Explanations by AIDA (Affective Intelligent Driving Agent)

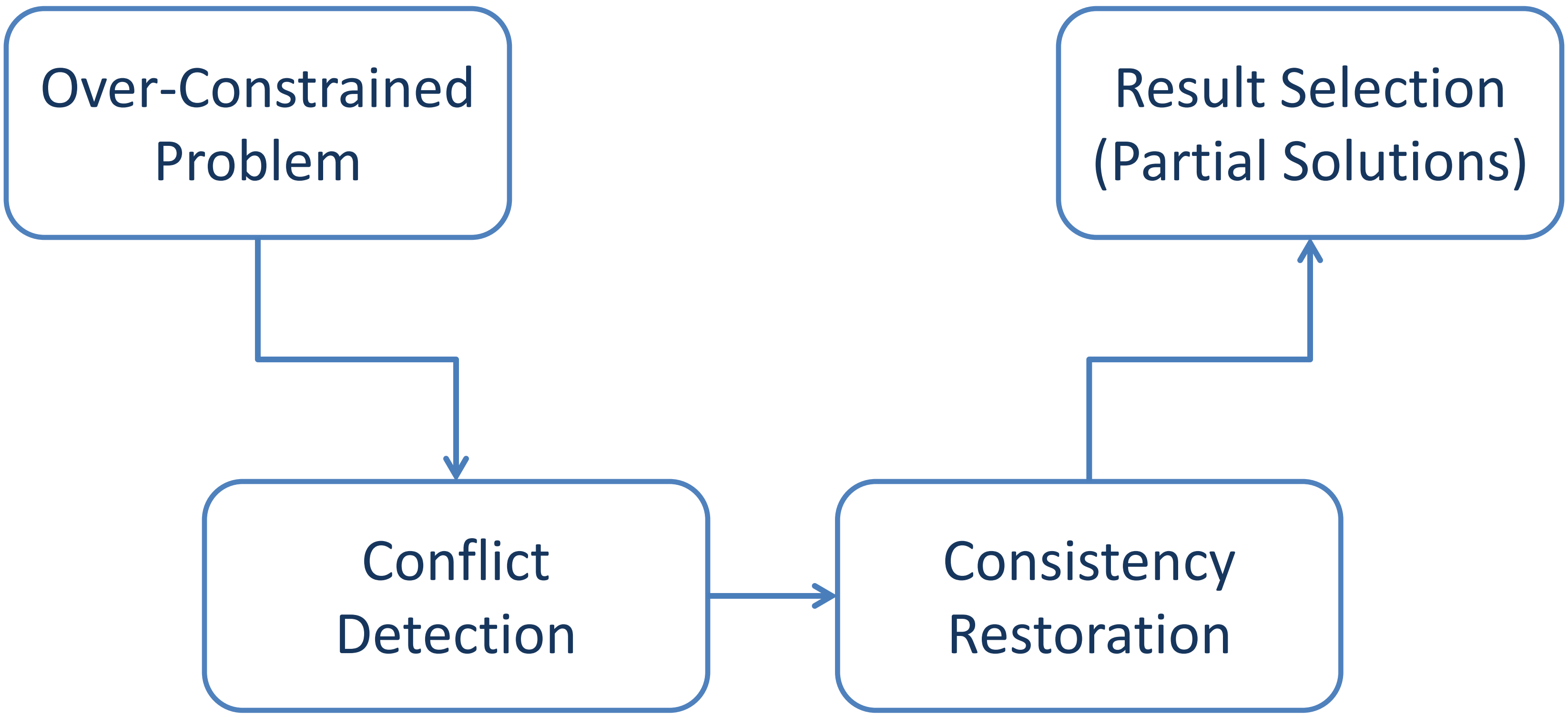
Nancy Foen, Peng Yu

May 9, 2011

Content

- Motivation
- Problem Definition
- System Architecture
- Diagnosis Algorithms
- Evaluation & Conclusion

Plan Diagnosis Review

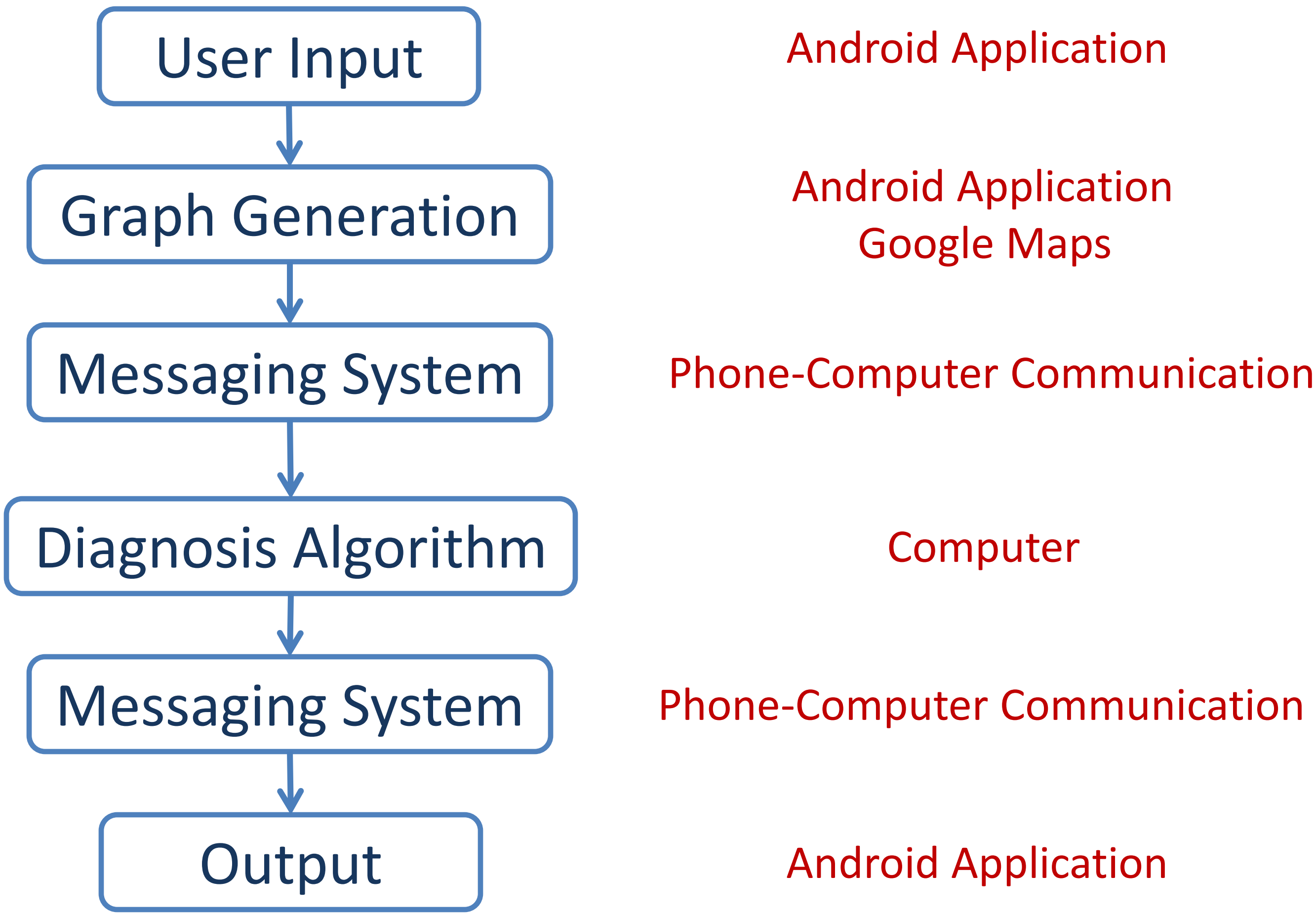


Problem Definition

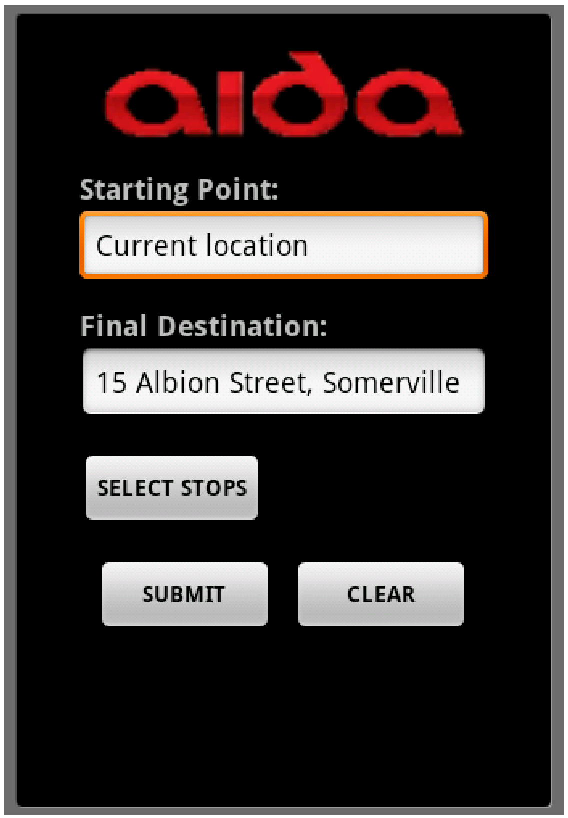
- Problem: Hard to visualize the applications of algorithms that generate diagnosis
- Goal: Provide a simple demo
  - Implemented on AIDA, a robotic driving assistant
  - Scenario:
    1. User inputs information (starting point, destination, stops and time constraint).
    2. AIDA checks the feasibility of the plan. If unfeasible, AIDA looks for alternative plans.
    3. AIDA returns the results.

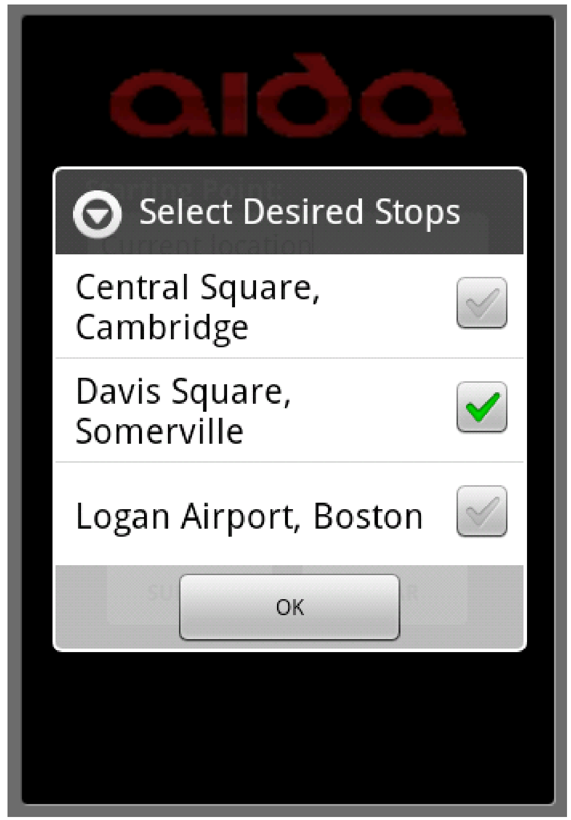
AIDA (Affective Intelligent Driving Agent)

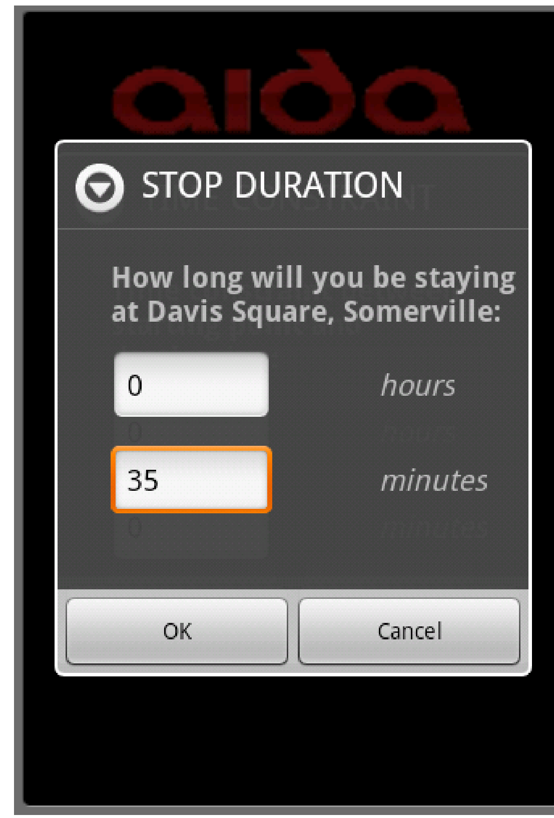
System Architecture



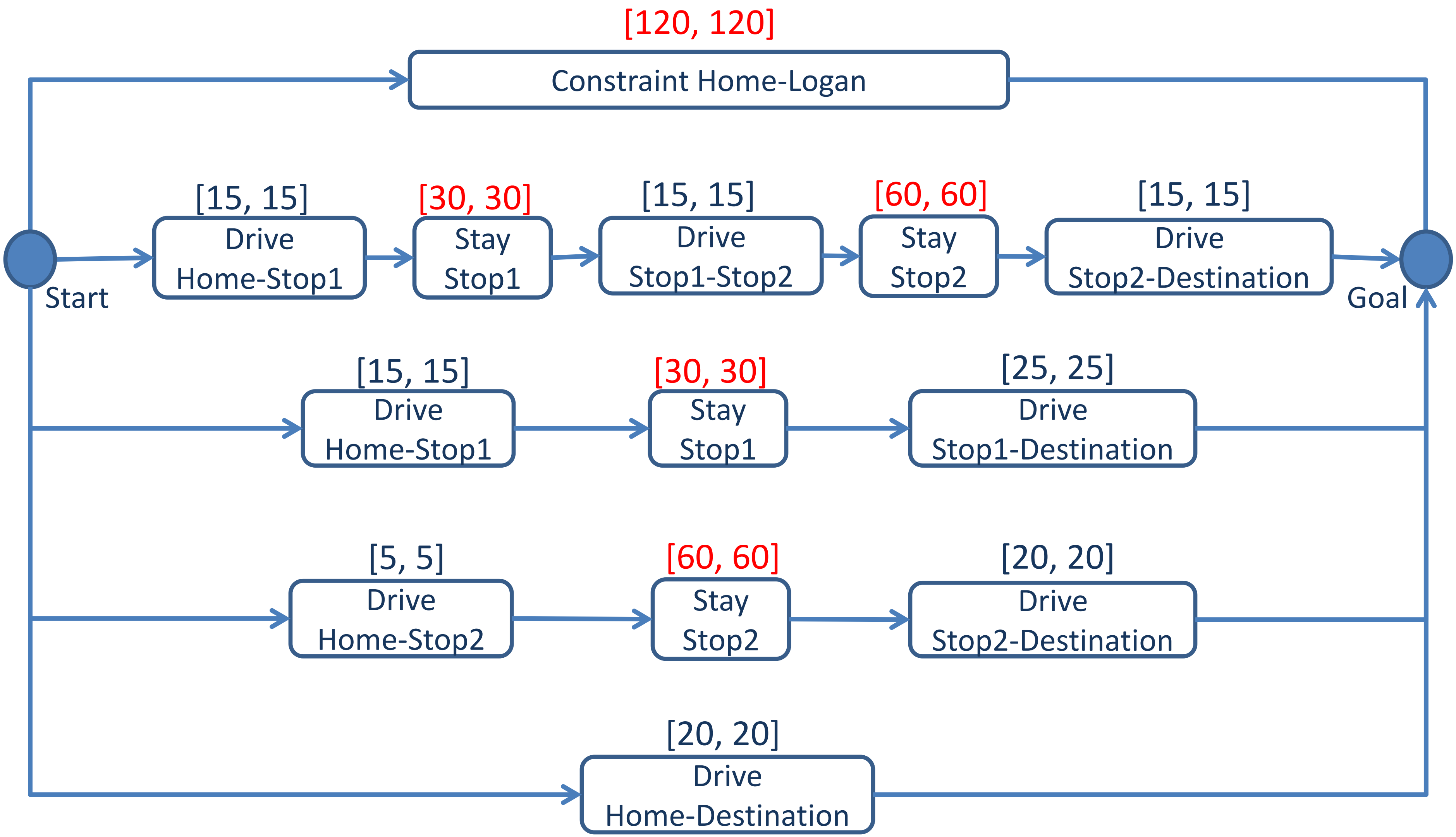
User Input


aida
Starting Point:
Current location
Final Destination:
15 Albion Street, Somerville
SELECT STOPS
SUBMIT
CLEAR


aida
Select Desired Stops
Central Square, Cambridge
Davis Square, Somerville
Logan Airport, Boston
OK


aida
STOP DURATION
How long will you be staying at Davis Square, Somerville:
0
hours
35
minutes
OK
Cancel

Graph Generation



Graph Generation

1. From user input:

   [STOP1, STOP2, STOP3] → [true, false, true]

2. Generate options

   True options = [STOP1, STOP3]

   Options = [true, true]
[true, false]
[false, true]
[false, false]

3. Create Graph

Messaging System

- Client (phone) – server (PC) system
- An XML file is created with the data on the graph generation and sent to server


```
<TPNS>
- <TPN>
 <FORMAT>Spec2</FORMAT>
 <NAME>Main.run</NAME>
 <START>0</START>
 <END>1</END>
 <DECISION>2</DECISION>

-<ARC>
 <START>4</START>
 <END>5</END>
 <PRIMITIVE>USER.drive("CURRENT_LOCATION","CENTRAL_SQUARE,_CAMBRIDGE")</PRIMITIVE>
 <COST>0.0</COST>
 <LOWERBOUND>5</LOWERBOUND>
 <UPPERBOUND>5</UPPERBOUND>
<\ARC>
 ....
```

Diagnosis Algorithms

- Objective:
  - Detect cause of failures in temporal plans.
  - Provide suggestions to recovery plan consistency.

- Detection:
  - RepresentativeXPlain by Barry O'Sullivan.
  - Conflict-Directed A* by Brian Williams.

- Recovery:
  - Continuous domain relaxation.

Representative Explanation

- Generate maximal relaxations through growing.
- Compute hitting sets of excluded constraints to generate new relaxation candidates.
- Stop iteration while all constraints appear in the relaxation + exclusion sets.
- Return:
  - Maximal Relaxation sets: ‘Do’
  - Exclusion sets: ‘Miss’

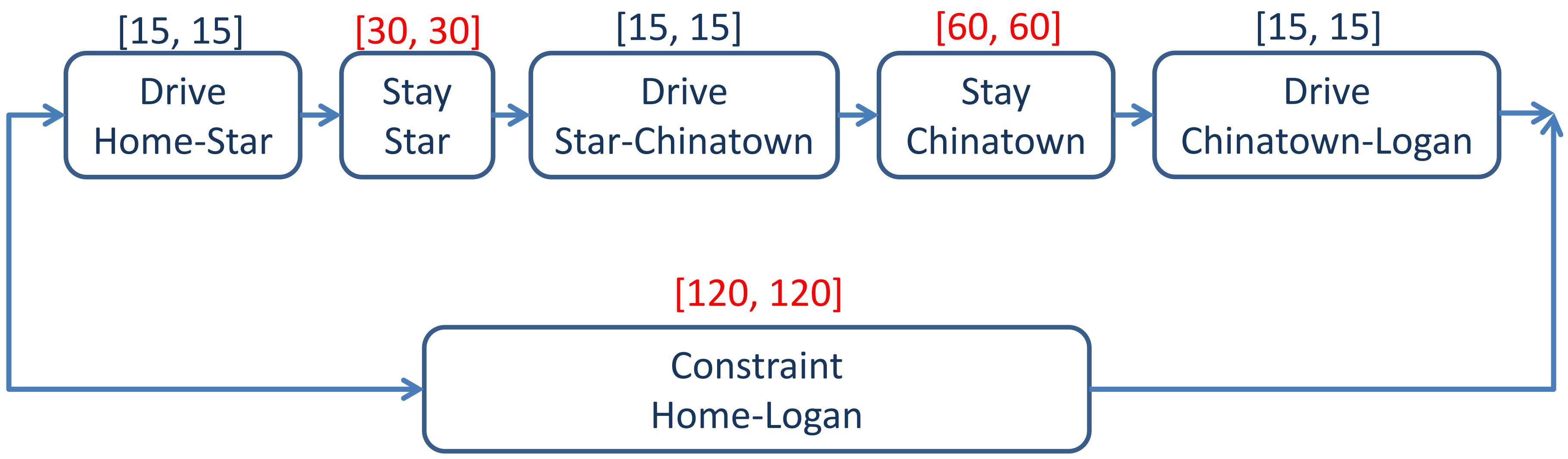
Representative Explanation

- Example:
  - Arrive at logan in 2 hrs, with a stop at Star market (30 mins) and chinatown (60 mins).
  - Driving time: Home-Star (15 mins); Star-Chinatown(15 mins) and Chinatown-Logan (15 mins).
- Output:
  - {stopStar = Yes; stopChinatown = No; onTime = Yes};
  - {stopStar = No; stopChinatown = Yes; onTime = Yes};
  - {stopStar = Yes; stopChinatown = Yes; onTime = No};
- What if the user want to keep all three constraints?

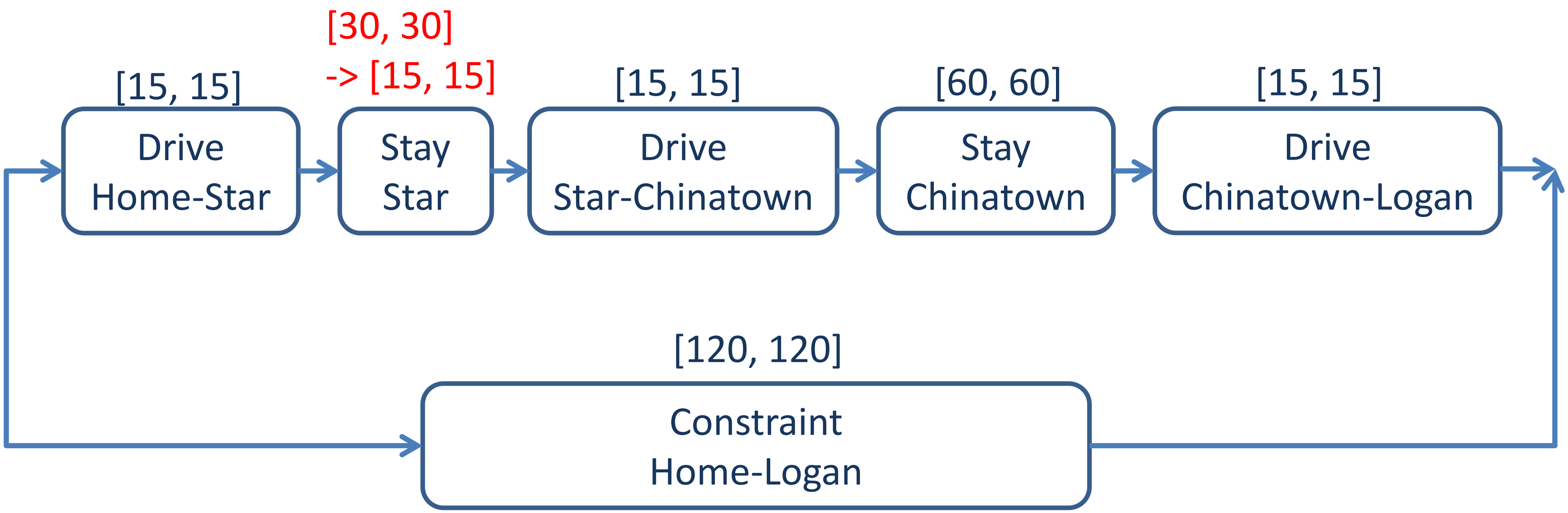
Continuous Relaxation

- Instead of removing an activity, we calculate a new feasible duration for it.
- Example:
  - Star Market: (30 mins) -> (15 mins).
  - Chinatown: (60 mins) -> (45 mins).
  - TotalDuration: (2 hrs) -> (2 hrs 15 mins)
- The new duration is calculated by APSP algorithm.

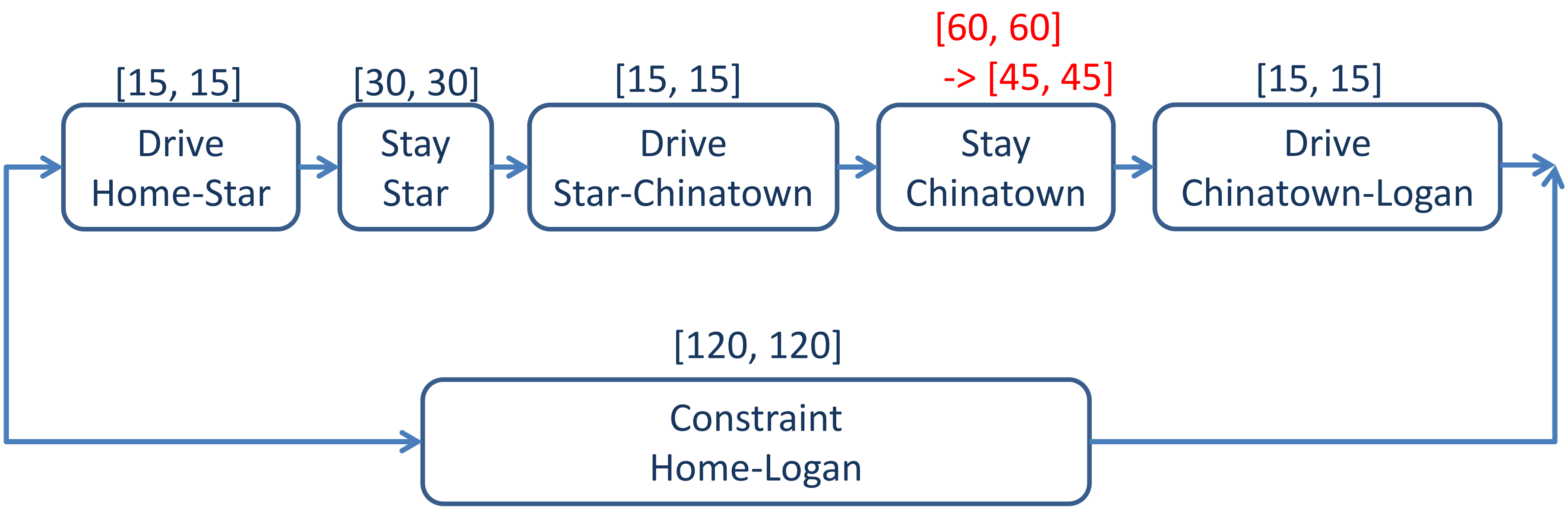
Continuous Relaxation



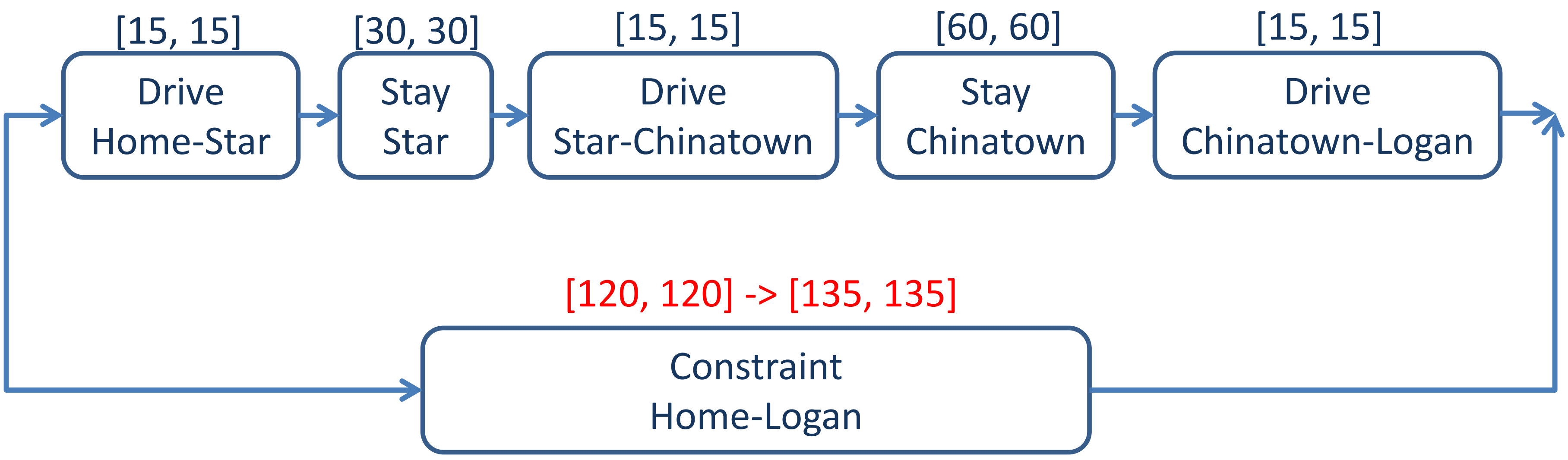
Continuous Relaxation



Continuous Relaxation



Continuous Relaxation



Evaluation of RepresentativeXPlain

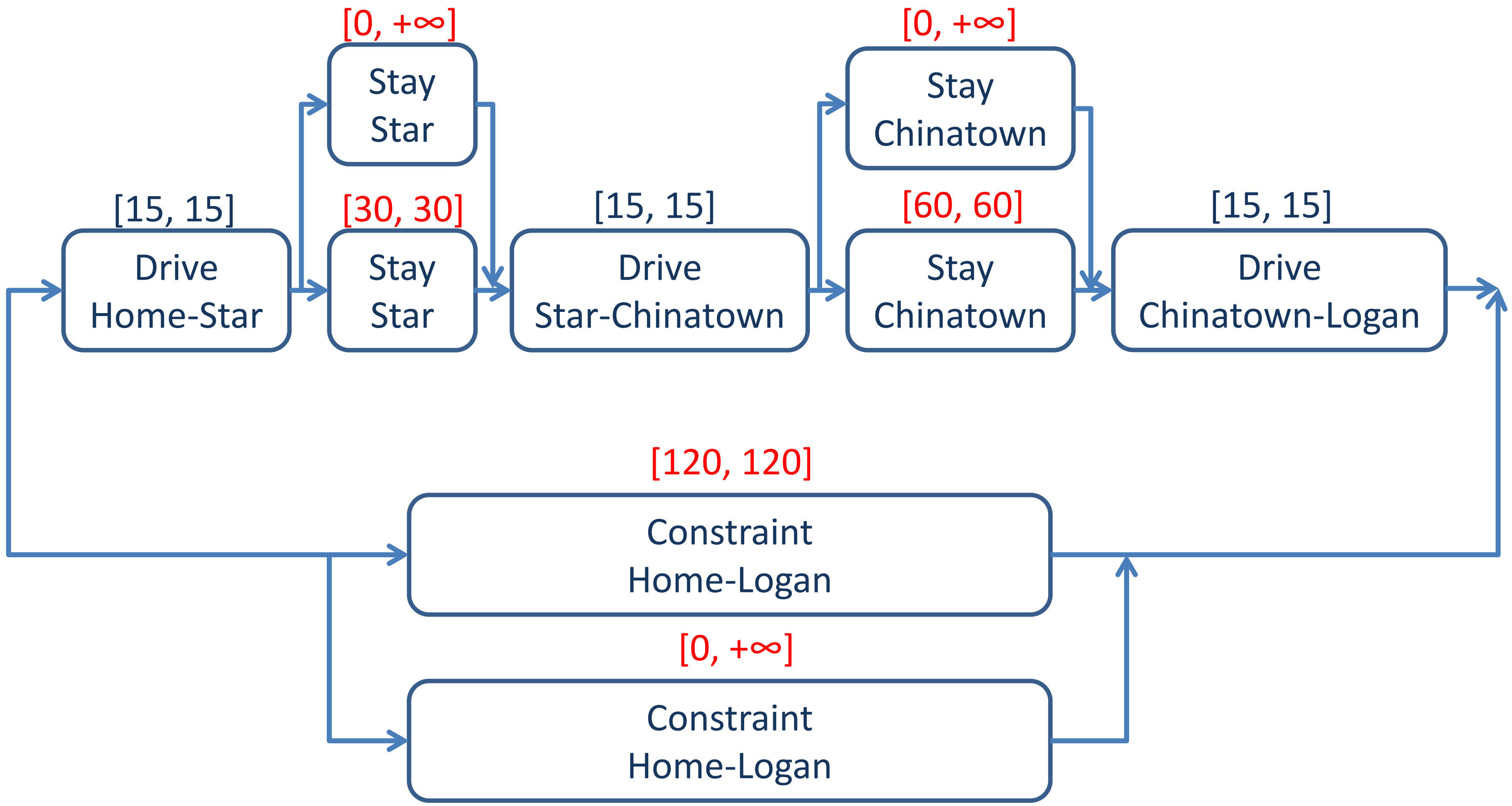
- Provide basic capability of plan diagnosis and recovery.
- Reduce result size and speed up diagnosis process*.
- However, there still room for improvements:
  - The growing process doesn't use any conflicts to prune repeated candidates.

- Use Conflict-Directed A* to find maximal relaxations

* Comparing to: 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), volume 3350 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

Problem Formulation

- Add additional choices to relaxable constraints.



Conflict-Directed Relaxation

- Add additional choices to relaxable constraints.
- Start search from the highest reward candidates.
  - If inconsistent, extract the conflict and move to the next candidate.
  - If consistent, add constraints with “Miss” assignments to the conflict set and move on.
- When candidates exhaust, returns the consistent candidates.

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = No; onTime = Yes};

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = No; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = No; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = Yes; onTime = No};

Conflict-Directed Relaxation

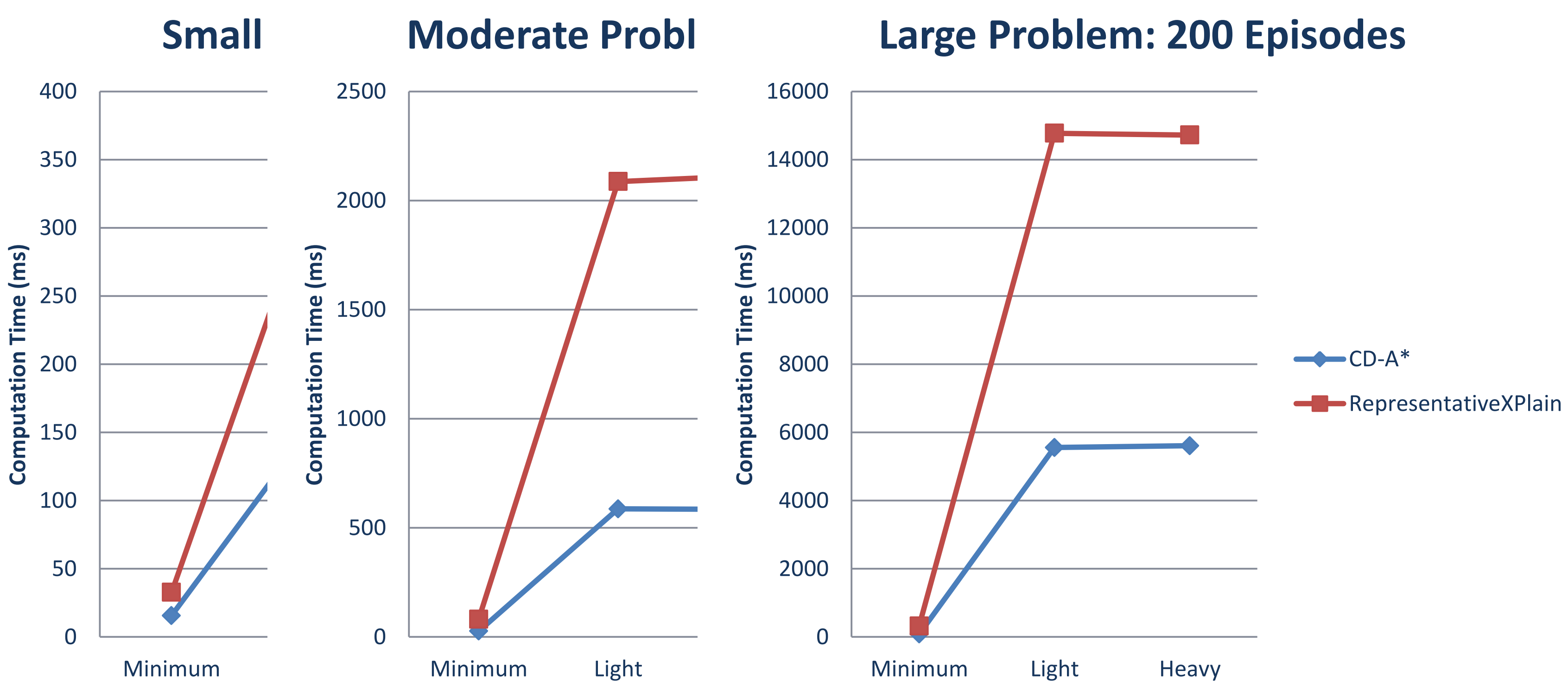
- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = No; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = Yes; onTime = No};
- {stopStar = No; stopChinatown = Yes; onTime = No};

Conflict-Directed Relaxation

- {stopStar = Yes; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = Yes; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = No; onTime = Yes};
- {stopStar = No; stopChinatown = No; onTime = Yes};
- {stopStar = Yes; stopChinatown = Yes; onTime = No};
- {stopStar = No; stopChinatown = Yes; onTime = No};

Performance Comparison

- Three test cases:
  - 50, 100 and 200 episodes.
- Three over-constrained levels:
  - Minimum, light and heavy.



Future Work

- Flexible Plan Diagnosis.
  - Have the diagnosis algorithm to handle all plan variations.
- Consider user preference.
- Performance improvements.

Questions