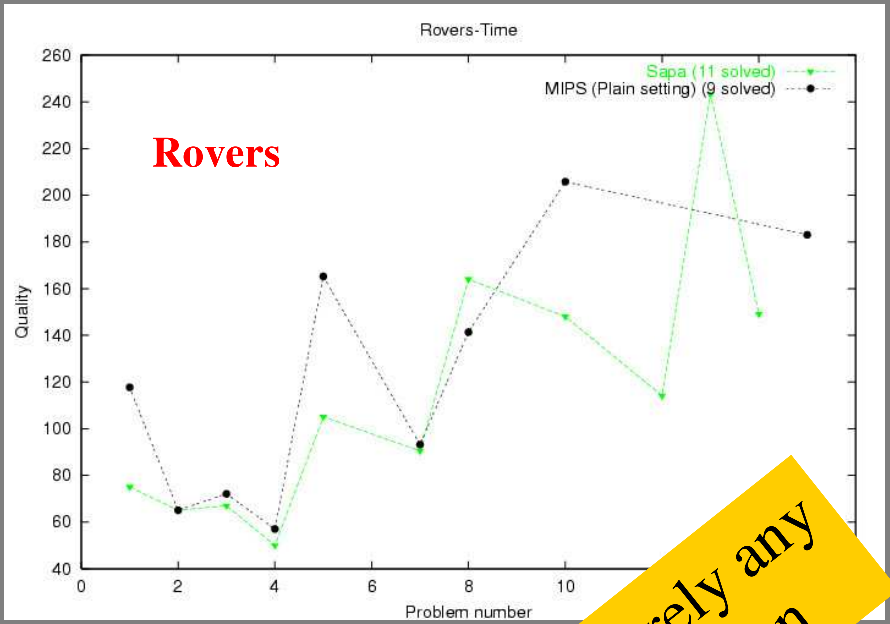
# Post-processing to get Order Constrained Plans (Partialization)

- Challenges:
  - More expressive action representation involving temporal and metric resource constraints
  - Variety of objective functions for partialization in terms of makespan, slack, fewest orderings
- Approach
  - Pose the problem as a CSOP
  - Develop greedy solutions based on specific value ordering starategies



# Sapa did well at IPC-2002

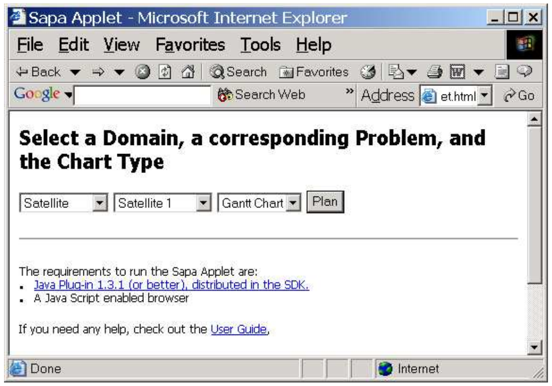


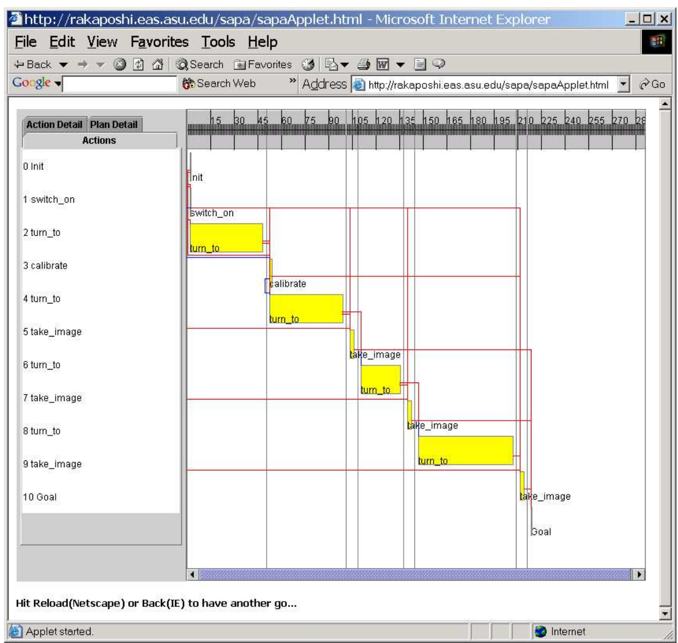



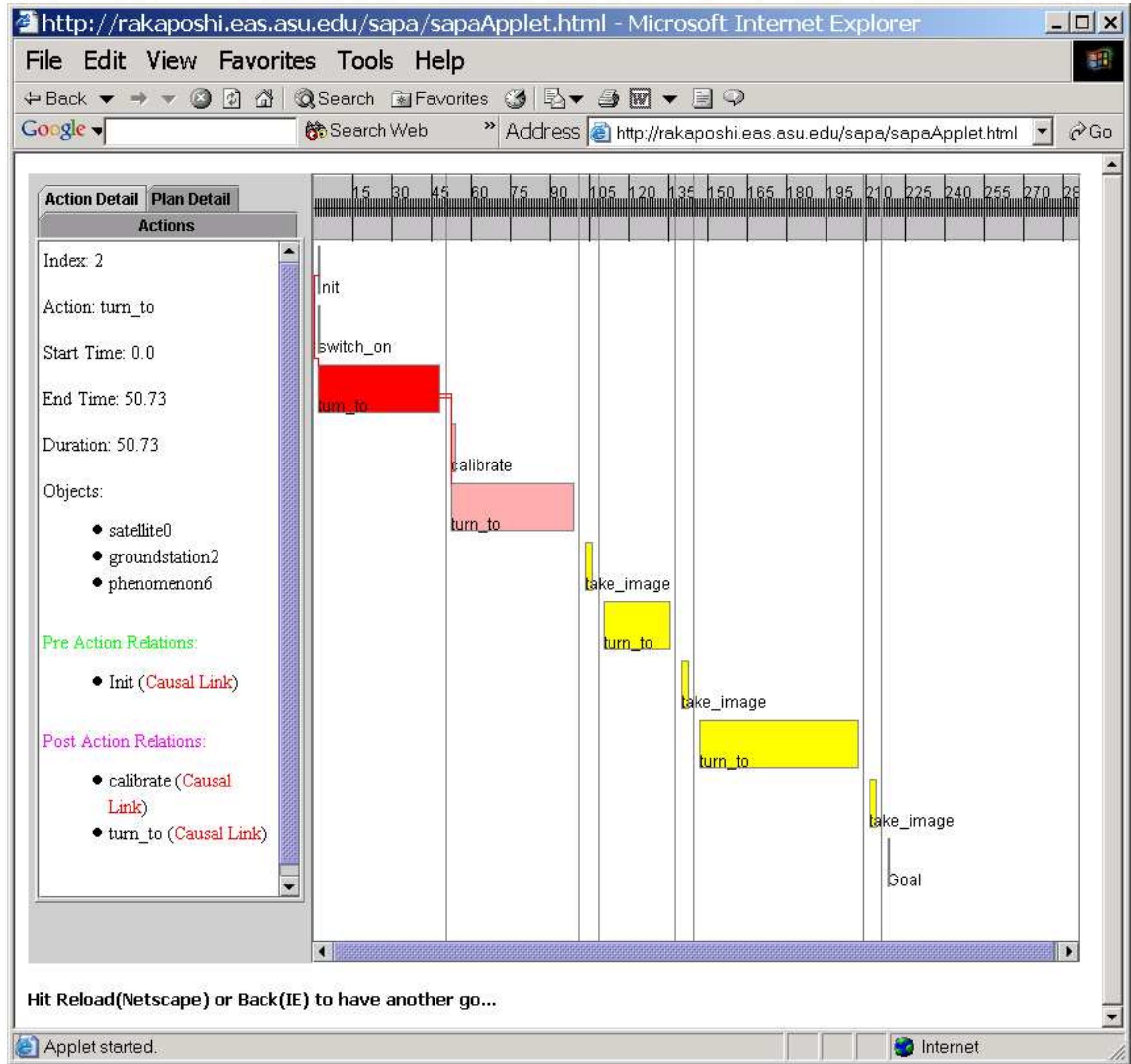

Best performance over the most expressive dom[...]
--Invited to contribute a paper to JA[...] issue

This with barely any code optimization

# SAPA is available online as JAVA Applet



# Results from IPC-2002

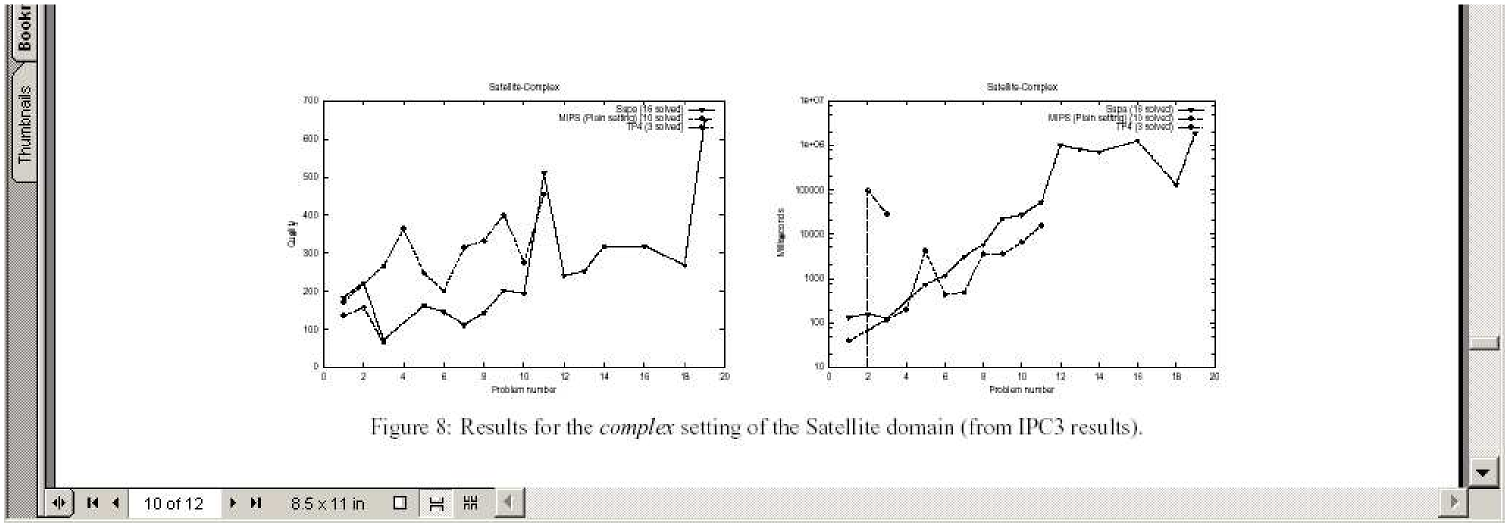



Figure 8: Results for the *complex* setting of the Satellite domain (from IPC3 results).

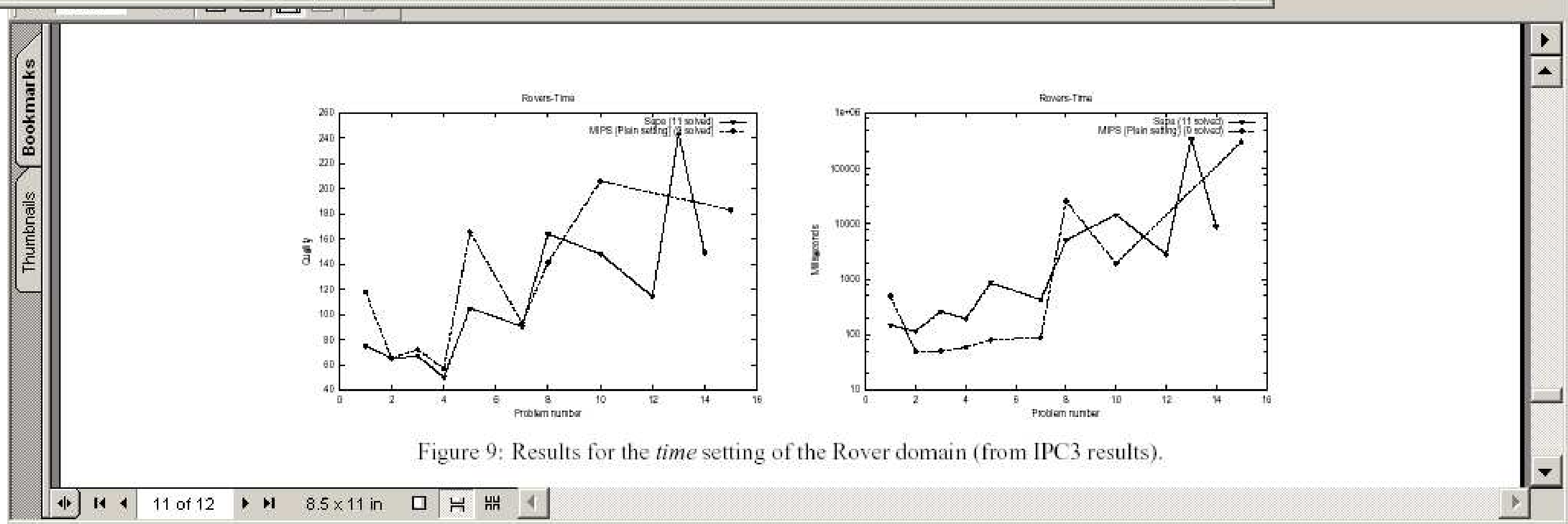



Figure 9: Results for the *time* setting of the Rover domain (from IPC3 results).

# Planned Extensions to SAPA

## Sapa

**Multi-objective:**
- ✓Cost: Money, resource
- ✓Time: makespan, slacks

**Additional:**
- ➢Flexibility
- ➢Robustness
- ➢Others?

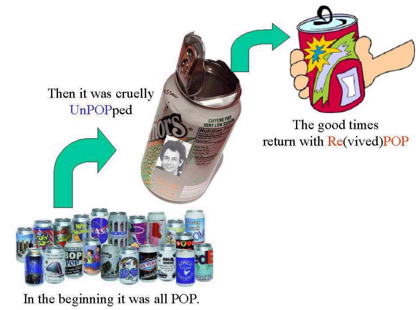
**Expressiveness:**
- ✓Temporal constraints (durative actions, goal dl)
- ✓Resource constraints

**Additional:**
- ➢Additional temp. const.
- ➢Exogenous events
- ➢Resource constraints
- ➢Others (logical)?

**Quality/Efficiency:**
- ✓Heuristics:
  - ✓Time-sensitive functions
  - ✓Adjustment techniques
- ✓Post-processing:
  - ✓Constraints for optimal p.p. with different objective func.
  - ✓Greedy p.p. in linear time

**Additional:**
- ➢Interleaved p.c/o.c plan generation
- ➢One-shot heuristic computation
- ➢Heuristics sensitive to other quality measures

# Other Accomplishments/Directions




- RePOP: Heuristic Control for partial order planners
  - RePOP was the first partial order planner that was able to outperform Graphplan
    - VHPOP [CMU], a planner based on RePOP ideas was the "best newcomer" at ICP-2002
  - Currently extending to metric-temporal planning (Re(ze)NO?)
- AltAlt-p: Generation of parallel plans in a heuristic state search framework: AltAlt-p
  - Outperforms Graphplan style planners in generating parallel plans
- Multi-Pegg: Supporting multi-objective search in Graphplan framework
  - Uses a memory-based variant of Graphplan called PEGG to efficiently search for multiple solutions
- Calt-alt?: Heuristics for Conformant/Contingent Planning
  - Generalizes planning graphs to extract heuristics for supporting conformant planning

Then it was cruelly UnPOPped


The good times return with Re(vived)POP


In the beginning it was all POP.

# Other Efforts

- AltAlt-P
  - Develop concurrent plans in a heuristic state search framework
  - Challenge: Projection over subsets of actions
  - Approach: Incremental "fattening" of most promising solution
  - Result: A heuristic state search planner that out-performs Graphplan style panners in parallel plan generation
- Multi-PEGG
  - Support multi-objective search in Graphplan framework (length, cost)
  - Challenge: Cannot stop after first solution
  - Approach: Use a memory-based variant of Graphplan, called PEGG to efficiently search for multiple solutions
  - Result: Multi-PEGG supports

# Heuristic Control of Metric Temporal Planning

Subbarao Kambhampati; Arizona State University [Cross Enterprise]

**Goal:** To develop a modular scalable architecture for supporing metric-temporal planning

## Objectives:

• Develop temporal planners that exploit the recent advances in classical planning

•To develop architectures that seamlessly integrate planners with schedulers

## Key Innovation:

•Concentration modular rather than monolithic architectures as a way to scale-up metric temporal planners

•Heuristics sensitive to multiple objectives (such as makespan, cost, temporal flexibility)

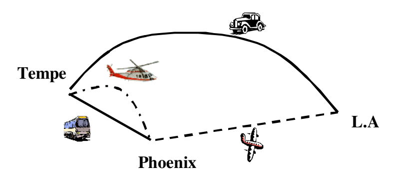
## NASA Relevance:

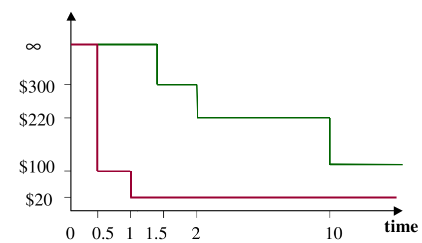
- Metric/temporal planning to support ground-based mission planning.

## Accomplishments to date:

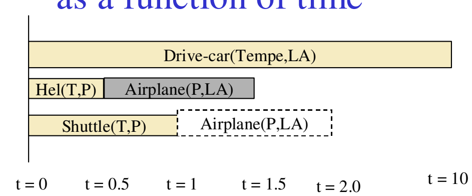
- SAPA was the best planner for NASA-inspired Rover and Satellite domains in International Planning Competition
- Presented papers on SAPA at ECP-2001, AIPS-2002
- Developed PEGG: a multi-objective planning algorithm for Graphplan
    - Presented a paper on PEGG at AIPS 2002 workshop on planning and scheduling with multiple criteria

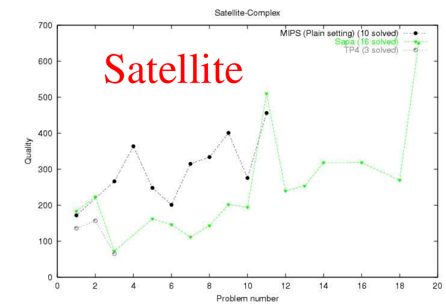
## Description/Schedule



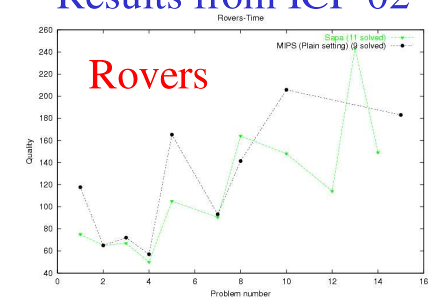




Heuristics tracking cost as a function of time




Results from ICP-02




Sapa was the best performer in the NASA-inspired domains at Intl. Planning Competition

## Schedule:

- Making SAPA implementation robust and user-friendly
- Integrate SAPA into a Realplan-like architecture where it cooperates with a real scheduler.

# Heuristic control of Metric Temporal Planners

Subbarao Kambhampati; Arizona State University [IS]

**Goal:** To develop heuristic control techniques for metric temporal planners

## Objectives:

• Generalize the heuristic control techniques in classical planning to least commitment planners and metric temporal planners

## Key Innovation:

• Exploiting automated reachability analysis to control any planners' search process

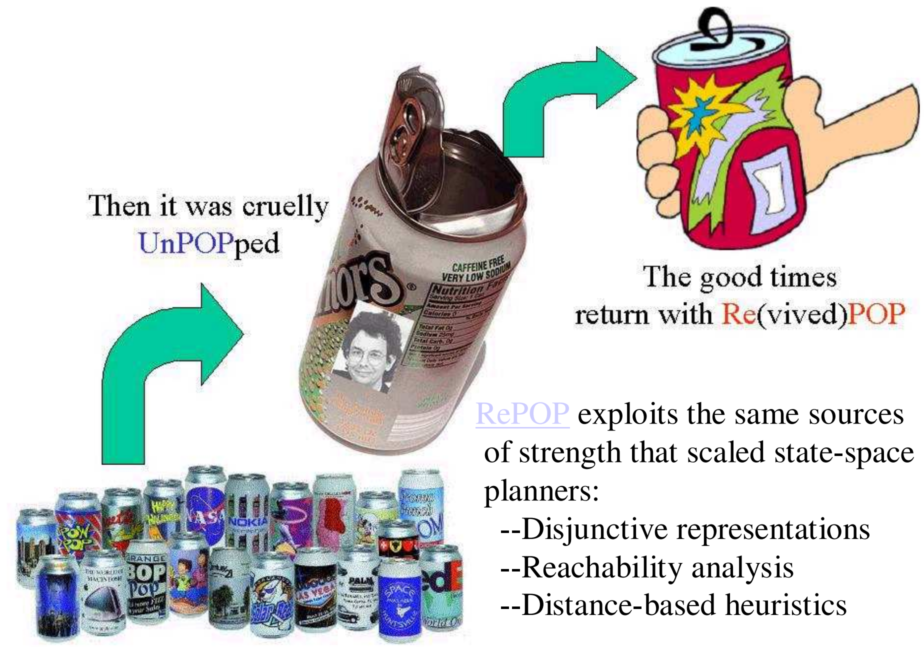
## NASA Relevance:

- HSTS planner demonstrates that least-commitment planners offer many advantages for mission planning. Search control for such planners is a critical issue.

## Accomplishments to date:

- Developed RePOP, a partial order least-commitment planner that shows dramatic scalability potential. [IJCAI-2001]
    - VHPOP, a CMU planner partially based on RePOP ideas was the "best new comer"(!) at IPC 2002
- Developed a method for partializing position-constrained plans
    - Presented at AIPS 2002 Temporal Planning Workshop

## Description/Schedule




RePOP exploits the same sources of strength that scaled state-space planners:
--Disjunctive representations
--Reachability analysis
--Distance-based heuristics

## Schedule:

• Generalize the heuristic techniques to work with partially instantiated plans, and plans with durative actions.