Planning as $X$

$X \in \{\text{SAT, CSP, ILP, ...}\}$

José Luis Ambite*

Complexity of Planning

- Domain-independent planning: PSPACE-complete or worse
  - (Chapman 1987; Bylander 1991; Backstrom 1993, Erol et al. 1994)

- Bounded-length planning: NP-complete
  - (Chenoweth 1991; Gupta and Nau 1992)

- Approximate planning: NP-complete or worse
  - (Selman 1994)
Compilation Idea

- Use any computational substrate that is (at least) NP-hard.

- Planning as:
  - SAT: Propositional Satisfiability
    - OBDD: Ordered Binary Decision Diagrams (Cimatti et al, 98)
  - CSP: Constraint Satisfaction
    - GP-CSP (Do & Kambhampati 2000)
  - ILP: Integer Linear Programming
    - Kautz & Walser 1999, Vossen et al 2000
  - ...

Planning as SAT

- Bounded-length planning can be formalized as propositional satisfiability (SAT)
- Plan = model (truth assignment) that satisfies logical constraints representing:
  - Initial state
  - Goal state
  - Domain axioms: actions, frame axioms, ...
  for a fixed plan length
- Logical spec such that any model is a valid plan
Architecture of a SAT-based planner

Problem Description
- Init State
- Goal State
- Actions

Compiler (encoding) → CNF → Simplifier (polynomial inference) → CNF

Increment plan length if unsatisfiable

Plan ← Decoder ← Simplifier

Solver (SAT engine/s)

Satisfying model
Parameters of SAT-based planner

- Encoding of Planning Problem into SAT
  - Frame Axioms
  - Action Encoding
- General Limited Inference: Simplification
- SAT Solver(s)
Encodings of Planning to SAT

- **Discrete Time**
  - Each proposition and action have a time parameter:
    - drive(truck1 a b) $\sim>$ drive(truck1 a b 3)
    - at(p a) $\sim>$ at(p a 0)

- **Common Axiom schemas:**
  - INIT: Initial state completely specified at time 0
  - GOAL: Goal state specified at time N
  - A $\Rightarrow$ P,E: Action implies preconditions and effects

- Don’t forget: propositional model!
  - drive(truck1 a b 3) = drive_truck1_a_b_3
Encodings of Planning to SAT
Common Schemas Example

- **INIT:** on(a b 0) ^ clear(a 0) ^ ...

- **GOAL:** on(a c 2)

- **A => P, E**

  Move(x y z)
  
  pre: clear(x) ^ clear(z) ^ on(x y)
  
  eff: on(x z) ^ not clear(z) ^ not on(x y)

  Move(a b c 1) => clear(a 0) ^ clear(b 0) ^ on(a b 0)
  Move(a b c 1) => on(a c 2) ^ not clear(a 2) ^
  
  not clear(b 2)
Encodings of Planning to SAT
Frame Axioms

[Ernst et al, IJCAI 1997]

- Classical: (McCarthy & Hayes 1969)
  - state what fluents are left unchanged by an action
  - clear(d i-1) ^ move(a b c i) => clear(d i+1)
  - Problem: if no action occurs at step i nothing can be inferred about propositions at level i+1
  - Sol: at-least-one axiom: at least one action occurs

- Explanatory: (Haas 1987)
  - State the causes for a fluent change
  - clear(d i-1) ^ not clear(d i+1) =>
    (move(a b d i) v move(a c d i) v ... move(c Table d i))
Encodings of Planning to SAT
Situation Calculus

- **Successor state axioms:**
  \[
  \text{At}(P_1 \text{ JFK } 1) \leftrightarrow \left[ \text{At}(P_1 \text{ JFK } 0) \land \neg \text{Fly}(P_1 \text{ JFK SFO } 0) \land \right.
  \left. \neg \text{Fly}(P_1 \text{ JFK LAX } 0) \land \ldots \right] \lor
  \text{Fly}(P_1 \text{ SFO JFK } 0) \lor \text{Fly}(P_1 \text{ LAX JFK } 0)
  \]

- **Preconditions axioms:**
  \[
  \text{Fly}(P_1 \text{ JFK SFO } 0) \rightarrow \text{At}(P_1 \text{ JFK } 0)
  \]

- **Excellent book on situation calculus:**
## Action Encoding

**Representation** | **One Propositional Variable per** | **Example**
--- | --- | ---
Simply-split | fully-instantiated action’s argument | Paint-Arg1-A ∧ Paint-Arg2-Red
Overloaded-split | fully-instantiated argument | Act-Paint ∧ Arg1-A ∧ Arg2-Red
Bitwise | Binary encodings of actions | Bit1 ∧ ~Bit2 ∧ Bit3 (Paint-A-Red = 5)
Encoding Sizes [Ernst et al, IJCAI 1997]

<table>
<thead>
<tr>
<th>Vars</th>
<th>Unfactored</th>
<th>Simple</th>
<th>Factored</th>
<th>Unfactored</th>
<th>Overloaded</th>
<th>Factored</th>
<th>Bitwise</th>
</tr>
</thead>
<tbody>
<tr>
<td>$nF+nA$</td>
<td>$nF+n\overline{Ops}A_0\overline{Dom}$</td>
<td>$nF+n\overline{Ops}A_0\overline{Dom}$</td>
<td>$nF+n(\overline{Ops}+A_0\overline{Dom})$</td>
<td>$nF+n(\overline{Ops}+A_0\overline{Dom}+1)$</td>
<td>$nF+n\log_2 A$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Classical

| $nF\overline{A}$ | AT-LEAST-ONE | $O(nF\overline{A}A_0)$ | $O(nF\overline{A}A_0)$ | $O(nF\overline{A}A_0+n\overline{Ops}\overline{Dom}^2A_0)$ | $O(nF\overline{A}A_0+n\overline{Dom}^2A_0)$ | $O(nF\overline{A}\log_2 A)$ |
| $nA_0^A, A$ | AT-LEAST-ONE, NO-PARTIAL | $O(nF\overline{A}A_0+n\overline{Ops}\overline{Dom}^2A_0)$ | $O(nF\overline{A}A_0+n\overline{Dom}^2A_0)$ |

Explanatory

| $nF\overline{A}$ | EXCLUSION | $O(nF\overline{A}A_0A^2)$ | $O(nF\overline{A}A_0A^2)$ | $O(nF\overline{A}A_0A^2)$ | $O(nF\overline{A}A_0A^2)$ | $O(nF\overline{A}\log_2 A)^4)$ |
| $nA_0^2$ | EXCLUSION, NO-PARTIAL | $O(nF\overline{A}A_0A^2)$ | $O(nF\overline{A}A_0A^2)$ |

Figure 4: Composition and worst case size of the encodings. The bitwise action representation yields the smallest number of variables, but the most clauses; regular actions are the exact opposite. All encodings INT, GOAL, A⇒P,E, and FRAME axioms. Any additional clauses are noted, and the total size for all clauses is given. The reported numbers are asymptotic numbers of literals (i.e., the product of numbers of clauses and clause sizes).

- $|Ops|$ number of operators
- $|Pred|$ number of predicate symbols
- $|Dom|$ number of constants in the domain
- $n$ number of odd time steps in plan (may be $< \text{plan length}$)
- $A_p$ max arity of predicates
- $A_o$ max arity of operators
- $A_r$ length of action representation (predicate symbols per action): regular = 1; simple split = $A_o$; overloaded split = $A_o+1$; bitwise = $\log_2 A$
- $A = |Ops||Dom|A_0$ number of ground actions
- $F = |Pred||Dom|^A_p$ number of ground fluents
- $P_o = O(F)$ max num fluents mentioned in operator

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Action Representation</th>
<th>Clauses</th>
<th>Clause size</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT</td>
<td>All</td>
<td>$F$</td>
<td>1</td>
</tr>
<tr>
<td>GOAL</td>
<td>All</td>
<td>arbitrary formula, typically small</td>
<td>$A_r+1$</td>
</tr>
<tr>
<td>A⇒P,E</td>
<td>All</td>
<td>$O(nF\overline{A})$</td>
<td>$A_r+2$</td>
</tr>
<tr>
<td>FRAME</td>
<td>Classical</td>
<td>$O(nF\overline{A})$</td>
<td>$A_r+2$</td>
</tr>
<tr>
<td></td>
<td>Explanatory</td>
<td>$O(nF\overline{A}^4)$</td>
<td>$A_r+2$</td>
</tr>
<tr>
<td></td>
<td>AT-LEAST-ONE</td>
<td>$O(n)$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>Overloaded factored</td>
<td>$O(n)$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>All other representations</td>
<td>$O(nA_0\overline{A}^4)$</td>
<td>$A_r+2$</td>
</tr>
</tbody>
</table>

| EXCLUSION | Simple factored | $O(n|Ops|\overline{Ops}+A_0\overline{Dom})$ | $A_r+2$ |
|           | Overloaded factored | $O(n|Ops|^2+A_0\overline{Dom})$ | $A_r+2$ |
|           | All other representations | $O(nA_0\overline{A}^4)$ | $A$ |

| NO-PARTIAL | Simple factored: | $O(n|Ops|\overline{Dom}|A_0)$ | $|Dom|+1$ |
|            | Overloaded Factored: | $O(n|Dom|A_0)$ | $|Dom|+1$ |
Encodings: Linear (sequential)

- Same as KS92
- Initial and Goal States
- Action implies both preconditions and its effects
- Only one action at a time
- Some action occurs at each time (allowing for do-nothing actions)
- Classical frame axioms
- Operator Splitting
[Kautz & Selman AAAI 96] Encodings: Graphplan-based

- Goal holds at last layer (time step)
- Initial state holds at layer 1
- Fact at level $i$ implies disjunction of all operators at level $i-1$ that have it as an add-effect
- Operators imply their preconditions
- Conflicting Actions (only action mutex explicit, fact mutex implicit)
Graphplan Encoding

Fact $\Rightarrow$ Act1 $\lor$ Act2
Act1 $\Rightarrow$ Pre1 $\land$ Pre2
$\neg$Act1 $\lor$ $\neg$Act2
[Kautz & Selman AAAI 96] Encodings: State-based

- Assert conditions for valid states
- Combines graphplan and linear
- Action implies both preconditions and its effects
- Conflicting Actions (only action mutex explicit, fact mutex implicit)
- Explanatory frame axioms
- Operator splitting
- Eliminate actions ($\rightarrow$ state transition axioms)
Algorithms for SAT

- Systematic (Complete: prove sat and unsat)
  - Davis-Putnam (1960)
  - DPLL (Davis Logemann Loveland, 1962)
  - Satz (Li & Anbulagan 1997)
  - Rel-Sat (Bayardo & Schrag 1997)
  - Chaff (Moskewicz et al 2001; Zhang&Malik CADE 2002)

- Stochastic (incomplete: cannot prove unsat)
  - GSAT (Selman et al 1992)
  - Walksat (Selman et al 1994)

- Randomized Systematic
  - Randomized Restarts (Gomes et al 1998)
DPPL Algorithm [Davis (Putnam) Logemann Loveland, 1962]

Procedure DPLL($\varphi$: CNF formula)

If $\varphi$ is empty return yes
Else if there is an empty clause in $\varphi$ return no
Else if there is a pure literal $u$ in $\varphi$
    return DPLL($\varphi(u)$)
Else if there is a unit clause $\{u\}$ in $\varphi$
    return DPLL($\varphi(u)$)
Else
    Choose a variable $v$ mentioned in $\varphi$
    If DPLL($\varphi(v)$) yes then return yes
    Else return DPLL($\varphi(\neg v)$)

[\varphi(u) \text{ means "set } u \text{ to true in } \varphi \text{ and simplify" } ]
Walksat

For $i=1$ to max-tries
    $A := \text{random truth assignment}$
    For $j=1$ to max-flips
        If solution?(A) then return A else
            $C := \text{random unsatisfied clause}$
            With probability $p$ flip a random variable in $C$
            With probability $(1- p)$ flip the variable in $C$
                that minimizes number of unsatisfied clauses
General Limited Inference
Formula Simplification

- Generated wff can be further simplified by consistency propagation techniques

- Compact (Crawford & Auton 1996)
  - unit propagation: $O(n)$ $P \land \neg P \lor Q \Rightarrow Q$
  - failed literal rule $O(n^2)$
    - if $Wff + \{ P \}$ unsat by unit propagation, then set $p$ to false
  - binary failed literal rule: $O(n^3)$
    - if $Wff + \{ P, Q \}$ unsat by unit propagation, then add $(\neg p \lor \neg q)$

- Experimentally reduces number of variables and clauses by 30% (Kautz & Selman 1999)
## General Limited Inference

<table>
<thead>
<tr>
<th>Problem</th>
<th>Vars</th>
<th>Percent vars set by</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>unit prop</td>
</tr>
<tr>
<td>bw.a</td>
<td>2452</td>
<td>10%</td>
</tr>
<tr>
<td>bw.b</td>
<td>6358</td>
<td>5%</td>
</tr>
<tr>
<td>bw.c</td>
<td>19158</td>
<td>2%</td>
</tr>
<tr>
<td>log.a</td>
<td>2709</td>
<td>2%</td>
</tr>
<tr>
<td>log.b</td>
<td>3287</td>
<td>2%</td>
</tr>
<tr>
<td>log.c</td>
<td>4197</td>
<td>2%</td>
</tr>
<tr>
<td>log.d</td>
<td>6151</td>
<td>1%</td>
</tr>
</tbody>
</table>
Randomized Systematic Solvers

- Stochastic local search solvers (Walksat)
  - when they work, scale well
  - cannot show unsat
  - fail on some domains

- Systematic solvers (Davis Putnam)
  - complete
  - seem to scale badly

- Can we combine best features of each approach?
Cost Distributions

- Consider *distribution of running times of backtrack search on a large set of “equivalent” problem instances*
  - renumber variables
  - change random seed used to break ties

- *Observation (Gomes 1997): distributions often have heavy tails*
  - infinite variance
  - mean increases without limit
  - probability of long runs decays by power law (Pareto-Levy), rather than exponentially (Normal)
Heavy Tails

- Bad scaling of systematic solvers can be caused by heavy tailed distributions
- Deterministic algorithms get stuck on particular instances
  - but that same instance might be easy for a different deterministic algorithm!
- Expected (mean) solution time increases without limit over large distributions
Heavy-Tailed Distributions
Erratic Mean Cost Behavior
Randomized systematic solvers

- Add noise to the heuristic branching (variable choice) function
- Cutoff and restart search after a fixed number of backtracks
  → Provably Eliminates heavy tails
- *In practice: rapid restarts with low cutoff can dramatically improve performance*
Rapid Restart Behavior
Increased Predictability

![Graph showing log solution time for different rocket and log solutions]

- **Satz**
- **Satz/Rand**

The graph illustrates the log solution time for various rocket and log solutions, with different lines representing Satz and Satz/Rand.
blackbox version 9B
command line:  blackbox -o logistics.pddl -f logistics_prob_d_len.pddl
   -solver compact -l -then satz -cutoff 25 -restart 10

----------------------------------------
Converting graph to wff
6151 variables
243652 clauses
Invoking simplifier compact
Variables undetermined: 4633
Non-unary clauses output: 139866
----------------------------------------
Invoking solver satz version satz-rand-2.1
Wff loaded
[1] begin restart
[1] reached cutoff 25 --- back to root
[2] begin restart
[2] reached cutoff 25 --- back to root
[3] begin restart
[3] reached cutoff 25 --- back to root
[4] begin restart
[4] reached cutoff 25 --- back to root
[5] begin restart
**** the instance is satisfiable ****
**** verification of solution is OK ****

total elapsed seconds = 25.930000
----------------------------------------
Begin plan
1 drive-truck.ny-truck.ny-central.ny-po.ny
...
Begin plan
1 drive-truck_ny-truck_ny-central.ny-po.ny
1 drive-truck_sf-truck_sf-airport_sf-po sf
1 load-truck_package5_bos-truck_bos-po
1 drive-truck_pgh-truck_pgh-airport_pgh-central.pgh
1 fly-airplane_airplane2_pgh-airport_sf-airport
1 load-truck_package6_bos-truck_bos-po
2 load-truck_package2_pgh-truck_pgh-pgh-central
2 load-truck_package4_ny-truck_ny-po
2 load-truck_package7_ny-truck_ny-po
2 drive-truck_bos-truck_bos-po_bos-airport_bos
2 load-airplane_package8_airplane2_sf-airport
2 drive-truck_la-truck_la-po_la-airport_la
3 fly-airplane_airplane2_sf-airport
3 unload-truck_package5_bos-truck_bos-airport
3 drive-truck_pgh-truck_pgh-central.pgh
3 fly-airplane_airplane1_pgh-airport_sf-airport
4 unload-truck_package3_pgh-truck_pgh-airport
4 unload-truck_package2_pgh-truck_pgh-airport
4 unload-truck_package4_ny-truck_ny-airport
4 load-airplane_package6_airplane2_bos-airport
4 drive-truck_la-truck_la-airport_la-po_la
4 drive-truck_bos-truck_bos-airport_bos-central_bos
4 unload-truck_package7_ny-truck_ny-airport
5 drive-truck_ny-truck_ny-po.ny
5 drive-truck_sf-truck_sf-po sf
5 drive-truck_la-truck_la-po_la-central_la
5 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
5 load-airplane_package2_airplane1_pgh-airport
5 fly-airplane_airplane2_bos-airport
6 drive-truck_sf-truck_sf-airport_sf-po sf
6 unload-airplane_package4_airplane2_sf-airport
6 load-airplane_package6_airplane2_sf-airport
6 drive-truck_la-truck_la-central_la-po_la
6 drive-truck_bos-truck_bos-po_bos-airport_bos
6 load-airplane_package7_airplane2_ny-airport
6 drive-truck_ny-truck_ny-po.ny
6 unload-airplane_package8_airplane2_ny-airport
6 fly-airplane_airplane1_pgh-airport_sf-airport
6 load-truck_package1_pgh-truck_pgh-po
7 fly-airplane_airplane2_ny-airport
7 load-truck_package9_sf-truck_sf-central
7 load-truck_package6_ny-truck_ny-airport
7 drive-truck_bos-truck_bos-airport_bos-central_bos
7 drive-truck_pgh-truck_pgh-po_pgh-airport_pgh
7 load-truck_package8_ny-truck_ny-airport
8 drive-truck_sf-truck_sf-central sf-po sf
8 fly-airplane_airplane2_la-airport_pgh-airport
8 unload-truck_package1_pgh-truck_pgh-airport
8 drive-truck_bos-truck_bos-central_bos-po_bos
8 drive-truck_ny-truck_ny-airport_ny-central.ny
8 fly-airplane_airplane1_bos-airport_la-airport
9 drive-truck_la-truck_la-po_la-airport_la
9 drive-truck_bos-truck_bos-airport_bos-po_pgh
9 unload-airplane_package8_ny-truck_ny-central
9 unload-airplane_package5_airplane2_pgh-airport
9 unload-airplane_package9_sf-truck_sf-po
9 unload-airplane_package3_airplane1_la-airport
9 unload-truck_package6_ny-truck_ny-central
10 drive-truck_pgh-truck_pgh-po_pgh
10 drive-truck_la-truck_la-po_la-airport_la
10 drive-truck_bos-truck_bos-po_bos-airport_bos
10 drive-truck_ny-truck_ny-central.ny-po.ny
11 drive-truck_la-truck_la-airport_la-central_la
11 drive-truck_bos-truck_bos-airport_bos-po_bos
11 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
11 drive-truck_ny-truck_ny-po.ny
11 drive-truck_bos-truck_bos-central_bos
11 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
11 drive-truck_ny-truck_ny-airport_ny-central
11 drive-truck_bos-truck_bos-po_bos-airport_bos
11 drive-truck_ny-truck_ny-central
12 drive-truck_pgh-truck_pgh-po_pgh
12 drive-truck_ny-truck_ny-po.ny
12 drive-truck_bos-truck_bos-central_bos
12 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
12 drive-truck_ny-truck_ny-airport_ny-central
12 drive-truck_bos-truck_bos-po_bos-airport_bos
12 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
12 drive-truck_ny-truck_ny-airport
13 drive-truck_pgh-truck_pgh-po_pgh
13 drive-truck_ny-truck_ny-po.ny
13 drive-truck_bos-truck_bos-central_bos
13 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
13 drive-truck_ny-truck_ny-airport
13 drive-truck_bos-truck_bos-po_bos-airport_bos
13 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
14 drive-truck_pgh-truck_pgh-po_pgh
14 drive-truck_ny-truck_ny-po.ny
14 drive-truck_bos-truck_bos-central_bos
14 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
14 drive-truck_ny-truck_ny-airport
14 drive-truck_bos-truck_bos-po_bos-airport_bos
14 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
End plan
Blackbox Results

Graph of the results for different blackbox algorithms:

- Graphplan
- BB-walksat
- BB-rand-sys
- Handcoded-walksat

The graph shows the performance of these algorithms on various problems labeled as rocket.a, rocket.b, log.a, log.b, log.c, and log.d.

Key metrics:
- 10^16 states
- 6,000 variables
- 125,000 clauses
Planning as CSP

Constraint-satisfaction problem (CSP)

Given:
- set of discrete variables,
- domains of the variables, and
- constraints on the specific values a set of variables can take in combination,

Find an assignment of values to all the variables which respects all constraints

- Compile the planning problem as a constraint-satisfaction problem (CSP)
- Use the planning graph to define a CSP
Representing the Planning Graph as a CSP

(a) Planning Graph

(b) DCSP

Variables: $G_1, \ldots, G_4, P_1 \ldots P_6$

Domains: $G_1: \{A_1\}, G_2: \{A_2\} G_3: \{A_3\} G_4: \{A_4\}$  
$P_1: \{A_5\} P_2: \{A_6, A_{11}\} P_3: \{A_7\} P_4: \{A_8, A_9\}$  
$P_5: \{A_{10}\} P_6: \{A_{10}\}$

Constraints (normal): $P_1 = A_5 \Rightarrow P_4 \neq A_9$  
$P_2 = A_6 \Rightarrow P_4 \neq A_8$  
$P_2 = A_{11} \Rightarrow P_3 \neq A_7$

Constraints (Activity): $G_1 = A_1 \Rightarrow Active\{P_1, P_2, P_3\}$  
$G_2 = A_2 \Rightarrow Active\{P_4\}$  
$G_3 = A_3 \Rightarrow Active\{P_5\}$  
$G_4 = A_4 \Rightarrow Active\{P_1, P_6\}$

Init State: $Active\{G_1, G_2, G_3, G_4\}$
Transforming a DCSP to a CSP

Variables: $G_1, \ldots, G_4, P_1 \ldots P_6$

Domain: $G_1: \{A_1\}, G_2: \{A_2\} G_3: \{A_3\} G_4: \{A_4\}$

$P_1: \{A_5\} P_2: \{A_6, A_{11}\} P_3: \{A_7\} P_4: \{A_8, A_9\}$

$P_5: \{A_{10}\} P_6: \{A_{10}\}$

Constraints (normal): $P_1 = A_5 \Rightarrow P_4 \neq A_9$

$P_2 = A_6 \Rightarrow P_4 \neq A_8$

$P_2 = A_{11} \Rightarrow P_3 \neq A_7$

Constraints (Activity): $G_1 = A_1 \Rightarrow \text{Active}\{P_1, P_2, P_3\}$

$G_2 = A_2 \Rightarrow \text{Active}\{P_4\}$

$G_3 = A_3 \Rightarrow \text{Active}\{P_5\}$

$G_4 = A_4 \Rightarrow \text{Active}\{P_1, P_6\}$

Init State: $\text{Active}\{G_1, G_2, G_3, G_4\}$

Variables: $G_1, \ldots, G_4, P_1 \ldots P_6$

Domain: $G_1: \{A_1, \bot\}, G_2: \{A_2, \bot\} G_3: \{A_3, \bot\} G_4: \{A_4, \bot\}$

$P_1: \{A_5, \bot\} P_2: \{A_6, A_{11}, \bot\} P_3: \{A_7, \bot\} P_4: \{A_8, A_9, \bot\}$

$P_5: \{A_{10}, \bot\} P_6: \{A_{10}, \bot\}$

Constraints (normal): $P_1 = A_5 \Rightarrow P_4 \neq A_9$

$P_2 = A_6 \Rightarrow P_4 \neq A_8$

$P_2 = A_{11} \Rightarrow P_3 \neq A_7$

Constraints (Activity): $G_1 = A_1 \Rightarrow P_1 \neq \bot \land P_2 \neq \bot \land P_3 \neq \bot$

$G_2 = A_2 \Rightarrow P_4 \neq \bot$

$G_3 = A_3 \Rightarrow P_5 \neq \bot$

$G_4 = A_4 \Rightarrow P_1 \neq \bot \land P_6 \neq \bot$

Init State: $G_1 \neq \bot \land G_2 \neq \bot \land G_3 \neq \bot \land G_4 \neq \bot$

(a) DCSP

(b) CSP
Compilation to CSP

CSP: Given a set of discrete variables, the domains of the variables, and constraints on the specific values a set of variables can take in combination, FIND an assignment of values to all the variables which respects all constraints

- Variables: Propositions (In-A-1, In-B-1, ..At-R-E-0 …)
- Domains: Actions supporting that proposition in the plan
  In-A-1 : { Load-A-1, #}
  At-R-E-1: {P-At-R-E-1, #}
- Constraints:
  - Mutual exclusion
    not [(In-A-1 = Load-A-1) & (At-R-M-1 = Fly-R-1)] ; etc..
  - Activation:
    In-A-1 != # & In-B-1 != # (Goals must have action assignments)
    In-A-1 = Load-A-1 => At-R-E-0 != #, At-A-E-0 != #
    (subgoal activation constraints)

[Do & Kambhampati, 2000]
CSP Encodings can be more compact: GP-CSP

<table>
<thead>
<tr>
<th>Problem</th>
<th>Graphplan</th>
<th>Satz</th>
<th>Relsat</th>
<th>GP-CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time (s)</td>
<td>mem</td>
<td>time(s)</td>
<td>mem</td>
</tr>
<tr>
<td>bw-12steps</td>
<td>0.42</td>
<td>1 M</td>
<td>8.17</td>
<td>64 M</td>
</tr>
<tr>
<td>bw-large-a</td>
<td>1.39</td>
<td>3 M</td>
<td>47.63</td>
<td>88 M</td>
</tr>
<tr>
<td>rocket-a</td>
<td>68</td>
<td>61 M</td>
<td>8.88</td>
<td>70 M</td>
</tr>
<tr>
<td>rocket-b</td>
<td>130</td>
<td>95 M</td>
<td>11.74</td>
<td>70 M</td>
</tr>
<tr>
<td>log-a</td>
<td>1771</td>
<td>177 M</td>
<td>7.05</td>
<td>72 M</td>
</tr>
<tr>
<td>log-b</td>
<td>787</td>
<td>80 M</td>
<td>16.13</td>
<td>79 M</td>
</tr>
<tr>
<td>hsp-bw-02</td>
<td>0.86</td>
<td>1 M</td>
<td>7.15</td>
<td>68 M</td>
</tr>
<tr>
<td>hsp-bw-03</td>
<td>5.06</td>
<td>24 M</td>
<td>&gt; 8 hs</td>
<td>-</td>
</tr>
<tr>
<td>hsp-bw-04</td>
<td>19.26</td>
<td>83 M</td>
<td>&gt; 8 hs</td>
<td>-</td>
</tr>
</tbody>
</table>

[Do & Kambhampati, 2000]
## GP-CSP Performance

<table>
<thead>
<tr>
<th>prob</th>
<th>GPCSP</th>
<th>Graphplan</th>
<th>Satz</th>
<th>Relsat</th>
<th>speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>prob</td>
<td>heu</td>
<td>time (s)</td>
<td>time (s)</td>
<td>time (s)</td>
</tr>
<tr>
<td>bw-12steps</td>
<td>dlc</td>
<td>0.63</td>
<td>0.17</td>
<td>3.96</td>
<td>1.60</td>
</tr>
<tr>
<td>bw-large-a</td>
<td>ldc</td>
<td>5.40</td>
<td>0.57</td>
<td>27.80</td>
<td>32.30</td>
</tr>
<tr>
<td>bw-large-b</td>
<td>ldc</td>
<td>661</td>
<td>71</td>
<td>&gt; 8hrs</td>
<td>901.55</td>
</tr>
<tr>
<td>rocket-a</td>
<td>dlc</td>
<td>1.22</td>
<td>43.13</td>
<td>3.81</td>
<td>5.27</td>
</tr>
<tr>
<td>rocket-b</td>
<td>dlc</td>
<td>2.33</td>
<td>87</td>
<td>5.91</td>
<td>8.39</td>
</tr>
<tr>
<td>log-a</td>
<td>dlc</td>
<td>0.95</td>
<td>842</td>
<td>2.88</td>
<td>1.11</td>
</tr>
<tr>
<td>log-b</td>
<td>dlc</td>
<td>19.10</td>
<td>390</td>
<td>7.73</td>
<td>22.03</td>
</tr>
<tr>
<td>log-c</td>
<td>dlc</td>
<td>24.27</td>
<td>&gt; 8hrs</td>
<td>308</td>
<td>77</td>
</tr>
<tr>
<td>log-d</td>
<td>dlc</td>
<td>84</td>
<td>&gt; 8hrs</td>
<td>15.99</td>
<td>199.38</td>
</tr>
<tr>
<td>hsp-bw-02</td>
<td>ldc</td>
<td>0.34</td>
<td>0.32</td>
<td>3.62</td>
<td>1.21</td>
</tr>
<tr>
<td>hsp-bw-03</td>
<td>ldc</td>
<td>1.63</td>
<td>2.14</td>
<td>&gt; 8hrs</td>
<td>130.77</td>
</tr>
<tr>
<td>hsp-bw-04</td>
<td>ldc</td>
<td>4.87</td>
<td>19.26</td>
<td>&gt; 8hrs</td>
<td>1682</td>
</tr>
<tr>
<td>grid-01</td>
<td>ldc</td>
<td>7.75</td>
<td>7.21</td>
<td>&gt; 8hrs</td>
<td>22.75</td>
</tr>
<tr>
<td>grid-02</td>
<td>ldc</td>
<td>6.36</td>
<td>5.30</td>
<td>&gt; 8hrs</td>
<td>21.45</td>
</tr>
<tr>
<td>grid-03</td>
<td>ldc</td>
<td>9.83</td>
<td>8.77</td>
<td>&gt; 8hrs</td>
<td>42.68</td>
</tr>
<tr>
<td>gripper-01</td>
<td>ldc</td>
<td>0.01</td>
<td>0.01</td>
<td>0.52</td>
<td>0.09</td>
</tr>
<tr>
<td>gripper-02</td>
<td>ldc</td>
<td>0.41</td>
<td>0.05</td>
<td>2.41</td>
<td>0.69</td>
</tr>
<tr>
<td>gripper-03</td>
<td>ldc</td>
<td>62</td>
<td>4.28</td>
<td>109.72</td>
<td>155.72</td>
</tr>
</tbody>
</table>
## GP-CSP Performance

<table>
<thead>
<tr>
<th>prob</th>
<th>GPCSP</th>
<th>Graphplan</th>
<th>Satz</th>
<th>ReliX</th>
<th>speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>prob</td>
<td>hele</td>
<td>time (s)</td>
<td>hele</td>
<td>time (s)</td>
<td>time (s)</td>
</tr>
<tr>
<td>low-1steps</td>
<td>dlc</td>
<td>0.63</td>
<td>0.17</td>
<td>3.06</td>
<td>1.60</td>
</tr>
<tr>
<td>low-large-a</td>
<td>lde</td>
<td>5.40</td>
<td>0.57</td>
<td>27.80</td>
<td>32.80</td>
</tr>
<tr>
<td>low-large-b</td>
<td>lde</td>
<td>691</td>
<td>71</td>
<td>&gt; 8hrs</td>
<td>900.65</td>
</tr>
<tr>
<td>rocket-a</td>
<td>dlc</td>
<td>1.22</td>
<td>3.13</td>
<td>3.81</td>
<td>5.27</td>
</tr>
<tr>
<td>rocket-b</td>
<td>dlc</td>
<td>2.33</td>
<td>87</td>
<td>5.61</td>
<td>8.29</td>
</tr>
<tr>
<td>lps-a</td>
<td>dlc</td>
<td>0.55</td>
<td>842</td>
<td>2.88</td>
<td>1.11</td>
</tr>
<tr>
<td>lps-b</td>
<td>dlc</td>
<td>20.09</td>
<td>390</td>
<td>7.73</td>
<td>22.03</td>
</tr>
<tr>
<td>lps-c</td>
<td>dlc</td>
<td>24.27</td>
<td>&gt; 8hrs</td>
<td>308</td>
<td>77</td>
</tr>
<tr>
<td>lps-d</td>
<td>dlc</td>
<td>84</td>
<td>&gt; 8hrs</td>
<td>15.99</td>
<td>199.88</td>
</tr>
<tr>
<td>lps-1</td>
<td>lde</td>
<td>0.34</td>
<td>0.32</td>
<td>3.02</td>
<td>1.21</td>
</tr>
<tr>
<td>lps-2</td>
<td>lde</td>
<td>1.63</td>
<td>2.14</td>
<td>&gt; 8hrs</td>
<td>139.77</td>
</tr>
<tr>
<td>lps-3</td>
<td>lde</td>
<td>1.67</td>
<td>18.26</td>
<td>&gt; 8hrs</td>
<td>1892.5</td>
</tr>
<tr>
<td>grid-61</td>
<td>lde</td>
<td>7.75</td>
<td>7.25</td>
<td>&gt; 8hrs</td>
<td>22.75</td>
</tr>
<tr>
<td>grid-62</td>
<td>lde</td>
<td>6.26</td>
<td>0.30</td>
<td>&gt; 8hrs</td>
<td>21.45</td>
</tr>
<tr>
<td>grid-63</td>
<td>lde</td>
<td>9.83</td>
<td>8.77</td>
<td>&gt; 8hrs</td>
<td>62.08</td>
</tr>
<tr>
<td>gripper-1</td>
<td>lde</td>
<td>0.05</td>
<td>0.06</td>
<td>0.32</td>
<td>0.09</td>
</tr>
<tr>
<td>gripper-2</td>
<td>lde</td>
<td>0.41</td>
<td>0.05</td>
<td>2.41</td>
<td>0.08</td>
</tr>
<tr>
<td>gripper-3</td>
<td>lde</td>
<td>0.82</td>
<td>0.82</td>
<td>1097.2</td>
<td>1567.2</td>
</tr>
<tr>
<td>hazmat-tower3</td>
<td>lde</td>
<td>0.19</td>
<td>0.04</td>
<td>1.96</td>
<td>0.32</td>
</tr>
<tr>
<td>hazmat-tower4</td>
<td>lde</td>
<td>0.68</td>
<td>0.45</td>
<td>12.58</td>
<td>54.69</td>
</tr>
<tr>
<td>hazmat-tower5</td>
<td>lde</td>
<td>0.06</td>
<td>0.12</td>
<td>&gt; 8hrs</td>
<td>&gt; 8hrs</td>
</tr>
<tr>
<td>bulldozer-1</td>
<td>lde</td>
<td>0.10</td>
<td>0.05</td>
<td>0.00</td>
<td>0.19</td>
</tr>
<tr>
<td>bulldozer-2</td>
<td>lde</td>
<td>0.11</td>
<td>0.09</td>
<td>0.01</td>
<td>0.19</td>
</tr>
<tr>
<td>bulldozer-3</td>
<td>lde</td>
<td>0.02</td>
<td>0.03</td>
<td>0.00</td>
<td>0.10</td>
</tr>
<tr>
<td>maritim-1</td>
<td>lde</td>
<td>0.53</td>
<td>0.55</td>
<td>1.22</td>
<td>0.80</td>
</tr>
<tr>
<td>maritim-2</td>
<td>lde</td>
<td>0.97</td>
<td>3.59</td>
<td>6.08</td>
<td>4.00</td>
</tr>
<tr>
<td>maritim-16</td>
<td>lde</td>
<td>3.58</td>
<td>3.17</td>
<td>6.08</td>
<td>4.25</td>
</tr>
<tr>
<td>mystery-2</td>
<td>lde</td>
<td>3.81</td>
<td>3.81</td>
<td>5.35</td>
<td>5.66</td>
</tr>
<tr>
<td>mystery-3</td>
<td>lde</td>
<td>0.20</td>
<td>0.43</td>
<td>0.00</td>
<td>0.41</td>
</tr>
<tr>
<td>mystery-25</td>
<td>lde</td>
<td>1.19</td>
<td>1.00</td>
<td>1.76</td>
<td>1.12</td>
</tr>
<tr>
<td>mystery-28</td>
<td>lde</td>
<td>4.05</td>
<td>0.34</td>
<td>2.78</td>
<td>2.07</td>
</tr>
<tr>
<td>mystery-39</td>
<td>lde</td>
<td>2.81</td>
<td>3.43</td>
<td>&gt; 8 hrs</td>
<td>&gt; 8 hrs</td>
</tr>
<tr>
<td>field-type-1</td>
<td>lde</td>
<td>0.12</td>
<td>0.12</td>
<td>0.09</td>
<td>0.19</td>
</tr>
<tr>
<td>field-type-2</td>
<td>lde</td>
<td>0.38</td>
<td>0.34</td>
<td>1.24</td>
<td>0.65</td>
</tr>
</tbody>
</table>