

***“Taming Numbers and Durations
in the
Model Checking Integrated Planning System***

by Stefan Edelkamp

David “Dave” Dagon

dagon@cc.gatech.edu

College o’ Computing
Georgia Institute of Technology

- New paper (from PuK2002)
- Available:
<http://www-is.informatik.uni-oldenburg.de/sauer/puk2002/papers/edelkamp.pdf>
- Model Checking Integrated Planning System (MIPS)
 - MIPS History:
 - AIPS-2000 Competition
 - AIPS-2002 Competition

- Early origins in heuristic symbolic exploration tools
- Used StaticBdd and Buddy C++ BDD package
- Added a parser and static analyzer to cluster atoms and reduce encoding
- 2002 planning competition (resulted in PDDL 2.1 and actions with durations)

Competition Categories

- 2002 Competition included:
 - Pure propositional planning
 - Planning with numerical resources
 - Planning with numerical resources and constant action duration
 - Planning with numerical resource, and variable action duration
 - Complex problems
- MIPS produced results in all 5 tracks.

Paper Contributions

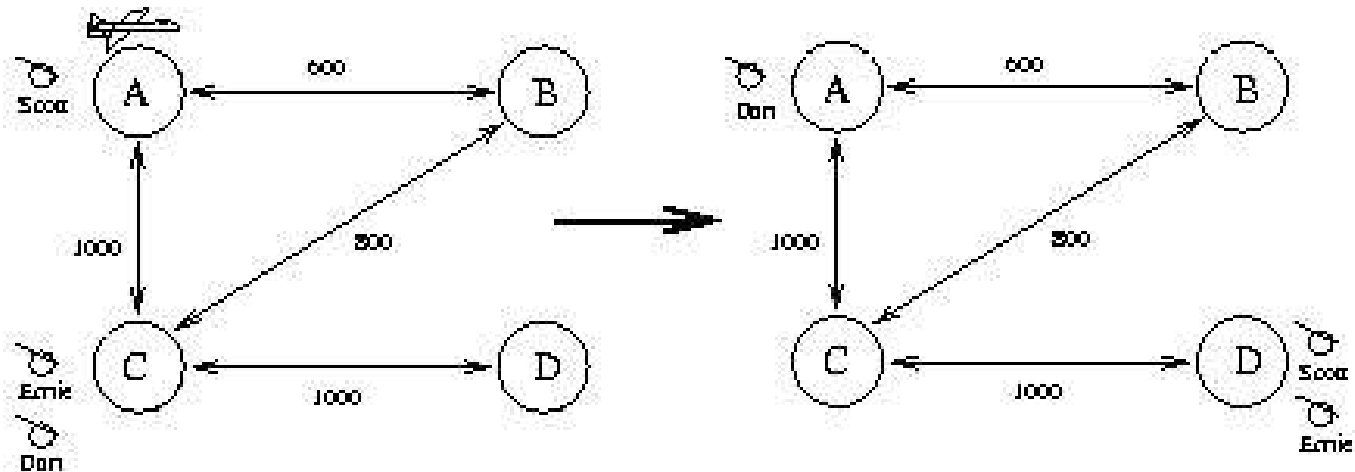
- Formal definitions of grounded propositional and numerical planning
- OO framework
- static analysis of data resources, symmetries, etc.
- temporal planning
- Graph viz tool

- Sets and Indices

Set	Descriptor	Examples
<i>OBJ</i>	objects	dan, city-a, plane
<i>TYPE</i>	object types	aircraft, person
<i>PRED</i>	predicates	(at ?a ?c), (in ?p ?a)
<i>FUNC</i>	numerical functions	(fuel ?a)
<i>ACT</i>	paramaterized actions	(board ?a ?p)
<i>IACT</i>	instantiated actions	(board plane scott)
$\mathcal{O} \subseteq IACT$	fluent operators	(board plane scope)
<i>IPRED</i>	instantiated preducates	(at plane city-b)
$\mathcal{F} \subseteq IPRED$	fluents	(at plane city-b)
<i>IFUNC</i>	instantiated functions	(distance city-a city-b)
$\mathcal{V} \subseteq IFUNC$	variables	(fuel plane), (total-time)

Example (Zeno Travel)

- Zeno travel problem

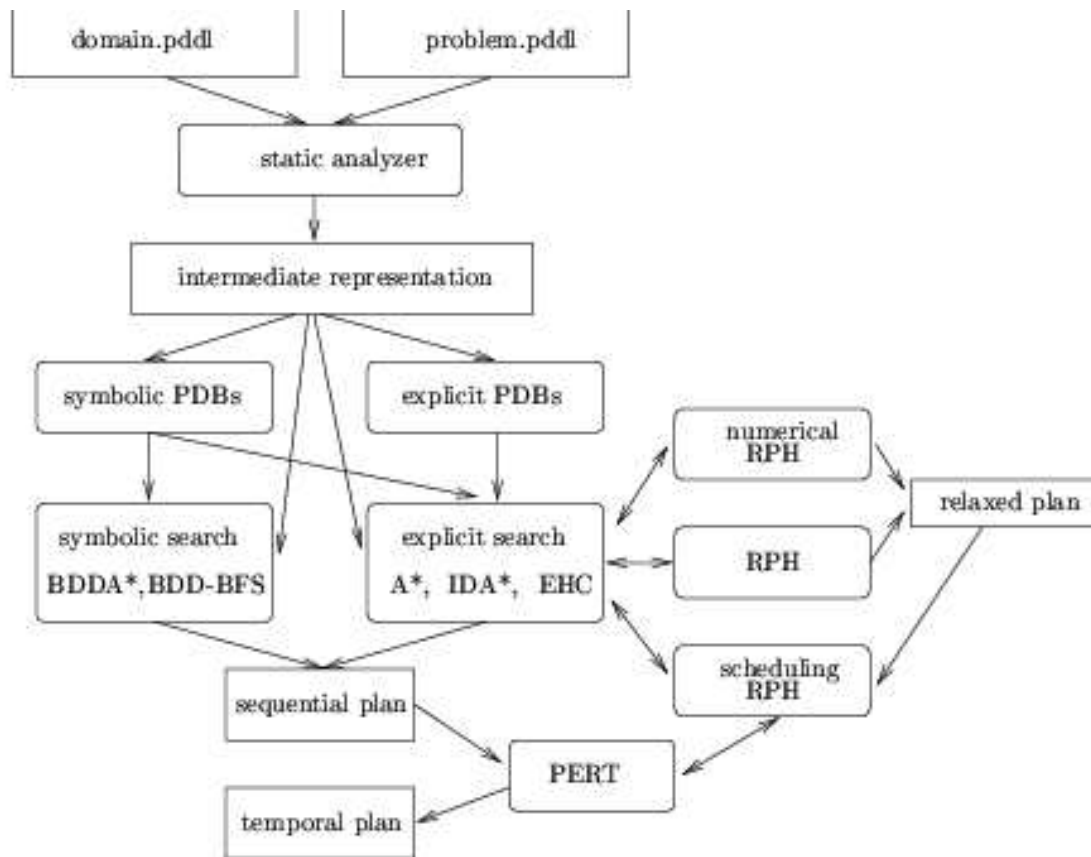


- (Show example of PDDL2.1 syntax)

- Static analysis phase examines domain and problem
- Performs efficient indexing (merge common states)
- Flattens temporal identifiers (merge timed prop. preconditions)
- Constants and fluents sorted out
- Grouping atoms and symmetries detected (efficient encoding)

- Heuristics
 - Relaxed planning heuristic (RPH) - Approx. number of steps to solve propositional plan; returns operators in relaxed plan
 - Numerical RPH - numerical extension; forward and backward approximation; multiple operator application
 - Pattern databases heuristic (PDB) - heuristic estimations; efficiently stored
 - Symbolic PDB - boolean representations of heuristic estimates
 - Scheduling RPH (or SRPH) - critical path analysis

- Diagram:



Exploration Algorithms

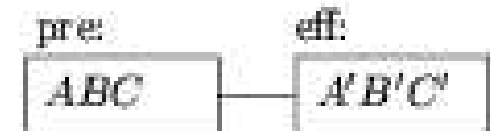
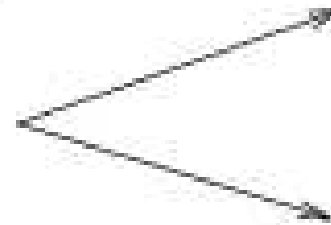
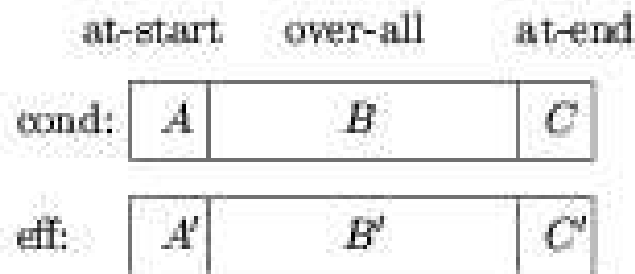
- Exploration Algorithms Include:
 - Weighted A*
 - Weighted Iterative-Deepening A*, (W)IDA*
 - Enforce Hill Climbing
 - Bidirectional Symbolic Breadth-First-Search (BDD-BFS)
 - Weighted Symbolic A*, BDDA*
 - Weak and Strong Planning

Temporal Planning

- PDDL2.1 has temporal modifiers:

at start preconditions and effects at invocation
over all invariable condition
at end finalization conditions

- Compiled:



- Thus, MIPS flattens temporal problem into planning with preconditions and effects.

Temporal Analysis

- Goal of planning competition was minimizing sequential plan length
- Operator Dependency – detects mutex/dependency relations among operators, to detect sequential tasks.
- Gives formal definition to precedence ordering
- If all operators are dependent, the problem is inherently sequential; no schedule is more optimal.
- Critical path analysis, using Project Evaluation and Review Technique (PERT)

- Critical Path Analysis

Procedure *Critical-Path*

Input: Sequence of operators O_1, \dots, O_k , precedence ordering \preceq_d

Output: Optimal parallel plan length $\max\{t_i + d(O_i) \mid O_i \in \{O_1, \dots, O_k\}$

for all $i \in \{1, \dots, k\}$

$e(O_i) = d(O_i)$

for all $j \in \{1, \dots, i - 1\}$

if $(O_j \preceq_d O_i)$

if $e(O_i) < e(O_j) + d(O_i)$

$e(O_i) \leftarrow e(O_j) + d(O_i)$

return $\max_{1 \leq i \leq k} e(O_i)$

- $e(O_i)$ represents the earliest end-of-time operator
- Variation on Dijkstra's single source shortest path algorithm

- MIPS also includes a visualization tool

- OO Framework for planning.
- forward chaining, heuristic, symbolic search planner
- Produces optimized parallel plans
- Temporal planning: sequential (totally ordered plans) optimized without solving NP-hard problems.