Incremental Search for Counterexample-Guided Cartesian Abstraction Refinement

Jendrik Seipp, Samuel von Allmen, Malte Helmert

October 28, 2020

University of Basel

In a Nutshell

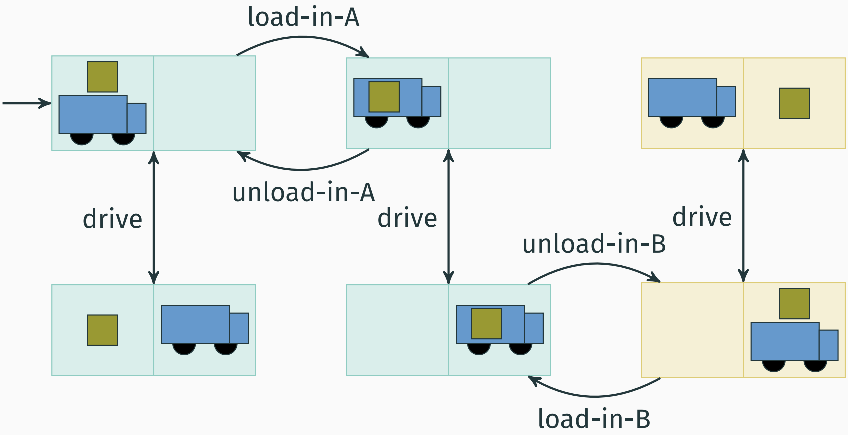
- optimal classical planning
- A^* search + abstraction heuristic
- counterexample-guided Cartesian abstraction refinement
- bottleneck: find shortest path
- incremental search: 1000x speedup

Counterexample-Guided Abstraction Refinement

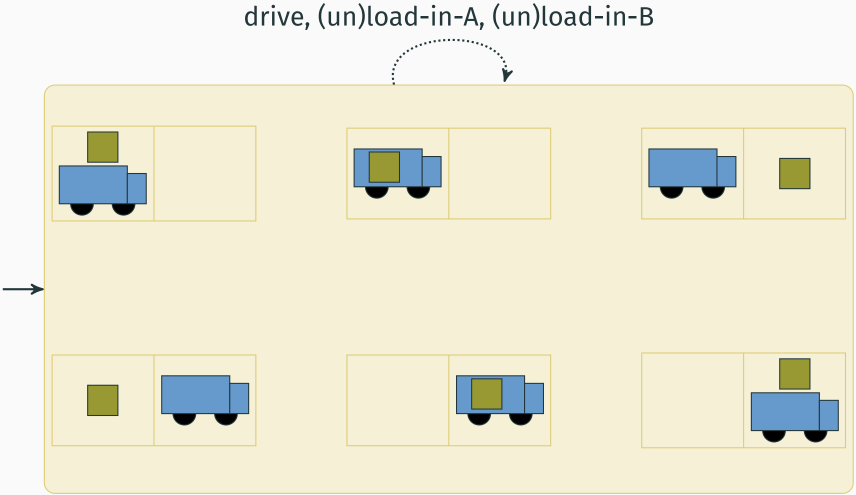
CEGAR

```
compute initial abstraction
until TERMINATE():
    find shortest path in abstraction
    if there is no path:
        return unsolvable
    find flaw in path
    if there is no flaw:
        return plan
    refine abstraction for flaw
return abstraction
```

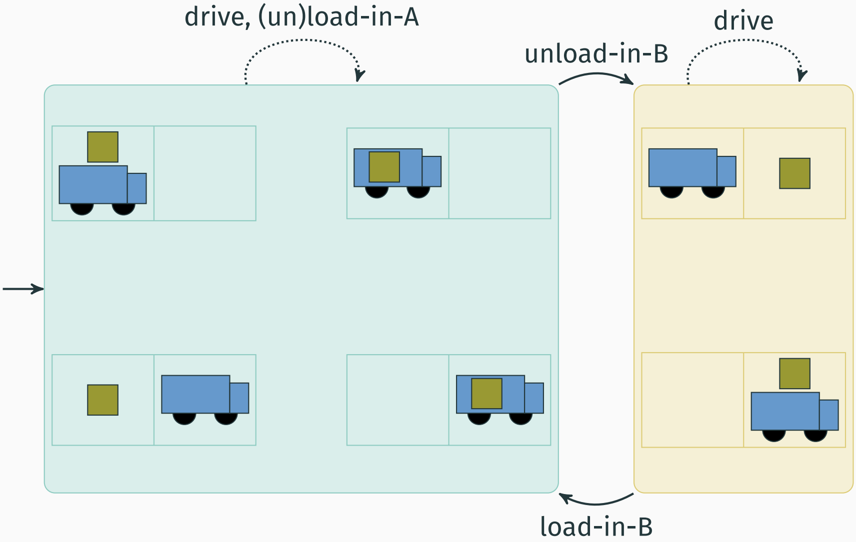
Example Task



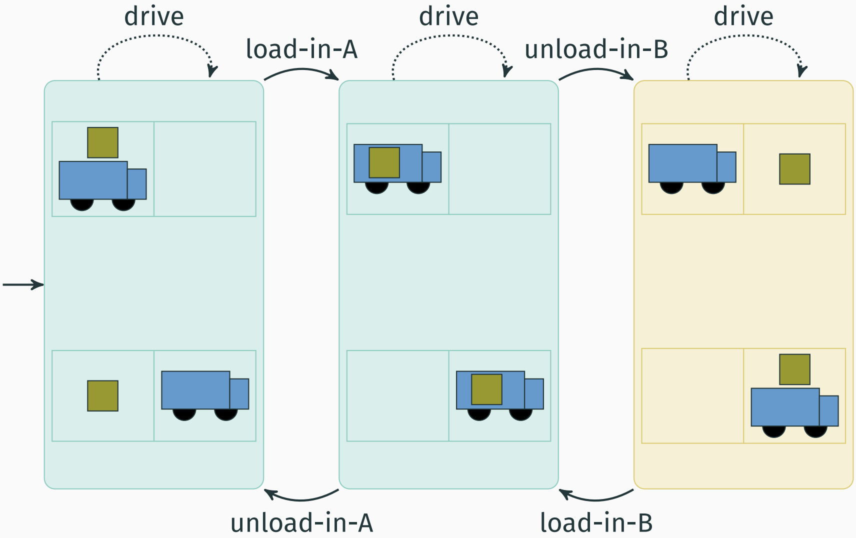
Abstraction Refinement



Abstraction Refinement



Abstraction Refinement



Bottleneck: Find Shortest Path

- Dijkstra's algorithm
- A^* search
- search times grow

Bottleneck: Find Shortest Path

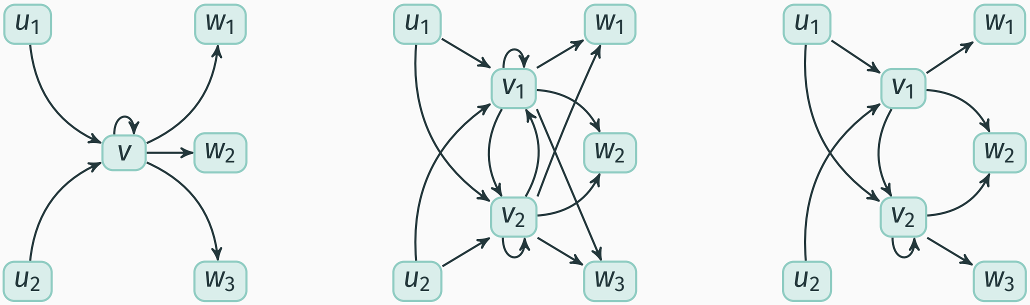
- Dijkstra's algorithm
- A^* search
- search times grow

$\rightarrow$ incremental search

Incremental Search

- dynamic shortest path
- add/remove transitions
- increase/decrease weights
- fixed set of states

Two-step Refinement for CEGAR



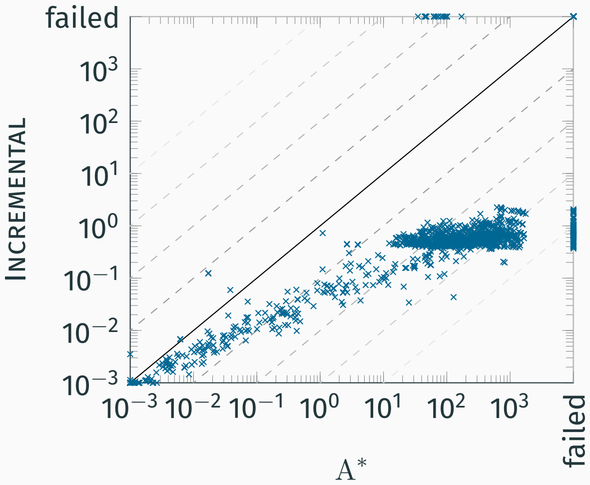
before splitting v | copy v | prune transitions

Incremental Search Algorithm

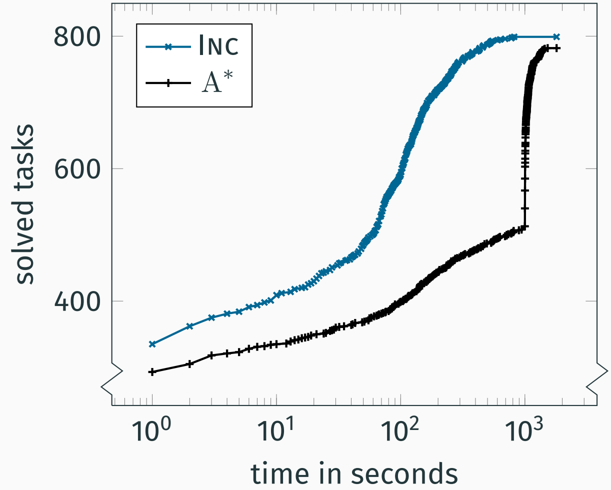
Increase (Frigioni et al., 2000)

- increasing weights, removing transitions
- shortest path tree
- algorithm:
  - reconnect ancestor states, mark rest dirty
  - run Dijkstra on dirty states

Time for Finding Shortest Paths



Solved Tasks Over Time



Summary

- CEGAR bottleneck: find shortest paths
- cast as dynamic shortest path problem
- drastic speedup