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
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
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 $\pi_P$
Disjoint Pattern Collections with CEGAR

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 \( \pi_P \)
  - For each \( P \in C \), look for flaws \( v \) of \( \pi_P \)
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 $\pi_P$
  - For each $P \in C$, look for flaws $v$ of $\pi_P$
  - 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: ⇒ single pattern
  - Randomly forbid variables for selection (blacklisting)
  - Keep track of progress (stagnation)
Outline

1. Disjoint Pattern Collections with CEGAR

2. Multiple CEGAR Runs

3. Experimental Results
## Coverage (SCP Heuristic) on IPC Benchmarks

<table>
<thead>
<tr>
<th>Competitors</th>
<th>SYS2</th>
<th>HC (900s)</th>
<th>CPC (100s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>981</td>
<td>965.4</td>
<td><strong>1033.5</strong></td>
</tr>
<tr>
<td>Constr. t.</td>
<td><strong>0.05</strong></td>
<td>4.97</td>
<td>103.82</td>
</tr>
</tbody>
</table>
## Coverage (SCP Heuristic) on IPC Benchmarks

<table>
<thead>
<tr>
<th>Competitors</th>
<th>SYS2</th>
<th>HC (900s)</th>
<th>CPC (100s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>981</td>
<td>965.4</td>
<td>1033.5</td>
</tr>
<tr>
<td>Constr. t.</td>
<td>0.05</td>
<td>4.97</td>
<td>103.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Single CEGAR</th>
<th>Coverage</th>
<th>Constr. t.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>946.6</td>
<td>0.48</td>
</tr>
</tbody>
</table>
## Experimental Results

### Coverage (SCP Heuristic) on IPC Benchmarks

<table>
<thead>
<tr>
<th>Competitors</th>
<th>SYS2</th>
<th>HC (900s)</th>
<th>CPC (100s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>981</td>
<td>965.4</td>
<td>1033.5</td>
</tr>
<tr>
<td>Constr. t.</td>
<td>0.05</td>
<td>4.97</td>
<td>103.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Single CEGAR</th>
<th>Multiple CEGAR</th>
<th>+ bl + stag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>946.6</td>
<td>1063.2</td>
<td>1087.2</td>
</tr>
<tr>
<td>Constr. t.</td>
<td>0.48</td>
<td>51.81</td>
<td>39.65</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th>SYS</th>
<th>HC1</th>
<th>CPC1</th>
<th>HC9</th>
<th>CPC9</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cov.</td>
<td>981</td>
<td>946.4</td>
<td>1033.5</td>
<td>965.4</td>
<td>1021.1</td>
<td>839</td>
</tr>
<tr>
<td>C.t.</td>
<td><strong>0.05</strong></td>
<td>3.05</td>
<td>103.82</td>
<td>4.97</td>
<td>876.21</td>
<td>5.49</td>
</tr>
</tbody>
</table>

## CEGAR

<table>
<thead>
<tr>
<th></th>
<th>single</th>
<th>multiple</th>
<th>sRCG</th>
<th>mRCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cov.</td>
<td>946.6</td>
<td><strong>1087.2</strong></td>
<td>758.8</td>
<td>1018.7</td>
</tr>
<tr>
<td>C.t.</td>
<td>0.48</td>
<td>39.65</td>
<td><strong>0.07</strong></td>
<td>10.07</td>
</tr>
</tbody>
</table>