Continuously Relaxing Over-constrained Conditional Temporal Problems

Massachusetts Institute of Technology

Peng Yu, Brian C. Williams

Massachusetts Institute of Technology

CSAIL

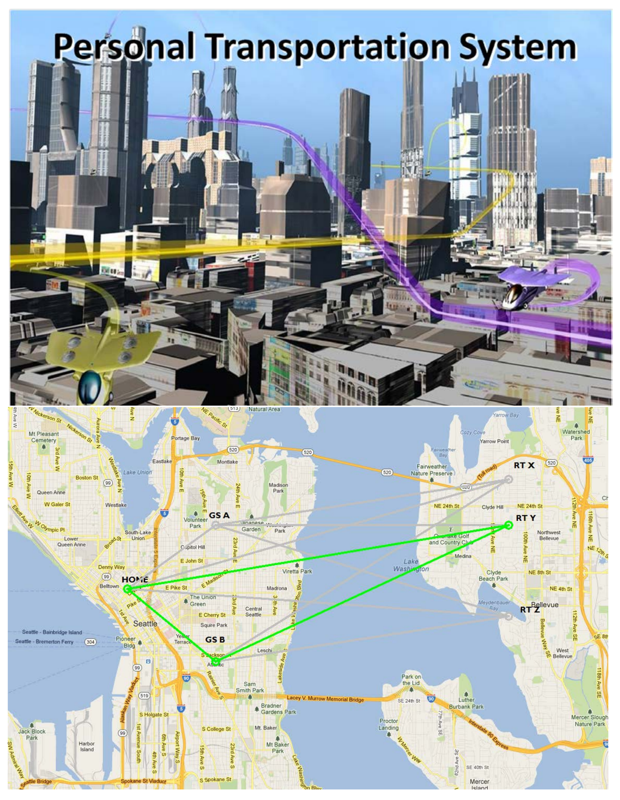
Motivation & Objective

Over-subscribed situations: when travelling, we want to do more activities than we have time for.

'I want to stop at a grocery store for 30 minutes, have lunch for 40 minutes, and get home within 60 minutes.'

Previous approach: an all-or-nothing strategy, in which some user requirements are "discretely" removed to make the situation feasible.

'You have to give up either shopping or lunch'.



Objective: Given an over-subscribed situation, continuously relax the requirements, while maximally preserving the user's goals:

'Delay your arrival by 5 minutes because of the extended travel time.'
'If you want to shop for 25 minutes, you can have lunch at restaurant Y for at most 55 minutes.'

Application: This project is in support of a travel advisor for the Personal Transportation System (PTS) project, a robotic air taxi.

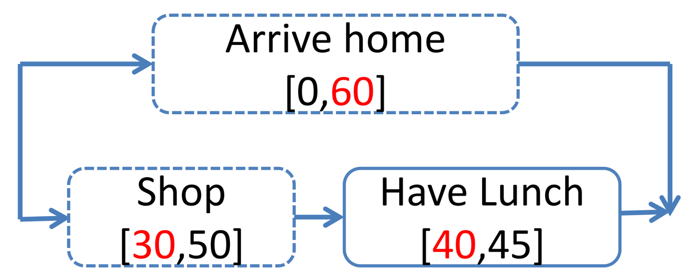
- We model travel plans using Controllable Conditional Temporal Problems (CCTP), in which a subset of temporal constraints can be relaxed, and choices are controlled by the user.

Discrete vs. Continuous Relaxations

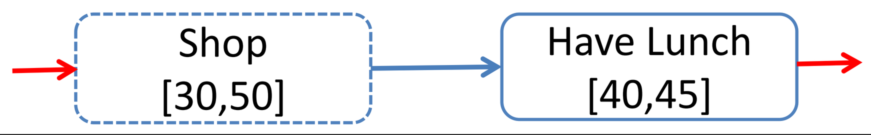
Discrete Relaxations

- Resolve over-constrained temporal problems by completely **suspending** constraints [1][2].
- $M \subseteq C$ *such that* $C \backslash M$ *is consistent.*

For example,



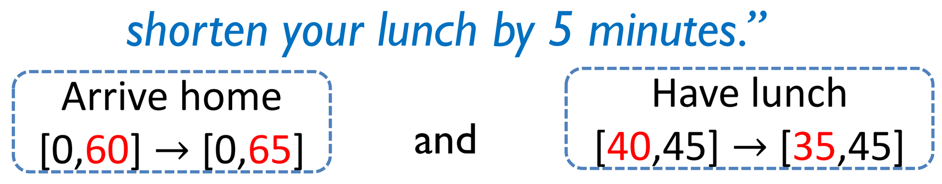
"Remove your trip duration requirement".



Continuous Relaxations

- Resolve by relaxing constraints **partially,** thus reducing the perturbation.
- A *continuous relaxation*, CR_i, weakens a temporal constraint $[LB, UB]$ to $[LB', UB']$ where $LB' \leq LB$ *and* $UB' \geq UB$.
- A valid set of continuous relaxations restores the consistency of a temporal problem.

"Delay your arrival by 5 minutes, and shorten your lunch by 5 minutes."



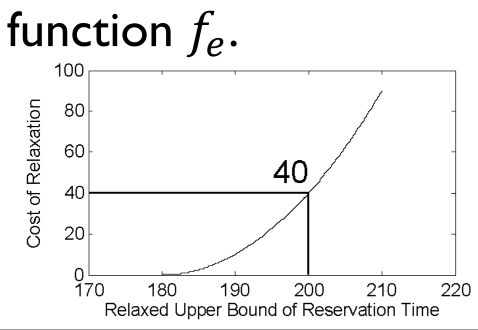
Defining Preferred Solutions

1) Each choice in the problem is mapped to a positive reward using function f_p, and computed using addition.

Store	A	40
	B	100
Lunch	X	70
	Y	80

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

2) Each *constraint relaxation* is mapped to a positive cost using function f_e.

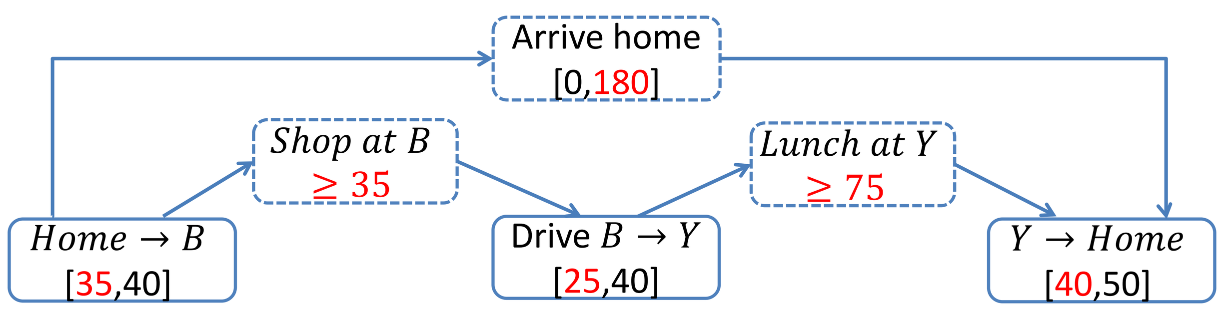


Relaxation: *TravelTime*$[0,180] \rightarrow [0,200]$
Cost: $f_e(200 - 180) = 40$

Generating Continuous Relaxations for "Continuous" Conflicts

Step 1: Learn Discrete Conflicts

- A discrete conflict composes of an inconsistent set of temporal constraints and their guard assignments.
- We learn conflicts from the negative cycles detected by temporal consistency algorithms. For example: **Store = B, Lunch = Y**, and:



Step 2: Map to Continuous Conflicts

- A continuous conflict is defined using an equality that connects a set of constraints and a negative value.
- We map a negative cycle and all its temporal constraints to a continuous conflict.

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

$$\boldsymbol{ArriveHome - Home \rightarrow B - Shop\ at\ B - DriveB \rightarrow Y - Lunch\ at\ Y - Y \rightarrow Home = -30}$$

Step 3: Solve for Minimal Continuous Relaxations

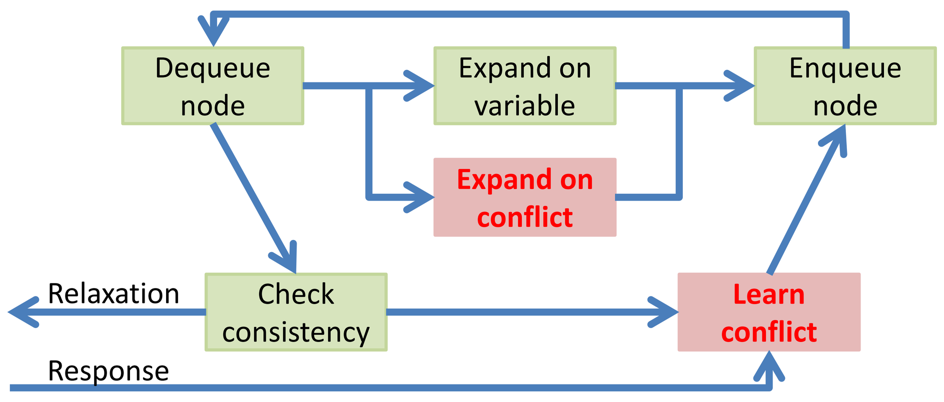
- A minimal continuous relaxation is defined through a set of linear inequalities based on a continuous conflict.
- The inequality only involves constraints that can be relaxed.

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

"This is the minimum amount of relaxation for resolving the conflict"

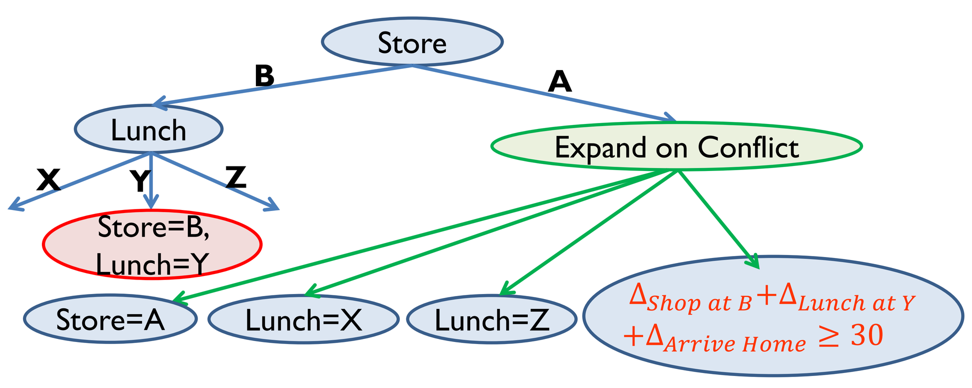
Best-first Enumeration of Relaxations

We developed the Best-first Conflict-Directed Relaxation (BCDR) algorithm for enumerating continuous relaxations. It generalizes Conflict-directed A*[3] with the continuous conflict resolution technique given above.



Each time a conflict is used to expand the search tree, we use both

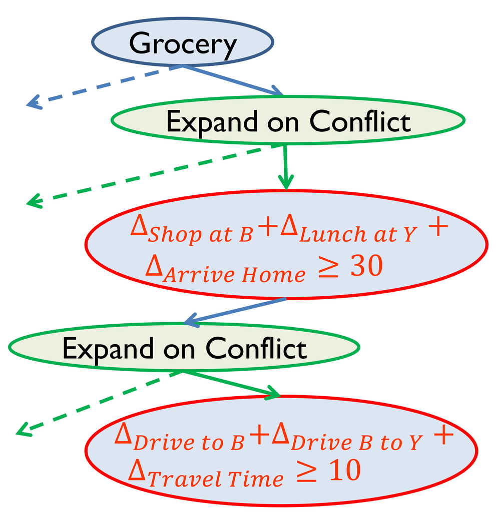
1. alternative assignments that suspend constraints in the conflict.
2. the minimal continuous relaxation to the conflict.



The utility of a search node is evaluated by the grounded continuous relaxation of the lowest cost, subject to all minimal relaxations on its branch.

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

$$\text{s.t. } \Delta_{Shop\ at\ B} + \Delta_{Lunch\ at\ Y} + \Delta_{Arrive\ Home} \geq 30$$
$$\Delta_{Drive\ to\ B} + \Delta_{Drive\ B\ to\ Y} + \Delta_{Travel} \geq 10$$

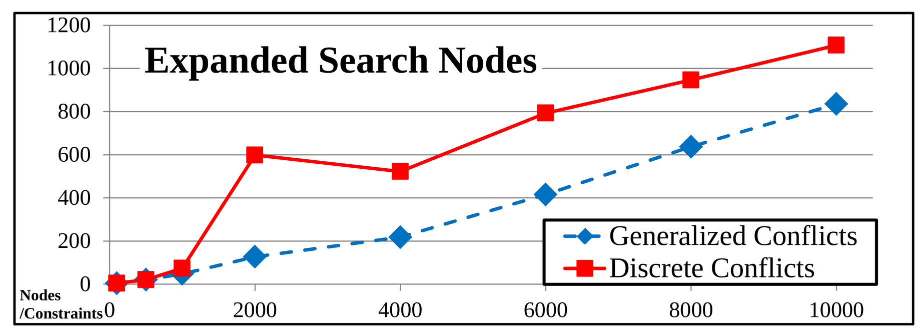


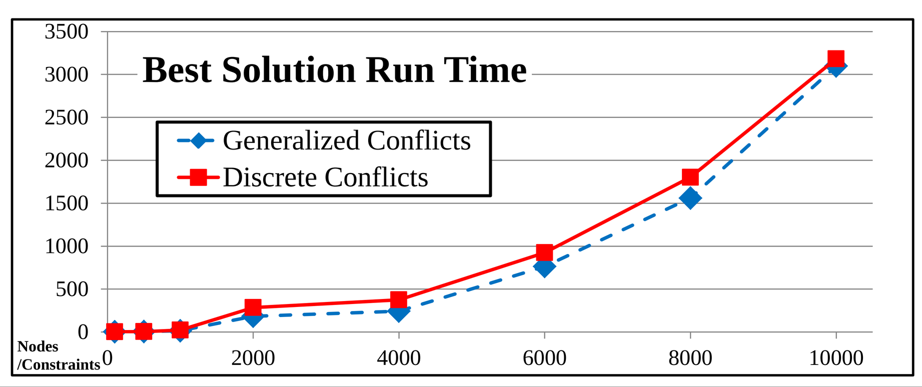
Applications & Experiments

1. Collaboratively diagnosing over-constrained travel plans for Personal Transportation System.
2. Mission advisory system for autonomous underwater vehicles.
3. Trip advisor for car-sharing network users.

- We simulated a car-sharing network in Boston using randomly generated car locations and destinations, by varying a. Number of reservations per car. b. Number of cars in the network. c. Number of activities per reservation. d. Number of alternative options per activity.
- Preferences over alternative activities and reservation time.

- We compared BCDR to a disjunctive linear problem algorithm[4] which only uses discrete conflict resolutions.
- The result shows that BCDR is more efficient in pruning the search space: the number of node expansions is significantly reduced.
- This technique can be implemented with Branch & Bound and other search strategies.





This research is funded by the Boeing Company grant MITBA-GTA-1

[1] J. Bailey and P. J. Stuckey. "Discovery of Minimal Unsatisfiable Subsets of Constraints Using Hitting Set Dualization." In *Proc. of the 7th International Symposium on Practical Aspects of Declarative Languages* (PADL05), pages 174-186, 2005.
[2] Michael D. Moffitt and Martha E. Pollack. Partial constraint satisfaction of disjunctive temporal problems. In *Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference* (FLAIRS-2005), 2005.
[3] B. Williams and R. Ragno. Conflict-directed A* and its role in model-based embedded systems. In *Discrete Applied Mathematics*, 155(12):1562–1595, 2007.
[4] Li, H., Williams, B. (2005) "Generalized Conflict Learning for Hybrid Discrete Linear Optimization," In *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming*, pp.415-429.