Resolving Over-constrained Temporal Problems with Uncertain Durations

ICAPS 2014, Portsmouth

Peng Yu, Cheng Fang, Brian Williams

June 26, 2014

Objective

We would like to **work with the users**

to **resolve** uncontrollable temporal problems

through making **trade-offs**

between **risk taken** and **temporal requirements**.

Robotic Personal Transportation System

- A personal air taxi with an intelligent trip advisor.

Key features

- Find alternative solutions that are **simple** and **preferred**.
- Provide **insights** into cause of failure and its resolution.
  - Minimize the perturbations;
  - Prioritize alternatives;
  - Explain the cause of failure;
  - Adapt incrementally to new constraints.

"Delay your arrival by 5 minutes".

"OK, then how about having lunch at restaurant Y".

"Because of the extended travel time".

"if you want to shop for at least 25 minutes, you can have lunch at restaurant Y for 55 minutes".

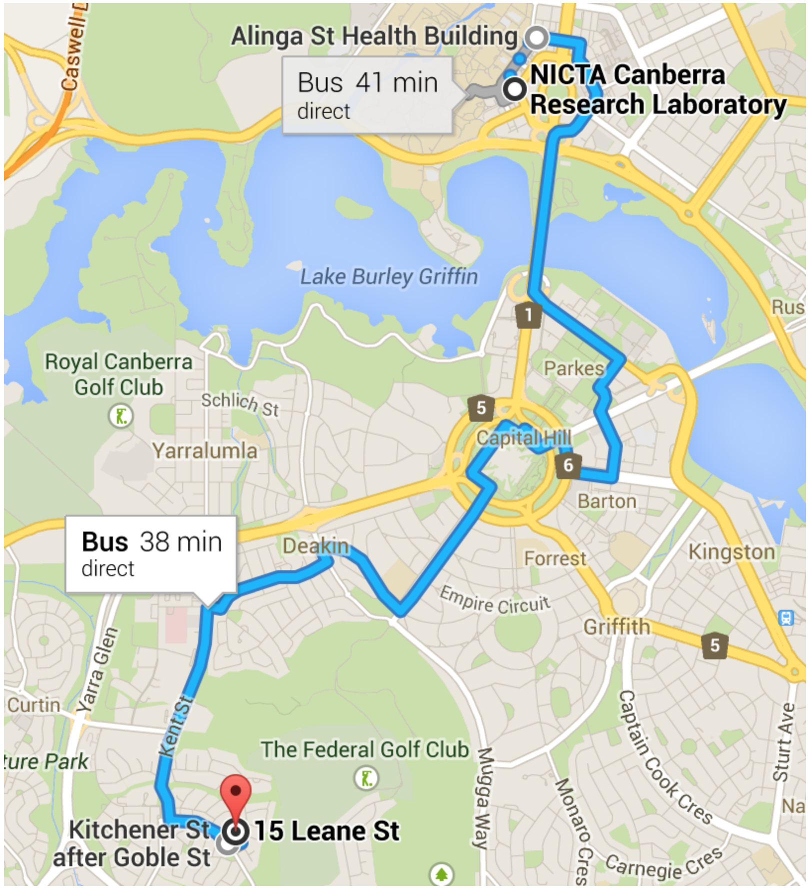
When There Is Uncertainty

- Uncertainty Sensitive Transit Advisor.

It is **6pm** now and Brian is leaving his office for home.

–He wants to **be home in 40 minutes**, and is only willing to take buses.

–Right now, he is looking up Google Map for directions…



Which Bus To Take

- Google Map returns two options (leaving office at 1800), ranked based on trip duration

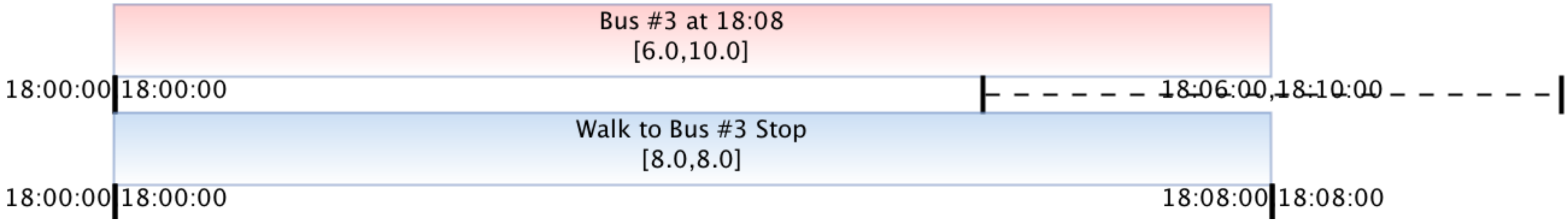
- Option 1:
  - Take the **18:08 Bus #3** (Ride time 23 mins).
  - Walking to departure stop: 8 mins.
  - Walking from arrival stop to home: 3 mins.

- Option 2:
  - Take the **18:11 Bus #934** (Ride time 26 mins).
  - Walking to departure stop: 10 minutes.
  - Walking from arrival stop to home: 3 minutes.

Uncertainty Affects Our Decision

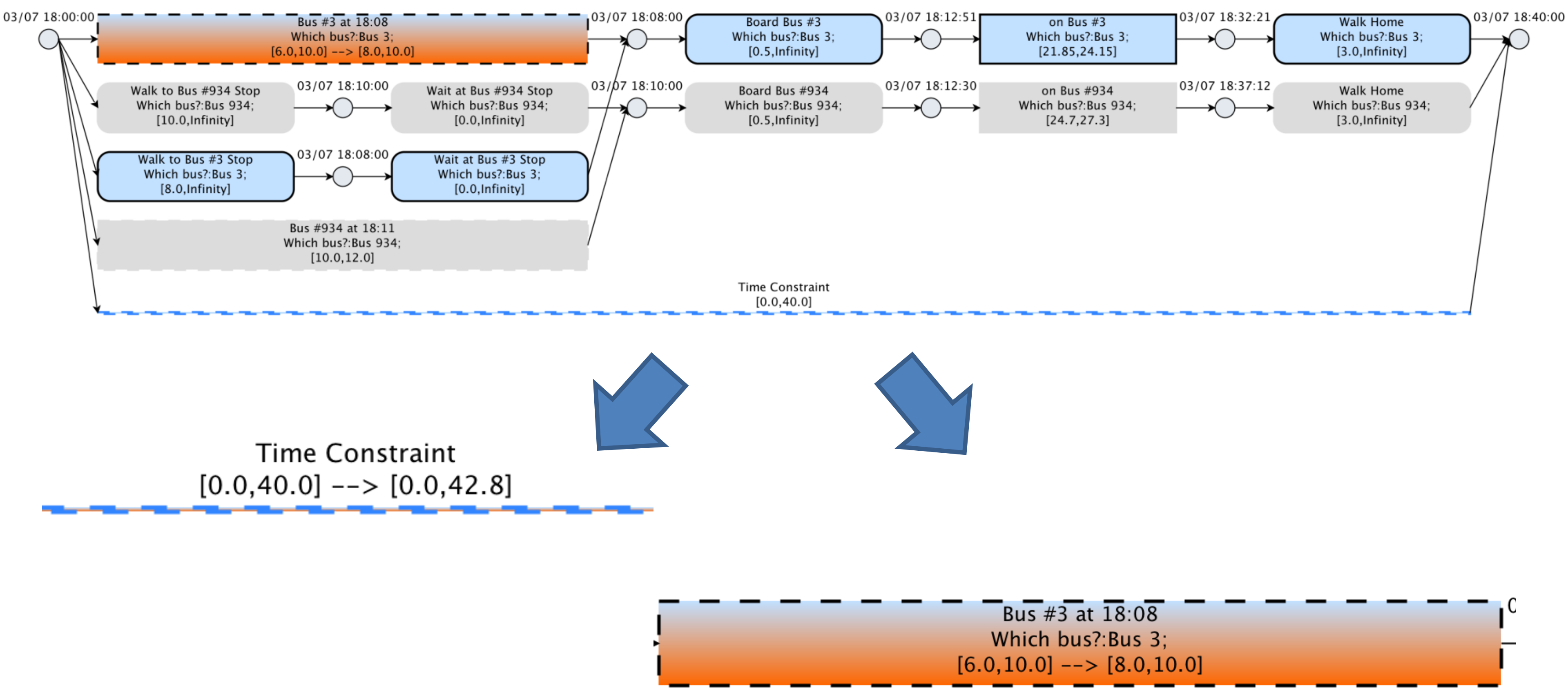
- Buses may be late or early:
  - Bus #3: 18:08 ± 2 minutes.
  - Bus #934: 18:11 ± 1 minute.

- Brian may miss the bus if he takes the Google preferred option.



Cope With the Uncertainty

- *"You can catch Bus #934 and arrive home* ***3 minutes late****."*
- *"Or, you can take Bus #3 and arrive home on time, but* ***taking the risk*** *of missing the bus, if it arrives early."*

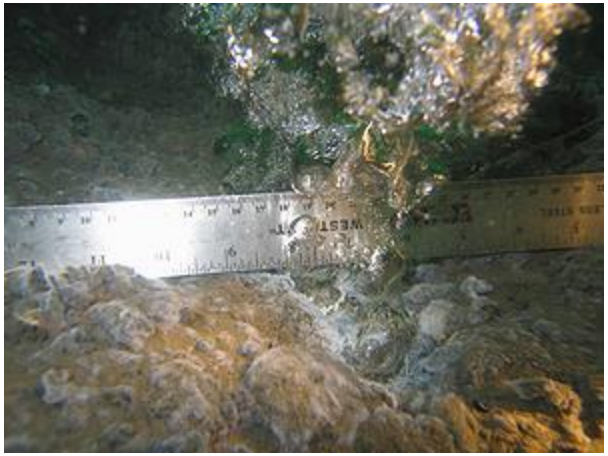


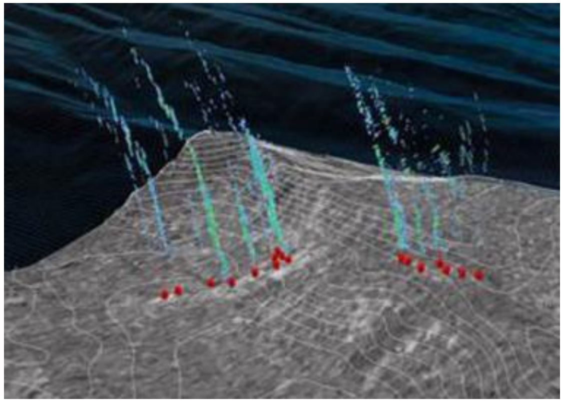
Mission Advisor for Woods Hole Oceanographic Inst.

- During an expedition cruise, the chief scientist needs assistance for planning and scheduling activities, especially when things go wrong.
  - Task sequencing and scheduling.
  - Goal relaxation and failure recovery.
  - Human resources and assets management.


A WHOI Mission

- Duration: Sep 26th – Oct 17th.
- Vessel: R/V Atlantis.
- Location: Along the coast between SF and LA.
- Objectives:
  - Find and sample methane seeps near the coast.
  - Locate and sample a 60 year-old DDT dumping site.
  - Recover and replace incubators on the seafloor.





A 3-day Plan From the Cruise

From: "David Valentine" <scil@atlantis.whoi.edu>
Subject: Plan of the day 9/28+
Date: Sat, September 28, 2013 6:37 am
To: pod@atlantis.whoi.edu

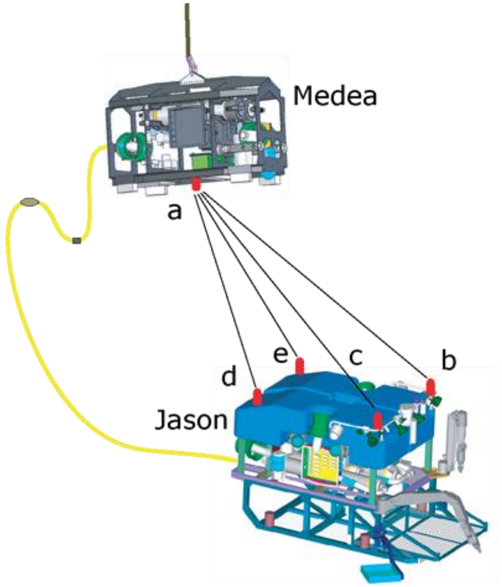
```
Draft Cruise Plan 9/28-9/30

______________________________________________________________________________
9/27/13
Sentry Dive at Partington Canyon        6hrs                      Target start time 2000
______________________________________________________________________________
9/28/13
Depart Partington Canyon                                              Estimated Departure Time 0230
Transit to Paull's Pingo                        27 hrs               ETA 0530 hrs (9/29/13)
Science Meeting 10AM!
Multibeam pass of SBB-2H Pockmark                          Line Z to Z'
Multibeam pass in Southern SB Channel                          Lines A to A', B to B', C to C'
and D to D'
Multibeam pass of SW Mounds area                                  Lines E to E' and F to F'
______________________________________________________________________________
9/29/13
Arrive at Paull's Pingo                                                  ETA 0530 hrs
Jason Operations at Paull's Pingo       15 hrs                 Deploy by 0730; 2 Elevator
Deployments
Transit to SW mounds                        1.5 hrs               Arrive SW Mounds ~2400
______________________________________________________________________________
9/30/13 and beyond
Sentry deployment at SW Mounds        16 hrs                Deploy at 0000
Jason Deployment at SW Mounds        24 hrs                Multiple Elevators
```

Everything can Go Wrong

- [Day 1] Jason failed after 30 min into its first dive, entered an uncontrollable spin and broke its optic fiber tether.

- [Day 1] The new camera installed on Sentry did not work well in low light situations. It had been replaced during its second dive.



Everything can Go Wrong

- [Day 2] Jason entered an uncontrollable spin and broke its optic fiber tether again during its second dive. It turned out that there is a bug in its newly updated code.

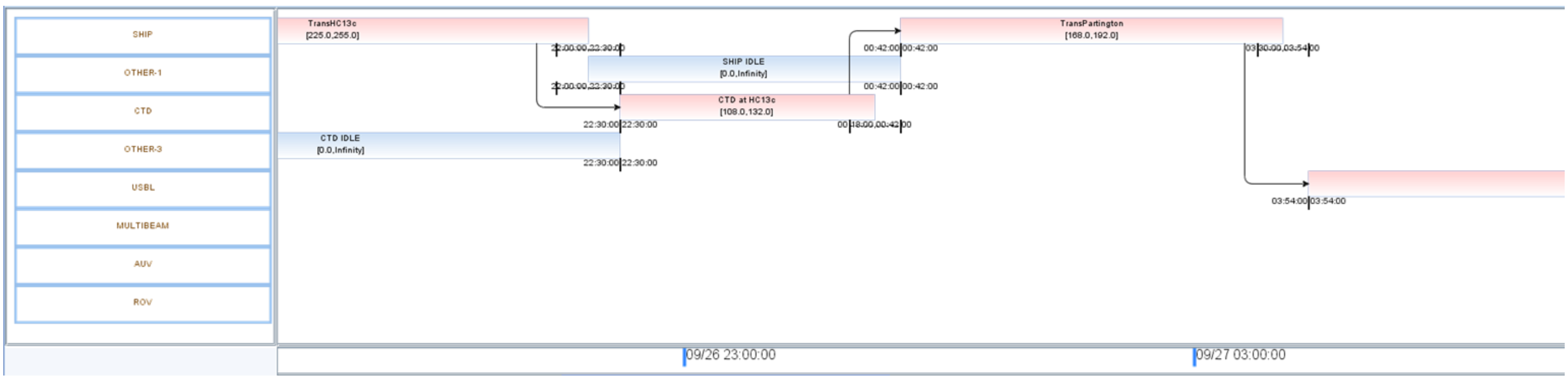
- [Day 3] Sentry' s mass spectrometer failed during its second dive. They sent Rich to Pittsburg to get it fixed.

… …

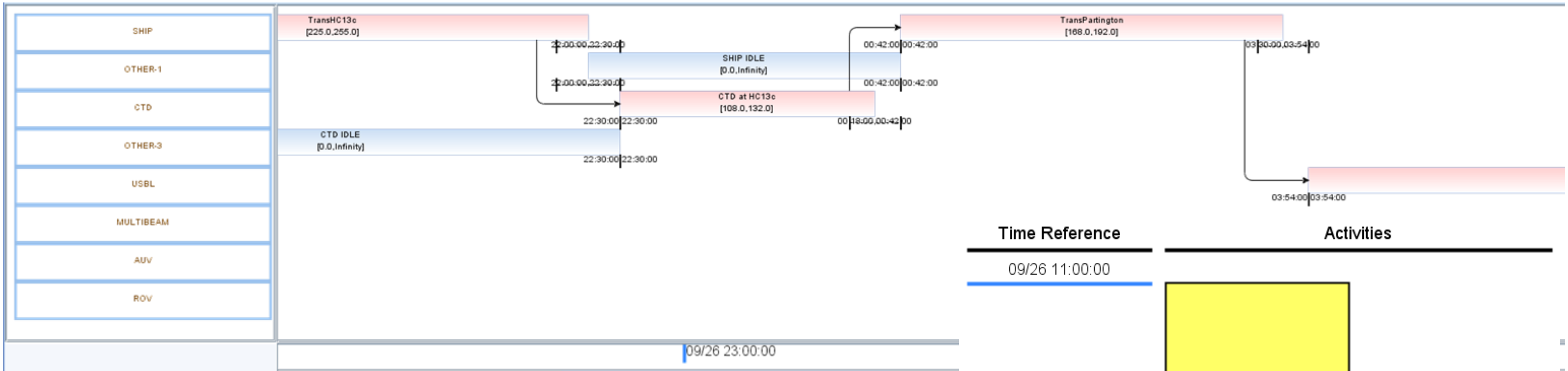
- [Day 7] Sentry aborted its mission 1 hour after launch. Atlantis aborted its mapping routes and went back to recover Sentry. The failure was caused by a burned wire.

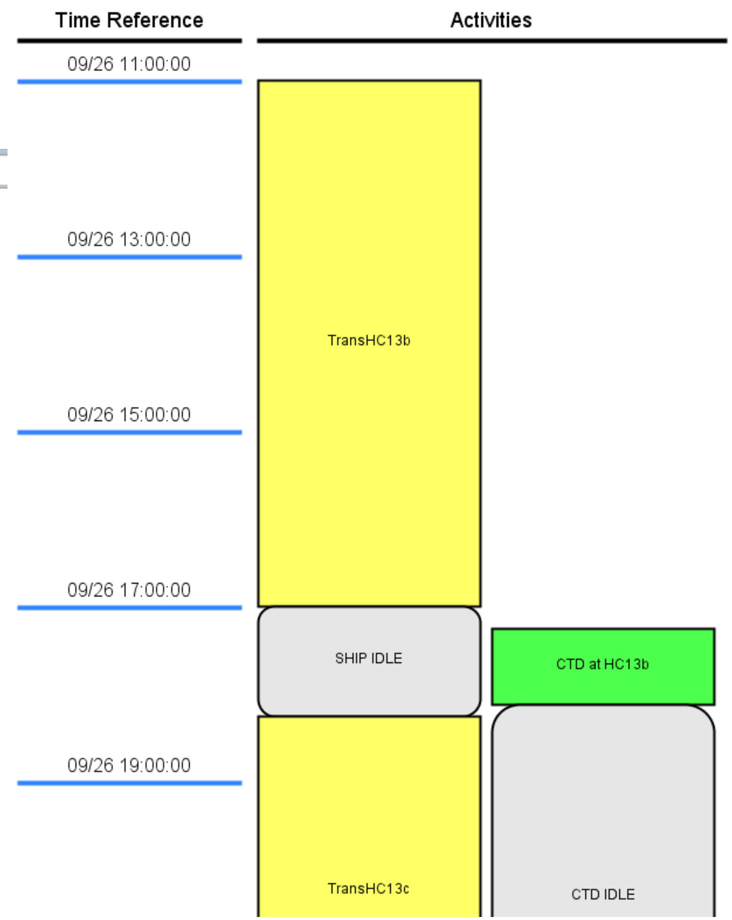
Our Deliverable

- A mission advisory system that assists the chief scientists of expeditions on the following tasks:
  - Scheduling Activities with Uncertainty.
  - Failure and Downtime Recovery Scheduling.
  - Assets Managements.



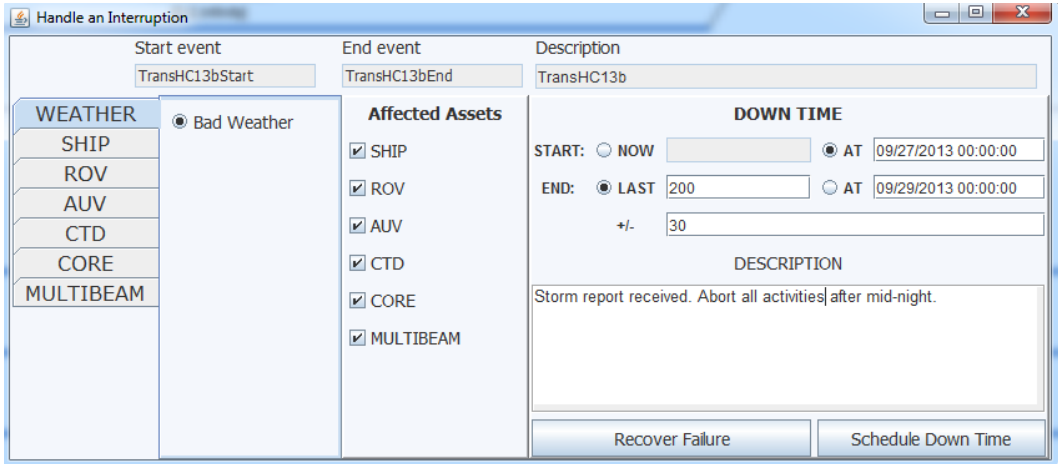
Scheduling Activities with Uncertainty

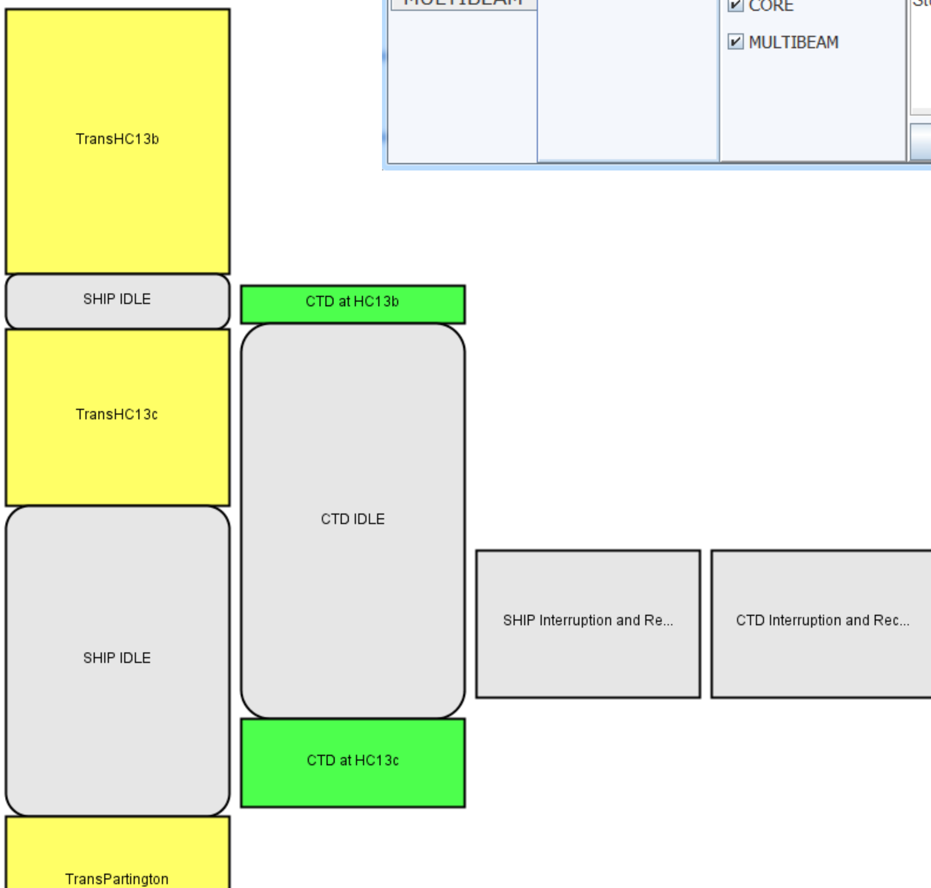




- Plan and schedule activities with temporal uncertainty.
- Make optimal decisions and relaxations between alternative goals and requirements.
- Present solutions using either a calendar or a Gantt chart (assets based).

Failure Recovery and Downtime Scheduling



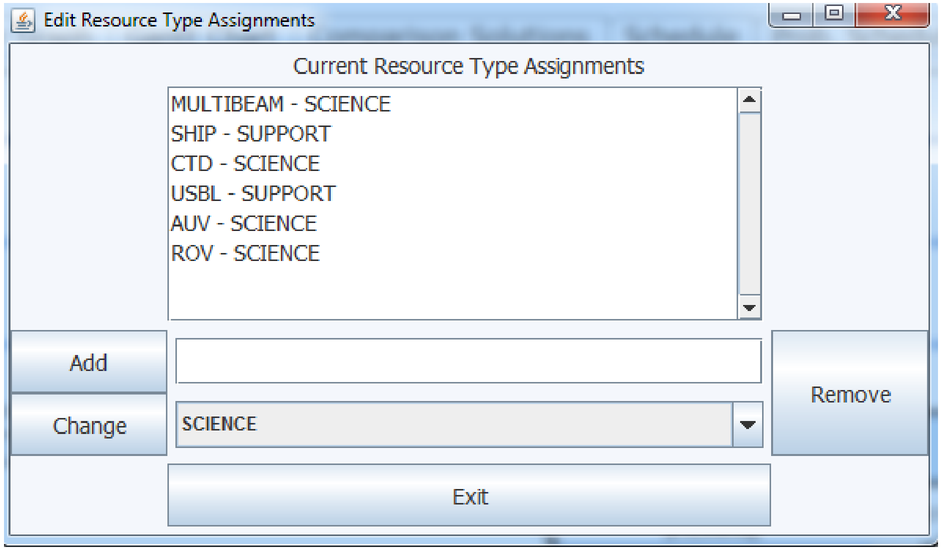


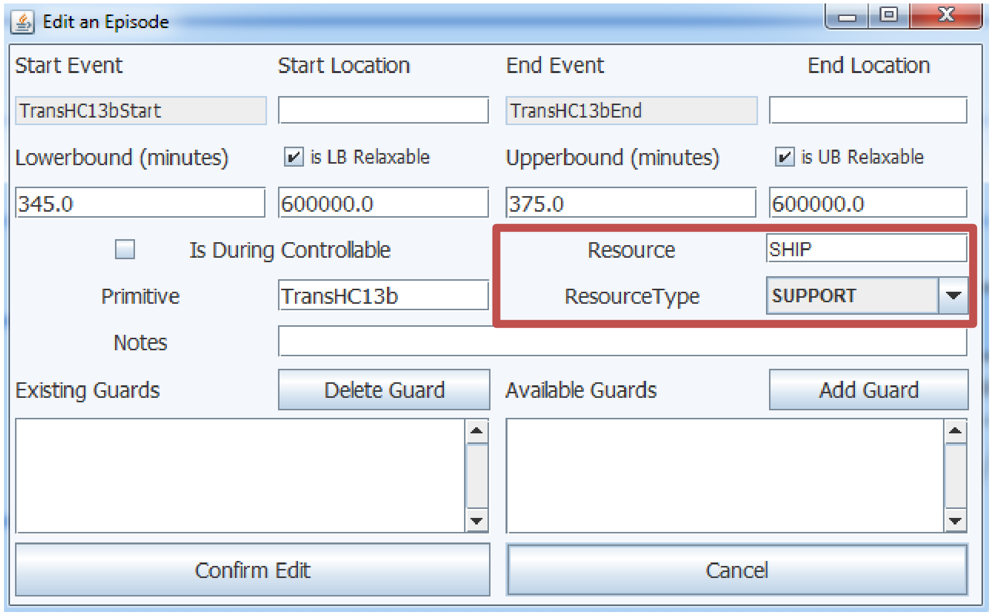
- When a problem/failure is happening or expected:
  - Work with the user to identify affected assets.
  - Identify affected tasks and reschedule them to avoid danger and failure.

Assets Managements

- Each activity is tagged with an asset that is required for its completion.

- This tag will be used to determine if this activity will be affected while an asset failure or down time is expected.





Contents

- Relaxations of Temporal Problems;
- Continuous Relaxation and Conflict Resolution;
- Restoring Controllability with Uncertainty Durations;
- Best-first Enumeration through Conflict-directed Relaxation;
- Empirical Evaluation.

Problem Formulation

- Model: (Over-constrained) Controllable Conditional Temporal Problems with Uncertainty.
  - All choices are **controllable**.
  - Allowing temporal constraints to be **relaxed**.
  - Allowing uncertain durations to be **tightened**.

- A solution is a tuple with:
  - A complete set of **decisions**.
  - A set of continuous **relaxations** for temporal constraints.
  - A set of continuous **tightening** for uncertain durations.

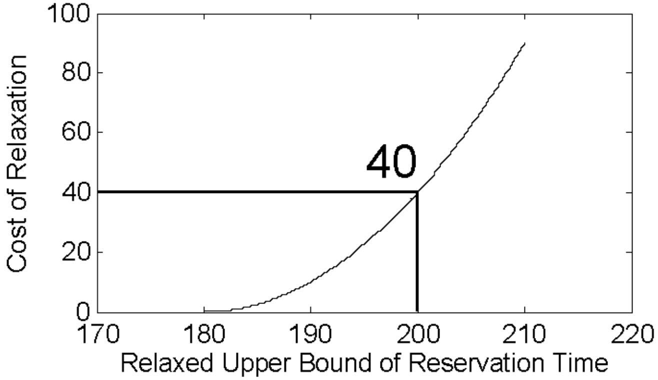
  such that the set of activated durations and constraints is consistent/controllable.

Define User Preferences

- Preference functions are defined over decisions, relaxation and tightening.
  - Each decision is mapped to a positive **reward** by function f_p.
  - Each relaxation/tightening is mapped to a positive **cost** by function f_e.

Store	A	40
	B	100
Lunch	X	70
	Y	80
	Z	30

Assignment :$\{Store = B, Lunch = Y\}$
Reward: $100 + 80 = 180$



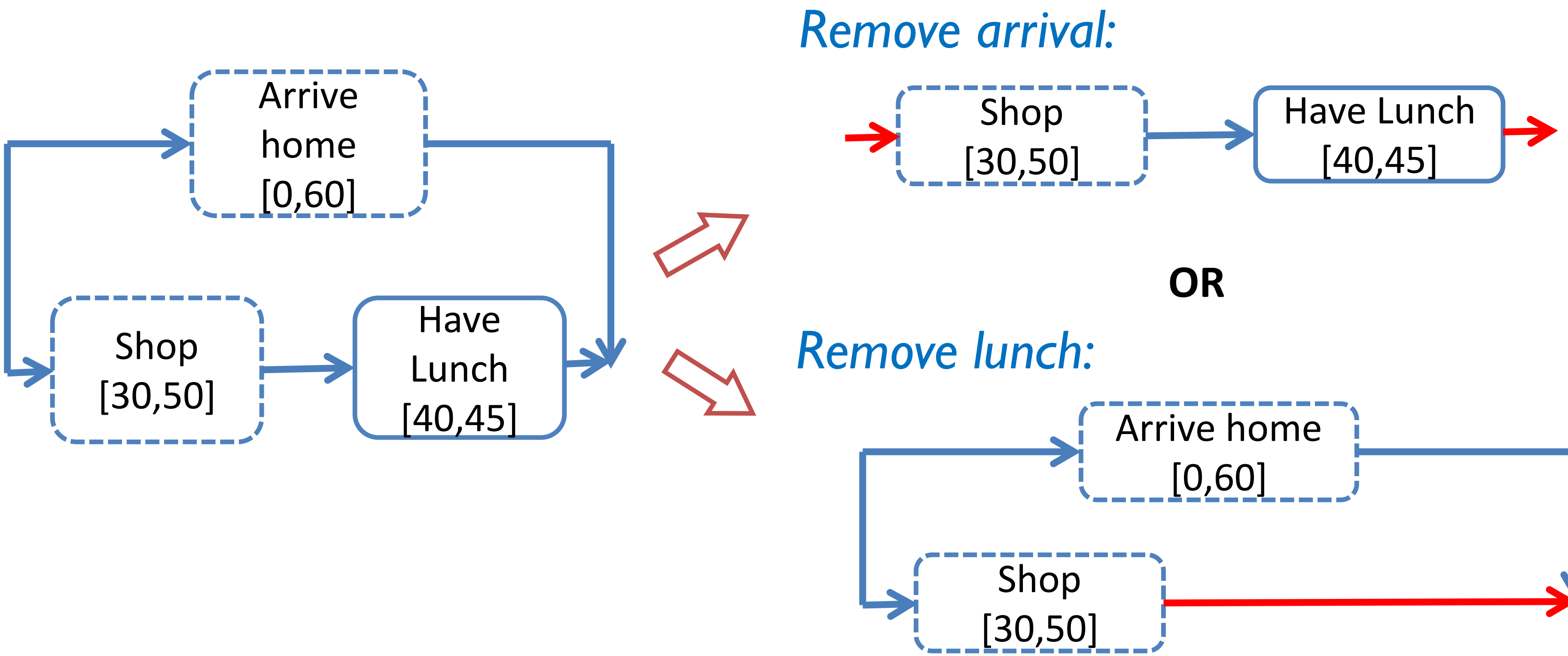
Relaxation: $Reservation[0,180] \rightarrow [0,200]$
Cost: $f_e(200 - 180) = 40$

Contents

- Relaxations of Conditional Temporal Problems;
- Continuous Relaxation and Conflict Resolution;
- Restoring Controllability with Uncertainty Durations;
- Best-first Enumeration through Conflict-directed Relaxation;
- Empirical Evaluation.

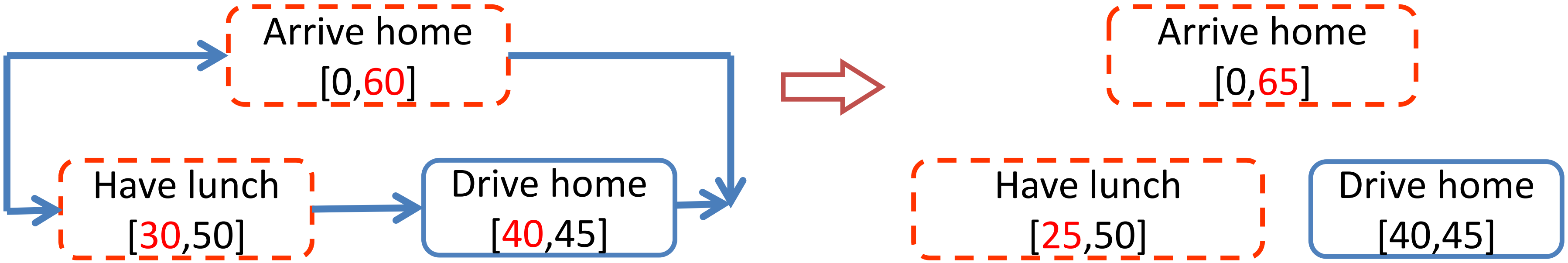
(Minimal) Discrete Relaxation

- Resolve over-constrained temporal problem C by **removing** constraints.
  - Resolved: $M \subseteq C$ *such that* $C \setminus M$ *is consistent*.
  - Minimal: $\forall c \in M \ (C \setminus M) \cup \{c\}$ *is inconsistent*.



Continuous Relaxation

- Relax a constraint partially by **continuously** modifying its temporal bounds:
  - A continuous relaxations, CR_i, weakens a temporal constraint: $[LB, UB] \rightarrow [LB', UB']$ where $LB' \leq LB \text{ and } UB' \geq UB$.
  - Continuous relaxations only apply to **relaxable** constraints.



"Shorten lunch to 25 minutes and delay arrival by 5 minutes"

Discrete vs. Continuous Relaxations

- Resolve a conflict by relaxing constraints **completely** or **partially**.

Conflict:

Store = B, Lunch = Y;
Home → B ≥ 35; Shop at B ≥ 35;
Drive B → Y ≥ 25; Lunch at Y ≥ 75;
Y → Home ≥ 40; Arrive Home ≤ 180.

Discrete Resolutions

Remove Shop at B ≥ 35;
Remove Lunch at Y ≥ 75;
Remove Arrive Home ≤ 180

Continuous Resolutions

Lunch at Y ≥ 45;
Arrive Home ≤ 210;
Shop at B ≥ 25 and Lunch at Y ≥ 55;
… …
and many more

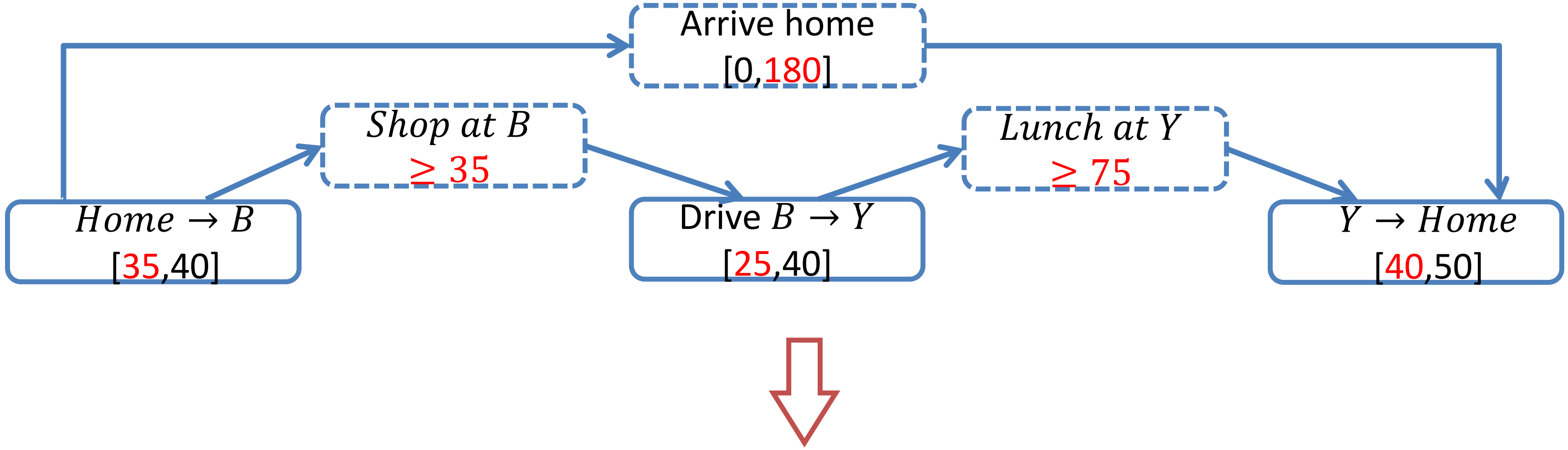
Contents

- Relaxations of Conditional Temporal Problems;
- Continuous Relaxation and Conflict Resolution;
- Restoring Controllability with Uncertainty Durations;
- Best-first Enumeration through Conflict-directed Relaxation;
- Empirical Evaluation.

1. Learn Discrete Conflicts

- A discrete conflict is an inconsistent set of temporal constraints.

Choosing Store=B and Lunch=Y produces:



Discrete Conflict:

Store = B;
Home → B ≥ 35;
Drive B → Y ≥ 25;
Y → Home ≥ 40;

Lunch = Y;
Shop at B ≥ 35;
Lunch at Y ≥ 75;
Arrive Home ≤ 180.

2. Weaken to Continuous Conflicts

- A continuous conflict is an equation formed from the discrete conflict.
- It specifies the deviation needed to resolve the conflict.

Discrete Conflict:

$\mathrm{HometoB} \geq 35;$
$\mathrm{ShopatB} \geq 35;$
$\mathrm{BtoY} \geq 25;$
$\mathrm{LunchatY} \geq 75;$
$\mathrm{YtoHome} \geq 40;$
$\mathrm{ArriveHome} \leq 180.$

⇨

Continuous Conflict:

$$ArriveHome - HometoB - ShopatB - BtoY - LunchatY - YtoHome = \mathbf{-30}$$

3. Map to Constituent Continuous Relaxations

- Relaxations specified by linear inequalities:

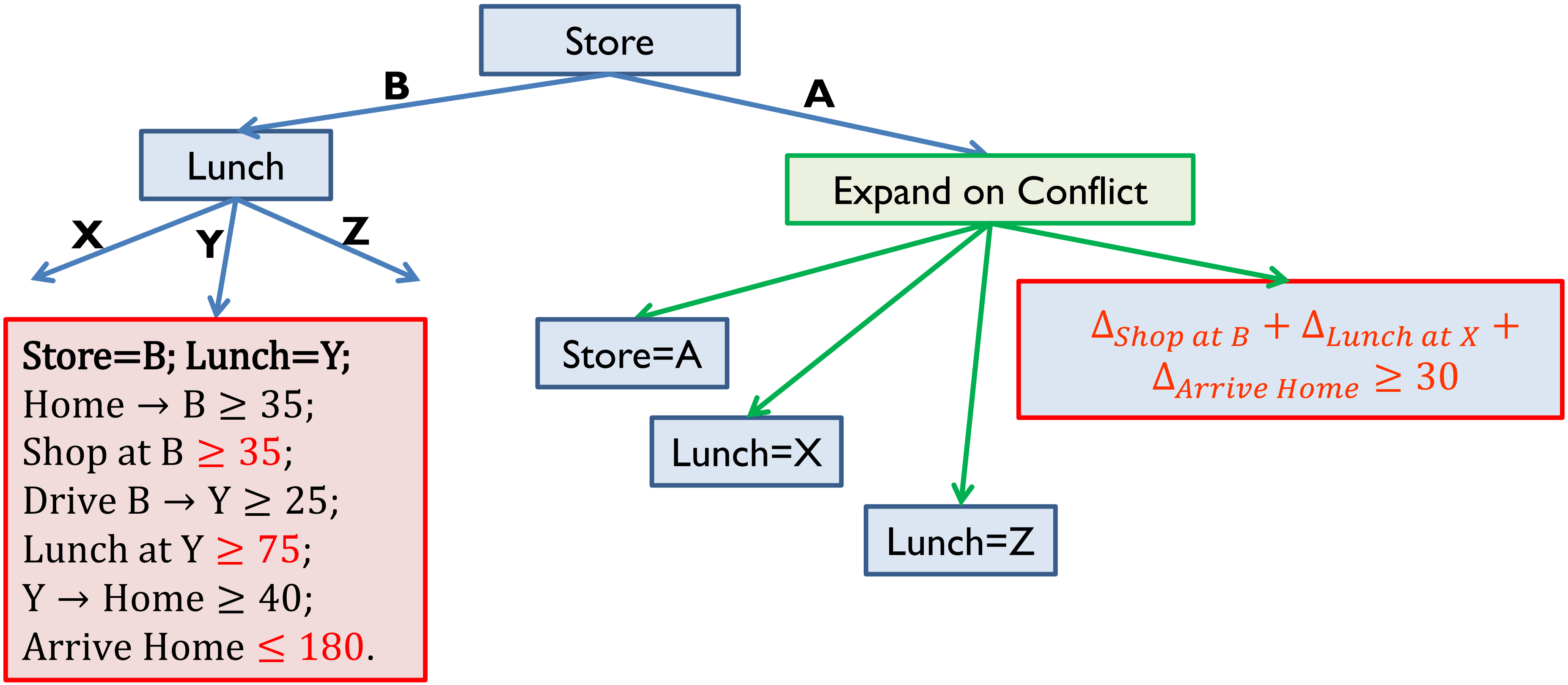
$$ArriveHome - HometoB - ShopatB - BtoY - LunchatY - YtoHome = \mathbf{-30}$$

$$\Downarrow$$

$$\Delta_{ShopatB} + \Delta_{LunchatY} + \Delta_{ArriveHome} \geq 30$$

Expand with Discrete and Continuous Resolutions

- If a node has an unresolved conflict, we expand it using both constituent **continuous** relaxation and **decisions** that deactivates its constraints (Yu and Williams, 2013).



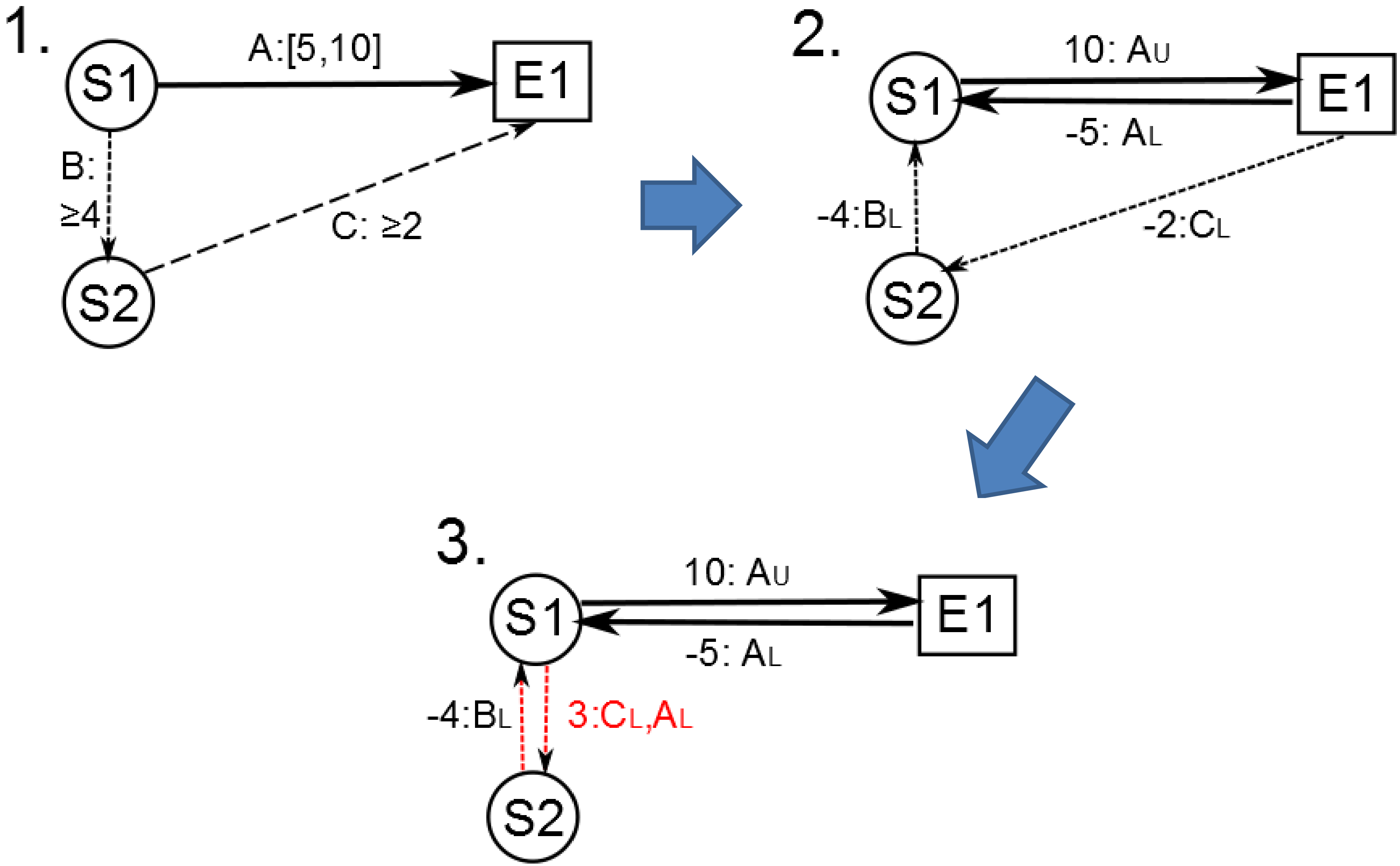
Contents

- Relaxations of Conditional Temporal Problems;
- Continuous Relaxation and Conflict Resolution;
- Restoring Controllability with Uncertainty Durations;
- Best-first Enumeration through Conflict-directed Relaxation;
- Empirical Evaluation.

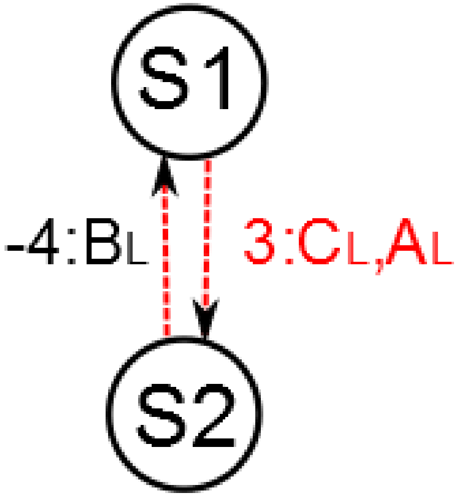
Learn Conflicts From Uncontrollable Problems

- Learning conflicts from controllability checking algorithms is more difficult.
  - For consistency checking, there is a **one-to-one mapping** between the distance edges and the bounds of constraints.
  - No such mapping exists for controllability checking (strong and dynamic) **due to the reduction procedures**, making it difficult to extract conflicts from the reduced graph.
- Key: during the reduction, **record the 'contribution'** of each constraint and duration in the temporal problem.

A Strong Controllability Example



Resolving Uncontrollable Conflicts

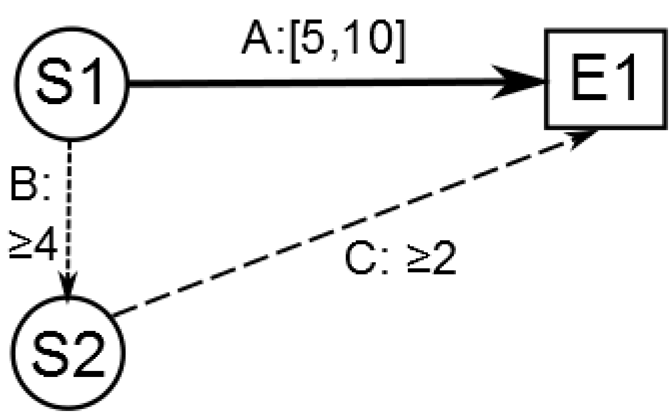


- Constraint for resolving continuous conflict (negative value -1):

$$\Delta C_L + \Delta B_L + \Delta A_L \geq 1$$

where:

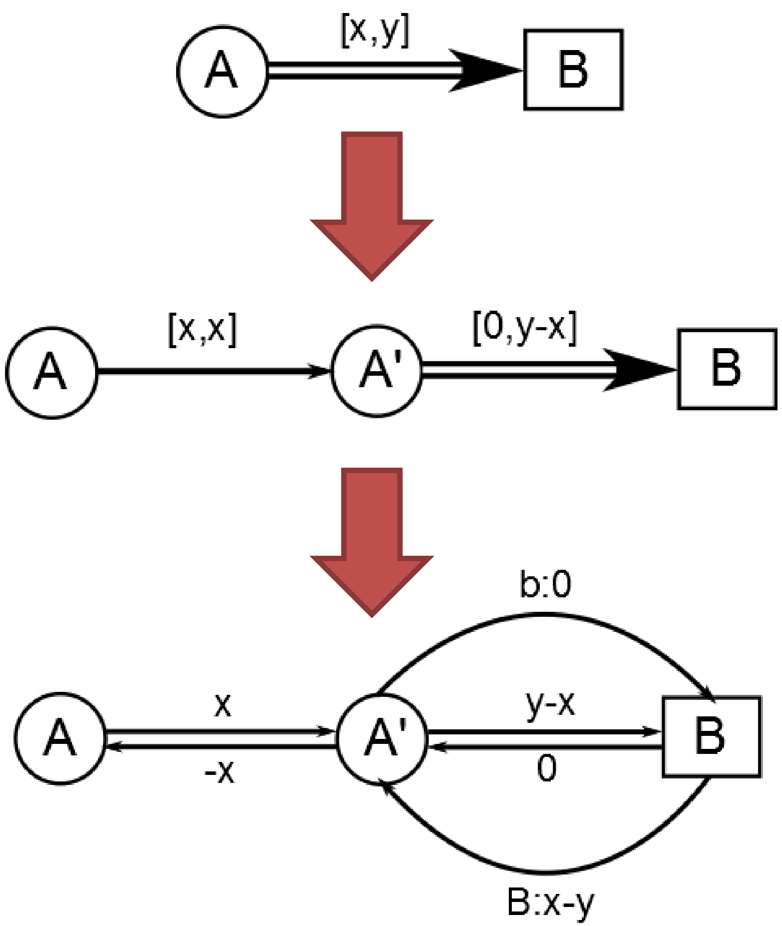
- $\Delta B_L, \Delta C_L$ are relaxations for B and C.
- ΔA_L is tightening for A.



and

$$\Delta A_L \leq 5$$

Learning Dynamically Uncontrollable Conflict



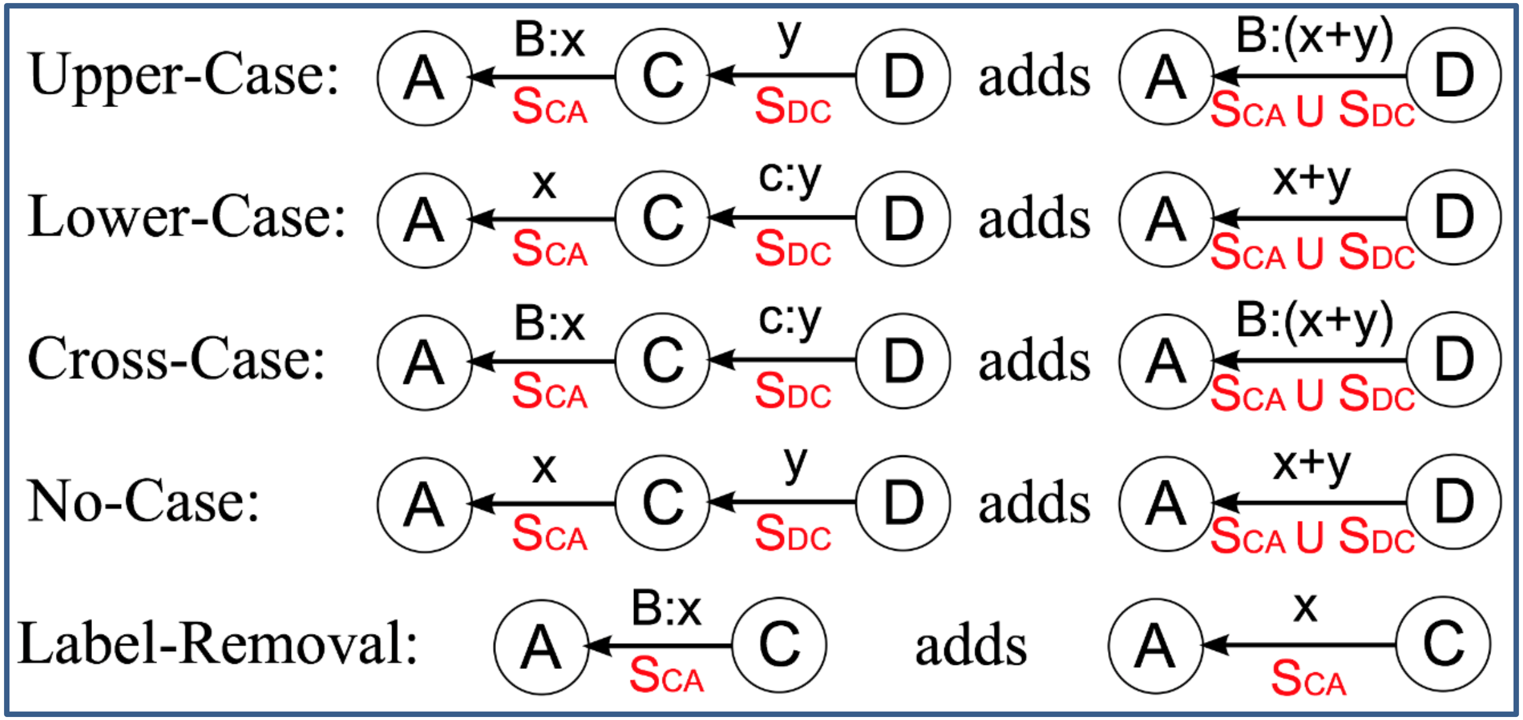
Record supporting constraints for both **requirement** and **conditional** edges while generating the directed graphs.

$A-(x)\rightarrow A'$	AB_{Lower}
$A'-(-x)\rightarrow A$	$-AB_{Lower}$
$A'-(y-x)\rightarrow B$	$-AB_{Upper}$ $-AB_{Lower}$
$A'-(b{:}\,0)\rightarrow B$	None
$B-(0)\rightarrow A'$	None
$B-\ (B{:}\,x-y)\rightarrow A'$	AB_{Upper}

Paul Morris. A structural characterization of temporal dynamic controllability. In Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming (CP-2006), pages 375–389, 2006

Learning Dynamically Uncontrollable Conflict

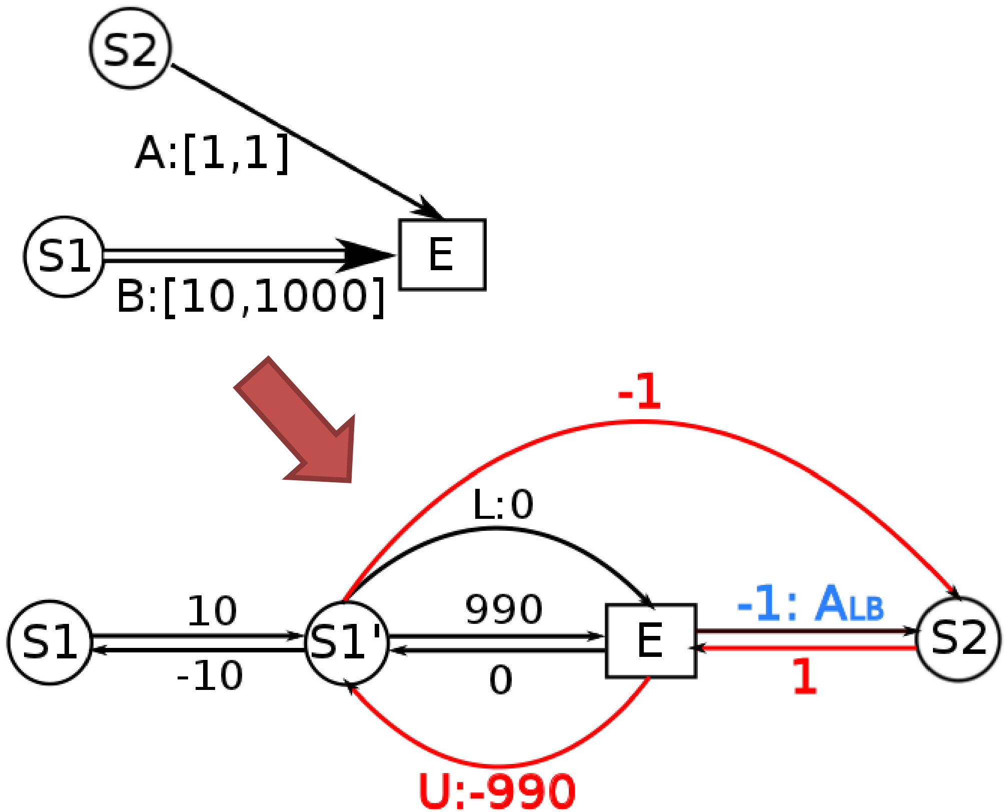
- **Record supporting** constraints and durations during the iterative reduction procedure.



- Note that a constraint may be recorded multiple times during reduction.

Another Way to Resolve DC Conflicts

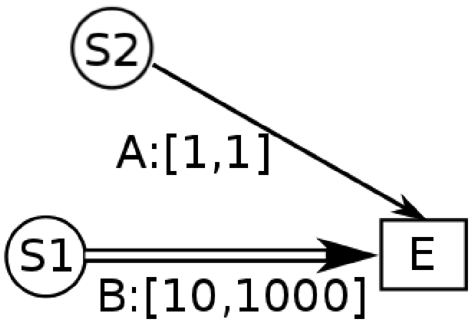
- We can resolve a conflict by **disabling** reductions that lead to edges in the negative loop*.

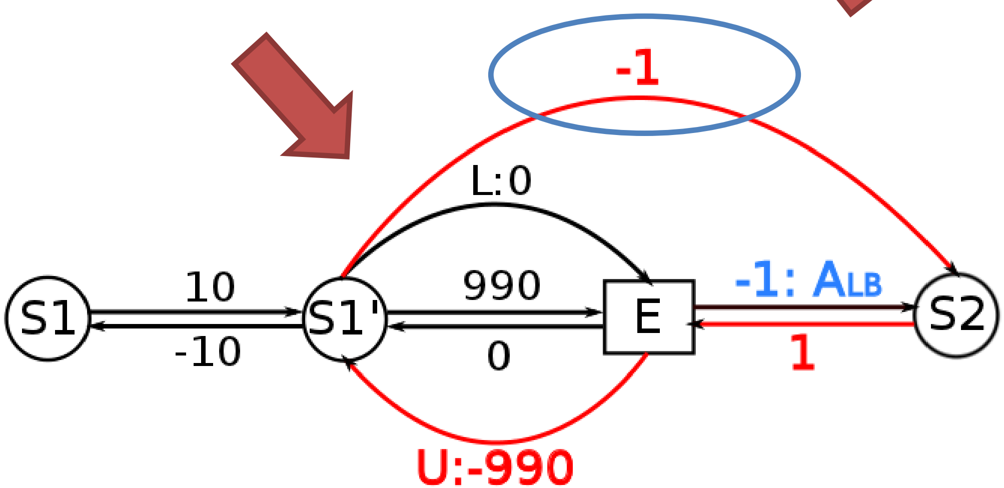


*A STNU is dynamically controllable if and only if it does not have a **semi-reducible** negative loop [Morris 2006].

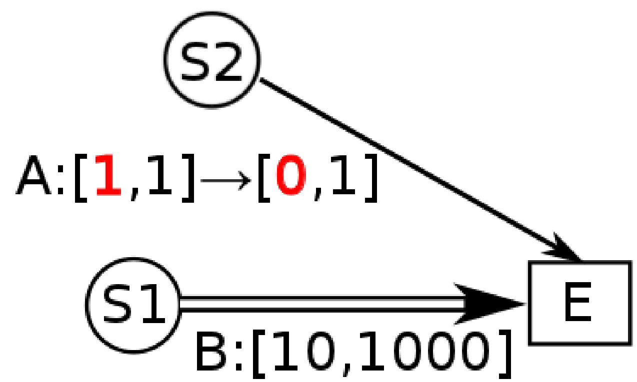
Another Way to Resolve DC Conflicts

- We can resolve a conflict by **disabling** reductions that lead to edges in the negative loop*.





Relax the lowerbound of A ***from 1 to 0*** *to disable the reduction*.



*A STNU is dynamically controllable if and only if it does not have a **semi-reducible** negative loop [Morris 2006].

Contents

- Relaxations of Conditional Temporal Problems;
- Continuous Relaxation and Conflict Resolution;
- Restoring Controllability with Uncertainty Durations;
- Best-first Enumeration through Conflict-directed Relaxation;
- Empirical Evaluation.

Generalize CDA* to Continuous Relaxations

- Conflict-Directed A* (Williams and Ragno, 2004) can be applied to discrete relaxation problems:
  - Efficiently prunes search space using learned conflicts.
  - Enumerates minimal discrete relaxations in best-first order.

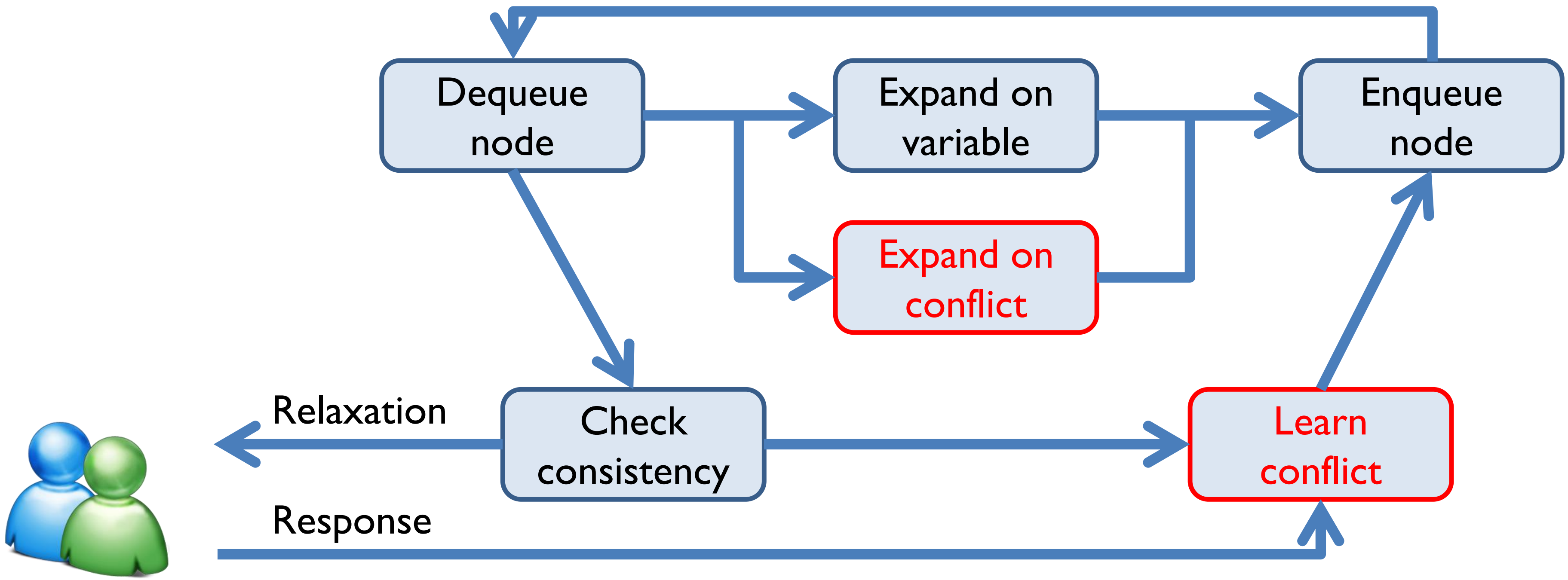
- To solve a relaxation problem:
  - Frame an equivalent constraint optimization problem.

    A. Discrete relaxation: add binary variables.

    B. Continuous relaxation/tightening: add non negative continuous variables.
  - The objective function represents the preference.

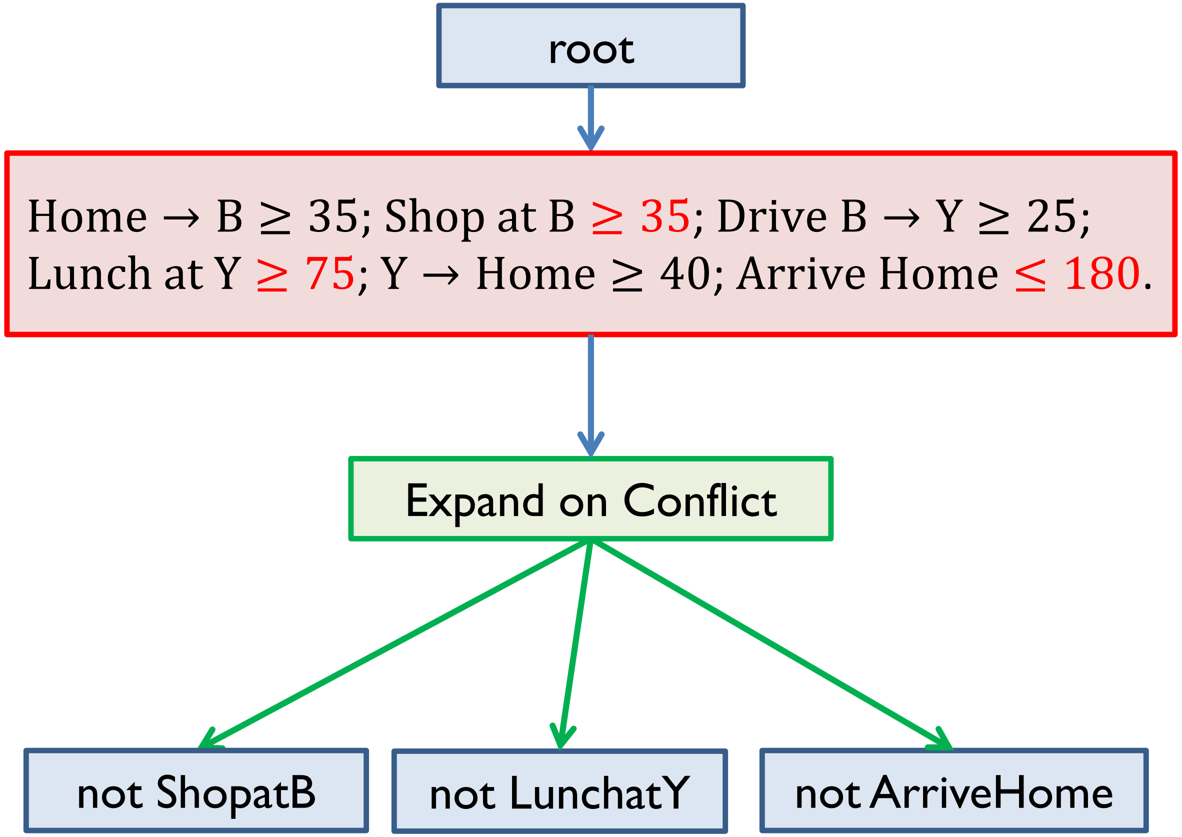
Best-first Conflict Directed Relaxation

- BCDR generalizes the conflict resolution procedure in CDA* to include constituent continuous relaxations.



Conflict-Directed A*

- Key Ideas:
  - Split on conflict;
  - Best-first enumeration.



CDA* with Constituent Continuous Relaxation

- Split a conflict using its constituent continuous relaxations.

root

↓

Home → B ≥ 35; Shop at B ≥ 35; Drive B → Y ≥ 25; Lunch at Y ≥ 75; Y → Home ≥ 40; Arrive Home ≤ 180.

↓

Expand on Conflict

↓

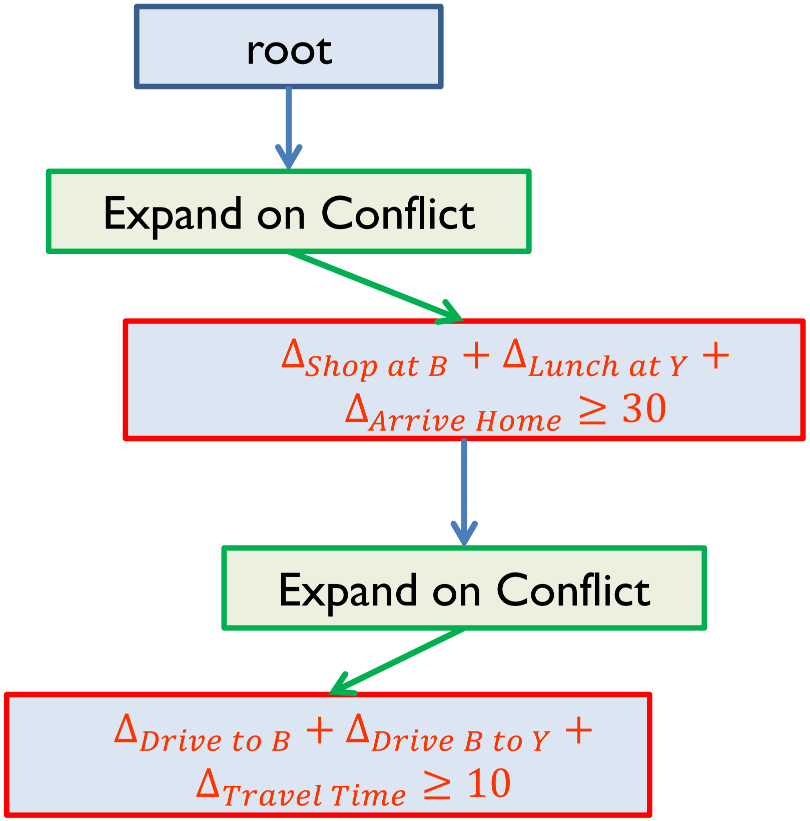
$$\Delta_{Shop\ at\ B} + \Delta_{Lunch\ at\ Y} + \Delta_{Arrive\ Home} \geq 30$$

$$\min\left(f\left(\Delta_{Shop\ at\ B}\right) + f\left(\Delta_{Lunch\ at\ Y}\right) + f\left(\Delta_{Arrive\ Home}\right)\right)$$

s.t. $\Delta_{Shop\ at\ B} + \Delta_{Lunch\ at\ Y} + \Delta_{Arrive\ Home} \geq 30$

Continuous Relaxations for Multiple Conflicts

- For two or more continuous relaxations on the same branch, the utility is determined by the grounded solution that respects both inequalities.



$$\min(f(\Delta_{Shop\ at\ B}) + f(\Delta_{Lunch\ at\ Y}) + f(\Delta_{Arrive\ Home}) + f(\Delta_{Drive\ to\ B}) + f(\Delta_{Drive\ B\ to\ Y}) + f(\Delta_{Travel\ Time}))$$

s.t. $\Delta_{Shop\ at\ B} + \Delta_{Lunch\ at\ Y} + \Delta_{Arrive\ Home} \geq 30$

and

$$\Delta_{Drive\ to\ B} + \Delta_{Drive\ B\ to\ Y} + \Delta_{Travel} \geq 10$$

Incorporating User Responses

- BCDR incrementally adapts to new requirements.
- These requirements are recorded as new conflicts.

Required Continuous Relaxations

No, I **do not** want to extend my reservation time.

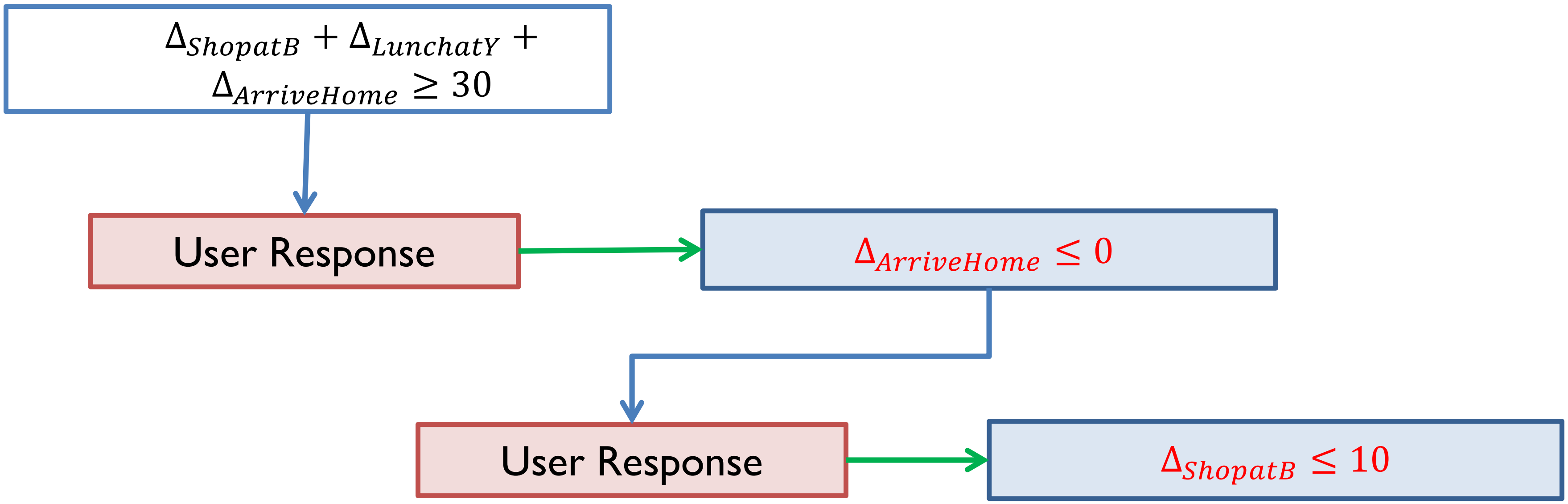
$\Delta_{Arrive\ Home} \leq 0;$

No, I want to spend **at least 25** minutes on shopping.

$\Delta_{Shop\ at\ B} \leq 10;$

New Requirements as Conflicts

- Expand search tree using user response conflicts.



$$\min(f(\Delta_{ShopatB}) + f(\Delta_{LunchatY}) + f(\Delta_{ArriveHome}))$$

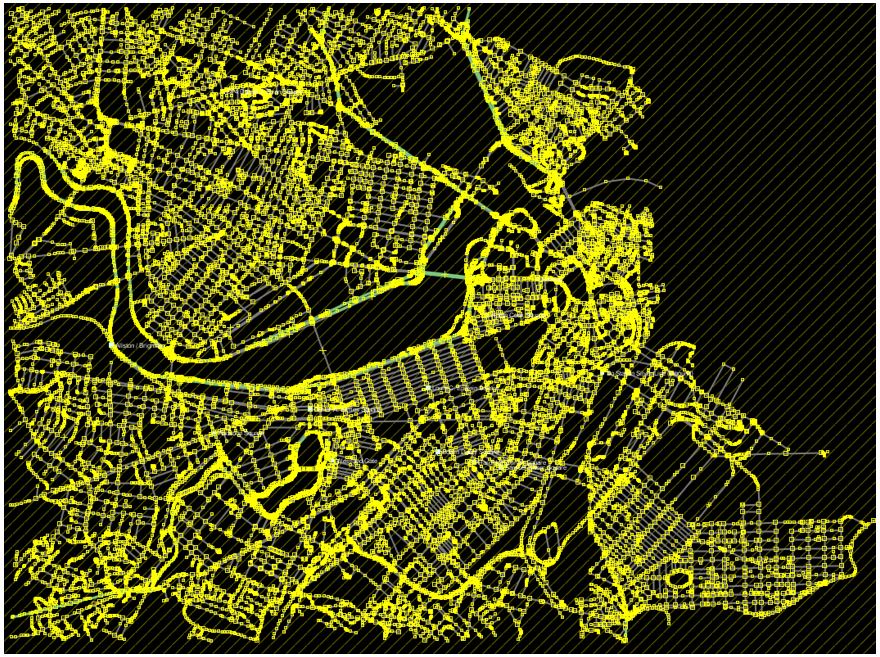
$$\text{s.t.} \quad \Delta_{ShopatB} + \Delta_{LunchatY} + \Delta_{ArriveHome} \geq 30;$$
$$\Delta_{ArriveHome} \leq 0;$$
$$\Delta_{ShopatB} \leq 10.$$

Contents

- Relaxations of Conditional Temporal Problems;
- Continuous Relaxation and Conflict Resolution;
- Restoring Controllability with Uncertainty Durations;
- Best-first Enumeration through Conflict-directed Relaxation;
- Empirical Evaluation.

Experiment Setup

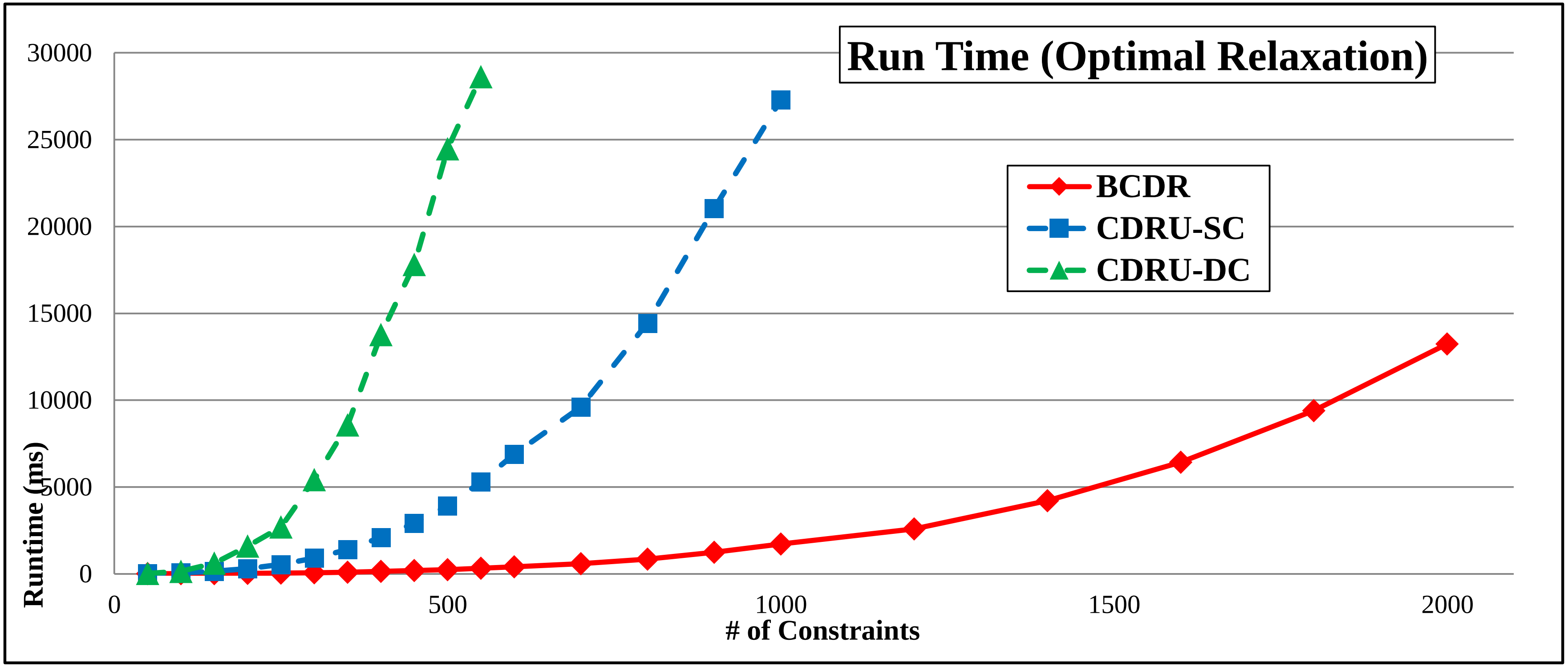
- We simulated a car-sharing network in Boston using randomly generated car locations and destinations.



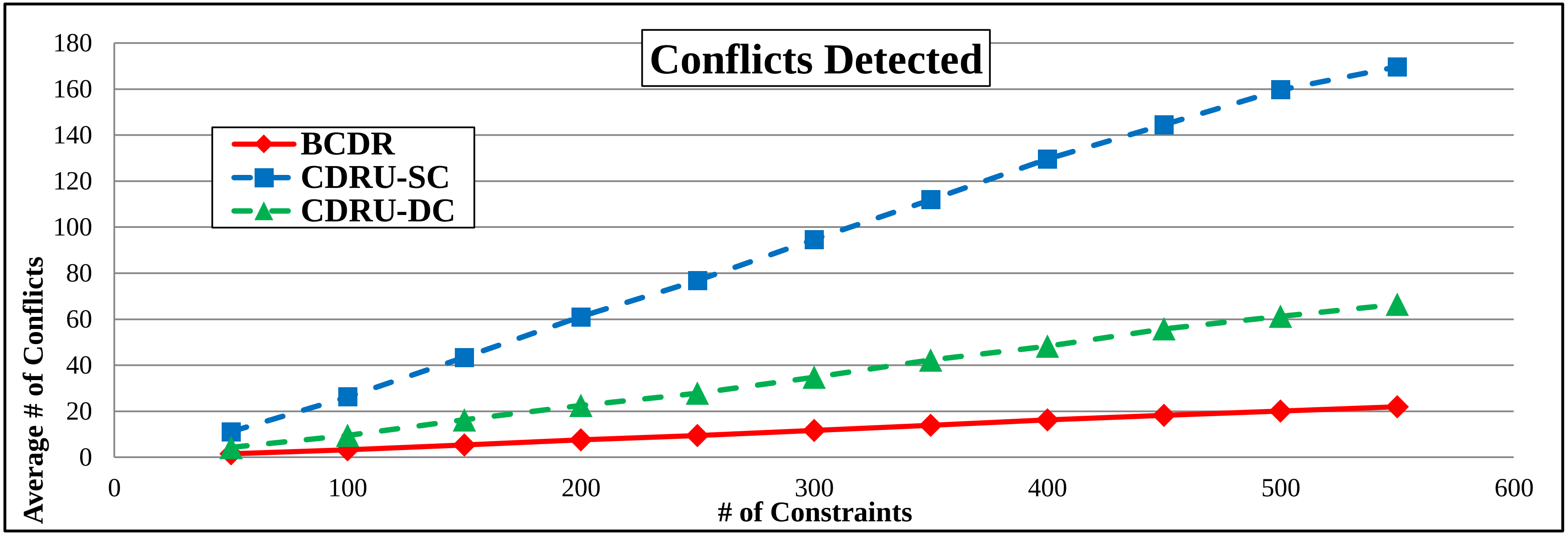
- Test cases are characterized by:
  - Number of reservations per car.
  - Number of cars in the network.
  - Number of activities per reservation.
  - Number of alternative options per activity.

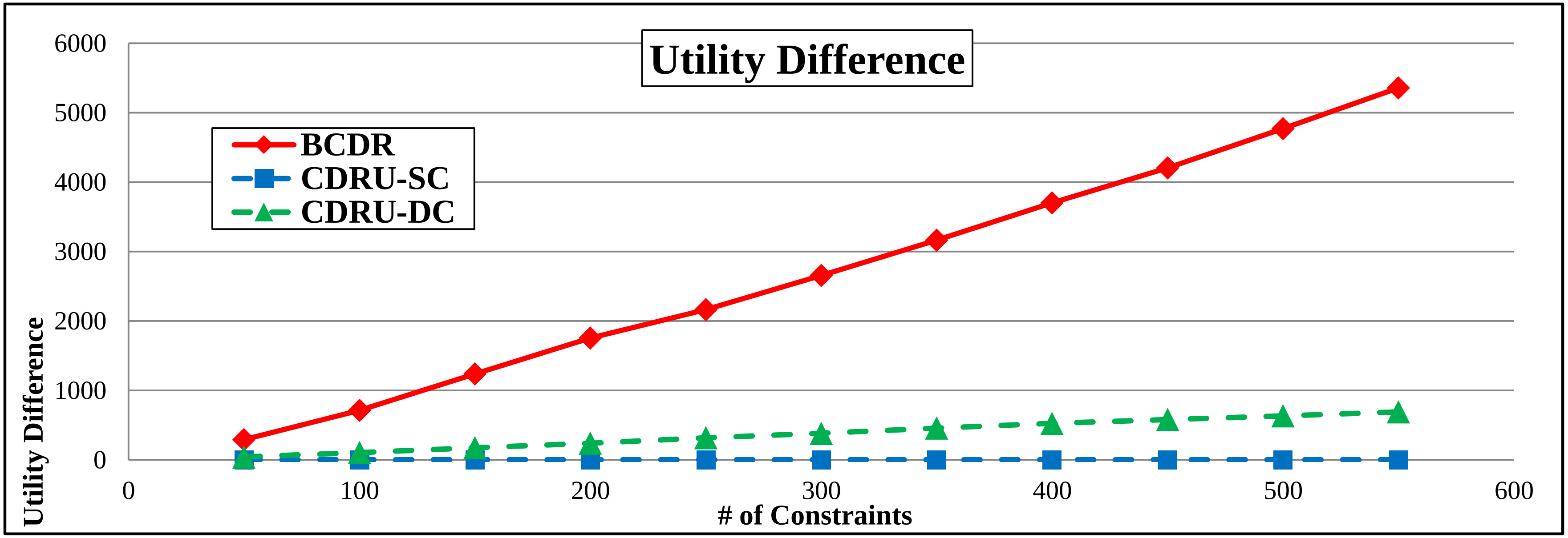
Empirical Results - Runtime

- We compare the performance of three algorithms:
  - BCDR (consistency).
  - CDRU-SC (strong controllability).
  - CDRU-DC (dynamic controllability).



Solution Utility and Conflicts Detected





Contributions

- Over-constrained temporal problems can be resolved by relaxing the temporal constraints continuously.
- The fundamental concepts of conflicts and minimal relaxations naturally generalize to the continuous case.
- The framework naturally extends to resolving uncontrollable problems with uncertain durations.
- We can efficiently enumerate discrete and continuous relaxations in best-first order, by generalizing the Conflict-Directed A* algorithm.

Acknowledgements

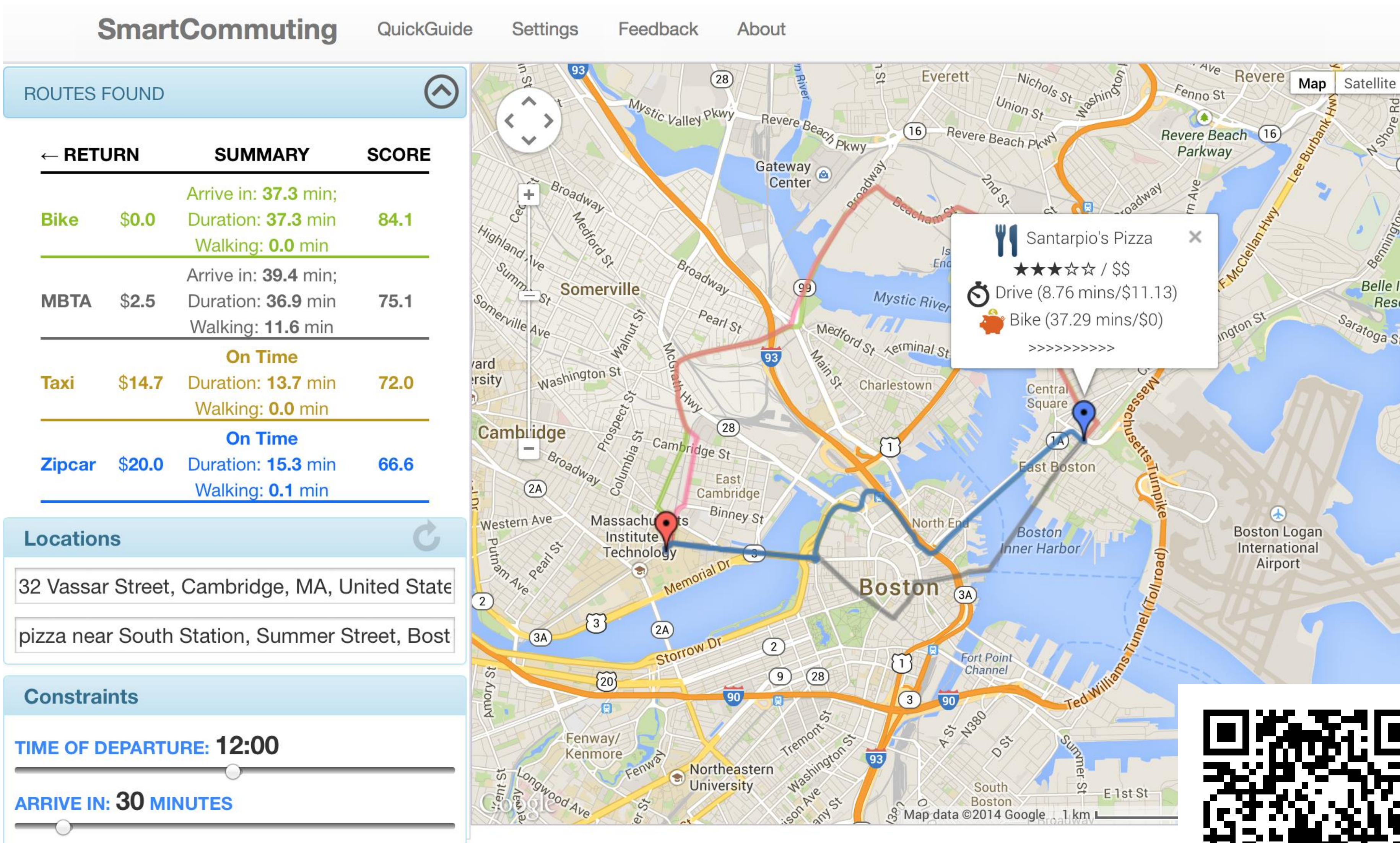
- This project is supported by the Boeing Company under contract MIT-BA-GTA-1, and by the Deep Submergence Lab at Woods Hole Oceanographic Institute.

- The authors want to thank Scott Smith, Ron Provine, Rich Camilli and Jing Cui for their help and valuable inputs on this project.

- The implementation and test cases can be downloaded from:

  http://people.csail.mit.edu/yupeng/software.html

A Transit Advisor Application



people.csail.mit.edu/yupeng/SmartCommuting/Boston.html