Counterexample-Guided Abstraction Refinement for Pattern Selection in Optimal Classical Planning

Alexander Rovner Silvan Sievers Malte Helmert

University of Basel, Switzerland

July 14, 2019

Motivation

- Pattern databases (PDB) for optimal planning:
  - Based on pattern collections (single PDBs don't scale)
  - Combining PDBs: e.g., canonical PDBs, cost partitioning
  - Pattern selection: e.g., hill climbing, genetic optimization

Motivation

- Pattern databases (PDB) for optimal planning:
  - Based on pattern collections (single PDBs don't scale)
  - Combining PDBs: e.g., canonical PDBs, cost partitioning
  - Pattern selection: e.g., hill climbing, genetic optimization
- This work: pattern selection (fixed combination: SCP)
  - Observation: existing algorithms relatively slow

Motivation

- Pattern databases (PDB) for optimal planning:
  - Based on pattern collections (single PDBs don't scale)
  - Combining PDBs: e.g., canonical PDBs, cost partitioning
  - Pattern selection: e.g., hill climbing, genetic optimization
- This work: pattern selection (fixed combination: SCP)
  - Observation: existing algorithms relatively slow
- Contribution: pattern selection based on the counterexample-guided abstraction refinement principle
  - Fast method
  - Only select useful patterns
  - Convergence

Outline

1. Disjoint Pattern Collections with CEGAR
2. Multiple CEGAR Runs
3. Experimental Results

Schematic Algorithm

- Disjoint collection: compromise between single patterns and arbitrary collections

Schematic Algorithm

- Disjoint collection: compromise between single patterns and arbitrary collections
- Initialize pattern collection C with one pattern per goal

Schematic Algorithm

- Disjoint collection: compromise between single patterns and arbitrary collections
- Initialize pattern collection C with one pattern per goal
- Repeat:
  - For each $P \in C$, compute abstract plan π_P

Schematic Algorithm

- Disjoint collection: compromise between single patterns and arbitrary collections
- Initialize pattern collection C with one pattern per goal
- Repeat:
  - For each $P \in C$, compute abstract plan π_P
  - For each $P \in C$, look for flaws v of π_P

Schematic Algorithm

- Disjoint collection: compromise between single patterns and arbitrary collections
- Initialize pattern collection C with one pattern per goal
- Repeat:
  - For each $P \in C$, compute abstract plan π_P
  - For each $P \in C$, look for flaws v of π_P
  - Select flaw $\langle P, v \rangle$ and refine C by adding v to C: add v to P or merge P with P' containing v

Outline

1. Disjoint Pattern Collections with CEGAR
2. Multiple CEGAR Runs
3. Experimental Results

Multiple CEGAR

- Repeatedly use CEGAR and combine all patterns

Multiple CEGAR

- Repeatedly use CEGAR and combine all patterns
- Diversification:
  - Restrict each iteration to single goal: $\Rightarrow$ single pattern
  - Randomly forbid variables for selection (blacklisting)
  - Keep track of progress (stagnation)

Outline

Coverage (SCP Heuristic) on IPC Benchmarks

	Competitors		
	SYS2	HC (900s)	CPC (100s)
Coverage	981	965.4	**1033.5**
Constr. t.	**0.05**	4.97	103.82

Coverage (SCP Heuristic) on IPC Benchmarks

Competitors

	SYS2	HC (900s)	CPC (100s)
Coverage	981	965.4	**1033.5**
Constr. t.	**0.05**	4.97	103.82

	Single CEGAR
Coverage	946.6
Constr. t.	**0.48**

Coverage (SCP Heuristic) on IPC Benchmarks

Competitors

	SYS2	HC (900s)	CPC (100s)
Coverage	981	965.4	**1033.5**
Constr. t.	**0.05**	4.97	103.82

	Single CEGAR	Multiple CEGAR	+ bl + stag
Coverage	946.6	1063.2	**1087.2**
Constr. t.	**0.48**	51.81	39.65

Conclusions

- CEGAR for pattern selection: fast algorithm
- State-of-the-art pattern selection (for explicit PDBs & until IJCAI)
- Future work: interleave pattern selection with cost partitioning

Results

Competitors						
	SYS	HC1	CPC1	HC9	CPC9	G
Cov.	981	946.4	1033.5	965.4	1021.1	839
C.t.	**0.05**	3.05	103.82	4.97	876.21	5.49

	CEGAR			
	single	multiple	sRCG	mRCG
Cov.	946.6	**1087.2**	758.8	1018.7
C.t.	0.48	39.65	**0.07**	10.07