Resolving Over-subscribed Temporal Planning Problems through Fluent Human-Robot Collaboration

Peng Yu

Nuance Sunnyvale Laboratory

June 11th, 2015

About Myself

- Peng Yu
- 5th year graduate student (3nd year PhD) at the Computer Science and Artificial Intelligence Lab of MIT.

- Master Thesis (February 2013):

Continuous Relaxation to Over-constrained Temporal Problems.

- PhD Thesis (June 2016, expected):

Resolving Over-subscribed Temporal Planning Problems through Fluent Human-Robot ***Collaboration***

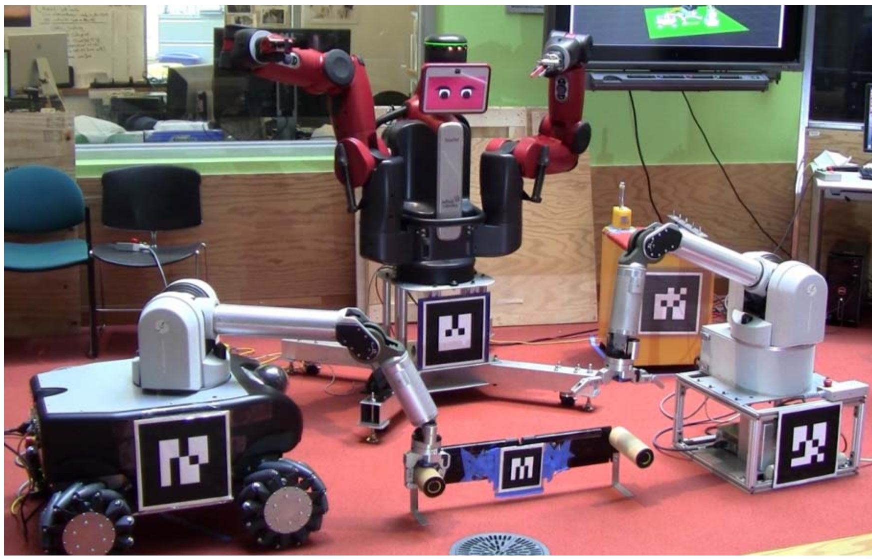
About My Research Group

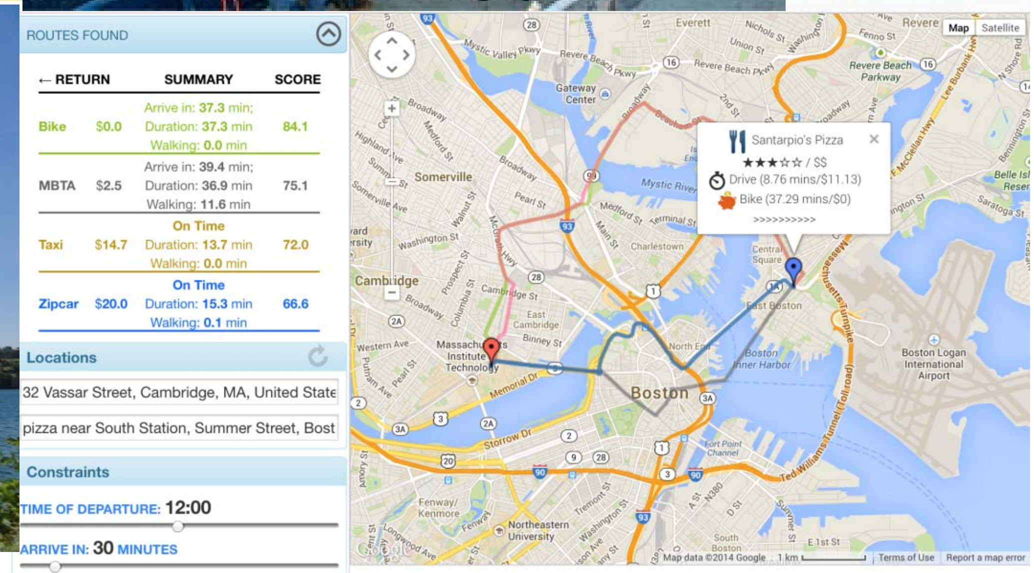
- Model-based Embedded and Robotic Systems:
  - PI: Brian Williams.

- Research focus: diagnosis, planning and scheduling.
  - Activity planning, execution and monitoring.
  - Risk-sensitive task scheduling and execution.
  - The diagnosis/state estimation of hybrid systems.

Applications

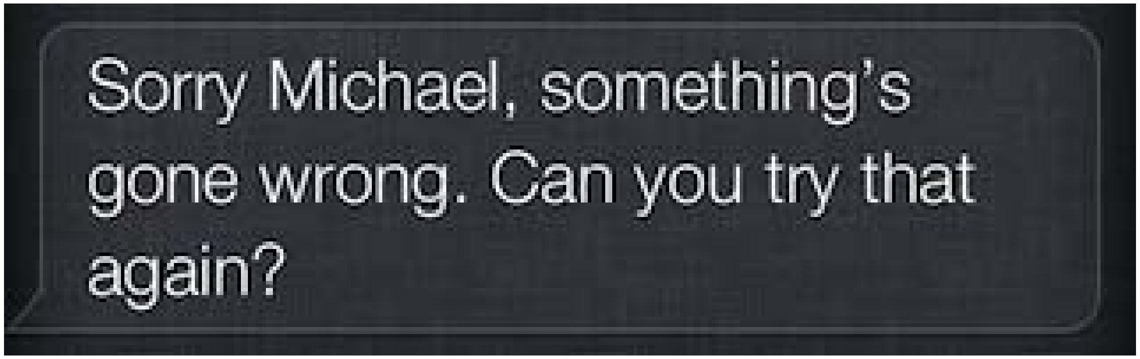
- Aerial, ground, underwater autonomous systems.





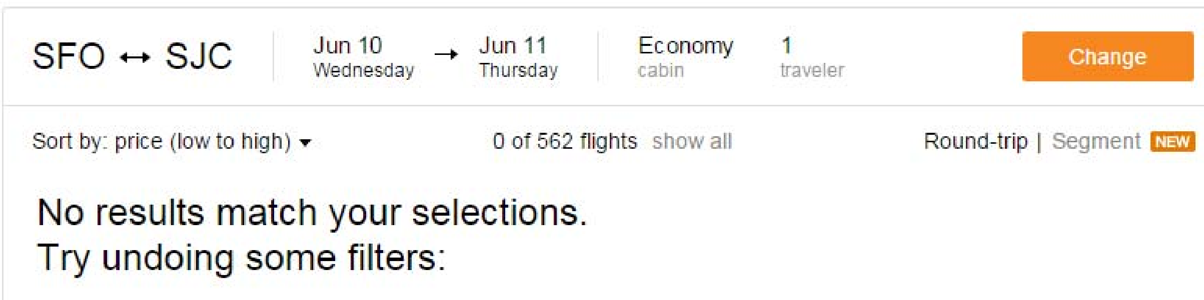
Challenges

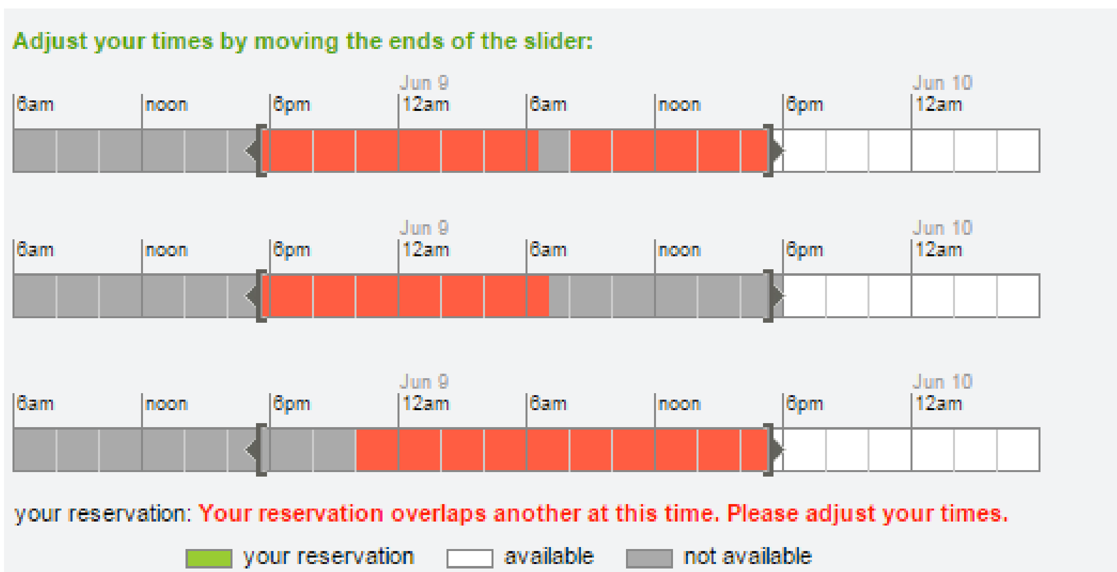
- These capable systems often become brittle when they fail to achieve what they were asked to do:
  - They are not good at communicating their failure.



Challenges

– They cannot provide much insight into the cause of the failure, and/or recommend useful recovery options.





Objectives

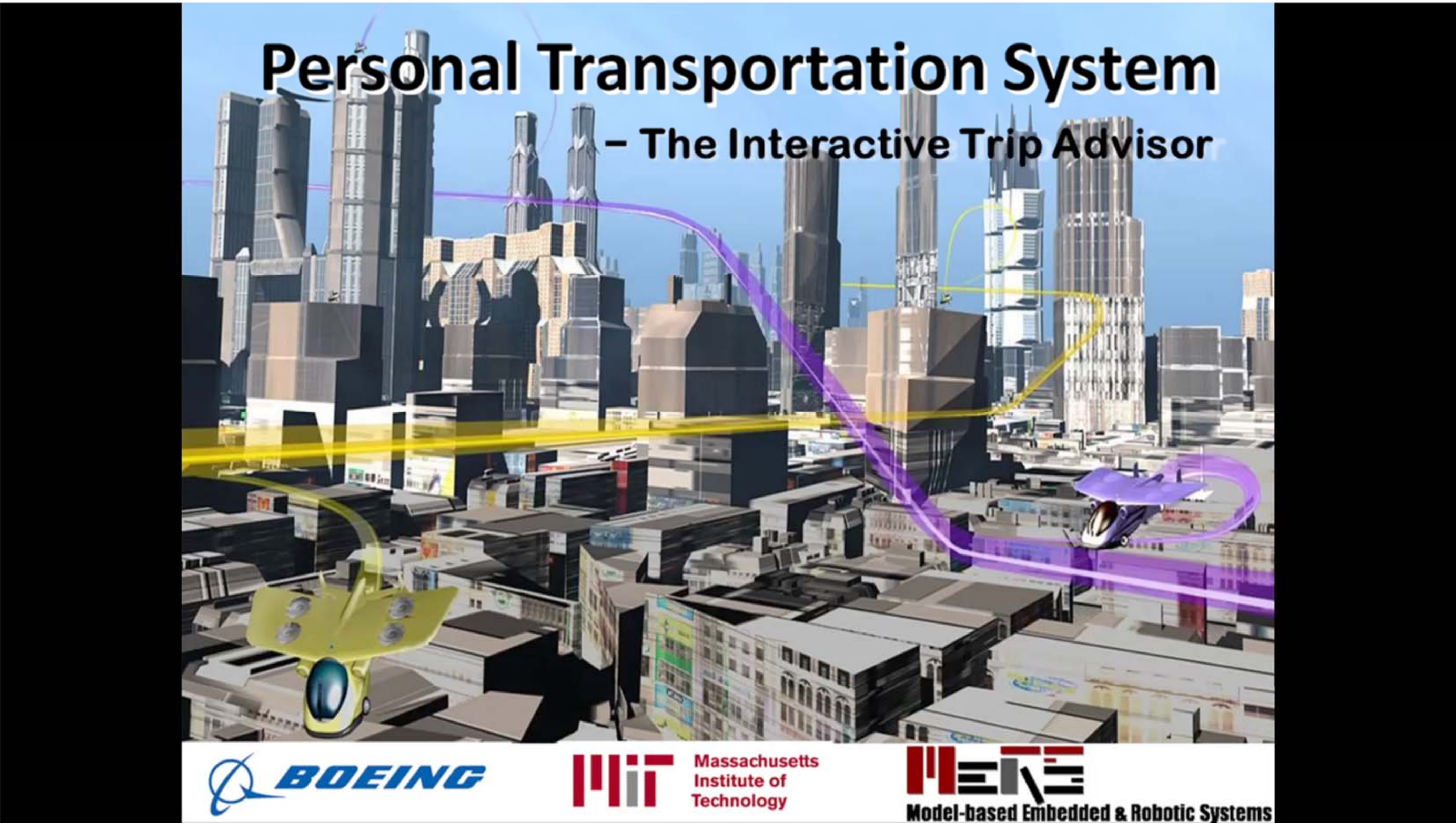
- Make autonomous systems more robust towards over-subscribed situations:
  - by detecting and **explaining** the cause of failure;
  - and recommending **preferred** alternative solutions.

- Collaborates with humans throughout the process:
  - To understand their goals and constraints.
  - To ask for preferences over trade-offs between alternatives.
  - To obtain assistance when running out of options.

Outline

- Problem:
  - Resolving over-subscribed scheduling & planning problems.
- Approach:
  - Uhura: a collaborative plan diagnosis assistant.
- Going real-world:
  - Being robust to uncertainty.
  - Automatically learning and prioritizing alternatives.
- Next step:
  - Effective collaboration with humans through dialog.

A Demo on the Personal Transportation System



Key features

- Find alternative solutions that are **simple** and **preferred**.
- Provide **insights** into cause of failure and its resolution.
  - Minimize the perturbations;

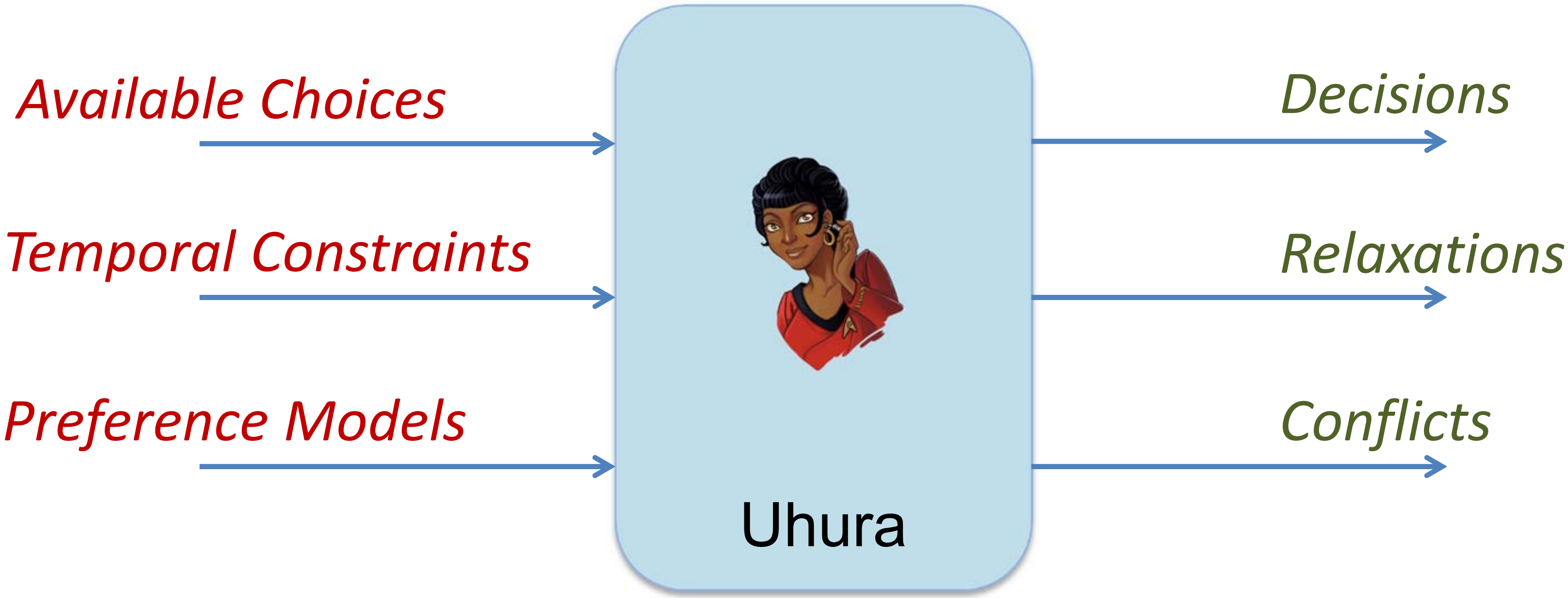
    "Delay your arrival by 5 minutes".
  - Prioritize alternatives;

    "OK, then how about having lunch at restaurant Y".
  - Explain the cause of failure;

    "Because of the extended travel time".
  - Adapt incrementally to new inputs.

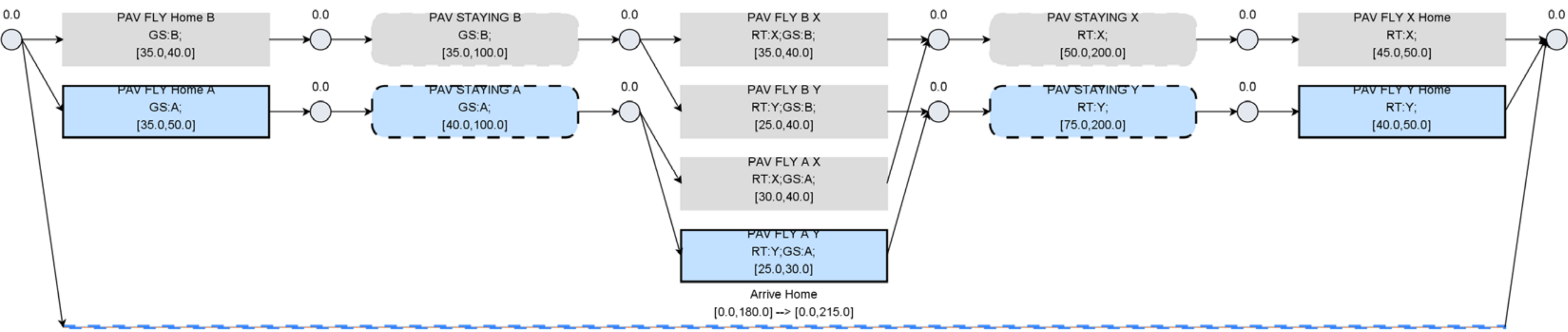
    "if you want to shop for at least 25 minutes, you can have lunch at restaurant Y for 55 minutes".

The Inputs and Outputs



Inputs

- A (over-subscribed) scheduling problem.
  - Encoded using *Temporal Constraints* and *Discrete Choices*.



  - Over-subscription: no choices and schedules can be found that meet all constraints in the problem.

Outputs

- A triple of decisions, conflicts and relaxations that enables a feasible schedule to be generated:
  - **Decisions**: Store=A, Restaurant=Y

    "*You can go to Store A and Restaurant Y.*"
  - **Relaxations**: ArriveHome[0,180]→[0,215]

    "But I have to delay your arrival by 35 minutes."
  - **Conflicts**: $\{Shop@A_{LB}, FlyAtoY_{LB}, Lunch@Y_{LB}, ArriveHome_{UB}, \ldots\}$

    "Because of the long travel time between them and your requirements on the minimum lunch time."

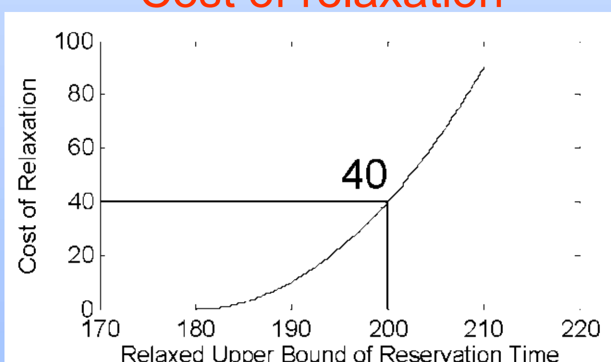
User Preferences and Feedbacks

Preference

Rewards of decisions

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

Cost of relaxation




Uhura

Feedback

"Lunch should be at least 25 minutes"

$\text{Lunch}_{LB} > 25$

"Return home on time."

$\Delta\text{ArriveHome}_{UB} = 0$

Outline

- Problem:
  - Resolving over-subscribed scheduling & planning problems.

- Approach:
  - Uhura: a collaborative plan diagnosis assistant.

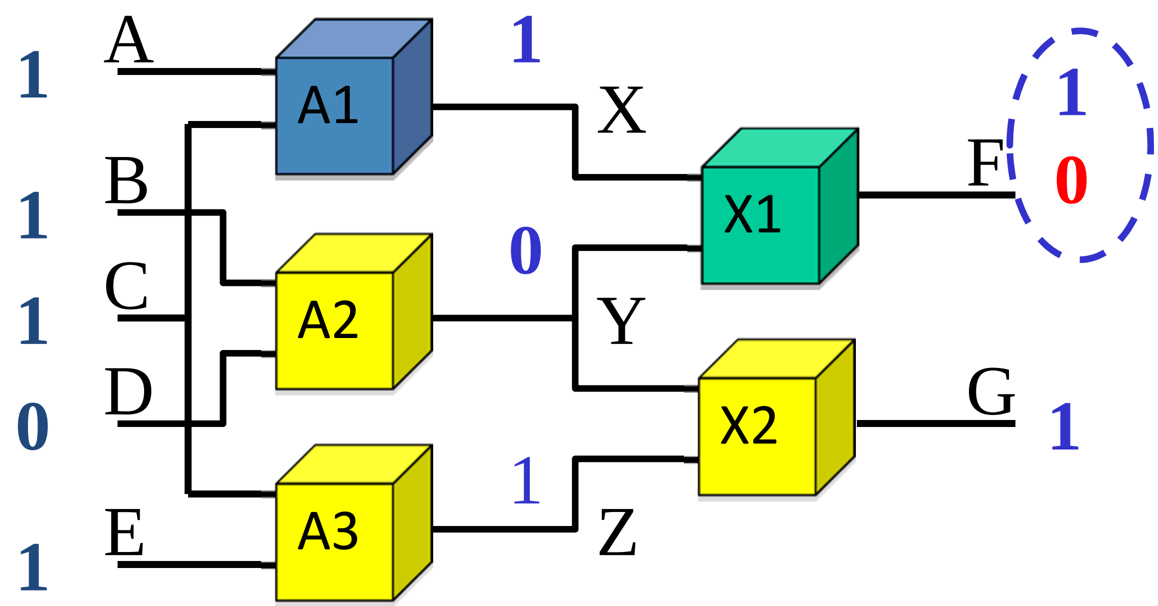
- Going real-world:
  - Being robust to uncertainty.
  - Automatically learning and prioritizing alternatives.

- Next step:
  - Effective collaboration with humans through dialog.

A Diagnosis Approach

- Given an over-subscribed scheduling problem, we need to:
  - **Detect** the conflicting sets of constraints.
  - **Relax** some constraints to resolve the conflicts.
  - **Enumerate** alternative solutions in best-first order.

- It is like solving a diagnosis problem, but on a broken schedule/plan instead of hardware.



Solving Diagnosis Problems

- General Diagnosis Engine (*de Kleer and Williams, 1987*) was developed to detect cause of failures and compute diagnoses.
  - It learns from infeasible mode assignments and use them to prune search domains.

- Conflict-Directed A* (Williams and Ragno, 2001) builds upon the ideas in GDE, and adds:
  - efficient pruning of search space by generalizing the learned conflicts.
  - enumeration of minimal diagnosis in best-first order.

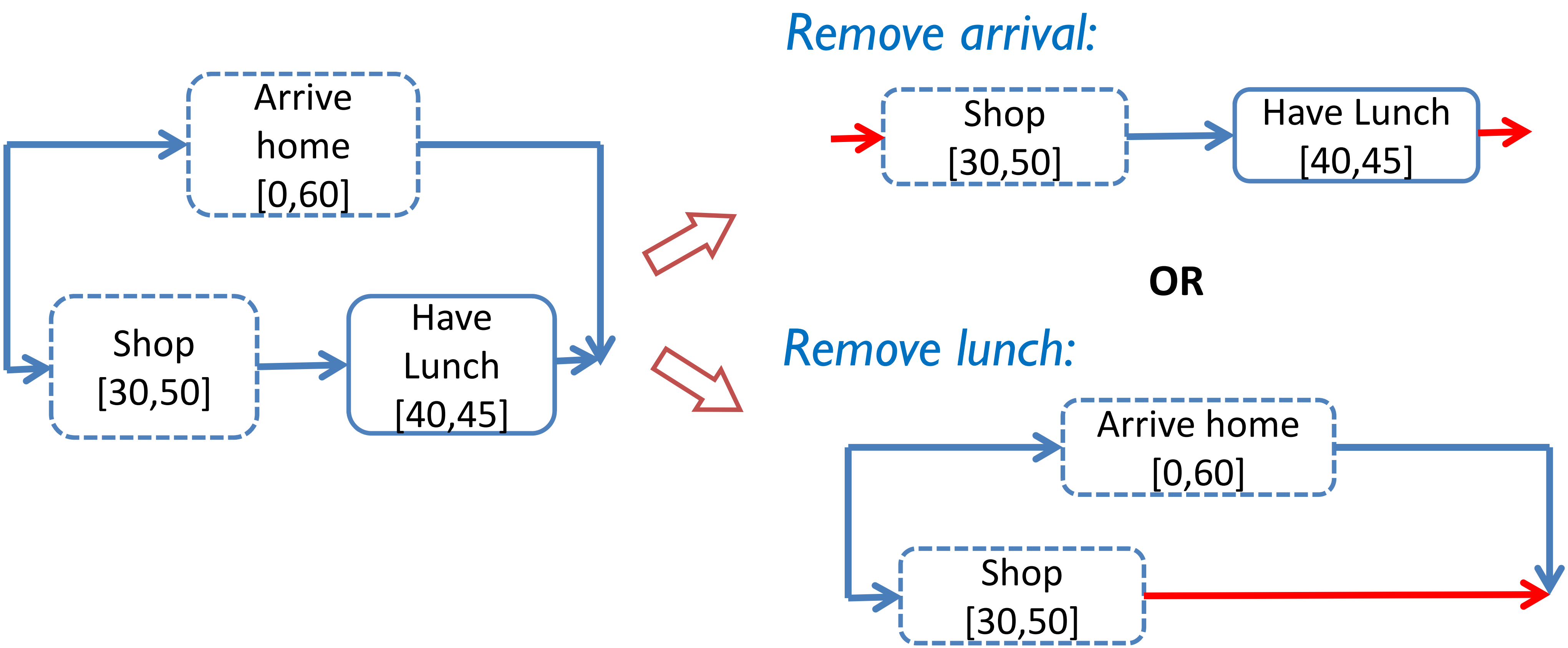
From Hardware Diagnosis to Temporal Relaxation

- Similar ideas have been used to solve inconsistent temporal constraint problems through **constraint suspension**.
  - Enumerate minimal relaxations (Previti and Marques-Silva, 2013).
  - Support user preferences (Peintner, Moffitt and Pollack, 2005).
  - Efficiently prune domain using learned conflicts (Bailey and Stucky, 2005).

- However, completely suspending a constraint is unnecessary in most situations.
  - Slightly weakening a constraint instead of a complete removal is often enough.

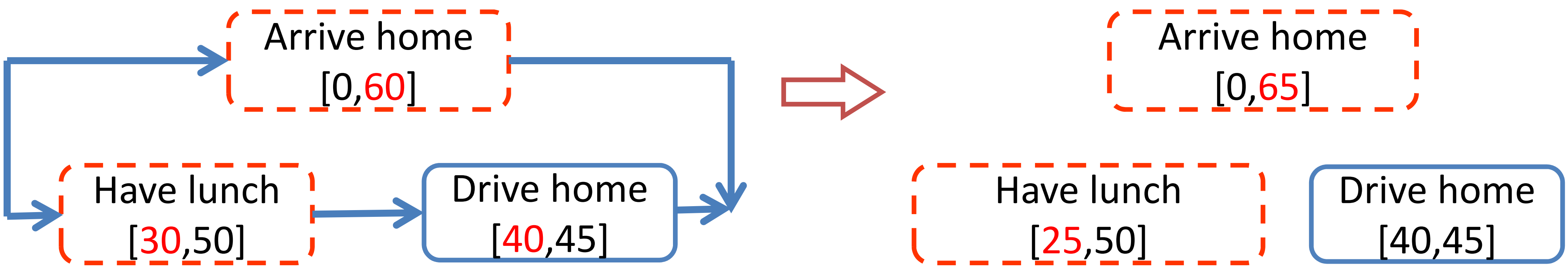
Discrete Relaxation

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



Continuous Relaxation

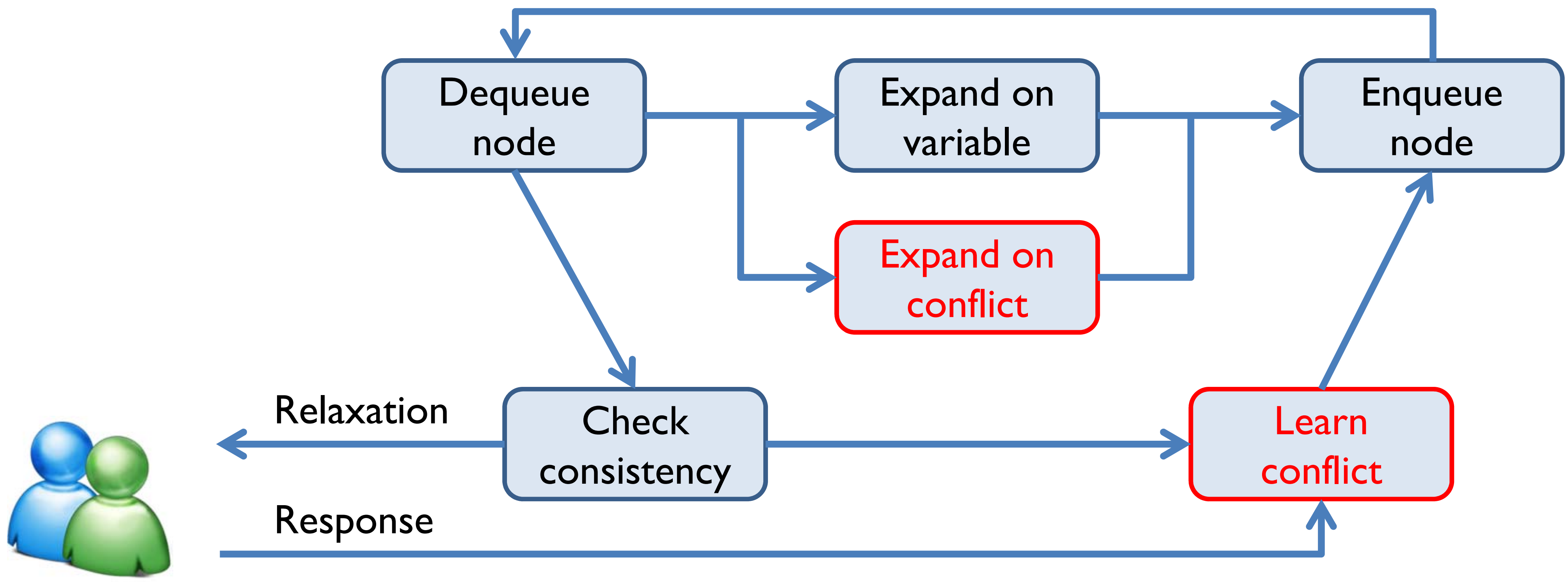
- Relax a constraint partially by **continuously** modifying its temporal bounds:
  - A continuous relaxations, CR_i, weakens a temporal constraint: $[LB, UB] \rightarrow [LB', UB']$ where $LB' \leq LB \; and \; UB' \geq UB$.
  - Continuous relaxations only apply to **relaxable** constraints.



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

Best-first Conflict-Directed Relaxation

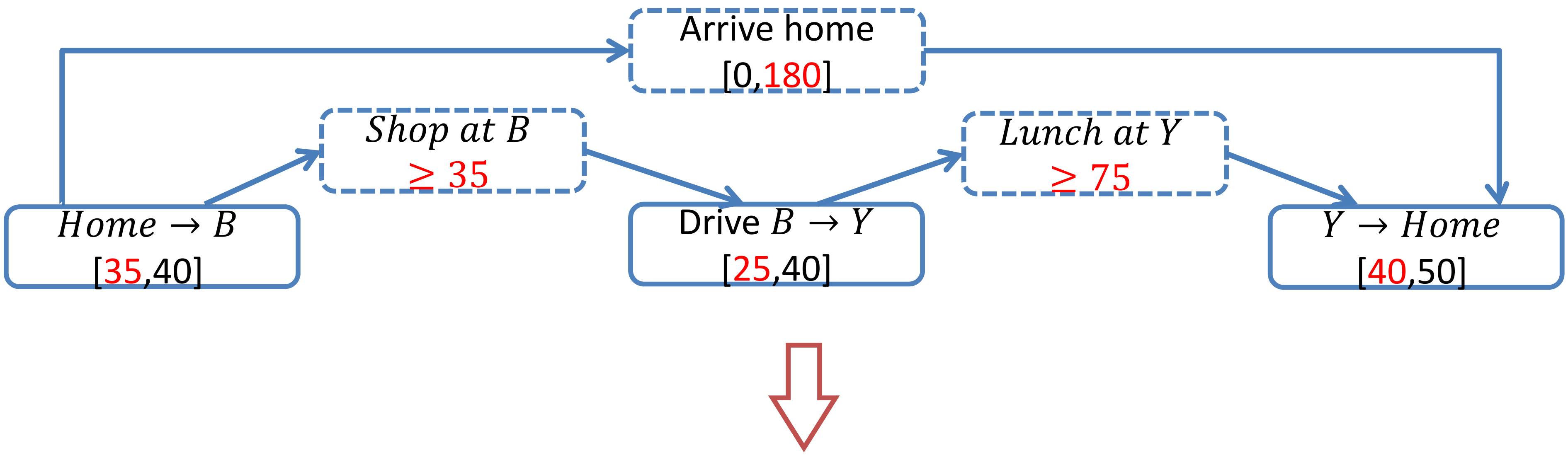
- We generalize the conflict resolution procedure in Conflict-Directed A* *(Williams and Ragno, 2004)* to enumerate continuous relaxations in best-first order.



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:

$$\begin{aligned} &\text{HometoB} \geq 35; \\ &\text{ShopatB} \geq 35; \\ &\text{BtoY} \geq 25; \\ &\text{LunchatY} \geq 75; \\ &\text{YtoHome} \geq 40; \\ &\text{ArriveHome} \leq 180. \end{aligned}$$

Continuous Conflict:

$$\begin{aligned} &ArriveHome - HometoB - ShopatB \\ &\quad -BtoY - LunchatY - YtoHome = \mathbf{-30} \end{aligned}$$

3. Map to Constituent Continuous Relaxations

- Relaxations specified by linear inequalities:

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

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

Discrete vs. Continuous Relaxations

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

Conflict:

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

Discrete Resolutions

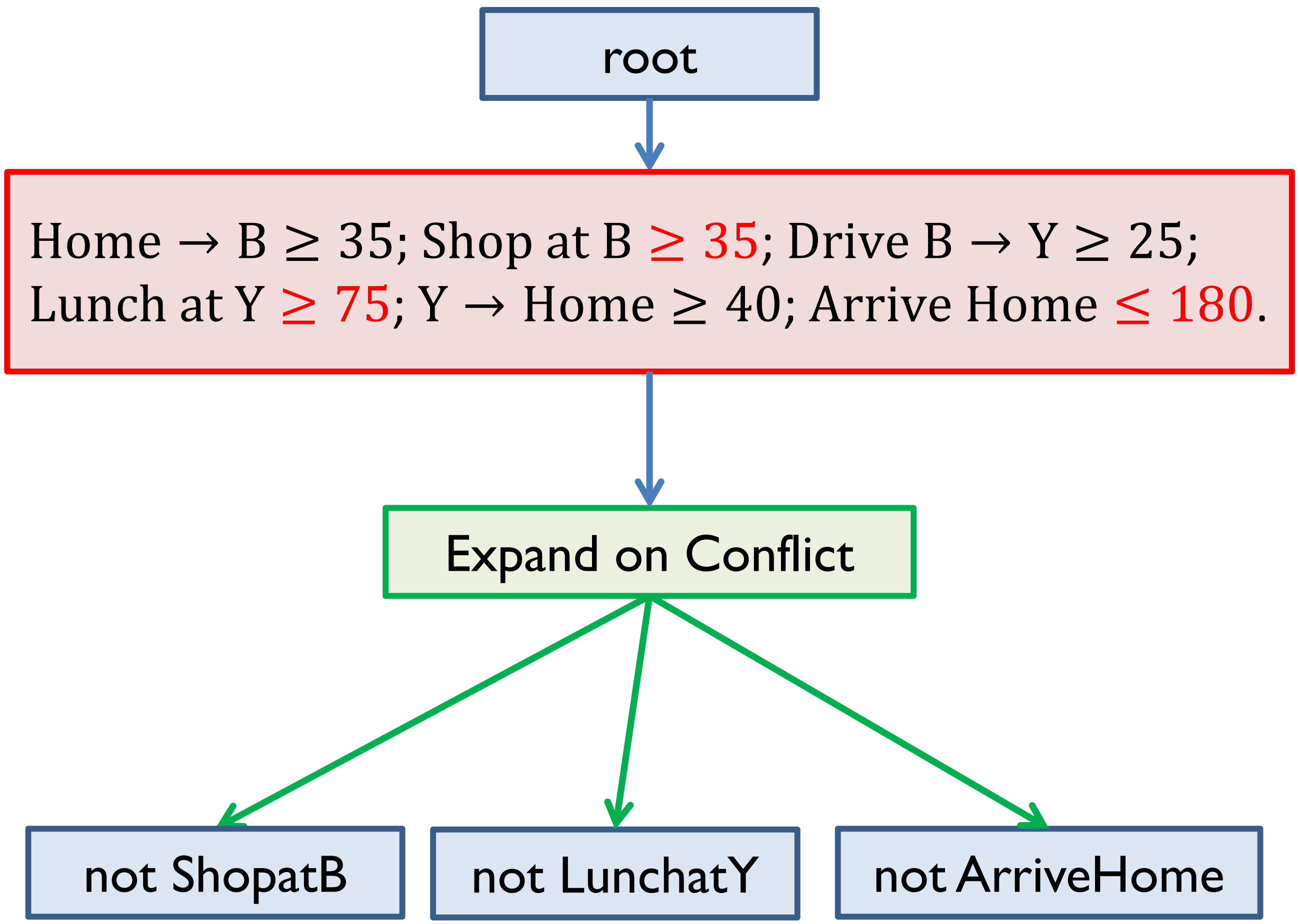
Remove Shop at B ≥ 35;
Remove Lunch at Y ≥ 75;
Remove Arrive Home ≤ 180

Continuous Resolutions

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

Conflict Resolution using Discrete Relaxation

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



Conflict Resolution using Continuous Relaxation

- Expand 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.

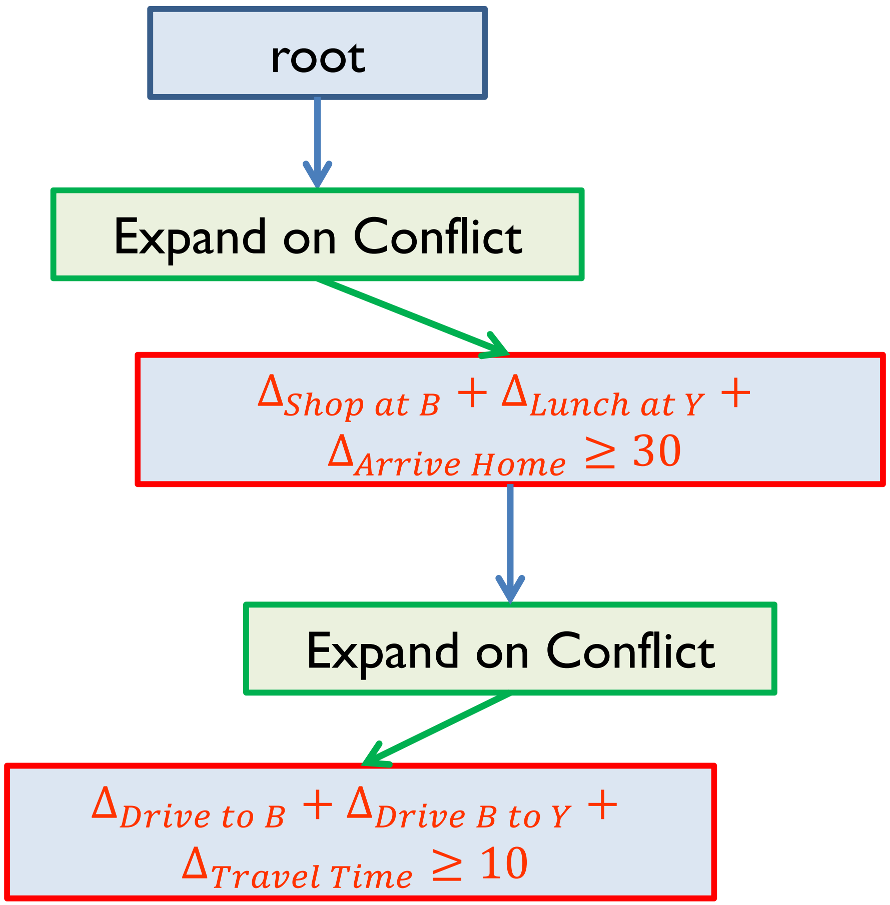
Expand on Conflict

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

$$\min(f(\Delta_{Shop\ at\ B}) + f(\Delta_{Lunch\ at\ Y}) + f(\Delta_{Arrive\ Home}))$$
$$\text{s.t. } \Delta_{Shop\ at\ B} + \Delta_{Lunch\ at\ Y} + \Delta_{Arrive\ Home} \geq 30$$

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.

No, I **do not** want to delay my arrival time.

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

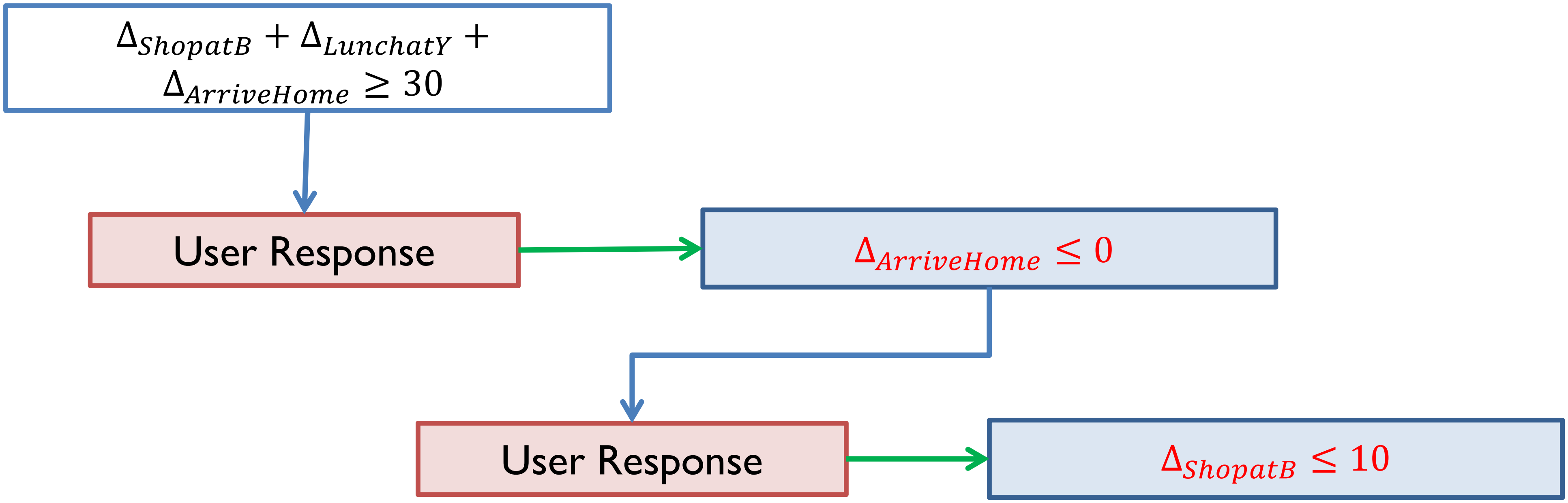
Required Continuous Relaxations

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

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

New Requirements as Conflicts

- Expand search tree using user response conflicts.

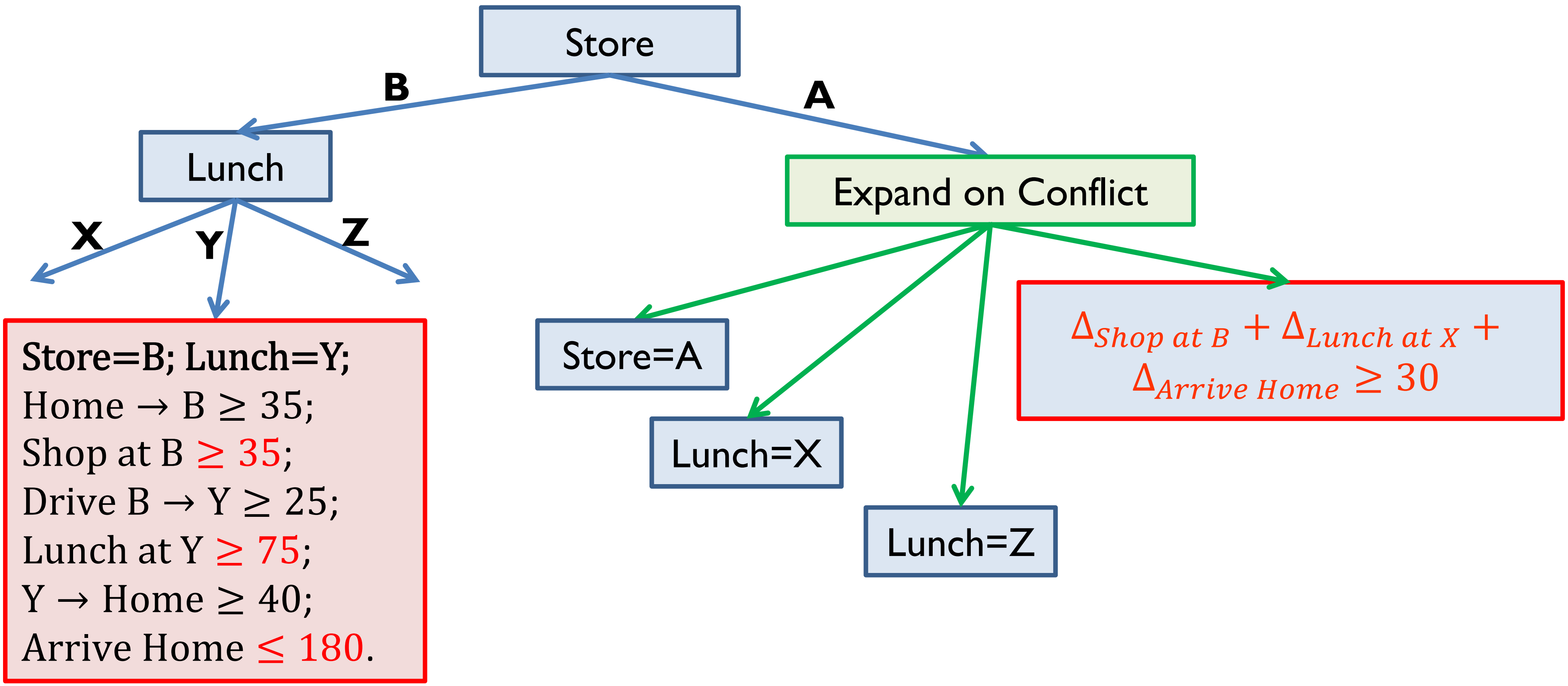


$$\min(f(\Delta_{ShopatB}) + f(\Delta_{LunchatY}) + f(\Delta_{ArriveHome}))$$

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

Split on Conflicts for Conditional Problems

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



Outline

- Problem:
  - Resolving **over-subscribed** scheduling & planning problems.
- Approach:
  - Uhura: a collaborative plan diagnosis assistant.
- Going real-world:
  - Being robust to uncertainty.
  - Automatically learning and prioritizing alternatives.
- Next step:
  - Effective collaboration with humans through dialog.

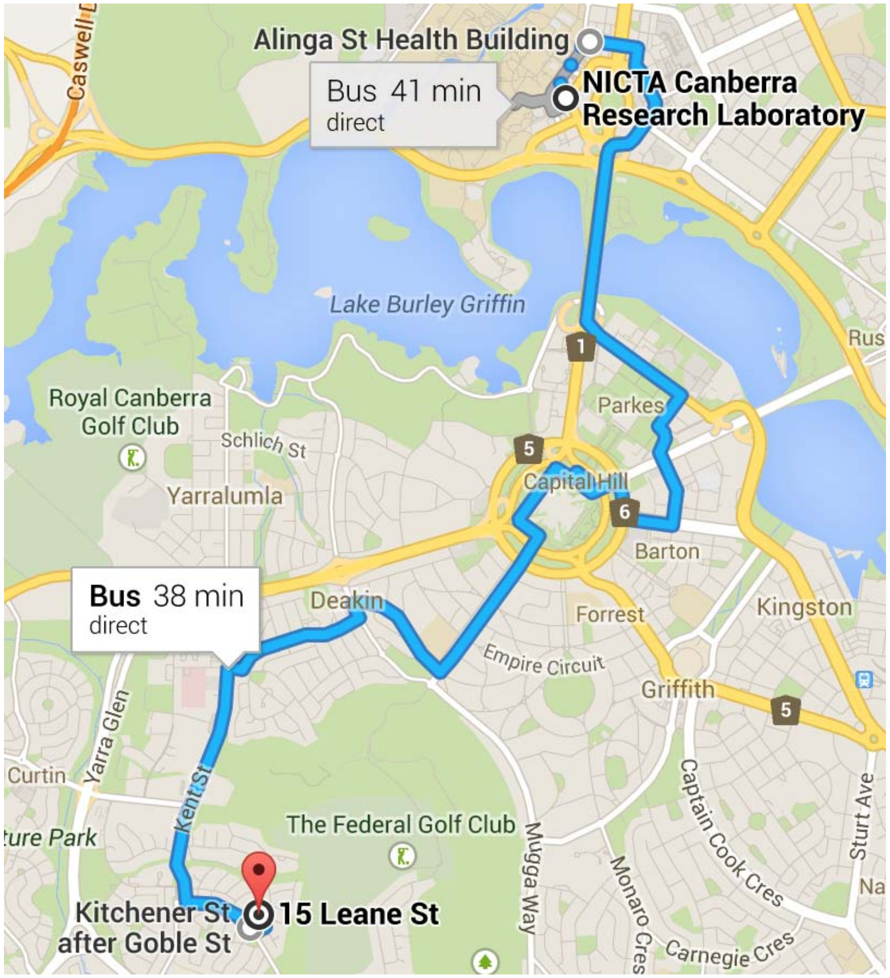
Uncertainty is Everywhere

- Planning a trip home.

It is **6pm** now and Brian is leaving NICTA Lab 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 lab 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.

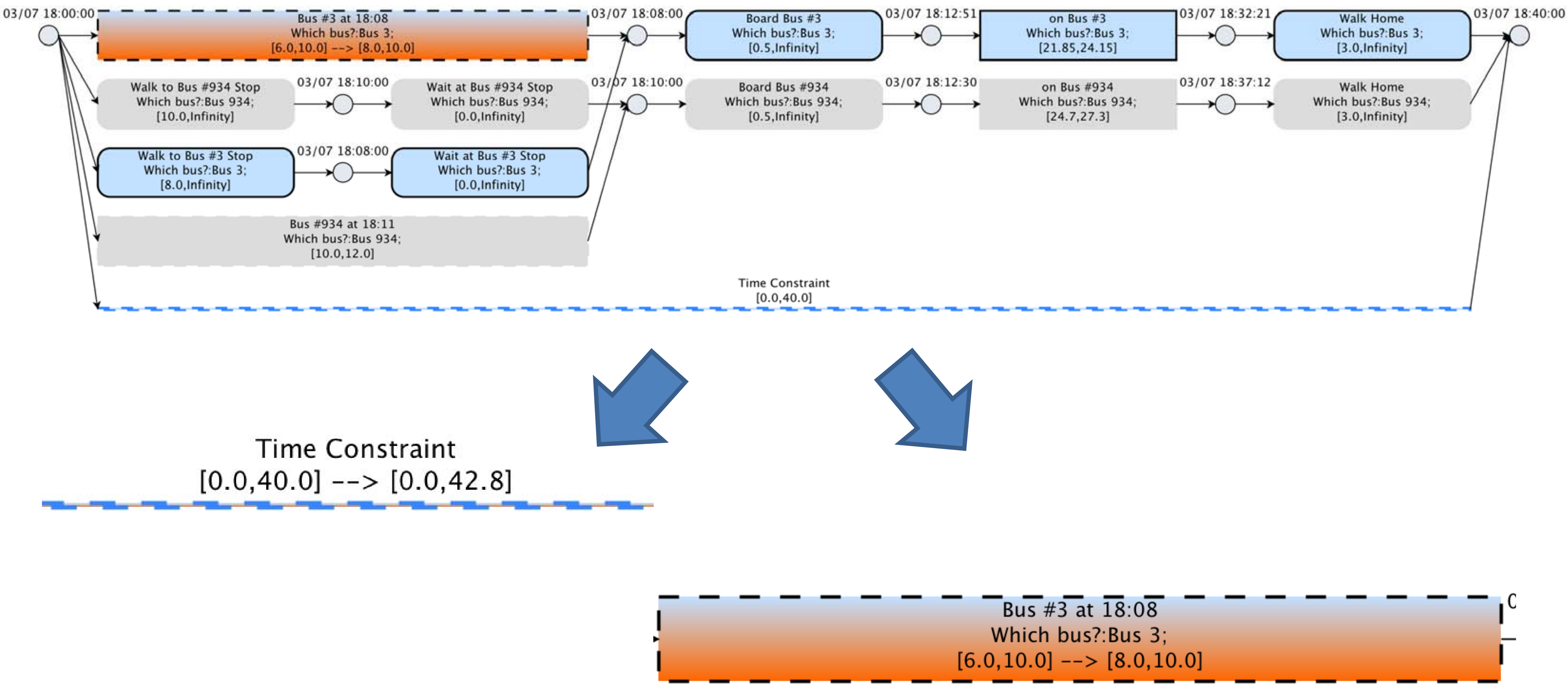
Uncertainty Affects Our Decision

- 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 recommended 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."*

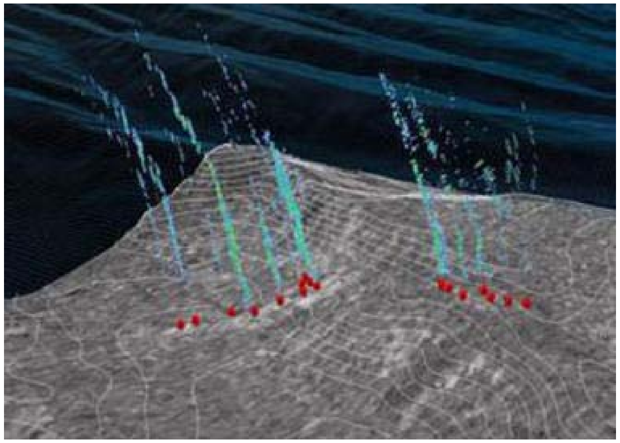


Mission Advisor for Deep-sea Explorations

- During an expedition, the oceanographers need assistance for sequencing and scheduling activities, especially when things go wrong.

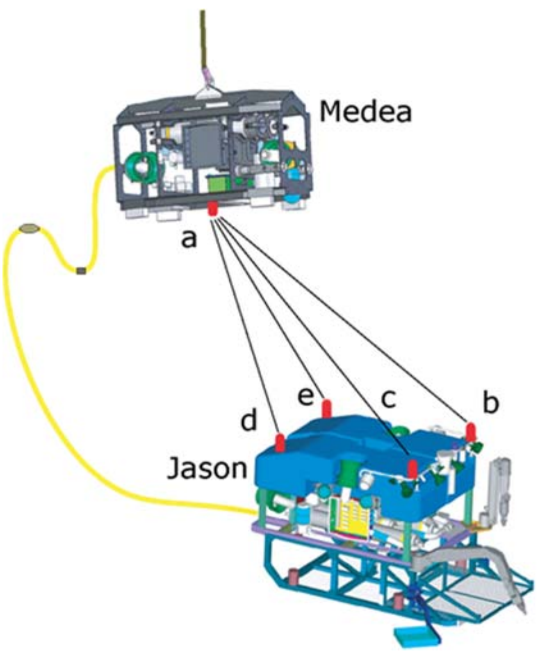
An Expedition Mission with WHOI

- Duration: Sep 26th – Oct 17th, 2013.
- 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.



Everything can Go Wrong

- [Day 1] The remote vehicle 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 the autonomous vehicle did not work well in low light situations. A replacement task had to be scheduled.

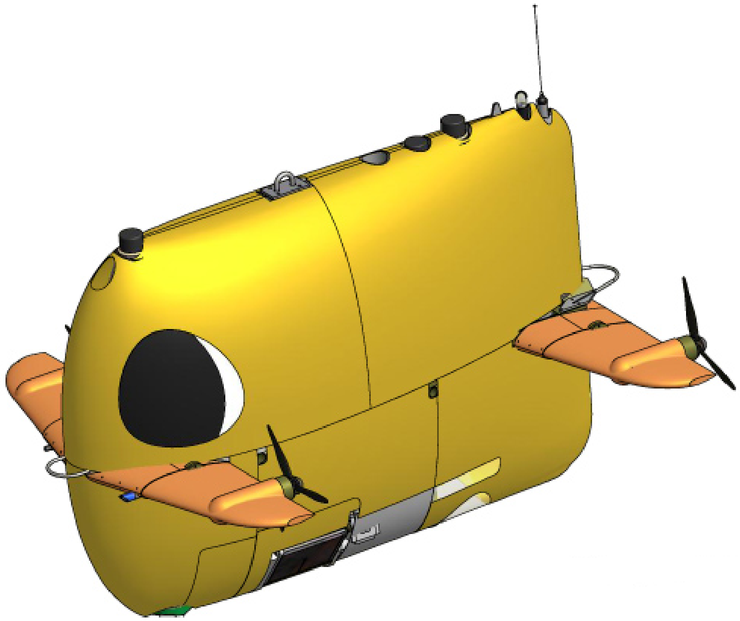


Everything can Go Wrong

- [Day 2] The remote vehicle broke its optic fiber tether again during its second dive. It turned out that there is a bug in its newly updated code.

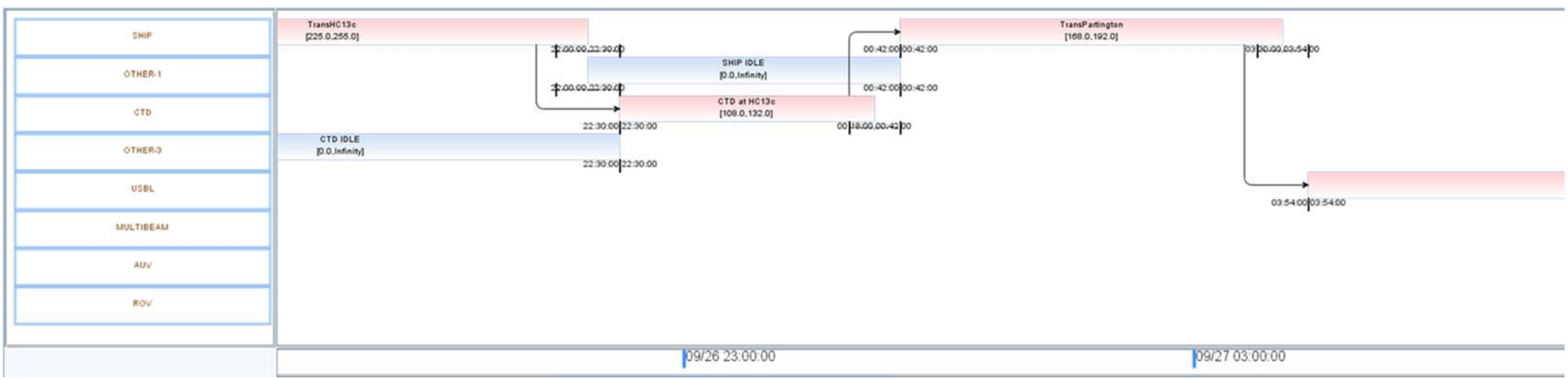
- [Day 3] The autonomous vehicle's mass spectrometer failed during its second dive. An engineer was sent to Pittsburg to get it fixed.

… …



Our Deliverable

- A mission advisory system ('Enterprise under the sea') that helps the oceanographers to better :
  - plan activities with large temporal and vehicle uncertainty;
  - fast recovery from unexpected failure and downtime;
  - effective management of resources.



A Short Intro of the Mission Advisor

Outline

- Problem:
  - Resolving **over-subscribed** scheduling & planning problems.
- Approach:
  - Uhura: a collaborative plan diagnosis assistant.
- Going real-world:
  - Being robust to uncertainty.
  - Automatically learning and prioritizing alternatives.
- Next step:
  - Effective collaboration with humans through dialog.

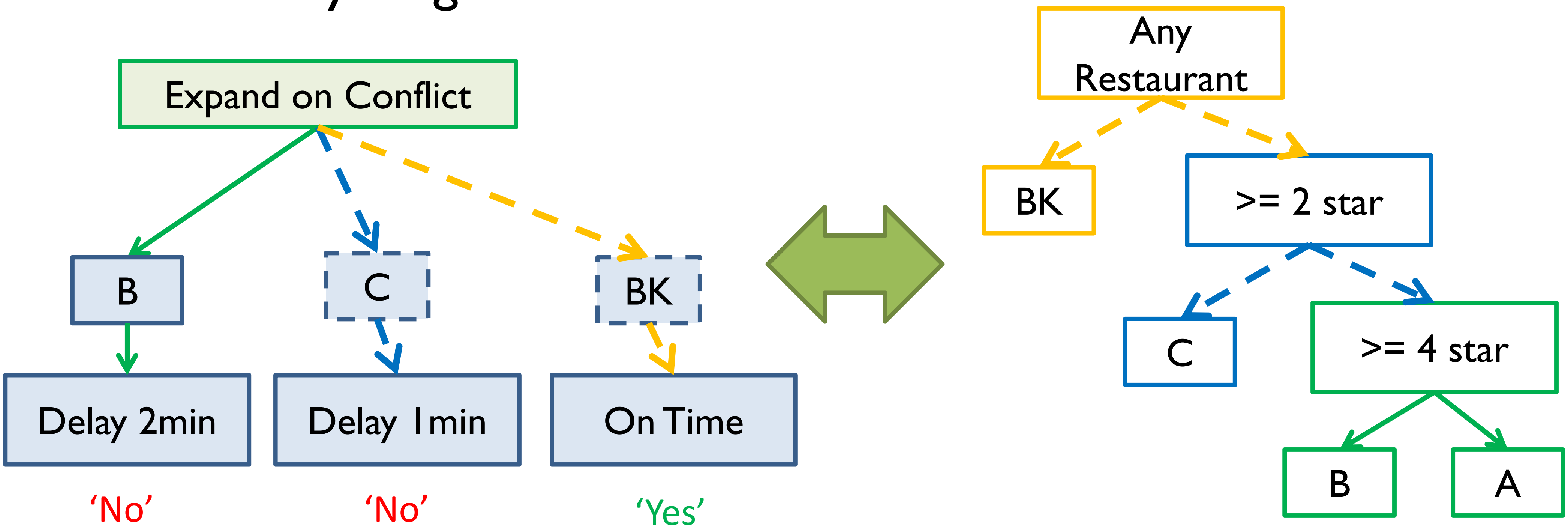
Beyond Temporal Relaxation

- Instead of relaxing temporal requirements, people also relax their goals in a different dimension:
  - "Chinese restaurant → Any Asian restaurant → Any restaurant".
  - "Whole Foods → Any organic grocery store".
  - "Star Wars → Any Sci-Fi movie → Any movie".
- We named it *Semantic Relaxation*:
  - Finding semantically similar goals to resolve conflicts in planning problems.

PTS Demo with Semantic Relaxation

Computing Semantic Relaxations

- We organize the restaurants in a taxonomy.
- Given a conflict, if no resolution can be found, query the taxonomy to get additional candidates.

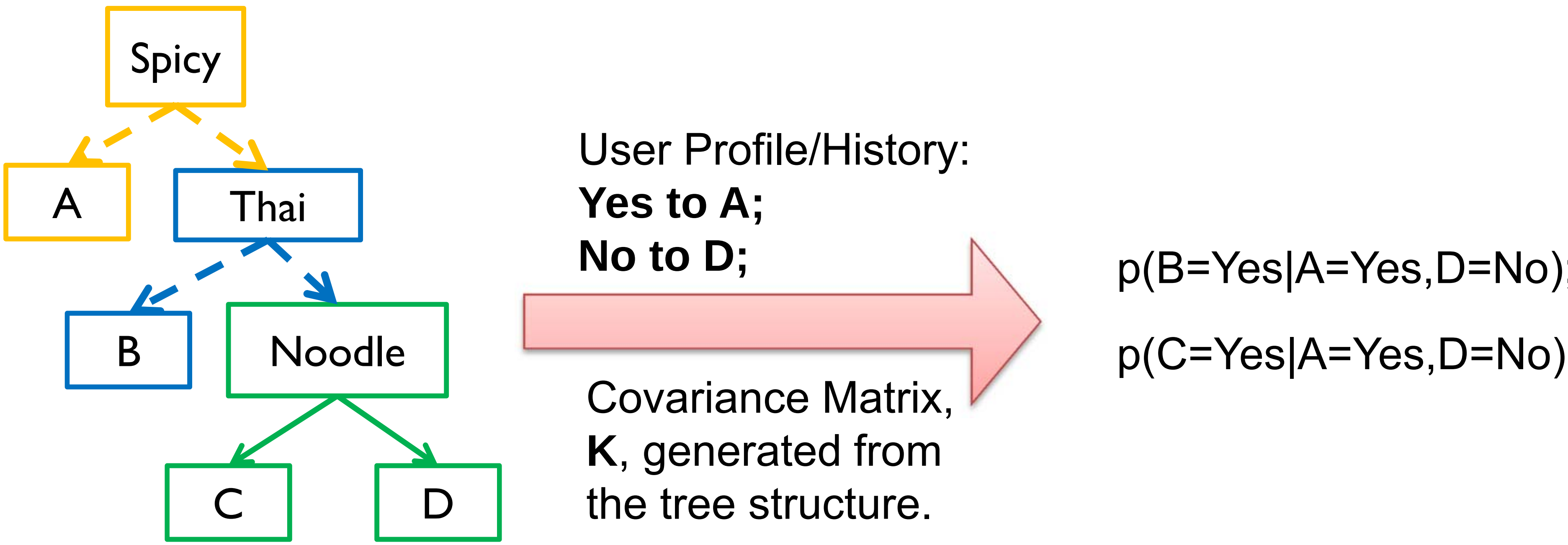


- But where can we get the taxonomy?
  - This is a much harder problem.

Generating the Taxonomy (Jonathan Raiman)

- Input: restaurant data in yelp.com/seattle (ratings, categories, descriptions and reviews).

- Output: a taxonomy* that organizes the alternatives.
  - It is also used with the user profile to prioritize alternatives.



*_Structured Statistical Models of Inductive Reasoning_, Kemp and Tenenbaum, 2009

Another Demo of Semantic Relaxation

Semantic Recommender

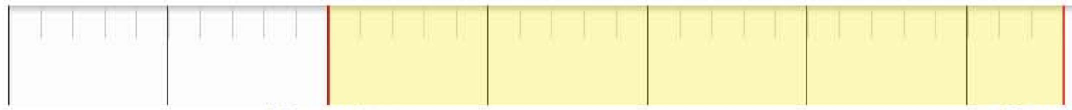
Voice recognition on

How can I help ?

Debug

just now

I suggest Espi's Sausage & Tocino Co. It has a higher rating. This trip will take 23 minutes longer. OK?



Espi's Sausage & Tocino Co 9.8 km

★★★★★ $$$$

Best categorized as filipino and meats

no better

1 minute ago

Let's go to Gourmet Noodle Bowl. Shall we proceed?

Gourmet Noodle Bowl 2.5 km

★★★★★ $$$$

Outline

- Problem:
  - Resolving over-subscribed scheduling & planning problems.
- Approach:
  - Uhura: a collaborative plan diagnosis assistant.
- Going real-world:
  - Being robust to uncertainty.
  - Automatically learning and prioritizing alternatives.
- Next step:
  - Effective collaboration with humans through dialog.

Uhura Cannot Work On Its Own

Available Choices

Temporal Constraints

Preference Models

Where does the model come from?

Uhura

Decisions

Relaxations

Conflicts

How to present the outputs?

Dialog Manager and Knowledge Base Integration

- From **dialog**: goals, preferences, requirements.
  - "I want to watch The Hobbit after dinner".
  - "I prefer the Italian restaurant over Burger King".
- From **knowledge base**: grounded goal and task models.
  - The duration of The Hobbit is 150±10 minutes.
  - There are three Italian restaurants and two Burger Kings in Sunnyvale.

Constraints

(Alternative) Goals

Preferences

More Effective Communication With the Users

- Enable the human to come back to the loop.

2012

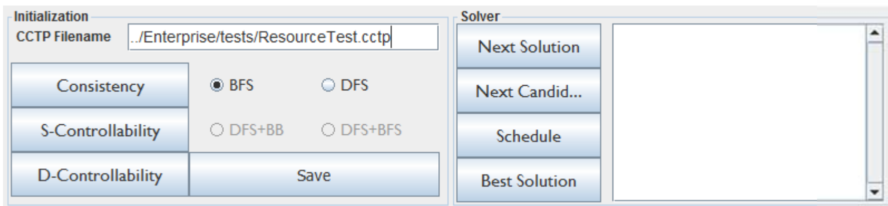
Uhura

Decisions

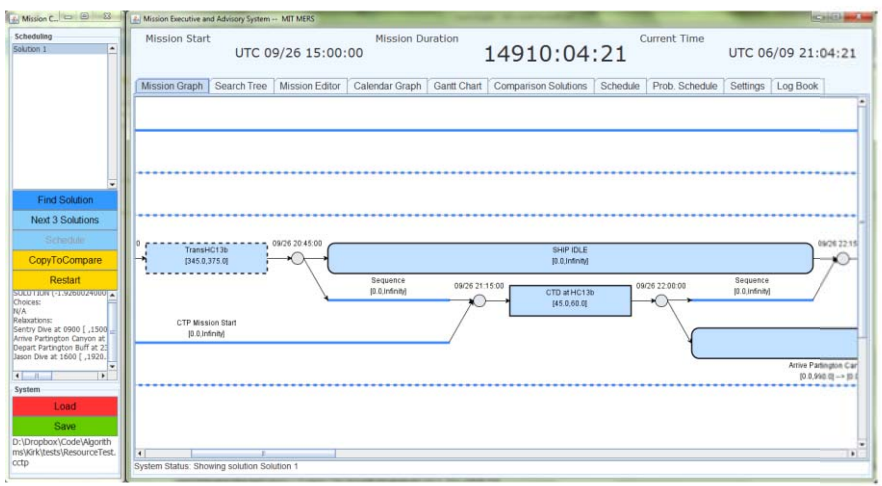
Relaxations

Conflicts

Feedback



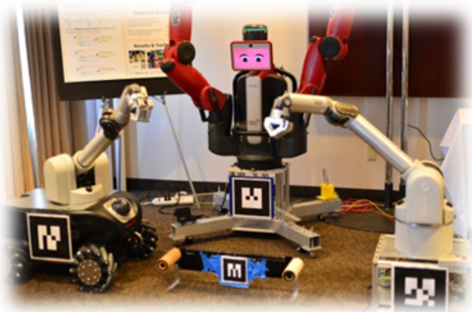
2013



2014

Summary and Remaining Work

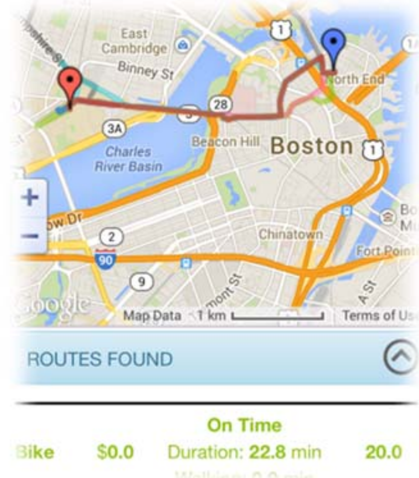
Collaborative Manufacturing[6]



Personal Air Vehicle[7]

Deep-sea Mission Advisor[4]

Personal Travel Assistant



Application

Research

Over-constrained *Conditional Temporal Problems*[1]

Chance-constrained ***Probabilistic*** *Temporal Problems*[3][5]

Temporal Problems with ***Uncertainty***[2]

Integration of ***semantic*** *and temporal relaxation*

- *Effective communication using a dialog interface.*
- *Model building through dialog and knowledge base integration.*

In collaboration with Jonathan Raiman, Cheng Fang, Szymon Sidor, Jing Cui, Steve Levine and Erez Karpas.

References

- [1] Peng Yu and Brian Williams, Continuously Relaxing Over-constrained Conditional Temporal Problems through Generalized Conflict Learning and Resolution, *Proceedings of the Twenty- third International Joint Conference on Artificial Intelligence (IJCAI-13)*, Beijing, 2013.

- [2] Peng Yu and Cheng Fang and Brian Williams, Resolving Uncontrollable Conditional Temporal Problems using Continuous Relaxations, *Proceedings of the Twenty-fourth International Conference on Automated Planning and Scheduling (ICAPS-14)*, Portsmouth, 2014 (Honorable Mention for Best Paper Award).

- [3] Cheng Fang and Peng Yu and Brian Williams, Chance-constrained Probabilistic Simple Temporal Problems, *In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14)*, Quebec City, 2014.

- [4] Peng Yu and Cheng Fang and Brian Williams, Resolving Over-constrained Probabilistic Temporal Problems through Chance Constraint Relaxation, *In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, Austin, 2015.

References

- [5] Jing Cui and Peng Yu and Cheng Fang and Patrik Haslum and Brian Williams, Optimising Bounds in Simple Temporal Networks with Uncertainty under Dynamic Controllability Constraints, *Proceedings of the Twenty-fifth International Conference on Automated Planning and Scheduling (ICAPS-2015),* Jerusalem, 2015.

- [6] Erez Karpas and Steve Levine and Peng Yu and Brian Williams, Robust Execution of Plans for Human-Robot Teams, *Proceedings of the Twenty-fifth International Conference on Automated Planning and Scheduling (ICAPS-2015),* Jerusalem, 2015.

- [7] Peng Yu, Continuous Relaxation of Over-constrained Temporal Plans, Master Thesis, Massachusetts Institute of Technology, February 2013.