Constraint Partitioning for Solving Planning Problems with Trajectory Constraints and Goal Preferences

Chih-Wei Hsu, Benjamin W. Wah, Ruoyun Huang, Yixin Chen

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
URL: http://manip.crhc.uiuc.edu

IJCAI-07

Outline

• Observation on constraint locality
• Partition-and-resolve approach
  • Partition a problem by its constraints into subproblems
• Observation in PDDL3 domains
• Architectures of SGPlan5
  • Handling soft goals
  • Handling trajectory constraints
  • Subproblem-level planning
• Conclusions
An Airport Planning Example

**Spatial Constraint Locality in AIRPORT**

Mutual exclusion between 2 subgoals:
2 planes cannot be on the same runway at the same time
Exploring Constraint Partitioning in Planning

Depicting Constraint Localities

Problem Formulation

Rovers-Propositional-P30

Constraints: mutual exclusions, trajectory constraints and goals
Variables: initial solution plan
Global constraints: constraints span across multiple subproblems

Attribute used for partitioning: subgoals

Outline

• Observation on constraint locality
• Partition-and-resolve approach
  • Partition a problem by its constraints into subproblems

Observation in PDDL3 domains
• Architectures of SGPlan\textsubscript{5}
  • Handling soft goals
  • Handling trajectory constraints
  • Subproblem-level planning
• Conclusions
Motivations of Approach

- Significantly smaller aggregate complexity of solving all subproblems
- Large reduction in solution time if global constraints can be resolved quickly
- Study of constraint locality and partitioning granularity to reduce the number of global constraints

![Graph showing exponential growth in solution time and convex relationship between number of partitions and total time]

**Subspace Partitioning**

**Partition** \( P \) by branching on the values of a variable

Solve \( P \) by choosing the correct path and by solving the subproblem

Overhead for solving each subproblem is similar to that of \( P \)
**Constraint Partitioning**

*Partition P* by its constraints into subproblems

Solve P by solving all the subproblems and by resolving those violated (active) global constraints

Overhead of each subproblem is substantially smaller

**Our Partition-and-Resolve Framework**

- Solving a subproblem
  - Satisfy local constraints
  - Minimize global objective
  - Satisfy (soft) global constraints
  - Increasing penalties on violated global constraints

Weighted active global constraints provide guidance in local subproblems

Similar solver as original problem

\[
\min_{z(1)} J(z)
\]
\[
s.t. \ h^{(1)}(z(1)) = 0
\]
\[
g^{(1)}(z(1)) \leq 0
\]

\[
\min_{z(n)} J(z)
\]
\[
s.t. \ h^{(n)}(z(n)) = 0
\]
\[
g^{(n)}(z(n)) \leq 0
\]
Outline

• Observation on constraint locality
• Partition-and-resolve approach
  • Partition a problem by its constraints into subproblems
• Observation in PDDL3 domains
• Architectures of SGPlan
  • Handling soft goals
  • Handling trajectory constraints
  • Subproblem-level planning
• Conclusions

Constraint Locality in PDDL3

• Planning benchmarks in PDDL 3.0
  • New features over PDDL 2.2
    • Soft goals or preferences on action preconditions
    • Hard and soft constraints on plan trajectory
• Different locality of constraints caused by new features
  • Poor locality by subgoal partitioning
    • Many global constraints due to implicit mutual exclusions
    • Trajectory constraints not considered in partitioning
**Multi-Valued Domain Formulation**

- Multi-valued state variables
  - Traditionally encoded in multiple binary facts
  - Now encode in one variable that can take multiple values
- Example
  - \( \text{AT(\text{TRUCK1}, \text{LOCATION1}), ..., AT(\text{TRUCK1}, \text{LOCATION8})} \) → \( \text{LOC(\text{TRUCK1}) = LOC1, ..., LOC(\text{TRUCK1}) = LOC8} \)
- More compact representation than STRIPS
- New problem representation for reducing the number of global constraints

**Constraint Locality by Other Variables**

- Guidance variable: variable in goal-state constraint
  - Ex: Stored quantity of a product
  - Cluster constraints by guidance variables
Granularity in Partitioning

- **Bottleneck variable**: variable in a group that other variables depend on
- **Ex**: Position of Truck in TPP; Position of Hoist in Storage

Table 1: Trade-offs on the number of partitions for the Trucks-TimeConstraints-20 instance.

<table>
<thead>
<tr>
<th>Partitioning Strategy</th>
<th>No Partitioning</th>
<th>Bottleneck</th>
<th>Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td># partitions</td>
<td>1</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td># global constraints</td>
<td>0</td>
<td>20</td>
<td>573</td>
</tr>
<tr>
<td>avg. # local const.</td>
<td>21274</td>
<td>1404</td>
<td>230.1</td>
</tr>
<tr>
<td>per subproblem time/subproblem</td>
<td>&gt;1800 sec.</td>
<td>2.04 sec.</td>
<td>0.16 sec.</td>
</tr>
</tbody>
</table>

Bottleneck: # partitions = min(# bottleneck var., # guidance var.)
Guidance: # partitions = # of guidance variables
Subgoal: subgoal partitioning

Summary of Partitioning Approach

- **Problem representation**: MDF
- **Partitioning attributes**: guidance variables
- **# partitions** = min(# guidance var., # bottleneck var.)
- **Cluster guidance variables using information of constraints/preferences**

Table 2: Average fraction of constraints that are active global constraints initially across all the instances of each IPC$^5$ QualitativePreferences domain under three partitioning strategies. (See keys in Table 1.)

<table>
<thead>
<tr>
<th>Domain</th>
<th>Bottleneck</th>
<th>Guidance</th>
<th>Subgoal</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPP</td>
<td>0.0153</td>
<td>0.0733</td>
<td>0.2892</td>
</tr>
<tr>
<td>OpenStacks</td>
<td>–</td>
<td>0.1728</td>
<td>0.1834</td>
</tr>
<tr>
<td>Trucks</td>
<td>–</td>
<td>0.0749</td>
<td>0.1623</td>
</tr>
<tr>
<td>Storage</td>
<td>0.0007</td>
<td>0.0032</td>
<td>0.0109</td>
</tr>
<tr>
<td>Rovers</td>
<td>0.0227</td>
<td>0.0464</td>
<td>0.0488</td>
</tr>
</tbody>
</table>
Outline

• Observation on constraint locality
• Partition-and-resolve approach
  • Partition a problem by its constraints into subproblems
• Observation in PDDL3 domains
• Architectures of SGPlan\textsubscript{5}
  • Handling soft goals
  • Handling trajectory constraints
  • Subproblem-level planning
• Conclusions

Architecture of SGPlan\textsubscript{5}

Preprocessing

Partitioning Strategy

Handling Soft Goals

Global Constraints on Subproblems

P\textsubscript{1} P\textsubscript{2} P\textsubscript{N}

Handling Trajectory Constraints

Penalty Update Strategy

Plan Evaluation

Temporal Engine

Trajectory Constraint Engine

Search Space Reduction

Granularity Control

Basic Planner

I\textsc{llinois}

Exploring Constraint Partitioning in Planning
Partitioning of Problems with Soft Goals

- State is a conjunctive list of state variables
- Naïve approach
  - Enumerate all reachable values of state variables in (soft and hard) goals
  - Choose assignment that optimizes plan metric
- To overcome high complexity of enumeration
  - Decompose enumeration according to locality of constraints in subgoals
  - Conduct heuristic search for large space

Handling Soft Goals

- If enumeration cost is small and metric has only soft-goal violations (on final state)
  - Enumerate and plan to optimal assignment
- Otherwise
  - Perform DFS with backtracking
    - Use plan metric value as heuristic value
    - Plan for each explored node

<table>
<thead>
<tr>
<th>Soft Const</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 \leq 3$</td>
<td>2</td>
</tr>
<tr>
<td>$X_2 = X_3$</td>
<td>3</td>
</tr>
<tr>
<td>$X_2 \leq X_1$</td>
<td>4</td>
</tr>
<tr>
<td>$X_3 \neq 5$</td>
<td>5</td>
</tr>
</tbody>
</table>
**Handling Trajectory Constraints**

- **General difficulties**
  - Huge complexity to enumerate over intermediate states
  - Difficult to detect violation of constraints beforehand
  - Need support from basic planner
- **Approach**
  - Adapt partition-and-resolve strategy and basic solver for trajectory constraints

**Handling Soft Trajectory Constraints**

- **Approach: relax-and-tighten**
  - Assign a penalty to each soft constraint
  - Initially set all penalties to zero
  - Call solver with modified heuristic function
  - Augment penalties of violated constraints
Subgoal-level planning

- Modified Metric-FF for PDDL3
  - Support temporal features and partitioned approach (done in IPC4)
  - New search heuristic MCDC, similar to that in Fast Downward
  - Penalize actions whose relaxed plan violates trajectory constraints

- Search-space reduction
  - Eliminate irrelevant actions that only relate to facts or subgoals in other subproblems (done in IPC4)
  - Prioritize actions that do not cause inconsistencies of bottleneck state variables (variable ordering)

- Subgoal-level decomposition techniques
  - Necessary if subproblem is too complicated
  - Incremental planning on subproblems with multiple subgoals

IPC5 Results (2006)

- SGPlan5 solved 871/978 (89%) of the benchmarks
- 1st prize in satisficing planners

Table 4: Number of instances solved by SGPlan5 with respect to the total number of instances (in parenthesis) in each domain variant. (— means no instances in that domain.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TTP</td>
<td>30(30)</td>
<td>79(80)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>18(30)</td>
<td>187(200)</td>
</tr>
<tr>
<td>OpenStacks</td>
<td>30(30)</td>
<td>40(40)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>—</td>
<td>—</td>
<td>110(110)</td>
</tr>
<tr>
<td>Trucks</td>
<td>28(30)</td>
<td>30(30)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>138(140)</td>
</tr>
<tr>
<td>Storage</td>
<td>30(30)</td>
<td>30(30)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>9(30)</td>
<td>129(150)</td>
</tr>
<tr>
<td>Pathways</td>
<td>30(30)</td>
<td>30(30)</td>
<td>30(30)</td>
<td>—</td>
<td>30(30)</td>
<td>—</td>
<td>120(120)</td>
</tr>
<tr>
<td>Rovers</td>
<td>40(40)</td>
<td>32(40)</td>
<td>20(20)</td>
<td>20(20)</td>
<td>—</td>
<td>—</td>
<td>112(120)</td>
</tr>
<tr>
<td>PipesWorld</td>
<td>30(50)</td>
<td>30(50)</td>
<td>—</td>
<td>—</td>
<td>15(18)</td>
<td>0(20)</td>
<td>75(138)</td>
</tr>
</tbody>
</table>
Comparison with Other IPC5 Planners

<table>
<thead>
<tr>
<th>Top Planners in IPC5 [46]</th>
<th>1\textsuperscript{st}/2\textsuperscript{nd} Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGPlan\textsubscript{5} [70]</td>
<td>32/3  35</td>
</tr>
<tr>
<td>Downward [58]</td>
<td>1/4  5</td>
</tr>
<tr>
<td>MIPS-BDD [34, 33]</td>
<td>0/1  1</td>
</tr>
<tr>
<td>MIPS-XXL [37, 36]</td>
<td>0/14 14</td>
</tr>
<tr>
<td>HPlan-P [7]</td>
<td>0/5  5</td>
</tr>
<tr>
<td>YochanPS [10, 30]</td>
<td>1/8  9</td>
</tr>
</tbody>
</table>

Conclusions

• Constraint partitioning is a powerful approach for exploiting constraint structure and for reducing complexity
  • Bottom-up resolution with guidance provided by top-level active global constraints
  • Using existing solvers to solve partitioned subproblems
• Planning with trajectory constraints and goal preferences
  • New partitioning strategy from MDF analysis: partitioning attributes and granularity control
  • Handling soft goals: enumeration or DFS
  • Heuristics for trajectory constraints
  • Integration of techniques
Questions?