Automatic Selection of Pattern Collections for Domain Independent Planning

MASTER THESIS PRESENTATION

Overview

- Introduction & Overview
- Background
- iPDB Heuristic
- PhO Heuristic
- Conclusion & Future Work

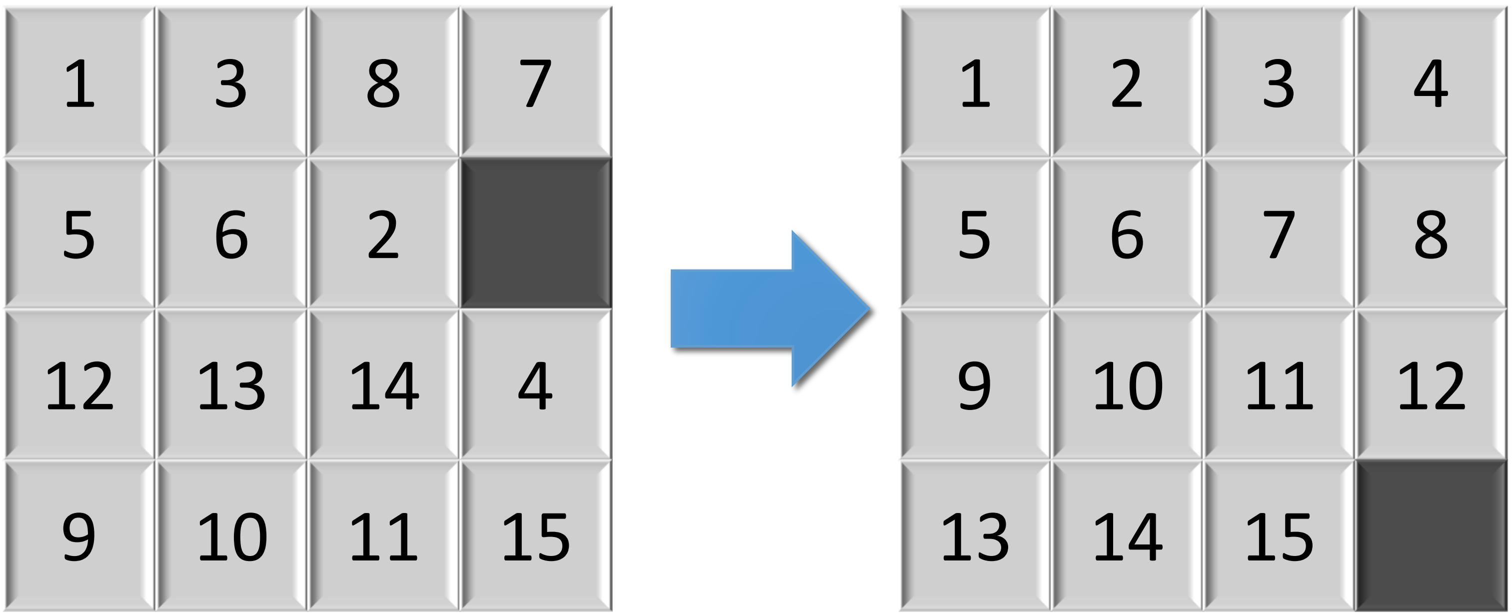
Background: Planning

- Planning task
- Variables
- States assign values to variables
- Initial state s_0 and goal state(s) s_*
- Operators have preconditions and effects

Background: Planning

- Heuristic search in state space
- Heuristic
  - Estimates cost to nearest goal
  - Admissible: never overestimates
- A* used as search algorithm
- IPC Benchmark & Coverage

Background: 15-Puzzle



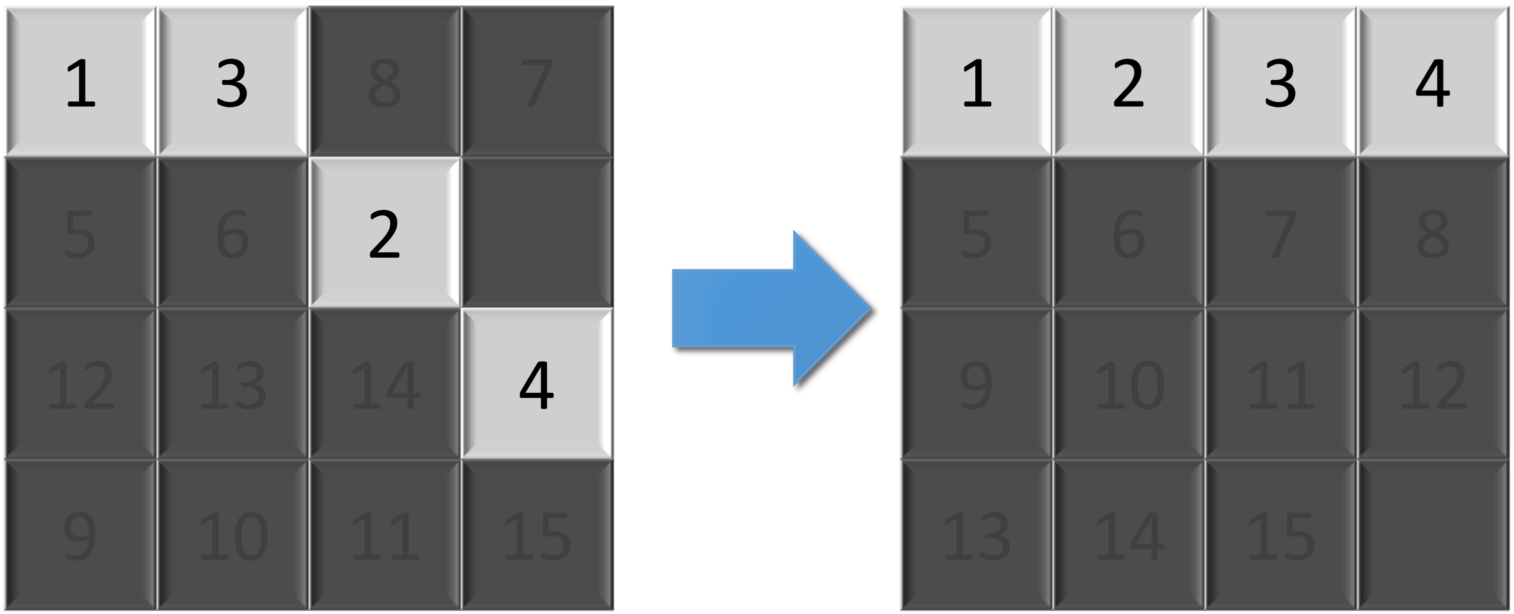
s_0 s_*

Background: Pattern Databases

- Pattern is a subset of variables $P \subseteq V$
- Simplified problem ignoring all variables not in the pattern P
- PDB contains cost of optimal plans for all states in the simplified problem
- Evaluating $h^P(s)$

Background: 15-Puzzle Abstraction

Pattern $P = \{T_1, T_2, T_3, T_4\}$



s_0

$h^P(s_0) = 5$

s_*

Background: Using multiple PDB heuristics

- Selection of a pattern collection
- Combination of heuristic values
  - Maximum of both heuristics
  - Sum of both heuristics if pattern are additive
- Two patterns are additive if there is no operator that affects both
- Canonical heuristic
  - Add where possible, maximum over sums

iPDB Heuristic

- Haslum et al. 2007
- Hill climbing algorithm in space of pattern collections
- Candidate patterns
- Samples
- Evaluated using the canonical heuristic
- Improvement

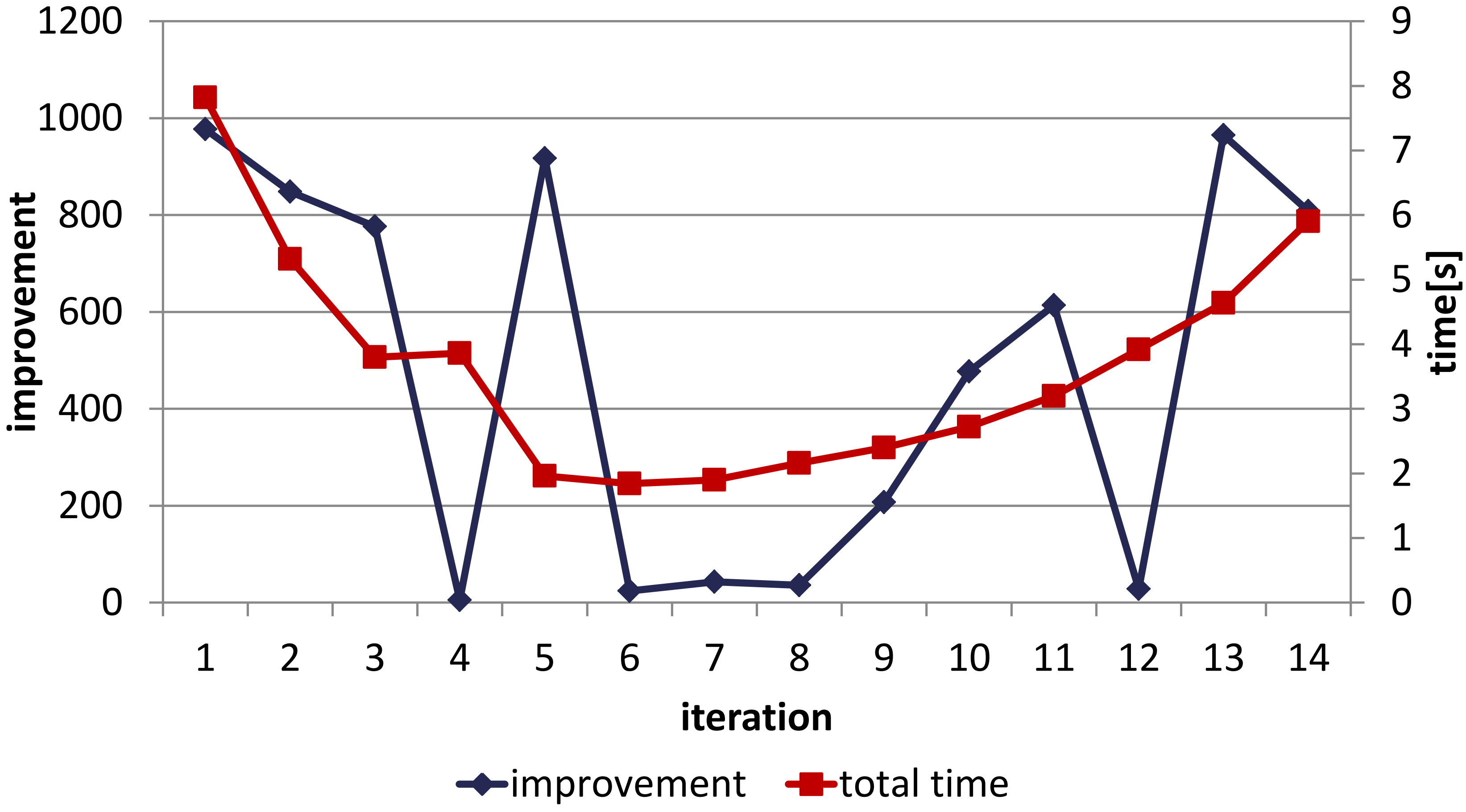
iPDB Heuristic: Hill Climbing

1. Add pattern for each goal variable to collection
2. Compute candidates
3. Repeat:
   1. Evaluate candidates
   2. Add best candidate to collection
   3. Compute new candidates
   4. Check stopping criterion

iPDB: Changes

- Time limit
- Improvement limit
- Ignoring candidates
- Changed sampling
- Increasing neighbourhood

iPDB: Improvement & Total Time



iPDB: Increasing the Neighbourhood

- Uses time limit of 900 seconds
- Add n new variables when creating candidates
- Variables must influence each other
- Memory issues

iPDB: Evaluation

- iPDB 2var and 3var include a time limit

Coverage			
iPDB base	iPDB time	iPDB 2var	iPDB 3var
664	684	706	698

PhO: Posthoc Optimization

- Pommerening et al. 2013
- Systematic pattern generation
- Evaluation: solves linear program (LP)
  - One variable per operator
  - One constraint per PDB
  - Depends on state

PhO: Changes

- Partial systematic sizes
- Dynamic systematic sizes
- Limiting number & total size of PDBs
- Limiting evaluation time
- Pruning unused constraints

PhO: Pruning Unused Constraints

- Samples
- Evaluate heuristic on samples
- Count how often each constraint was "useful"
- Remove constraints that were never "useful"

PhO: Evaluation

Coverage	
PhO base	PhO prune
590	598

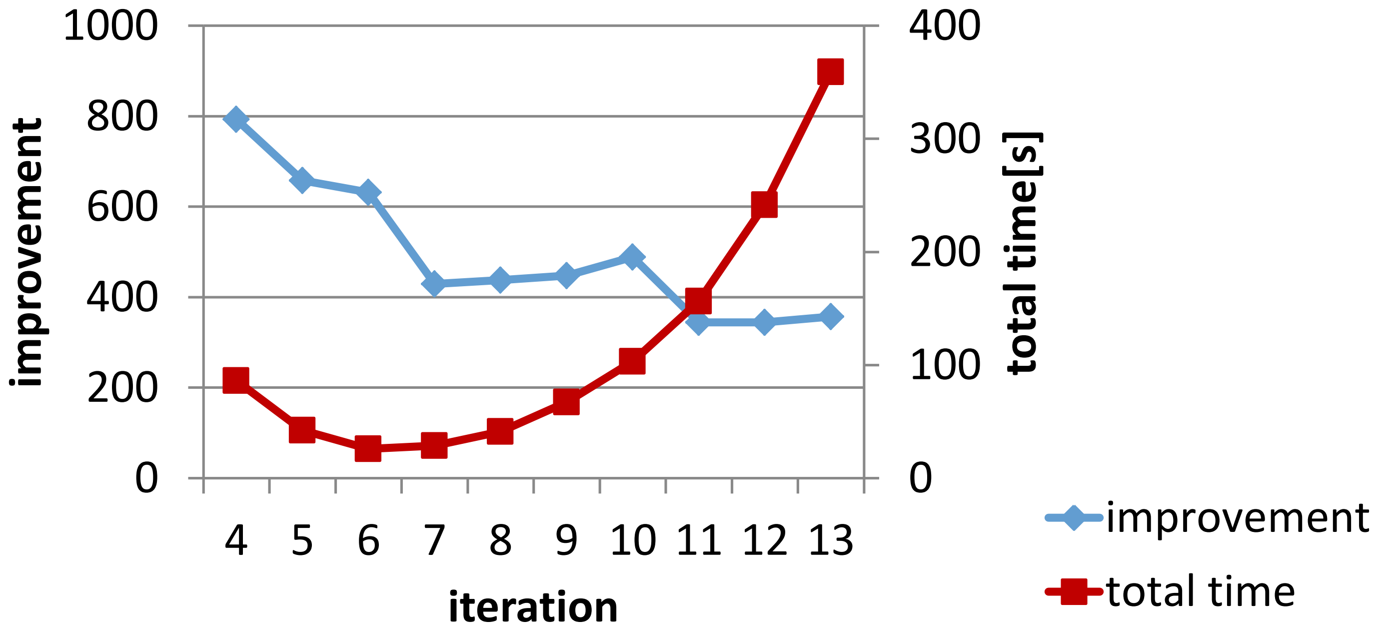
Conclusion & Future Work

- Viability of abstraction heuristics
- Further investigate
  - Better limit for iPDB
  - Better pruning of constraints for PhO

	iPDB 2var	PhO prune	LM-cut
Coverage	706	598	763
Avg. coverage percentage	54,38%	45,19%	53,07%

Additional Data: iPDB

	base	dyn.	time	2var	3var	ign.	FF
finished	1134	1360	1360	1361	1345	1308	1388
coverage	664	675	684	706	698	705	703



Additional Data: PhO

	base (k=2)	eval.	PhO prune
coverage	590	591	598