Planning and Acting in Incomplete Domains

Christopher Weber and Daniel Bryce

Motivation

- Train Agents cheaply
- Perfect Knowledge Engineering is Costly
- Plan and Act with Incomplete Domain Knowledge
- Focus on Incomplete Actions, but complete set of Propositions

Anatomy of an Incomplete Domain

- STRIPS: (P,A,I,G)
  - P: Set of propositions
  - A: Incomplete Actions
    - Known: Preconditions, Adds, Deletes
    - Possible: Preconditions, Adds, Deletes
    - Impossible: Preconditions, Adds, Deletes
  - I: Initial State
  - G: Goal

PARC Printer

```
(:action HtmOverBlack-Move-A4
 :parameters ( ?sheet - sheet_t )
 :precondition (and (clear) (Available HtmOverBlack-RSRC)
                    (Sheetsize ?sheet A4)
                    (Location ?sheet HtmOverBlack_Entry-EndCap_Exit))
 :effect (and (not (Available HtmOverBlack-RSRC))
              (Location ?sheet HtmOverBlack_Exit-Down_TopEntry)
              (not (Location ?sheet HtmOverBlack_Entry-EndCap_Exit))
              (Available HtmOverBlack-RSRC))
 :poss-effect (and (not (clear))))
```

Plan Semantics

- Optimistic:
  - Ignore possible preconditions and deletes.
  - Keep possible adds.
- Pessimistic:
  - Assume possible preconditions and deletes.
  - Ignore possible adds.
- Cautiously Optimistic:
  - Don't assume anything
  - Count models under which plan succeeds

Synthesizing Plans

- Optimistic/Pessimistic
  - Translate Instance to STRIPS instance
    - Use favorite classical planner
- Cautiously Optimistic
  - Translate to Conformant Planning
    - Use favorite Conformant Planner (count models)
  - Default Planner
    - Count models
    - Count failure diagnoses

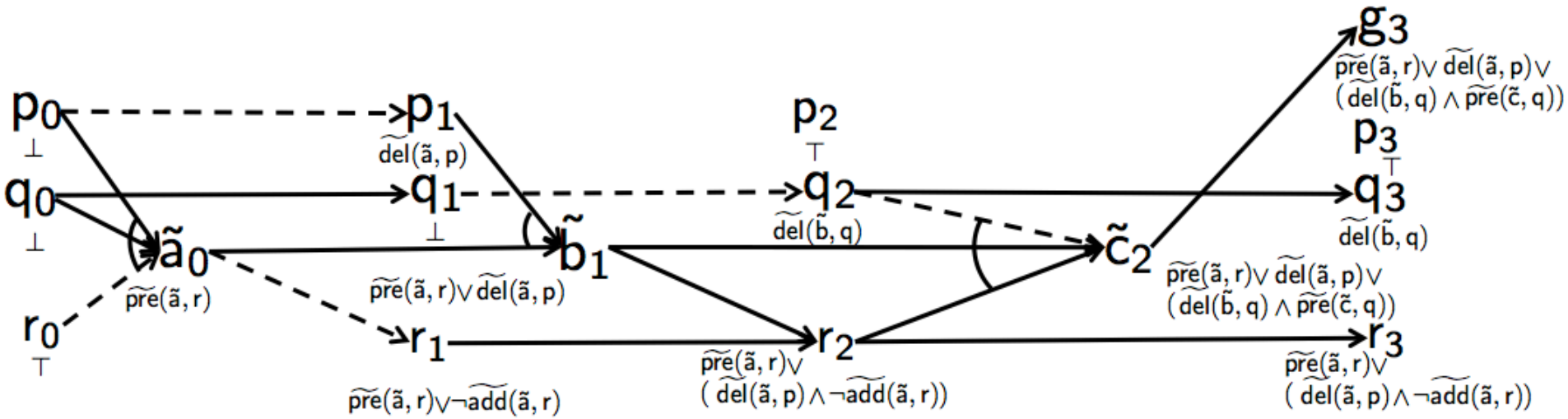
Conformant Planning

- Action Incompleteness → State Incompleteness
  - New propositions (e.g., pre(a,p), add(b, q), …)
  - Initial Belief State (e.g., unknown(pre(a,p)))
  - Condition each effect on incompleteness
    - -pre(a,p), r, s => g
    - pre(a,p), p, r, s => g
    - add(b,q), m, n => q
- Must be Conformant Probabilistic to count (weighted) models
  - Conformant Non-deterministic requires strong plans

DeFault

- Extend [Garland and Lesh, AAAI02] and [Robertson and Bryce, HDIP09] from single fault to multi-fault.
- Foundations in Model-Based Diagnosis [Reiter, AIJ87] and Assumption-Based Truth Maintenance Systems [de Kleer AIJ86].

Plan Failure Explanations



Propositions = {p,q,r,g}
Incomplete Actions = {a,b,c}
Initial State = {p, q}
Goal = {g}
Incomplete Features = {pre(a, r), add(a, r), del(a, p), del(b, q), pre(c, q)}
Complete Features: pre(a) = {p, q};
pre(b) = {r}, add(b) = {r};
pre(c) = {r}, add(c) = {g}.
State Sequence: (s0 = {p,q}, s1 = {p,q,r}, s2 = {q,r}, s3 = {q,r,g})

Propagating Failure Explanations

$$d(\tilde{a}_t) = d(\tilde{a}_{t-1}) \vee \bigvee_{p \in \text{pre}(\tilde{a})} d(p_t) \vee \bigvee_{p \in \widetilde{\text{pre}}(\tilde{a}_t)} (d(p_t) \wedge \widetilde{\text{pre}}(\tilde{a}_t, p))$$

$$d(p_{t+1}) = \begin{cases} d(p_t) \wedge d(\tilde{a}_t) & : p \in \text{add}(\tilde{a}_t) \\ d(p_t) \wedge (d(\tilde{a}_t) \vee \neg\widetilde{\text{add}}(\tilde{a}_t, p)) & : p \in \widetilde{\text{add}}(\tilde{a}_t) \\ \top & : p \in \text{del}(\tilde{a}_t) \\ d(p_t) \vee \widetilde{\text{del}}(\tilde{a}_t, p) & : p \in \widetilde{\text{del}}(\tilde{a}_t) \\ d(p_t) & : \text{otherwise} \end{cases}$$

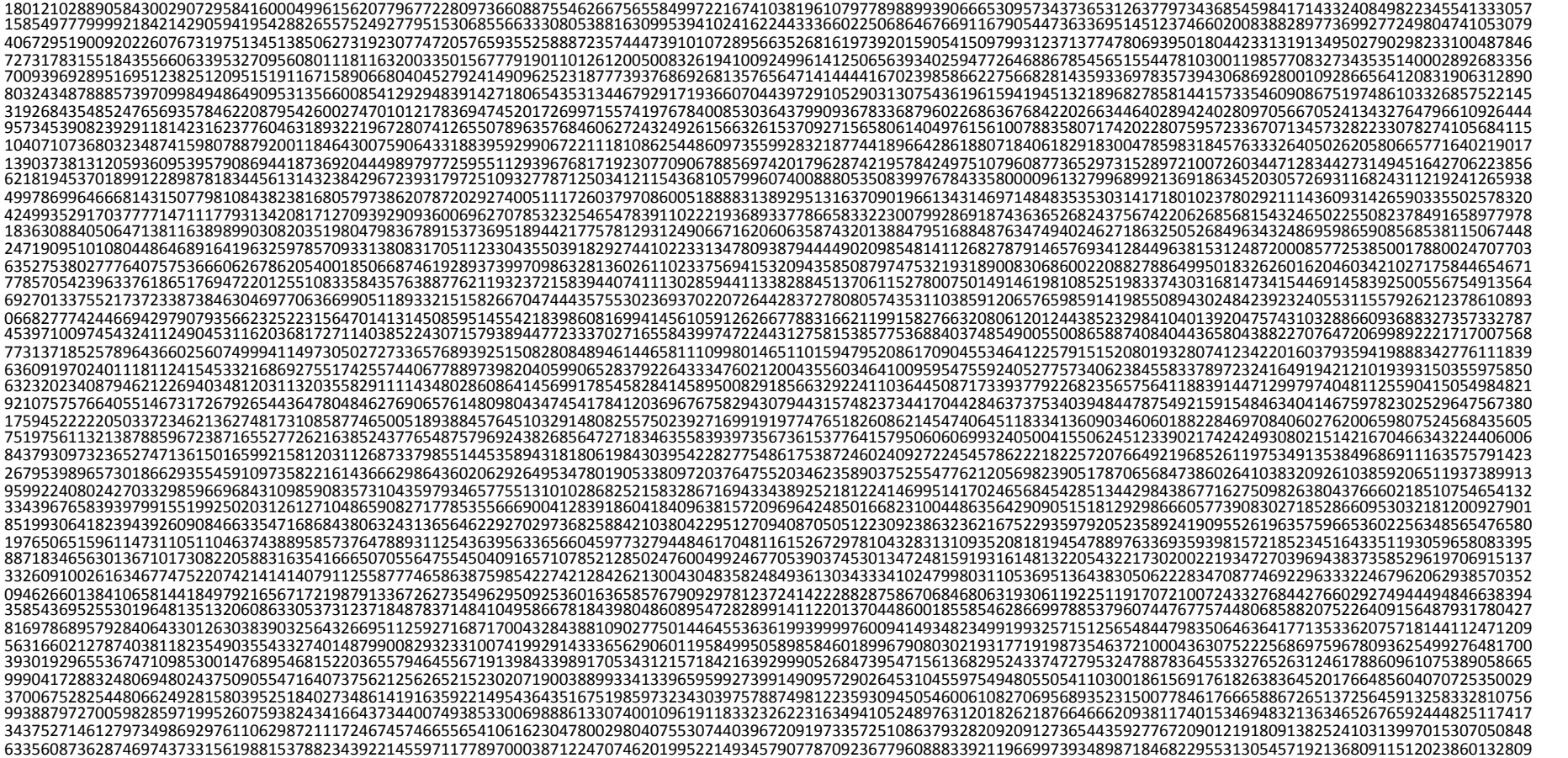
Counting Models and Diagnoses

$$pre(a, r) \vee del(a, p) \vee (del(b, q) \wedge pre(c, q))$$

- Propositional Models: 26
- PI1: 2
- PI2: 1

  $pre(a, r) \vee del(a, p)$
- Propositional Models: 24
- PI1: 2
- PI2: 0

Models (26834 Incomplete Features)



Heuristics

- Optimistic/Pessimistic
  - FF Heuristic
- Cautiously Optimistic
  - Propagate PFE's, ignore deletes

$$d(\tilde{a}_{t+k}) = \bigvee_{p \in \text{pre}(\tilde{a})} d(p_{t+k}) \vee \bigvee_{p \in \widetilde{\text{pre}}(\tilde{a})} (d(p_{t+k}) \wedge \widetilde{\text{pre}}(\tilde{a}, p))$$

$$d(p_{t+k+1}) = \begin{cases} d(a_{t+k}(p)) & : p \in \text{add}(a_{t+k}(p)) \\ d(a_{t+k}(p)) \vee \neg\widetilde{\text{add}}(a_{t+k}(p), p) & : p \in \widetilde{\text{add}}(a_{t+k}(p)) \end{cases}$$

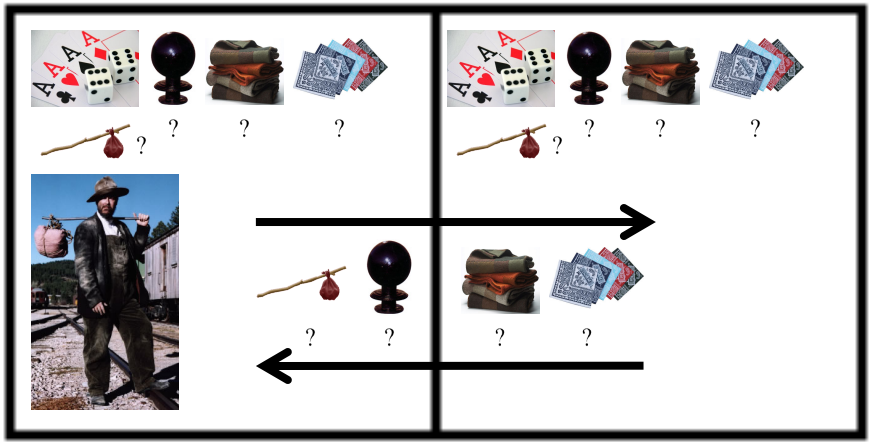
Heuristics Cont'd

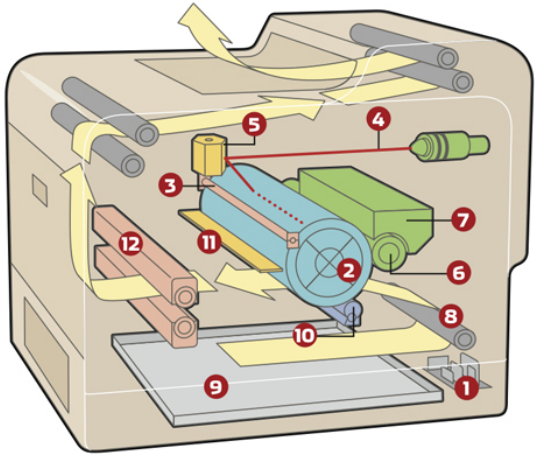
- Select Supporting action for each proposition
  - h^M Fewest Models of effect PFE
  - h^{PI} Fewest Prime Implicants (Diagnoses) of effect PFE
- Extract Relaxed Plan using chosen actions

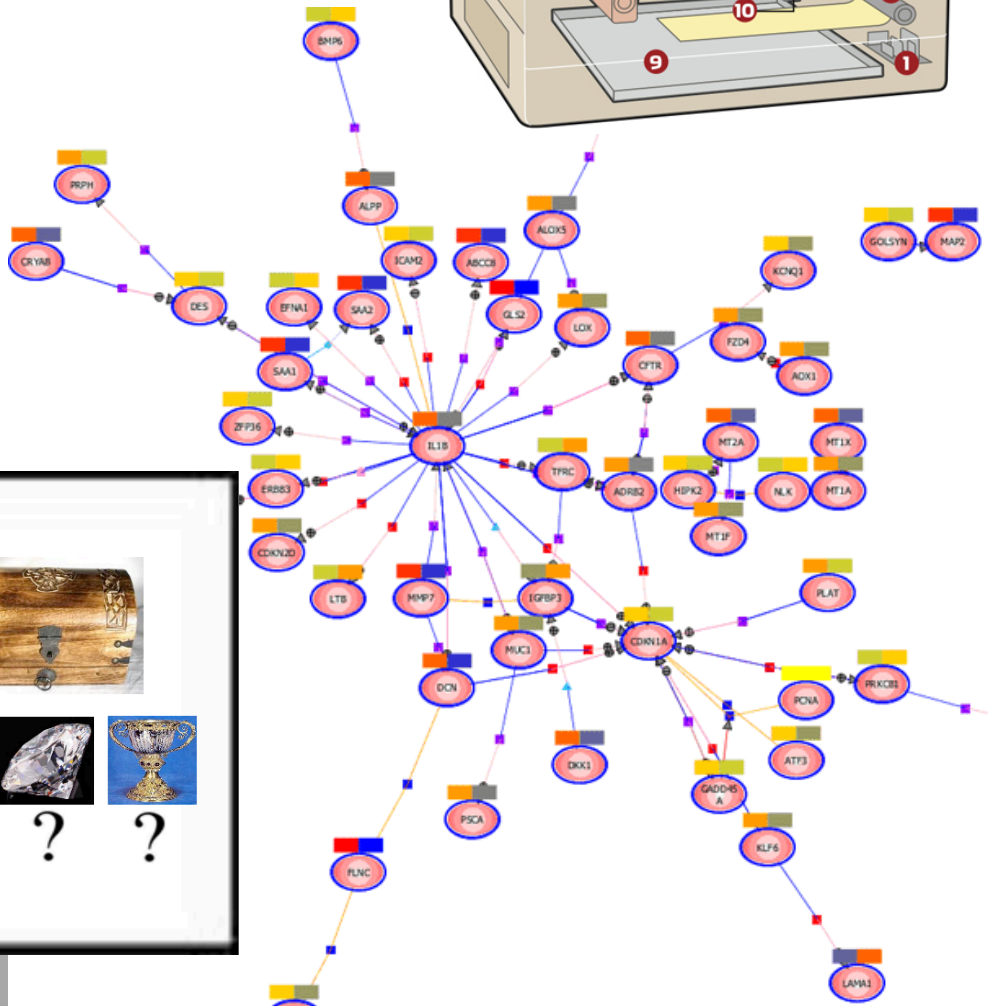
DeFault Planner

- Dual-Queue GBFS w/ Preferred Operators
- Return First Solution
- All code identical except heuristic functions and search nodes
- Compute plan quality after search
- Pure Java
- JDD BDD package

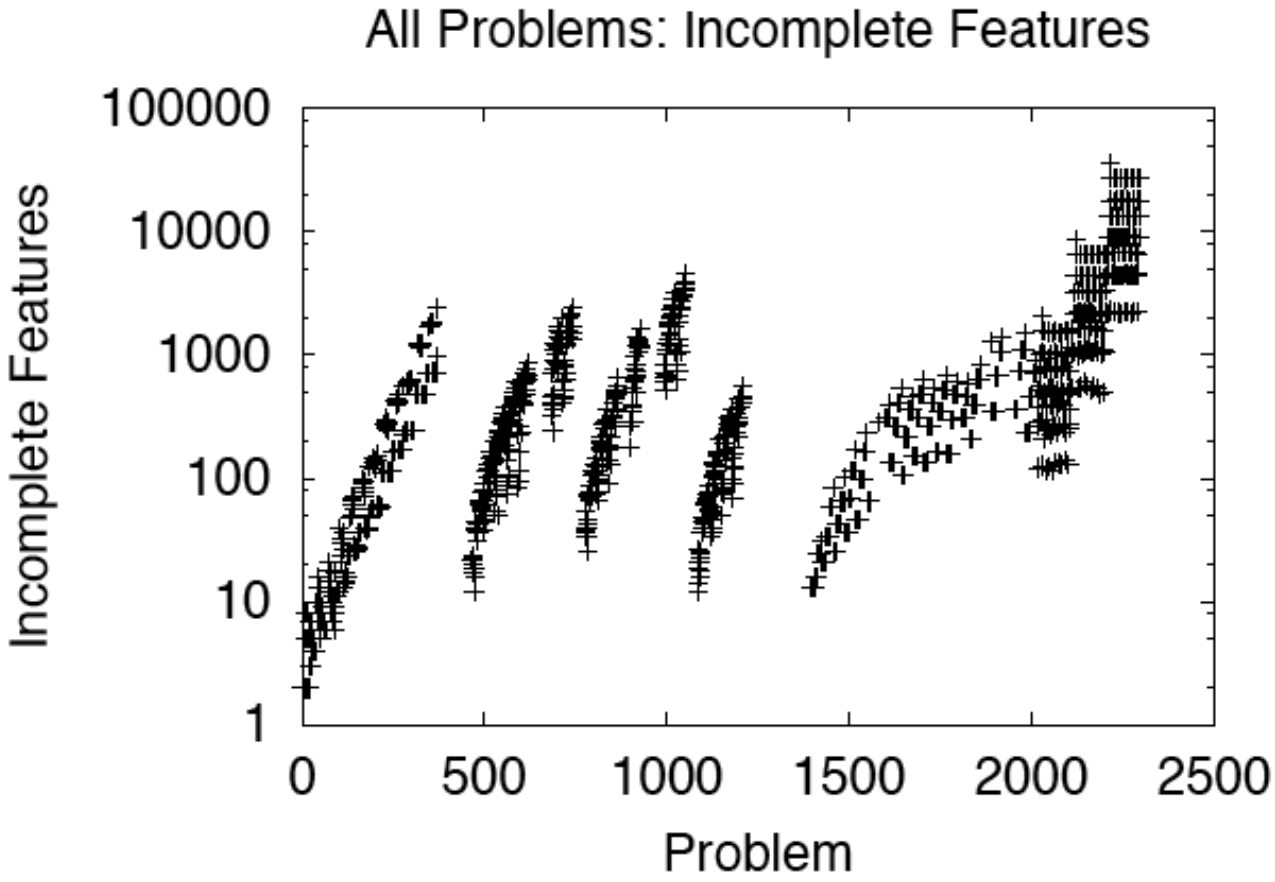
Domains




1
2
3
4
5
6
7
8
9
10
11
12


BMP6
PRPH
ALPP
ALOX5
CRYAB
ICAM2
ABCC8
KCNQ1
GOLSYN
MAP2
DES
EFNA1
SAA2
GLS2
LOX
FZD4
SAA1
CFTR
AOX1
ZFP36
IL1B
MT2A
MT1X
ERBB3
TFRC
ADRB2
HIPK2
NLK
MT1A
MT1F
CDKN2D
LTB
MMP7
IGFBP3
PLAT
MUC1
CDKN1A
DCN
PRKCBI
PCNA
DKK1
ATF3
GADD45A
PSCA
KLF6
FLNC
LAMA1

Incomplete Features



Questions

- Q1: Does reasoning about incompleteness lead to high quality plans?
- Q2: Does counting prime implicants perform better than counting models?
- Q3: As the number of incomplete features grows, does stronger reasoning about incompleteness help?

Q1: Quality
(# Better Quality)

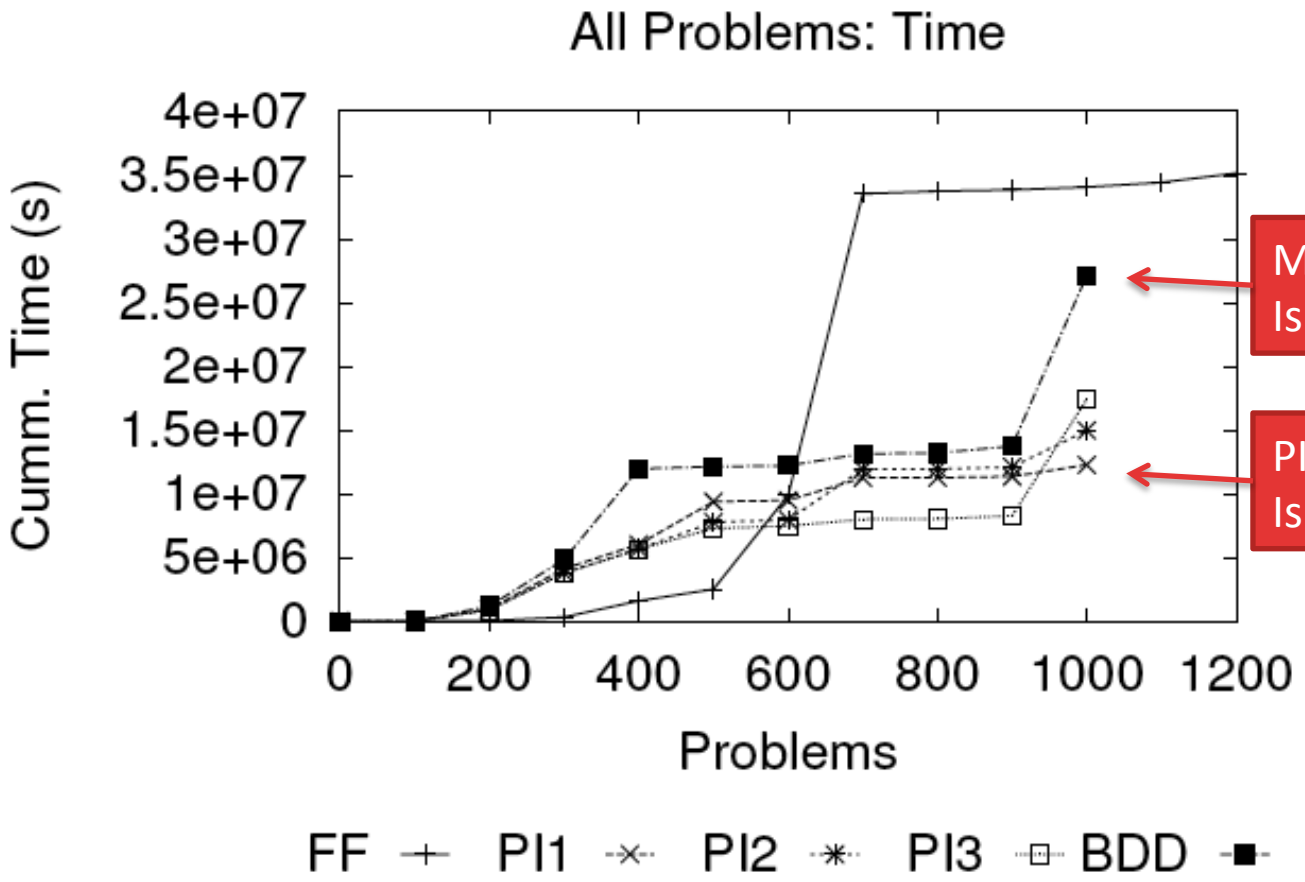
	FF	PI1	PI2	PI3	BDD
FF	0	148	157	158	121
PI1	**603**	0	**82**	**80**	**204**
PI2	**591**	75	0	48	**205**
PI3	**564**	59	**52**	0	**194**
BDD	**494**	185	189	186	0

Q2: Counting Performance (Problems Solved)

Domain	FF	PI1	PI2	PI3	BDD	POND
PARCprinter 0.25	130	83	85	**86**	80	10
PARCprinter 0.5	130	87	**88**	87	80	0
PARCprinter 0.75	130	82	**83**	81	80	0
PARCprinter 1.0	13	**10**	9	9	8	0
Parcprinter	403	262	**265**	263	248	10
Barter 0.25	150	106	128	**129**	108	60
Barter 0.5	150	134	**137**	134	118	45
Barter 0.75	150	**140**	138	137	111	27
Barter 1.0	15	**14**	**14**	**14**	11	2
Barter	465	394	**417**	414	348	155
Pathways 0.25	160	**40**	**40**	**40**	**40**	19
Pathways 0.5	160	**70**	60	50	60	13
Pathways 0.75	170	**60**	50	40	**60**	12
Pathways 1.0	19	5	6	6	**7**	2
Pathways	509	**175**	156	136	167	46

Domain	FF	PI1	PI2	PI3	BDD	POND
Bridges1 0.25	33	**19**	**19**	**19**	**19**	2
Bridges1 0.5	33	**15**	**15**	**15**	**15**	2
Bridges1 0.75	32	**18**	17	17	**18**	2
Bridges1 1.0	4	**2**	**2**	1	**2**	1
Bridges2 0.25	29	15	**16**	**16**	**16**	3
Bridges2 0.5	31	16	12	12	**19**	3
Bridges2 0.75	31	13	14	15	**16**	2
Bridges2 1.0	4	1	1	0	**2**	1
Bridges3 0.25	36	25	25	25	**26**	1
Bridges3 0.5	37	22	22	22	**23**	2
Bridges3 0.75	38	**25**	**25**	24	**25**	1
Bridges3 1.0	4	**3**	**3**	**3**	2	1
Bridges	312	174	171	169	**183**	21
Total	1689	1005	**1009**	982	946	232

Q2: Counting Performance (Cumulative Solving Time)



Q3: Counting and Features (Problems Solved)

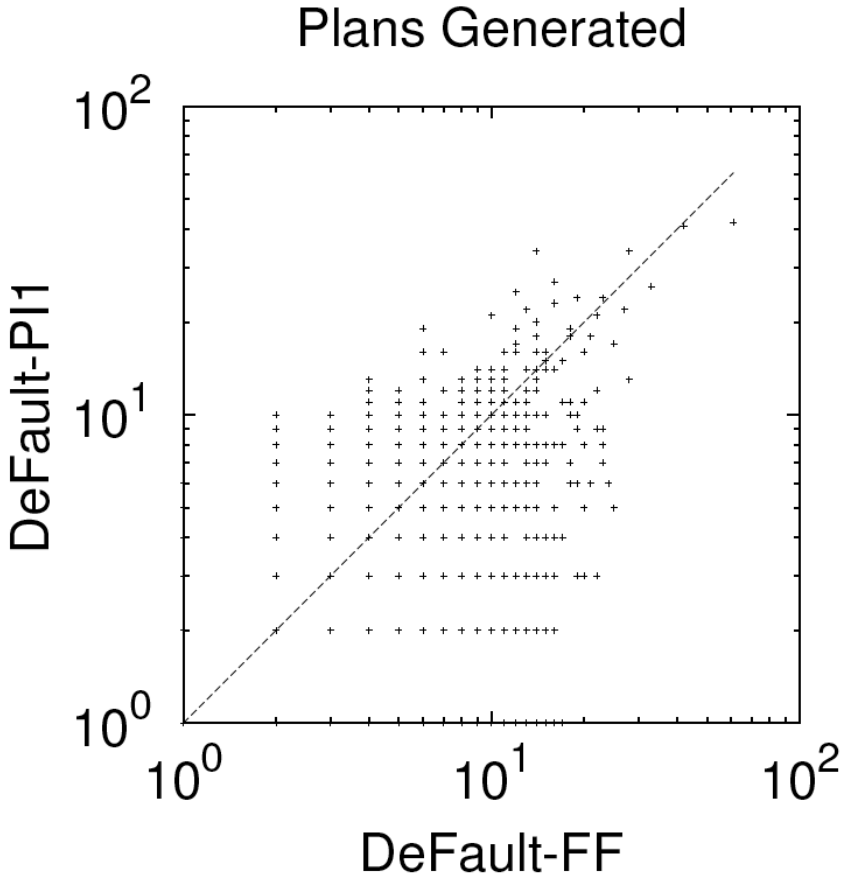
Domain	FF	PI1	PI2	PI3	BDD	POND
PARCprinter 0.25	130	83	85	**86**	80	10
PARCprinter 0.5	130	87	**88**	87	80	0
PARCprinter 0.75	130	82	**83**	81	80	0
PARCprinter 1.0	13	**10**	9	9	8	0
Parcprinter	403	262	**265**	263	248	10
Barter 0.25	150	106	128	**129**	108	60
Barter 0.5	150	134	**137**	134	118	45
Barter 0.75	150	**140**	138	137	111	27
Barter 1.0	15	**14**	**14**	**14**	11	2
Barter	465	394	**417**	414	348	155
Pathways 0.25	160	**40**	**40**	**40**	**40**	19
Pathways 0.5	160	**70**	60	50	60	13
Pathways 0.75	170	**60**	50	40	**60**	12
Pathways 1.0	19	5	6	6	**7**	2
Pathways	509	**175**	156	136	167	46

Domain	FF	PI1	PI2	PI3	BDD	POND
Bridges1 0.25	33	**19**	**19**	**19**	**19**	2
Bridges1 0.5	33	**15**	**15**	**15**	**15**	2
Bridges1 0.75	32	**18**	17	17	**18**	2
Bridges1 1.0	4	**2**	**2**	1	**2**	1
Bridges2 0.25	29	15	**16**	**16**	**16**	3
Bridges2 0.5	31	16	12	12	**19**	3
Bridges2 0.75	31	13	14	15	**16**	2
Bridges2 1.0	4	1	1	0	**2**	1
Bridges3 0.25	36	25	25	25	**26**	1
Bridges3 0.5	37	22	22	22	**23**	2
Bridges3 0.75	38	**25**	**25**	24	**25**	1
Bridges3 1.0	4	**3**	**3**	**3**	2	1
Bridges	312	174	171	169	**183**	21
Total	1689	1005	**1009**	982	946	232

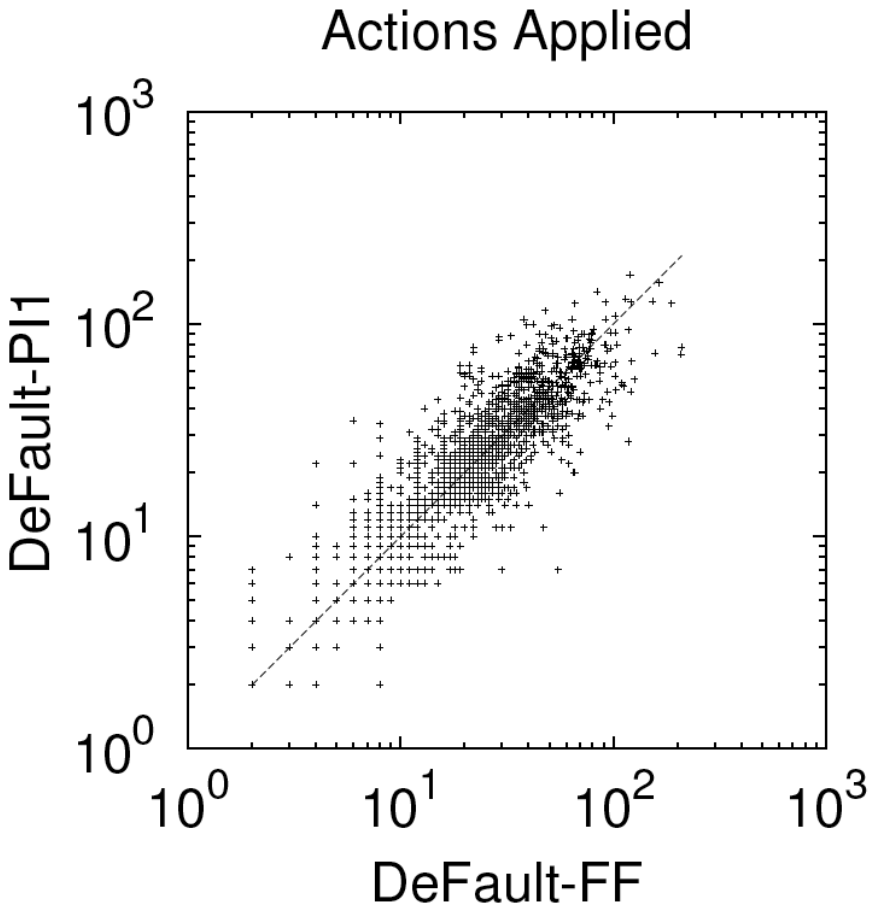
Execution in Incomplete Domains

- [Chang and Amir, ICAPS06]
  - Plan optimistically much like a classical planner
  - Learn adds and dels
  - Emphasizes Compact Filtering
- Goalie
  - Plan with DeFault
  - Learn pre, add, and dels
  - Emphasizes Robust Planning
- Q4: Does reasoning about incompleteness reduce the number of execution failures during execution?

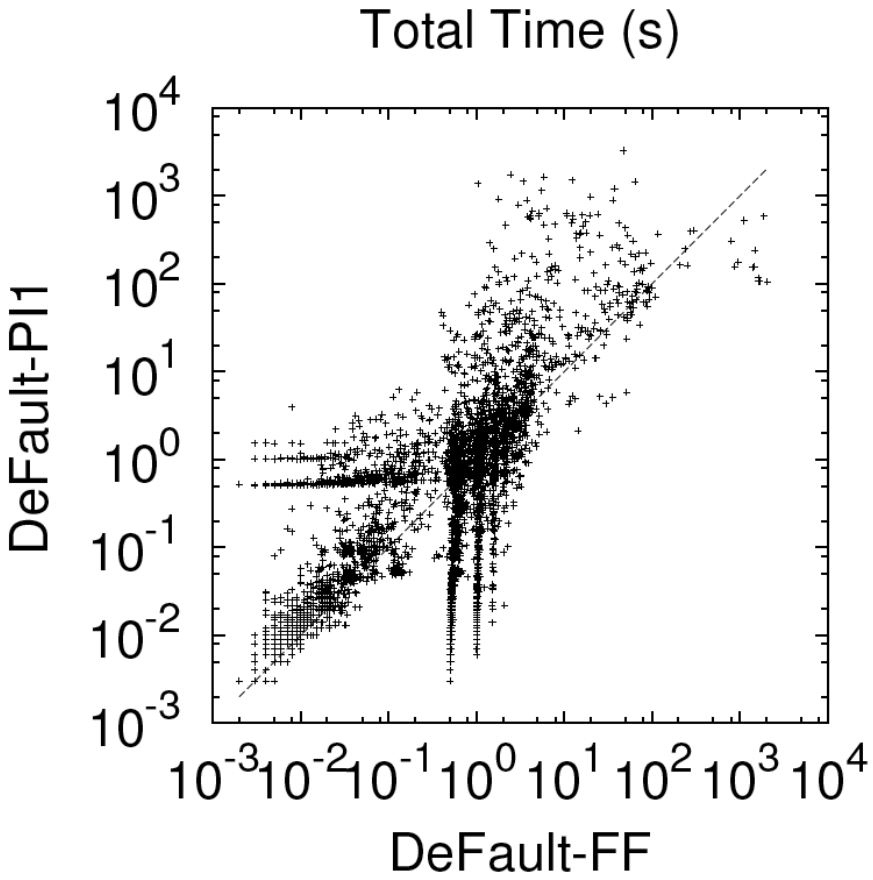
Q4: Planner Invocations



Actions Applied



Total Time



Summary

- First approach to planning in incomplete STRIPS domains
- Counting diagnoses (PIs) is faster than model counting and returns comparable quality plans
- Executing robust Incomplete plans requires less re-planning

Future Work

- More types of models
  - Conditional Effects, Temporal, First-Order, …
- Correlated Incompleteness
- Priors
- Alternative Representations of Failure Expl.
- Approximate Belief States
- Knowledge Acquisition

Thanks

DeFault and Goalie Java source code and domain generators available online

http://www.cs.usu.edu/~danbryce/software/default.jar