Planning as X
$X \in$ {SAT, CSP, ILP, …}

José Luis Ambite*

[* Some slides are taken from presentations by Kautz, Selman, Weld, and Kambhampati. Please visit their websites:
http://www.cs.washington.edu/homes/kautz/ http://www.cs.cornell.edu/home/selman/
http://www.cs.washington.edu/homes/weld/ http://rakaposhi.eas.asu.edu/rao.html
]

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

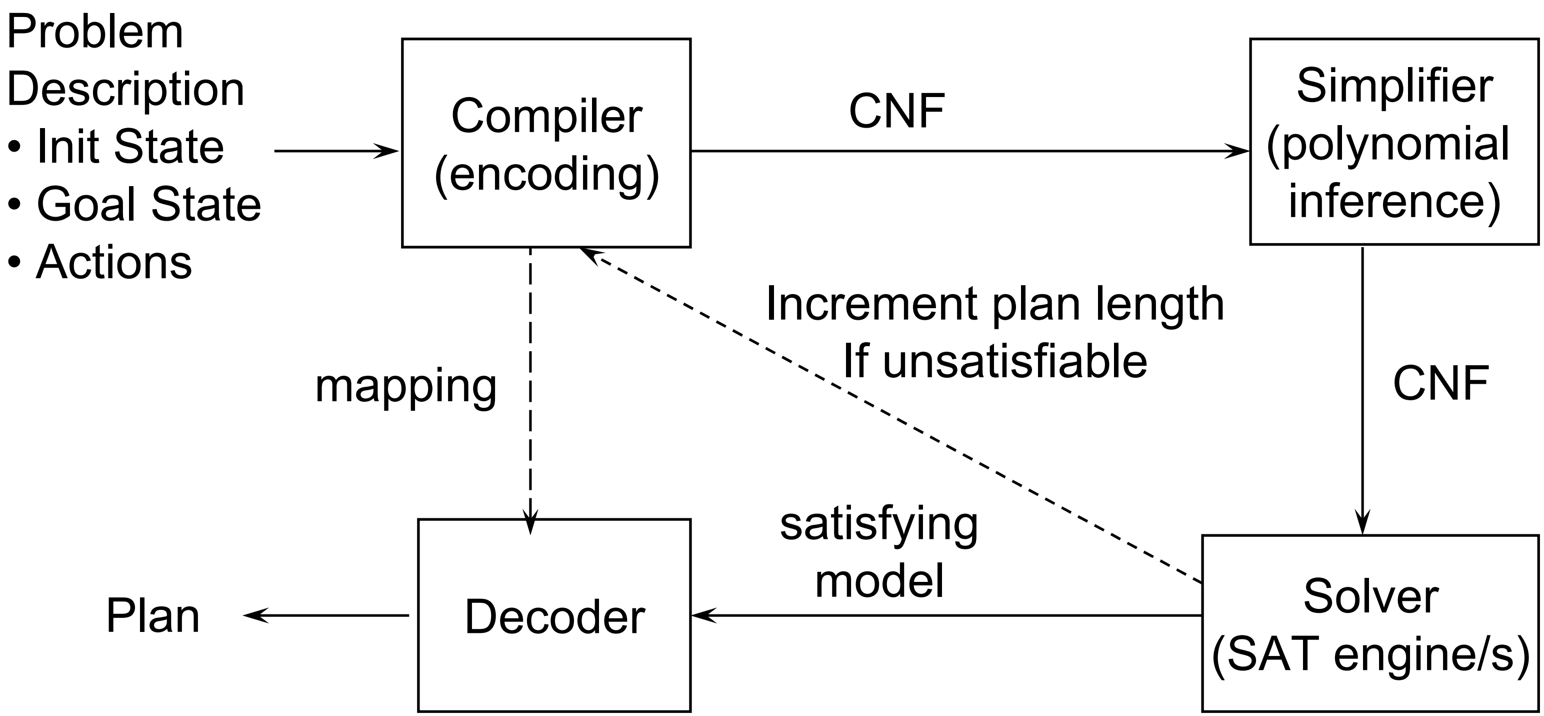
- Use any computational substrate that is (at least) NP-hard.
- Planning as:
  - SAT: Propositional Satisfiability
    - SATPLAN, Blackbox (Kautz&Selman, 1992, 1996, 1999)
    - OBDD: Ordered Binary Decision Diagrams (Cimatti et al, 98)
  - CSP: Constraint Satisfaction
    - GP-CSP (Do & Kambhampati 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



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) ~> drive(truck1 a b 3)
  - at(p a) ~> at(p a 0)
- Common Axiom schemas:
  - INIT: Initial state completely specified at time 0
  - GOAL: Goal state specified at time N
  - A => 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 [Ernst et al, IJCAI 1997]

- 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:

  $At(P_1\ JFK\ 1) \leftrightarrow [\ At(P_1\ JFK\ 0) \wedge \neg Fly(P_1\ JFK\ SFO\ 0) \wedge \neg Fly(P_1\ JFK\ LAX\ 0) \wedge \ldots\ ] \vee Fly(P_1\ SFO\ JFK\ 0) \vee Fly(P_1\ LAX\ JFK\ 0)$

- Preconditions axioms:

  $Fly(P_1\ JFK\ SFO\ 0) \rightarrow At(P_1\ JFK\ 0)$

- Excellent book on situation calculus:

  Reiter, "Logic in Action", 2001.

Action Encoding

[Ernst et al, IJCAI 1997]

Representation	One Propositional Variable per	Example
Regular	fully-instantiated action	Paint-A-Red, Paint-A-Blue, Move-A-Table
Simply-split	fully-instantiated action's argument	Paint-Arg1-A $\wedge$ Paint-Arg2-Red
Overloaded-split	fully-instantiated argument	Act-Paint $\wedge$ Arg1-A $\wedge$ Arg2-Red
Bitwise	Binary encodings of actions	Bit1 $\wedge$ ~Bit2 $\wedge$ Bit3 (*Paint-A-Red = 5*)

more vars

more clses

Encoding Sizes [Ernst et al, IJCAI 1997]

	Action representation					
	Regular	Simple		Overloaded		Bitwise
		Unfactored	Factored	Unfactored	Factored	
Vars	$n\mathcal{F}+n\mathcal{A}$	$n\mathcal{F}+n\|Ops\|A_o\|Dom\|$	$n\mathcal{F}+n\|Ops\|A_o\|Dom\|$	$n\mathcal{F}+n(\|Ops\|+A_o\|Dom\|)$	$n\mathcal{F}+n(\|Ops\|+A_o\|Dom\|+1)$	$n\mathcal{F}+n\log_2\mathcal{A}$
Classical	AT-LEAST-ONE $O(n\mathcal{F}\mathcal{A})$	AT-LEAST-ONE $O(n\mathcal{F}\mathcal{A}A_o + nA_o{}^{\mathcal{A}}\mathcal{A})$	AT-LEAST-ONE, NO-PARTIAL $O(n\mathcal{F}\mathcal{A}A_o + n\|Ops\|\|Dom\|^2A_o)$	AT-LEAST-ONE $O(n\mathcal{F}\mathcal{A}A_o + nA_o{}^{\mathcal{A}}\mathcal{A})$	AT-LEAST-ONE, NO-PARTIAL $O(n\mathcal{F}\mathcal{A}A_o + n\|Dom\|^2A_o)$	$O(n\mathcal{F}\mathcal{A}\log_2\mathcal{A})$
Explanatory	EXCLUSION $O(n\mathcal{F}\mathcal{A} + n\mathcal{A}^2)$	EXCLUSION $O(n\mathcal{F}A_o{}^{\mathcal{A}} + n(A_o\mathcal{A})^2)$	EXCLUSION, NO-PARTIAL $O(n\mathcal{F}A_o{}^{\mathcal{A}} + n\|Ops\|^2\|Dom\|^2A_o)$	EXCLUSION $O(n\mathcal{F}(\mathcal{A}A_o)^2 + n\mathcal{F}A_o{}^{\mathcal{A}}\mathcal{A})$	EXCLUSION, NO-PARTIAL $O(n\mathcal{F}A_o{}^{\mathcal{A}}\mathcal{A} + n\|Dom\|^2(A_o + \|Ops\|^2))$	$O(n\mathcal{F}(\log_2\mathcal{A})^{\mathcal{A}})$

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 INIT, 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 $<$ 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 $= \lceil \log_2 \mathcal{A} \rceil$
$\mathcal{A}$	$= \|Ops\|\|Dom\|^{A_o}$ number of ground actions
$\mathcal{F}$	$= \|Pred\|\|Dom\|^{A_p}$ number of ground fluents
P_o	$= O(\mathcal{F})$ max num fluents mentioned in operator

Axiom	Action Representation	Clauses	Clause size
INIT	All	$\mathcal{F}$	1
GOAL	All	arbitrary formula, typically small	
A⇒P,E	All	$O(nP_o\mathcal{A})$	$A_r + 1$
FRAME	Classical	$O(n\mathcal{F}\mathcal{A})$	$A_r + 2$
	Explanatory	$O(n\mathcal{F}A_r{}^{\mathcal{A}})$	$O(\mathcal{A})$
AT-LEAST-ONE	Simple factored	$O(n)$	$\|Ops\|\|Dom\|$
	Overloaded factored	$O(n)$	$\|Ops\|$
	All other representations	$O(nA_r{}^{\mathcal{A}})$	$\mathcal{A}$
EXCLUSION	Simple factored	$O(n\|Ops\|(\|Ops\| + A_o - 1)\|Dom\|^2)$	2
	Overloaded factored	$O(n(\|Ops\|^2 + A_o\|Dom\|^2))$	2
	All other representations	$O(n(A_r\mathcal{A})^2)$	2
NO-PARTIAL	Simple Factored:	$O(n\|Ops\|\|Dom\|A_o)$	$\|Dom\| + 1$
	Overloaded Factored:	$O(n\|Dom\|(A_o + 1))$	$\|Dom\| + 1$

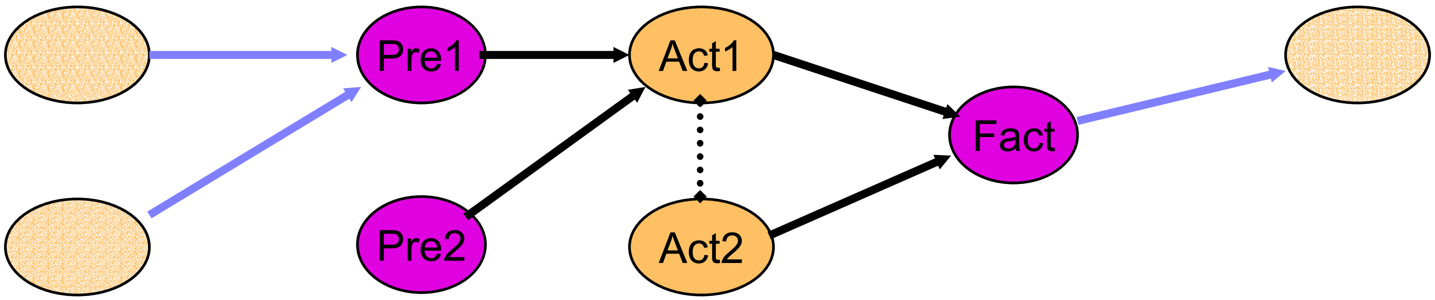
[Kautz & Selman AAAI 96] Encodings: Linear (sequential)

- 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 disjuntion of all operators at level i–1 that have it as an add-efffect
- Operators imply their preconditions
- Conflicting Actions (only action mutex explicit, fact mutex implicit)

Graphplan Encoding



$$Fact \Rightarrow Act1 \vee Act2$$

$$Act1 \Rightarrow Pre1 \wedge Pre2$$

$$\neg Act1 \vee \neg Act2$$

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

[φ(u) means "set u to true in φ and simplify"]

Walksat

For i=1 to max-tries

A:= random truth assigment

For j=1 to max-flips

If solution?(A) **then** return A **else**

C:= 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 ^ ~P v Q => 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 (not p V not q)
- Experimentally reduces number of variables and clauses by 30% (Kautz&Selman 1999)

General Limited Inference

Problem	Vars	Percent vars set by		
		unit prop	failed lit	binary failed
bw.a	2452	10%	100%	100%
bw.b	6358	5%	43%	99%
bw.c	19158	2%	33%	99%
log.a	2709	2%	36%	45%
log.b	3287	2%	24%	30%
log.c	4197	2%	23%	27%
log.d	6151	1%	25%	33%

Randomized Sytematic 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?

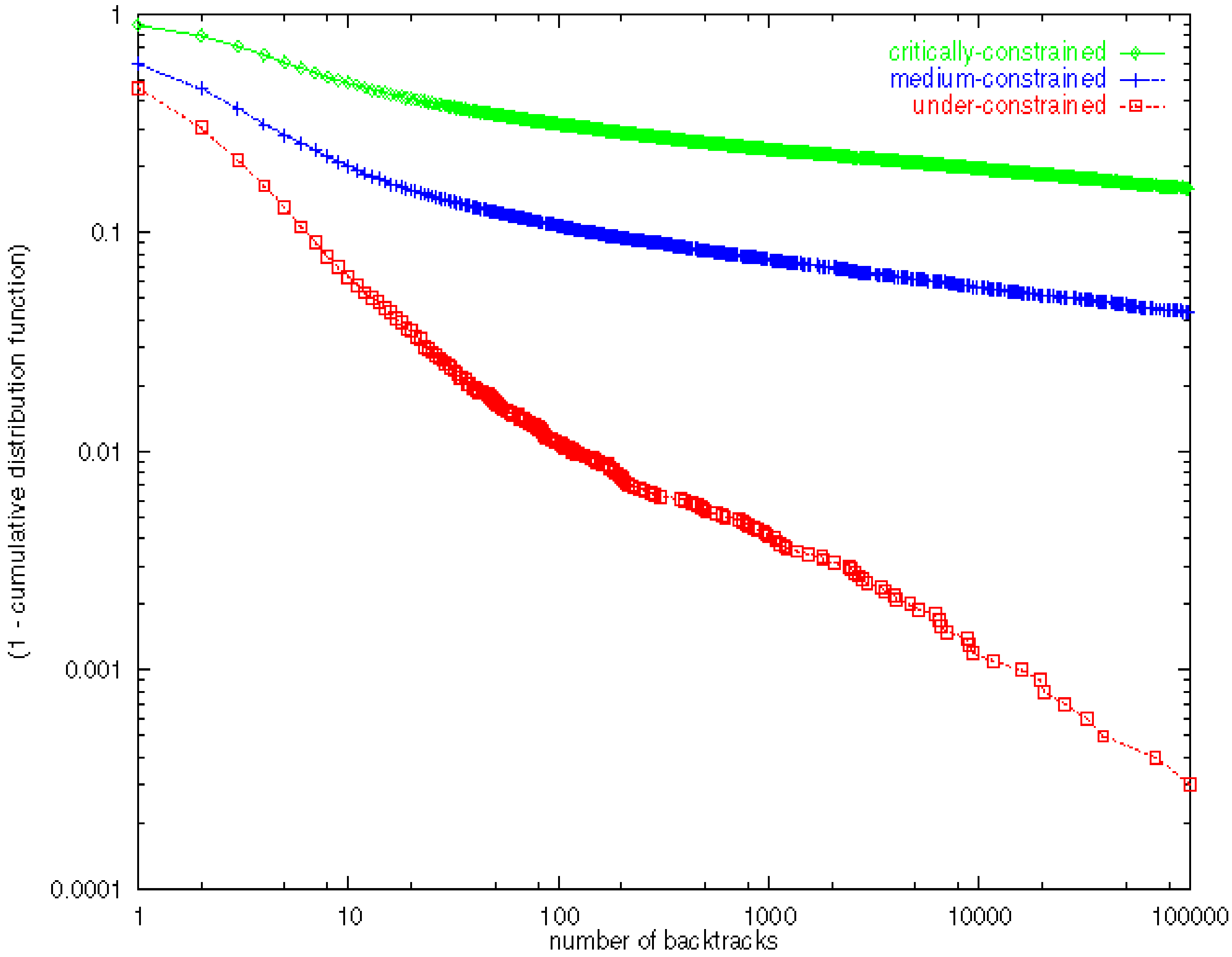
Cost Distributions

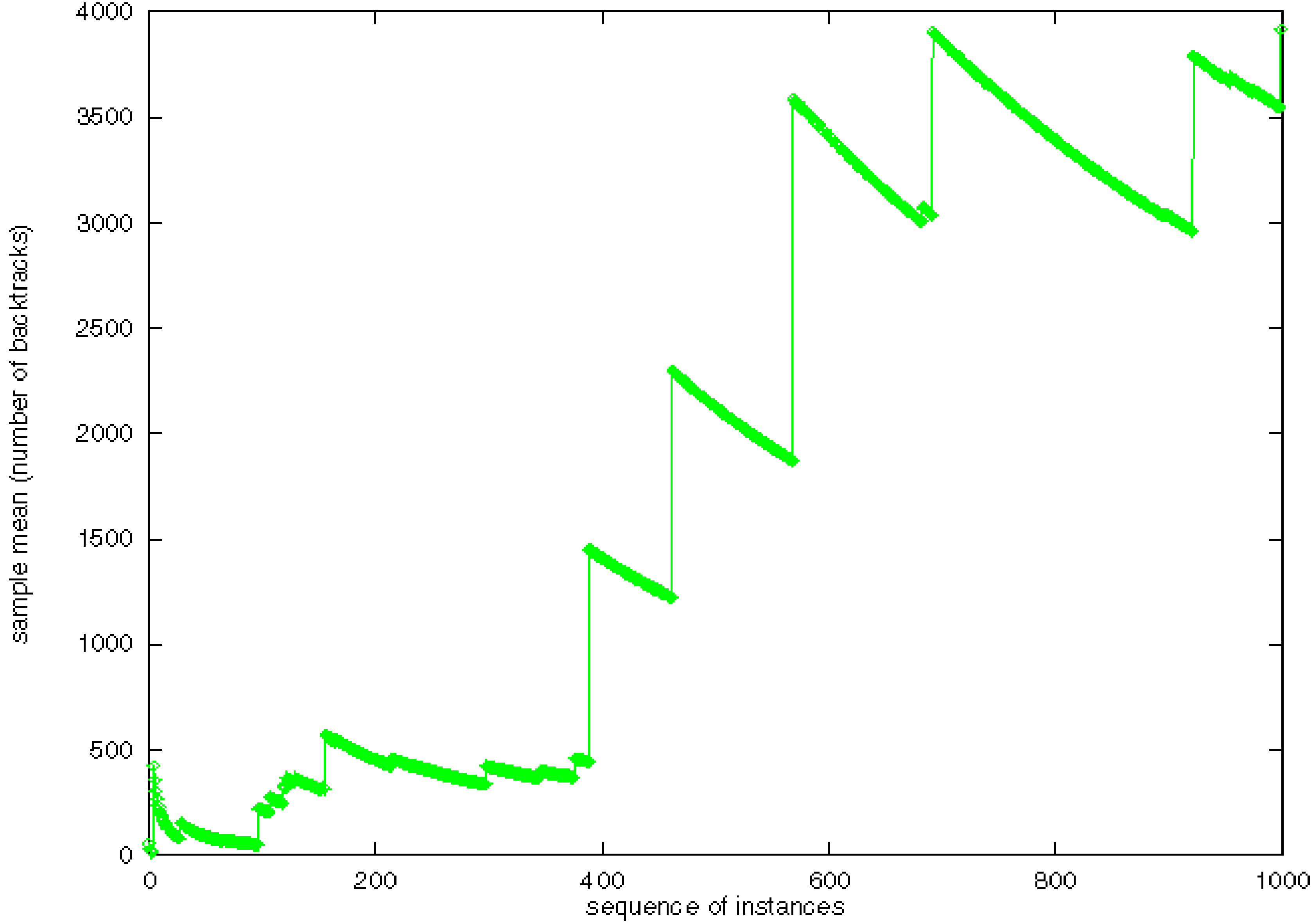
- Consider distribution of running times of backtrack search on a large set of “equivalent” problem instances
  - renumber variables
  - change random seed used to break ties
- *Observation (Gomes 1997): distributions often have heavy tails*
  - infinite variance
  - mean increases without limit
  - 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

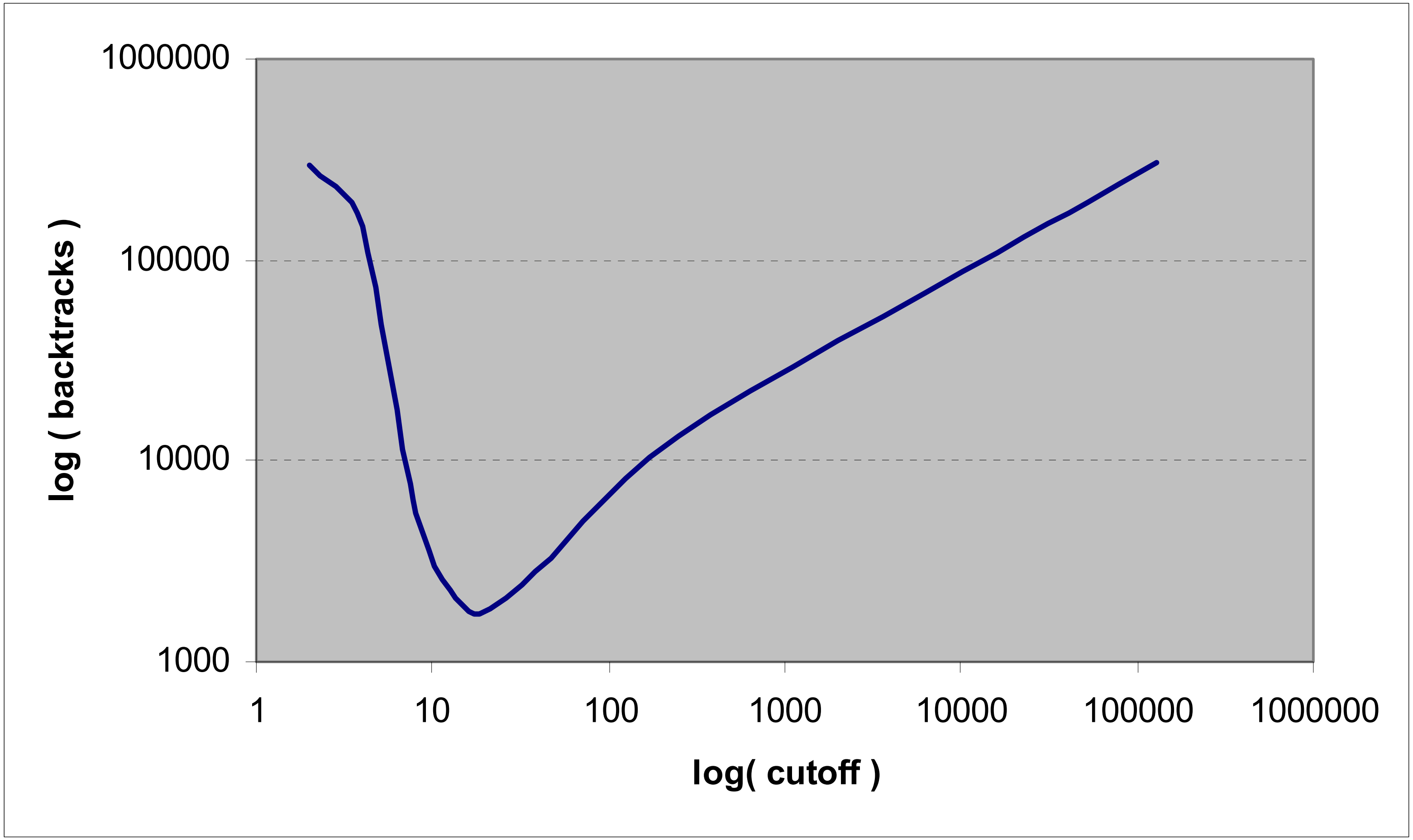
Randomized systematic solvers

- Add noise to the heuristic branching (variable choice) function
- Cutoff and restart search after a fixed number of backtracks

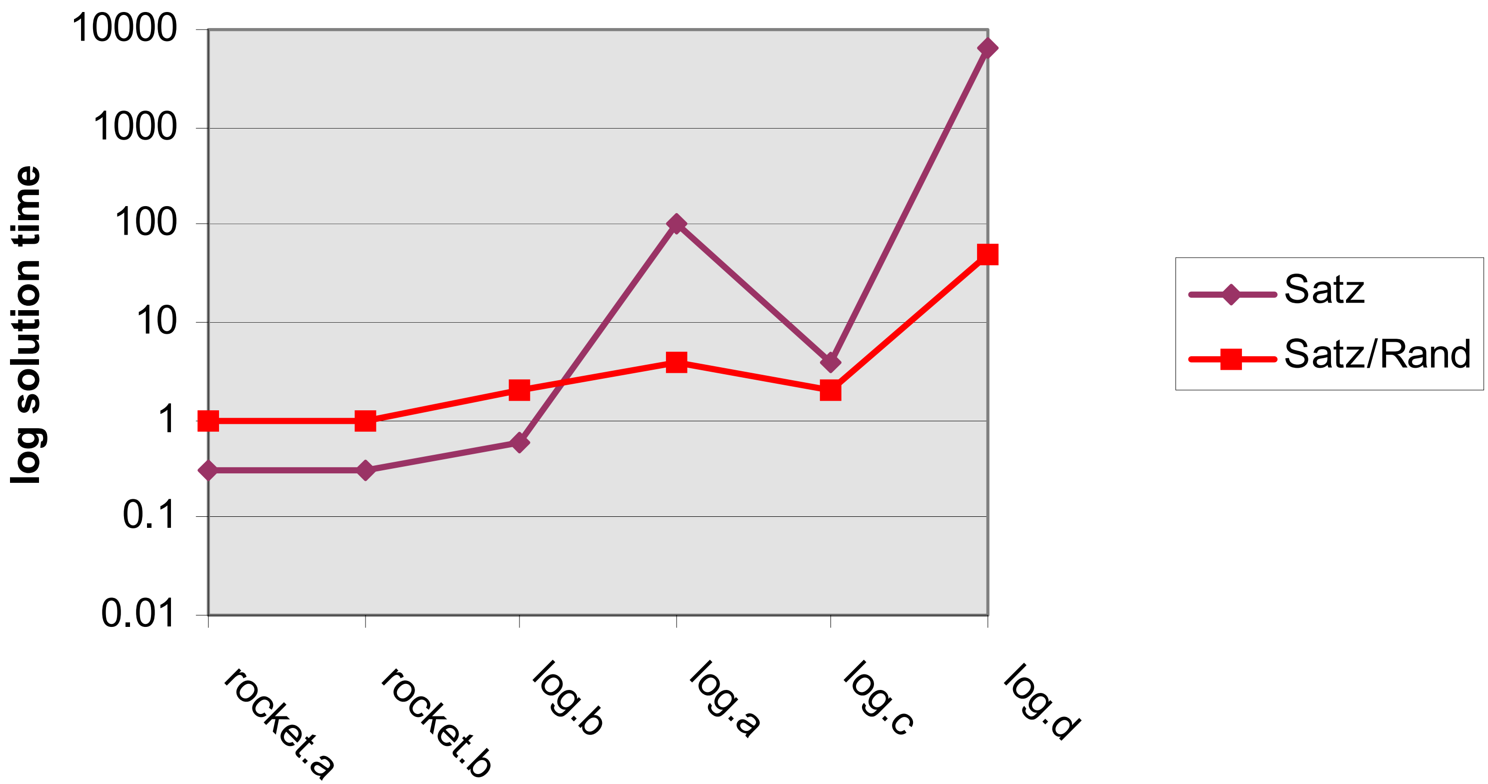
→ Provably Eliminates heavy tails

- *In practice: rapid restarts with low cutoff can dramatically improve performance*

Rapid Restart Behavior



Increased Predictability



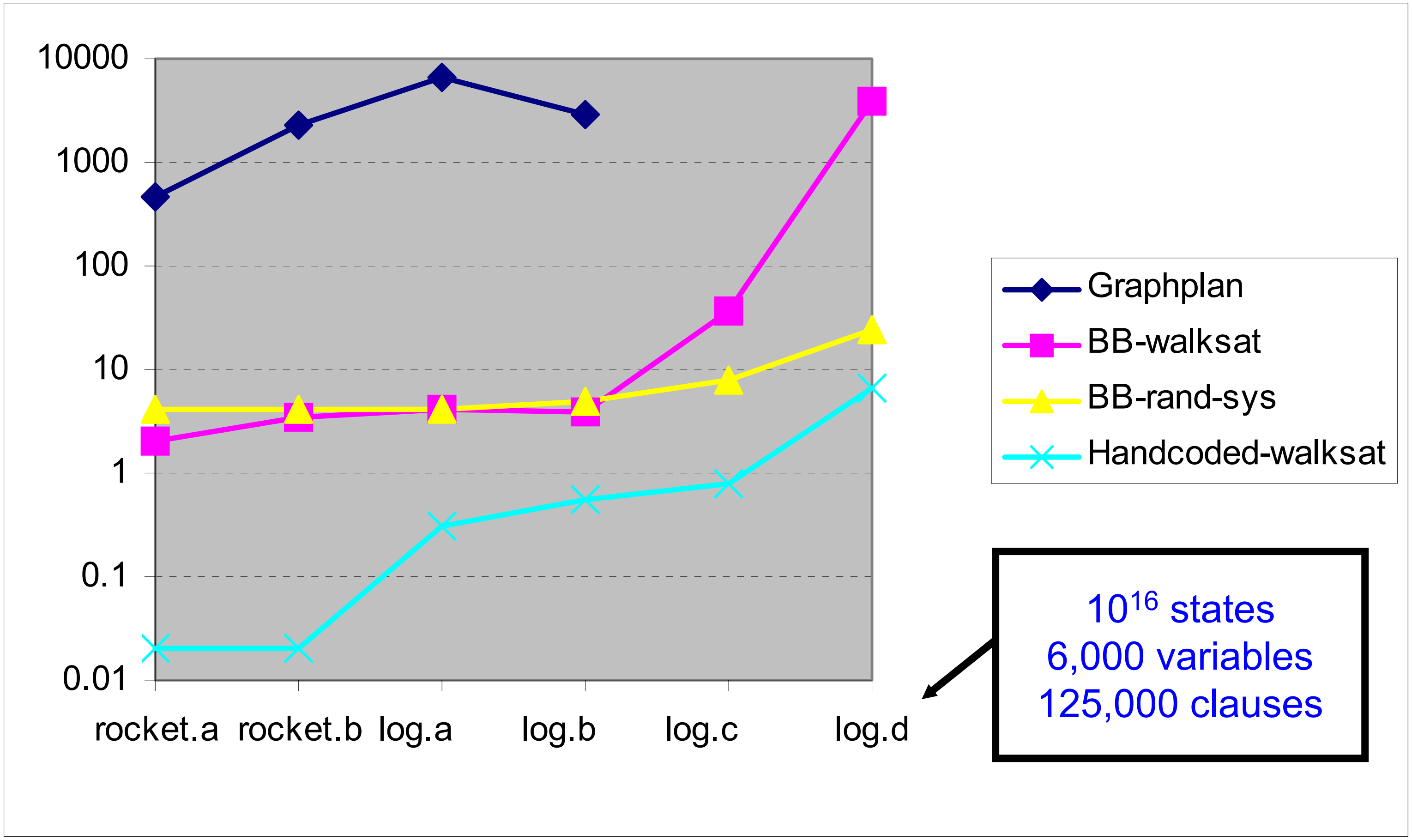
```
blackbox version 9B
command line:  blackbox -o logistics.pddl -f logistics_prob_d_len.pddl
  -solver compact -l -then satz -cutoff 25 -restart 10

----------------------------------------------------
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-central
2 load-truck_package4_ny-truck_ny-po
2 load-truck_package7_ny-truck_ny-po
2 load-truck_package3_pgh-truck_pgh-central
2 drive-truck_bos-truck_bos-po_bos-airport_bos
2 load-airplane_package8_airplane2_sf-airport
2 fly-airplane_airplane1_pgh-airport_sf-airport
2 drive-truck_la-truck_la-po_la-airport_la
3 fly-airplane_airplane2_sf-airport_bos-airport
3 unload-truck_package6_bos-truck_bos-airport
3 drive-truck_pgh-truck_pgh-central_pgh-airport_pgh
3 fly-airplane_airplane1_sf-airport_pgh-airport
3 unload-truck_package5_bos-truck_bos-airport
3 drive-truck_ny-truck_ny-po_ny-airport_ny
3 drive-truck_sf-truck_sf-po_sf-airport_sf
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 load-airplane_package5_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-airport_ny-po_ny
5 drive-truck_bos-truck_bos-central_bos-po_bos
5 load-airplane_package2_airplane1_pgh-airport
5 drive-truck_la-truck_la-po_la-central_la
5 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
5 load-airplane_package3_airplane1_pgh-airport
5 fly-airplane_airplane2_bos-airport_ny-airport
6 drive-truck_sf-truck_sf-airport_sf-central_sf
6 unload-airplane_package6_airplane2_ny-airport
6 load-airplane_package4_airplane2_ny-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-airport_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_la-airport
7 fly-airplane_airplane1_sf-airport_bos-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
8 drive-truck_la-truck_la-po_la-airport_la
9 unload-airplane_package7_airplane2_pgh-airport
9 unload-truck_package8_ny-truck_ny-central
9 unload-airplane_package5_airplane2_pgh-airport
9 unload-truck_package9_sf-truck_sf-po
9 unload-airplane_package3_airplane1_la-airport
9 unload-truck_package6_ny-truck_ny-central
9 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
9 load-airplane_package1_airplane2_pgh-airport
10 drive-truck_ny-truck_ny-central_ny-po_ny
10 fly-airplane_airplane2_pgh-airport_bos-airport
10 load-truck_package3_la-truck_la-airport
10 fly-airplane_airplane1_la-airport_ny-airport
10 drive-truck_pgh-truck_pgh-po_pgh-airport_pgh
11 drive-truck_bos-truck_bos-po_bos-airport_bos
11 drive-truck_ny-truck_ny-po_ny-airport_ny
11 unload-airplane_package2_airplane1_ny-airport
11 drive-truck_la-truck_la-airport_la-central_la
11 drive-truck_sf-truck_sf-po_sf-airport_sf
11 unload-airplane_package1_airplane2_bos-airport
11 load-truck_package7_pgh-truck_pgh-airport
11 load-truck_package5_pgh-truck_pgh-airport
12 drive-truck_sf-truck_sf-airport_sf-po_sf
12 load-truck_package1_bos-truck_bos-airport
12 fly-airplane_airplane2_bos-airport_la-airport
12 load-truck_package2_ny-truck_ny-airport
12 fly-airplane_airplane1_ny-airport_pgh-airport
12 drive-truck_pgh-truck_pgh-airport_pgh-po_pgh
12 unload-truck_package3_la-truck_la-central
13 drive-truck_ny-truck_ny-airport_ny-po_ny
13 load-truck_package3_la-truck_la-central
13 load-truck_package9_sf-truck_sf-po
13 drive-truck_bos-truck_bos-airport_bos-po_bos
13 unload-truck_package5_pgh-truck_pgh-po
13 unload-airplane_package4_airplane2_la-airport
14 unload-truck_package9_sf-truck_sf-po
14 unload-truck_package1_bos-truck_bos-po
14 unload-truck_package7_pgh-truck_pgh-po
14 unload-truck_package2_ny-truck_ny-po
14 unload-truck_package3_la-truck_la-central
End plan
```

Blackbox Results



10^{16} states
6,000 variables
125,000 clauses

Planning as CSP

Constraint-satisfaction problem (CSP)

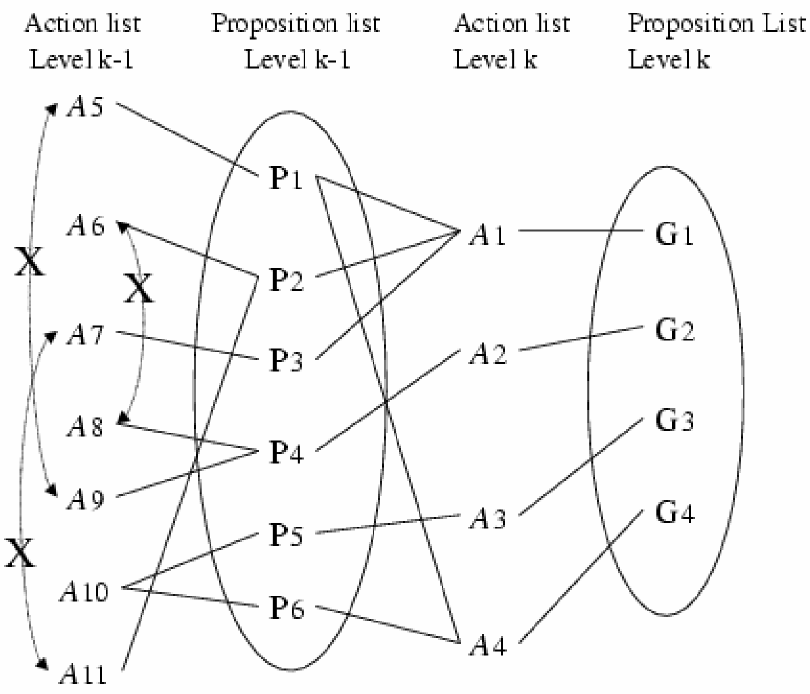
Given:

- 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

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

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: $Active\{G_1, G_2, G_3, G_4\}$

(b) DCSP

Transforming a DCSP to a CSP

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

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: $Active\{G_1, G_2, G_3, G_4\}$

(a) DCSP

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

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

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): $G_1 = A_1 \Rightarrow P_1 \neq \perp \wedge P_2 \neq \perp \wedge P_3 \neq \perp$
$G_2 = A_2 \Rightarrow P_4 \neq \perp$
$G_3 = A_3 \Rightarrow P_5 \neq \perp$
$G_4 = A_4 \Rightarrow P_1 \neq \perp \wedge P_6 \neq \perp$

Init State: $G_1 \neq \perp \wedge G_2 \neq \perp \wedge G_3 \neq \perp \wedge G_4 \neq \perp$

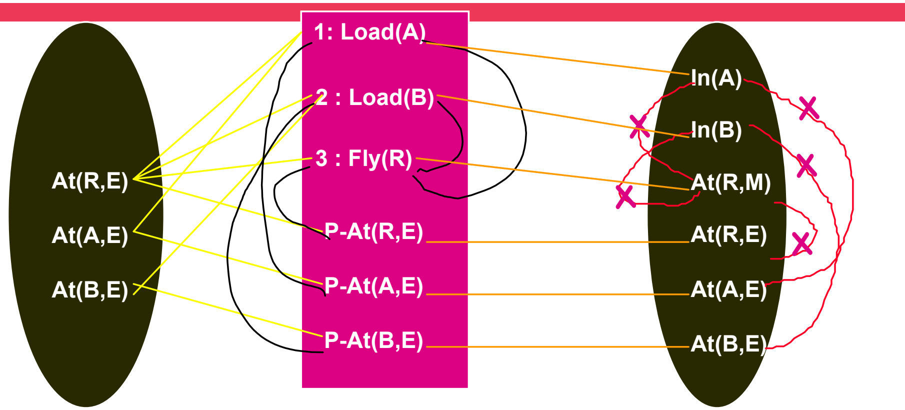
(b) CSP

Compilation to CSP

[Do & Kambhampati, 2000]

Goals: In(A),In(B)

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)

CSP Encodings can be more compact: GP-CSP

	Graphplan		Satz		Relsat		GP-CSP	
Problem	time (s)	mem	time(s)	mem	time (s)	mem	time (s)	mem
bw-12steps	0.42	1 M	8.17	64 M	3.06	70 M	1.96	3M
bw-large-a	1.39	3 M	47.63	88 M	29.87	87 M	1.2	11M
rocket-a	68	61 M	8.88	70 M	8.98	73 M	4.01	3M
rocket-b	130	95 M	11.74	70 M	17.86	71 M	6.19	4 M
log-a	1771	177 M	7.05	72 M	4.40	76 M	3.34	4M
log-b	787	80 M	16.13	79 M	46.24	80 M	110	4.5M
hsp-bw-02	0.86	1 M	7.15	68 M	2.47	66 M	.89	4.5 M
hsp-bw-03	5.06	24 M	> 8 hs	-	194	121 M	4.47	13 M
hsp-bw-04	19.26	83 M	> 8 hs	-	1682	154 M	39.57	64 M

[Do & Kambhampati, 2000]

GP-CSP Performance

	GPCSP		Graphplan	Satz	Relsat	speedup		
prob	heu	time (s)	time (s)	time (s)	time (s)	Graphplan	Satz	Relsat
bw-12steps	dlc	0.63	0.17	3.96	1.60	0.27	6.29	2.54
bw-large-a	ldc	5.40	0.57	27.80	32.30	0.11	5.15	5.98
bw-large-b	ldc	661	71	$>$ 8hrs	901.55	0.11	$>$ 43.57	1.36
rocket-a	dlc	1.22	43.13	3.81	5.27	35.35	3.12	4.32
rocket-b	dlc	2.33	87	5.91	8.39	37.34	2.54	3.60
log-a	dlc	0.95	842	2.88	1.11	886.32	3.03	1.17
log-b	dlc	19.10	390	7.73	22.03	20.42	0.40	1.15
log-c	dlc	24.27	$>$ 8hrs	308	77	$>$ 1187	12.69	3.17
log-d	dlc	84	$>$ 8hrs	15.99	199.38	$>$ 382.86	0.19	2.37
hsp-bw-02	ldc	0.34	0.32	3.62	1.21	0.94	10.65	3.56
hsp-bw-03	ldc	1.63	2.14	$>$ 8hrs	130.77	1.31	$>$ 17669	80.22
hsp-bw-04	ldc	4.87	19.26	$>$ 8hrs	1682	3.95	$>$ 5914	345.38
grid-01	ldc	7.75	7.21	$>$ 8hrs	22.75	0.93	$>$ 3716	2.94
grid-02	ldc	6.36	6.30	$>$ 8hrs	21.45	0.99	$>$ 4528	3.37
grid-03	ldc	9.83	8.77	$>$ 8hrs	42.68	0.89	$>$ 2930	4.34
gripper-01	ldc	0.01	0.01	0.52	0.09	1.00	52.00	9.00
gripper-02	ldc	0.41	0.05	2.41	0.69	0.12	5.88	1.68
gripper-03	ldc	62	4.28	109.72	155.72	0.07	1.77	2.51

GP-CSP Performance

	GPCSP		Graphplan	Satz	Relsat	speedup		
prob	heu	time (s)	time (s)	time (s)	time (s)	Graphplan	Satz	Relsat
bw-12steps	dlc	0.63	0.17	3.96	1.60	0.27	6.29	2.54
bw-large-a	ldc	5.40	0.57	27.80	32.30	0.11	5.15	5.98
bw-large-b	ldc	661	71	> 8hrs	901.55	0.11	> 43.57	1.36
rocket-a	dlc	1.22	43.13	3.81	5.27	35.35	3.12	4.32
rocket-b	dlc	2.33	87	5.91	8.39	37.34	2.54	3.60
log-a	dlc	0.95	842	2.88	1.11	886.32	3.03	1.17
log-b	dlc	19.10	390	7.73	22.03	20.42	0.40	1.15
log-c	dlc	24.27	> 8hrs	308	77	> 1187	12.69	3.17
log-d	dlc	84	> 8hrs	15.99	199.38	> 382.86	0.19	2.37
hsp-bw-02	ldc	0.34	0.32	3.62	1.21	0.94	10.65	3.56
hsp-bw-03	ldc	1.63	2.14	> 8hrs	130.77	1.31	> 17669	80.22
hsp-bw-04	ldc	4.87	19.26	> 8hrs	1682	3.95	> 5914	345.38
grid-01	ldc	7.75	7.21	> 8hrs	22.75	0.93	> 3716	2.94
grid-02	ldc	6.36	6.30	> 8hrs	21.45	0.99	> 4528	3.37
grid-03	ldc	9.83	8.77	> 8hrs	42.68	0.89	> 2930	4.34
gripper-01	ldc	0.01	0.01	0.52	0.09	1.00	52.00	9.00
gripper-02	ldc	0.41	0.05	2.41	0.69	0.12	5.88	1.68
gripper-03	ldc	62	4.28	109.72	155.72	0.07	1.77	2.51
hanoi-tower3	ldc	0.10	0.04	1.96	0.42	0.40	19.60	4.20
hanoi-tower4	ldc	9.87	0.45	12.58	54.68	0.05	1.27	5.54
hanoi-tower5	ldc	990	47.42	> 8hrs	> 8hrs	0.05	> 29.09	> 29.09
bulldozer-1	ldc	0.10	0.08	0.80	0.19	0.80	8.00	1.90
bulldozer-2	dlc	0.11	0.09	0.61	0.19	0.82	5.55	1.73
bulldozer-3	ldc	0.03	0.03	0.50	0.10	1.00	16.67	3.33
mprime-1	ldc	0.53	0.56	1.22	0.80	1.06	2.30	1.51
mprime-2	ldc	4.07	3.91	6.08	4.90	0.96	1.49	1.20
mprime-16	ldc	3.58	3.17	6.68	4.25	0.89	1.87	1.19
mystery-2	ldc	3.91	3.81	5.35	5.66	0.97	1.37	1.45
mystery-3	dlc	0.39	0.43	0.80	0.41	1.10	2.05	1.05
mystery-26	dlc	1.19	1.09	1.76	1.12	0.92	1.48	0.94
mystery-28	dlc	9.65	0.34	2.78	2.37	0.04	0.29	0.25
mystery-30	ldc	4.81	3.42	> 8 hrs	9.28	0.71	> 5988	1.93
frid-typed-1	dlc	0.12	0.12	0.59	0.19	1.00	4.92	1.58
frid-typed-2	dlc	0.38	0.34	1.34	0.65	0.89	3.53	1.71