Evaluating Planning Algorithms

Jörg Hoffmann

INRIA
Nancy, France

June 8, 2011

Outline

- **Evaluation? What's that?**
- On the IPC and other bizarre rituals
- Do your homework!
- Buried beneath tons of data
- The black art of counting black sheep
- Understanding the world

Evaluation? What's that?

What are the advantages – and the disadvantages!!! – of the technique I'm proposing here?

- **Empirical:** Data on examples
- **Theoretical:** If A then B
- **Applied:** It is/will be in real-world use at X
  (and they're earning $$$ with it)

Theory is the *only* way to ever truly generalize beyond examples!

Applied Evaluation

Good luck!

Don't lose sight of the big picture:

- What am I doing and why am I doing it?
- Who would be using this in practice and for what?
- *What is the added value of planning here?*
- Excellent example: [Ruml et al, JAIR'11]

Parenthesis: Automatic Planning

Is FF automatic? Yes or No?

Correct answer: No.

- You've got to give it the PDDL first
- It's all a matter of cost-for-input vs. usefulness-of-output!!
- "Applied" Web Service Composition (ca. 1001 papers):

  "Services annotated as planning actions, planner composes more complex/useful service automatically."
- Yeah great, but who's gonna write the "annotation"?

Theoretical Evaluation

From standards . . .

- Is it sound? (What do you mean, “no”?)
- Is it complete?
- Can it sing and dance?

. . . to excitement!

- “The representational power of Merge-and-Shrink strictly dominates that of PDBs” (Helmert et al, ICAPS’07)
- “Our compilation of conformant planning is exponential only in conformant width” (Palacios&Geffner, JAIR’09)
- “Our polynomial-time action-cost partitioning provides the tightest possible lower bound” (Katz&Domshlak, ICAPS’08)

- Often more feasible: look at individual domains ([Hoffmann, ICAPS’11; Nissim&Hoffmann&Helmert, IJCAI’11])

Empirical Evaluation

This is "easy" . . .

- Run technique on examples (well, implement it first . . .)
- Report data

. . . but the devil is in the details!

- How/against whom do I run it?
- How do I analyze and report the results?
- *How do I understand what's going on?*

Outline

- Evaluation? What's that?
- **On the IPC and other bizarre rituals**
- Do your homework!
- Buried beneath tons of data
- The black art of counting black sheep
- Understanding the world

The Four Commandments

1. Run IPC benchmarks.
2. Unless you run all, run the most recent ones.
3. Time-out is 30 minutes.
4. Compare to the most recent winner.

Commandment 1:
Run IPC benchmarks.

Natural Language Sentence Generation
[Koller&Petrick,CompInt'11]:

> *"While some of the planners did an impressive job of controlling the complexity of the search, we also found that all the planners we tested spent too much time on preprocessing to be useful."*

- Pre-processing difficulties are not considered in IPC
- IPC benchmarks "spoon-feed" existing planner implementations
- Ergo: pre-instantiation etc. has gone completely unquestioned since almost a decade!

- Generally: IPC benchmarks created to suit IPC conditions

Commandment 1: (continued)
Run IPC benchmarks.

A hypothetical conversation: (any resemblance to real conversations is purely coincidental)

Two researchers, X and Y, in front of a whiteboard. The whiteboard is covered with a mixture of haphazard drawings and 1st order logic, all partly crossed out and over-written.

Says X: *"Hm, yes, looks interesting."*

Says Y: *"But will it be useful in practice?"*

Says X: *"Well, let's look at what it does in a simple transportation domain with fuel usage."*

Says Y: *"But is that in the IPC benchmarks?"*

- **IPC = *some* interesting challenges, not *all* of them!!!**
- Later: IPC benchmarks not good for counting sheep . . .

Commandment 2:
Unless you run all, run the most recent ones.

Well. Plain nonsense, no?

- In what way are the recent ones “better”?
- What are “good” or “bad” benchmarks anyway?
- Is a benchmark better if it takes more time to solve?
- If so, note that Mystery and Mprime, e.g., are still tough nuts

- Yes, Scanalyzer is better than “Monkey-and-bananas” . . .
- . . . but this doesn’t apply to the whole history of the IPC!

Commandment 3:
Time-out is 30 minutes.

Natural Language Sentence Generation:
Need plan in split seconds.
Creating business processes at SAP [Hoffmann et al, AAAI'10]:
Need plan in split seconds.
Controlling printers at Xerox [Ruml et al, JAIR'11]:
Need plan in split seconds.
Video games [Sturtevant, "Dragon Age: Origins"]:
Need plan in split seconds.
Vacuum cleaners, football, DARPA Grand Challenge, ...

- Many planning applications take real-time decisions
- In others, planning models are not precise/exhaustive enough to enable exact/full solution ...
- ... and hence a human user waits online for the plan!
- Anybody knows an application *not* falling into these classes?

Commandment 4:
Compare to the most recent winner.

Some example data:

Domain	#instances	LM-cut	M&S-bop
Gripper	20	6	20
Miconic	150	140	55
Σ	170	146	75

- IPC-domain=Miconic $\Longrightarrow$ "and the winner is . . . LM-cut!"
- IPC-domain=Gripper $\Longrightarrow$ "and the winner is . . . M&S-bop!"
- IPC-domain=Both? Let's reverse the #instances . . .

- Performance is a function of the benchmarks used!

- IPC organizers make every effort to avoid the detrimental consequences . . .
- . . . still the best planner for *your* context may be someone else

IPC Summary

IPC Pros:

- Standard language (up to 90s, every planner had its own input . . .)
- Large set of standard benchmarks; standard competitive setting
- Awards and excitement

IPC Con 1: not nearly as important as it's made out to be!

- Setting not representative of (most?) applications
- Many domains, but impossible to cover *everything*
- "Award" is (a) a very blunt "results summary" and (b) a function of the benchmarks

IPC Con 2: very particular experiment design!

- Spoon-feeds current planners to increase participation and match their performance
- Challenges search not anything else (pre-processing . . .)
- No controlled scaling (scales everything at once)

Take-Home Message

- IPC-style experiments setup is a tradition . . .
- . . . sticking to which is suited as a standard for comparing competitive performance.

- But not for anything else!
- (On top of usual IPC tests) do whatever is suited for determining advantages/disadvantages in *your* context!

- . . . and please don't be that reviewer.

Outline

- Evaluation? What's that?
- On the IPC and other bizarre rituals
- **Do your homework!**
- Buried beneath tons of data
- The black art of counting black sheep
- Understanding the world

Homework

Some (simple?) rules to heed in experimentation (with planning systems).

- (Read Malte's papers, do whatever he does.)
- Look at Toby Walsh's web page:
  `http://www.cse.unsw.edu.au/~tw/empirical.html`
- IJCAI'01 tutorial on empirical methods in AI:
  `http://www.cse.unsw.edu.au/~tw/ijcai2001.ppt`
- "How Not To Do It":
  `http://www.cse.unsw.edu.au/~tw/hownotto.pdf`
- Paul Cohen, "Empirical Methods for AI", MIT Press, 1995

The Four Commandments, Revisited

1. Have a hypothesis.
2. Be careful (with statistics/raw data/cut-offs/summarization).
3. Don't change two things at once!!!
4. Report negative results!!!

Commandment 1: Have a hypothesis.

What am I trying to show?

- Trivial? I reviewed lots of papers where this wasn't clear or where the experiment design wasn't suitable.
- No names here . . . anyone knows an example from myself?
- Cohen, survey of 150 AAAI papers: "Only 16% of the papers offered anything that might be interpreted as a question or a hypothesis."

- No issue if all you investigate is competitive performance . . .

  "**H1***: FF is faster than HSP.*"

- . . . more interesting if you wish to dig deeper!

  "**H2***: FF is faster than HSP because of helpful actions pruning.*"

Hypothesis Testing in a Nutshell

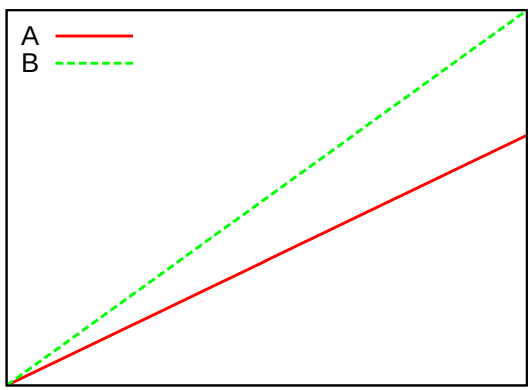
From IJCAI'01 tutorial:

- Example: toss a coin ten times, observe 8 heads. Is the coin fair, i.e., what is its long run behavior? And what is your residual uncertainty?
- You say, "If the coin were fair, then eight or more heads is pretty unlikely, so I think the coin isnt fair."
- Like proof by contradiction: Assert the opposite (the coin is fair), show that the sample result (8 heads) has low probability p, reject the assertion with residual uncertainty related to p.
- For a comprehensive overview, please consult IJCAI'01 tutorial
- For full details, consult a book ...

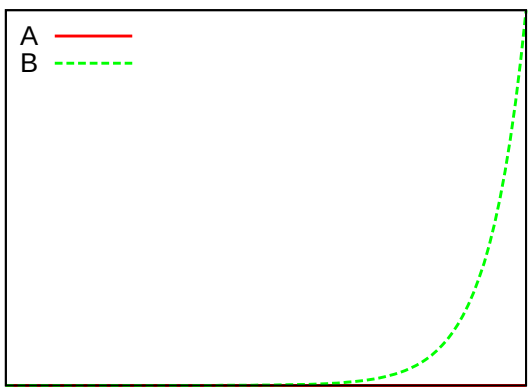
Commandment 2(a): Be careful with statistics.

Am I using the right statistical test?

- *Are the underlying assumptions justified?*
- My first exposure to statistics: is A faster than B in a domain?
- Ran "Dependent t-test for paired samples": $t = \frac{\overline{X_D}}{s_D / \sqrt{n}}$



"yes"



"no"

- This test has no notion of "scaling" ...
- ... and assumes that X_D follows a normal distribution

Commandment 2(b): Be careful with raw data.

Look at the raw data, not only at summaries!

- *Is there a phenomenon not visible at summary level?*
- Example: “exceptionally hard cases” in search – rare cases several orders of magnitude harder than similar instances
- Aka “Heavy-tailed behavior” [Carla Gomes et al, CP’97, . . .]
- Does not appear in median, may not be evident in mean!
- Quotes Gent et al “How To Not Do It”/IJCAI’01 tutorial:

  “We missed them until they hit us on the head when experiments crashed. Old data on smaller problems showed clear behaviour.”

  “We thought the program had crashed so we killed the job . . . the next day the same thing happened with new data, and we realized that some problems were remarkably difficult.”

Commandment 2(c): Be careful with cut-offs.

From IJCAI'01 tutorial:

Wind speed vs. forest fire containment time (max 150 hours):

3	120	55	79	10	140	26	15	110	12
6	78	61	58	81	71	57	21		
9	62	48	21	55	101				

What's the problem??

Cut-offs may bias the sample!

- A lot of high wind fires take $>$ 150 hours to contain ...
- ... those that don't are similar to low wind fires

- This kind of thing may happen in search just as well

Commandment 2(d): Be careful with summarization.

The best summarization method depends on the situation.

- Median: sample point "in the middle of" distribution
- Is often more robust than the mean
- (Well, can be a mixed blessing – heavy-tails)
- Especially funny: mean of ratios, like $\frac{runtime(A)}{runtime(B)}$
- Arithmetic mean of 2 and 0.5 is 1.25 . . . !
- Thus for data A=2,B=1; A=1,B=2 we get that A is "better" than B since mean of $\frac{A}{B} > 1$ . . . *and vice versa for* $\frac{B}{A}$ *. . . !*
- [Example due to Malte Helmert]
- Geometric mean: $\sqrt[n]{D_1 * \cdots * D_n}$

Commandment 3: Don't change two things at once!!!

- You will not know where the new behavior comes from
- Trivial? I've seen various papers proposing search heuristic A vs. old B, and then compared planners X and Y where X used A on search C, and Y used B on search D.

- If you wish to know the effect of options $O_1, \ldots, O_n$, then you need to run experiments on each configuration $C \in O_1 \times \cdots \times O_n$
- Called "ablation studies" or "factorial experiment"
- Simplified: $C \in \{o_1\} \times \cdots \times \{o_{k-1}\} \times O_k \times \{o_{k+1}\} \times \cdots \times \{o_n\}$
- However, option-interactions are often important!

- Examples: [Hoffmann&Nebel, JAIR'01 Sec 8.3.2; Röger&Helmert, ICAPS'10]

Ablation studies are the ONLY means to evaluate YOUR NEW IDEA, not only whether in sum it "beats" a completely different technique!

Commandment 4: Report negative results!!!

What are the advantages – and the disadvantages!!! – of the technique I'm proposing here?

- In the good old days, "cherry-picking" was not only a travellers' job in Australia ...
- (Even better now, no? "4 out of 40" ...)
- Gold medal for "not hiding bad results" goes to Patrik Haslum

- Negative results can be illuminating ...

  (e.g. FF JAIR'01 paper shows uselessness in rnd SAT formulas)
- ... and outright exciting!

  (e.g. [Domshlak&Hoffmann&Sabharwal, JAIR'09]: hopeless results spiced up by observation that "abstraction can never improve the best-case resolution refutation size")

A Cooking Recipe

1. Define objectives and hypotheses
2. Design experiment to meet these
   - 2.1 Avoid biasing outcome by settings, e.g. cut-offs
   - 2.2 To distinguish A from B, change nothing but A and B
3. Run limited samples to calibrate parameters
4. Run experiment
5. Look at raw data to get intuitive understanding
6. Design data analysis
   - 6.1 Be careful to properly use summarization/statistics
7. Understand analysis outcome
8. **if** unexpected behavior **then** goto 1
9. **if** something fishy **then** goto 2
10. **if** conclusions not crystal clear **then** goto 3
11. Report all results including negative ones

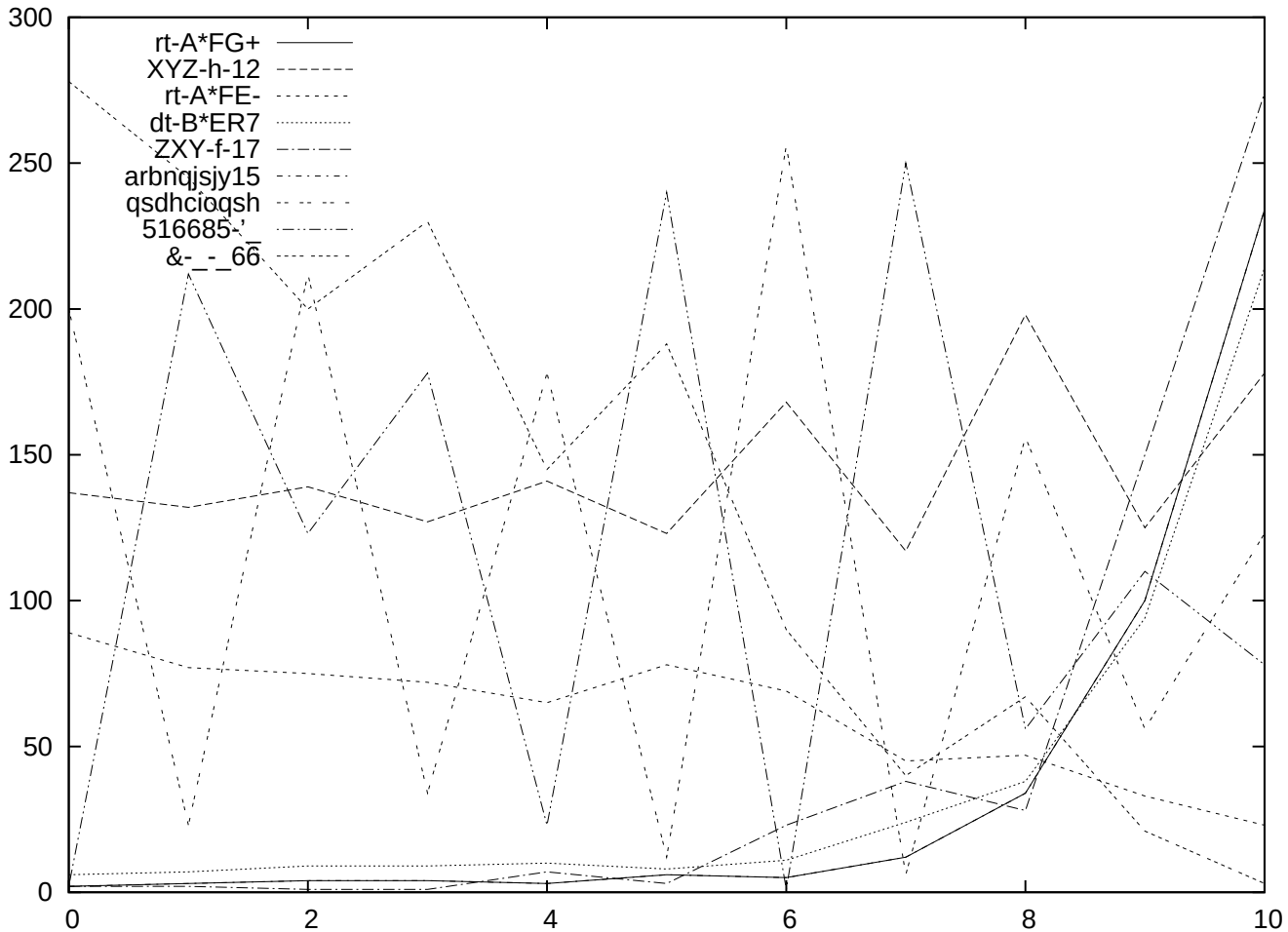
Outline

- Evaluation? What's that?
- On the IPC and other bizarre rituals
- Do your homework!
- **Buried beneath tons of data**
- The black art of counting black sheep
- Understanding the world

Buried beneath tons of data

- Anybody can generate 7 GB of data . . .
- . . . or much more than that,
  in case you're doing a factorial experiment . . .
- *. . . but how to extract the relevant observations?*
- . . . and present them within 2 pages conference paper?
- Yes of course you need to summarize . . .
- . . . but how to? $\Longrightarrow$ *understand* first!
- Vicious circle: need to summarize in order to understand in order to decide how to summarize . . .
- Take evolutionary approach

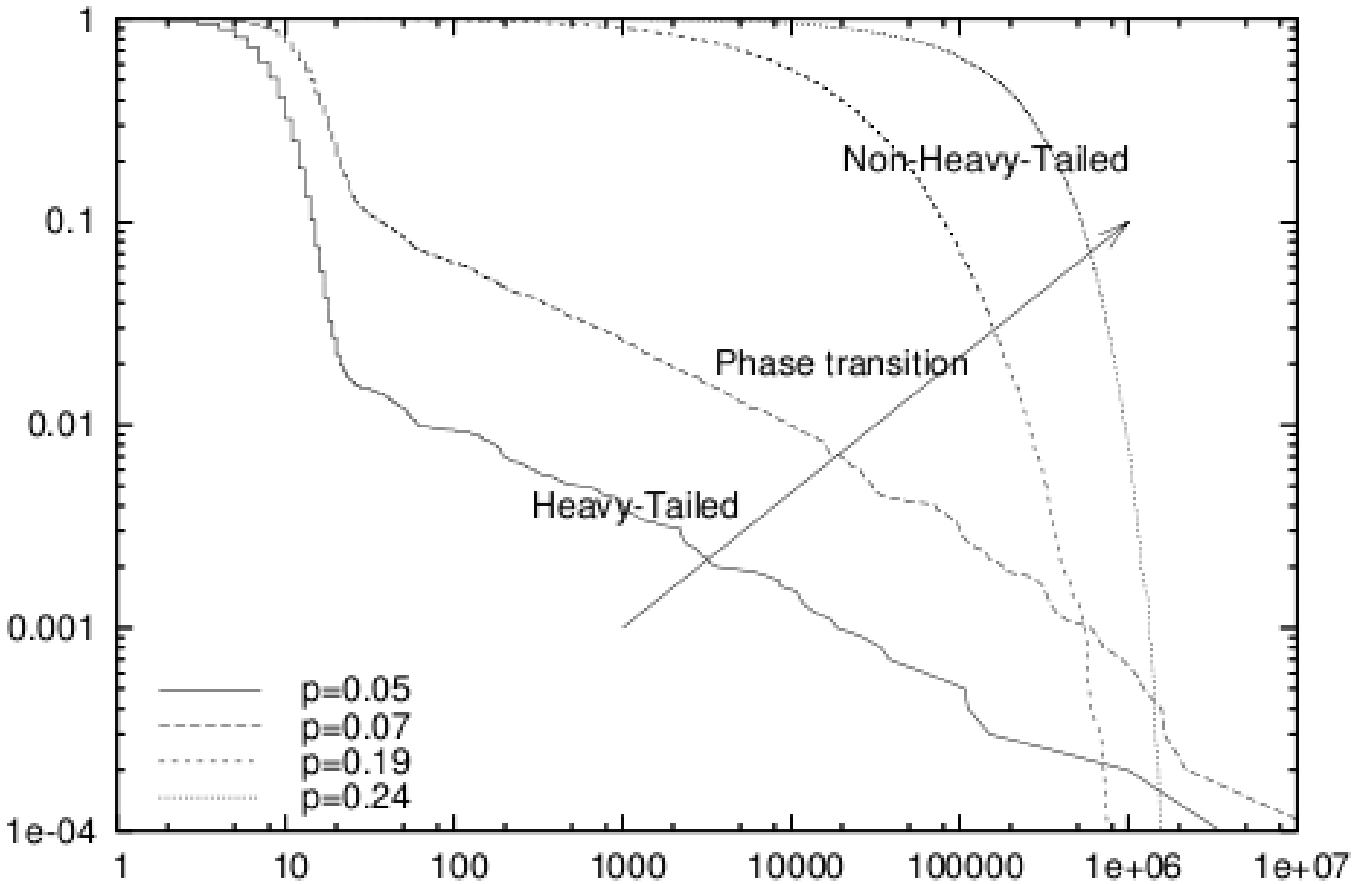
Burying the reader beneath tons of data . . .



Burying the reader beneath tons of data . . .

Domain	Lorg	Linv	Aorg	Ainv	INorg	INinv	IR	BestOf
Airport	0.706	1.360	0.716	1.187	0.743	1.028	0.879	2.182
Assembly	0.835	0.835	1.082	0.850	1.082	0.851	0.835	1.268
Blocks	0.996	1.141	0.997	1.154	0.996	1.049	1.127	1.374
Depots	0.882	2.048	1.499	1.194	0.787	1.633	1.042	2.624
Driverlog	1.197	1.197	1.100	1.100	1.245	1.114	1.090	1.269
FreeCell	0.946	1.007	1.004	1.000	0.948	0.985	0.928	1.455
Grid	0.994	0.987	1.025	2.521	1.513	1.212	2.189	2.647
Log	0.802	0.858	0.805	0.858	0.800	0.812	0.858	0.890
Mic-ADL	0.999	0.996	1.061	1.045	1.000	1.003	1.082	1.186
Mic-Sim	1.209	1.198	1.602	3.554	1.018	1.126	3.554	3.554
Mic-STR	2.591	2.843	1.451	2.843	1.083	1.118	2.843	2.846
MPrime	0.818	0.818	0.813	0.818	0.795	0.795	0.818	0.818
Mystery	0.871	0.871	0.860	0.871	0.826	0.826	0.871	0.871
Openst	1.028	1.029	1.036	1.030	1.017	1.030	1.057	1.063
Optical-T	0.596	0.596	0.662	0.434	0.688	0.600	0.358	0.688
Pathways	1.042	1.042	1.042	1.000	1.042	0.999	1.044	1.060
Philos.	0.722	0.722	0.054	0.133	0.671	0.146	0.041	0.759
Pipe-NoT	0.856	0.647	0.737	0.960	0.853	0.734	0.745	1.807
Pipe-T	0.752	0.846	0.723	0.839	0.899	0.688	0.789	1.770
PSR-L	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.001
PSR-M	1.008	1.000	1.000	1.000	1.000	1.000	1.000	1.010
PSR-S	1.007	0.994	1.012	1.003	1.016	1.005	1.020	1.043
Rovers	1.000	0.992	0.878	0.991	0.994	0.979	1.004	1.216
Satellite	1.028	1.024	0.920	1.146	1.036	1.011	1.225	1.272
Schedule	0.956	1.052	1.942	1.044	0.993	1.035	1.026	2.262
Storage	0.882	0.699	0.929	0.785	0.909	0.781	0.958	1.025
TPP	2.742	2.815	2.814	2.519	2.702	2.707	2.905	3.381
Trucks	1.160	1.161	0.785	2.526	0.674	1.686	1.564	3.634
Zenotravel	0.848	0.848	0.848	0.848	0.848	0.848	0.848	0.848

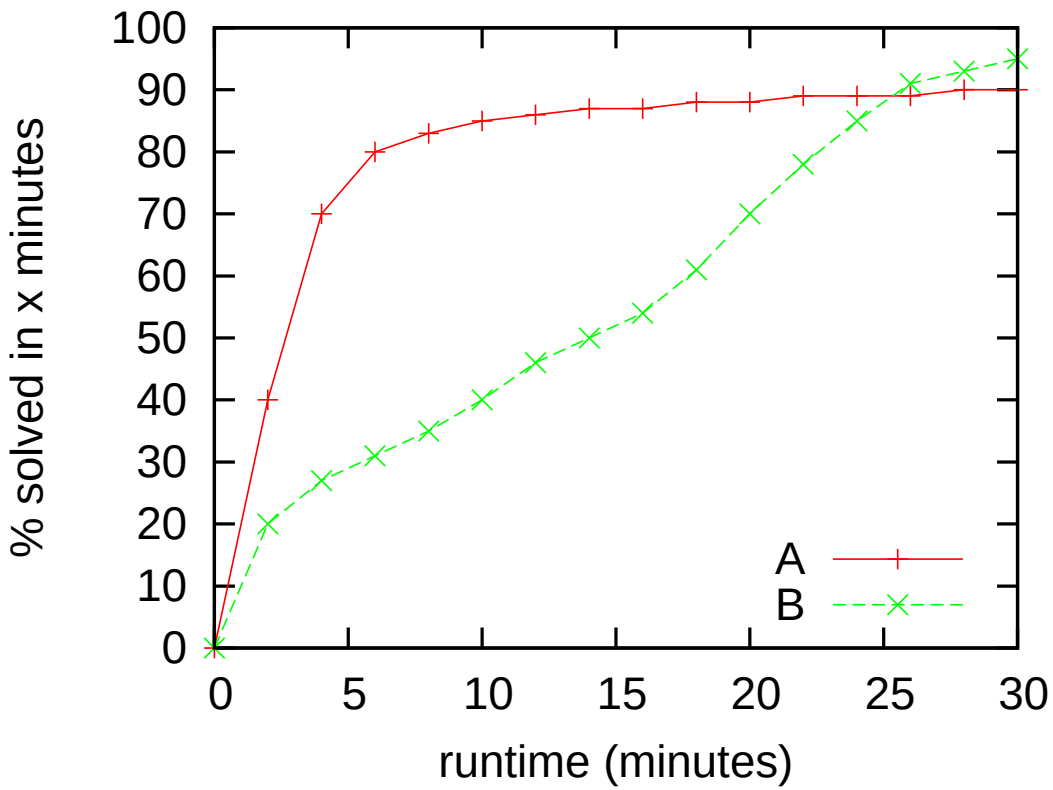
. . . showing clearly the relevant observations!



[Gomes et al., Constraints'05]

Coverage

Planner A	Planner B
90%	95%



Factorial Experiments $C \in O_1 \times \cdots \times O_n$ Example 1

- [Hoffmann&Nebel, JAIR'01]
- Interpolating between FF and HSP:
  - $O_1 = \{h^{FF}, h^{add}\}$
  - $O_2 = \{\textit{Enforced Hill-climbing}, \textit{Hill-climbing}\}$
  - $O_3 = \{\textit{Helpful actions}, \textit{None}\}$
- $2^3 = 8$ combinations
- Not too bad?

Interpolating between FF and HSP

How Not To Do It:

	Hill-climbing				Enforced Hill-climbing			
	All Actions		Helpful Actions		All Actions		Helpful Actions	
	time	length	time	length	time	length	time	length
HSP distance	9.5%	4.2%	9.5%	0.0%	4.2%	4.2%	9.5%	0.0%
FF distance	19.0%	23.8%	23.8%	23.8%	38.1%	28.6%	33.3%	33.3%

Figure 10: Comparison of related planners when using HSP estimates in difference to using FF goal distance estimates. Performance is compared in terms of *percentage of domains in our test suite* where the corresponding distance estimate resulted in significantly better performance than the other one.

(From initial JAIR submission; “significantly better” decided by hand)

Interpolating between FF and HSP

Significant per-domain improvements/deteriorations:

	F				E				H			
domain	−−	−E	H−	HE	−−	−F	H−	HF	−−	−F	E−	EF
Assembly	+	+	+	+		−	+	+	+	+	+	+
Blocksworld-3ops	+		+	−	+	−	+	+	+	+	+	+
Blocksworld-4ops		+			−	−	−			−	+	+
Briefcaseworld	+	−			−	−			−	−	−	−
Bulldozer	+	+	−		−	−	−		−	−	−	−
Freecell	+	+	+	+	+	+	+	+	+	+	+	+
Fridge	−				−	−	−	−	+	+	+	+
Grid	+		+	+	+		+	+	+	+	+	+
Gripper	+	+	+	+	−	+	+	+	+	+	+	+
Hanoi		+			+	+	+	+				
Logistics	−	−	+			−	+	+	+	+	+	+
Miconic-ADL		+		+	+	+	+	+		+	+	+
Miconic-SIMPLE	+	+	+		+	+	+	+	+	+	+	+
Miconic-STRIPS	+	+	+	+		+	+	+	+	+	+	+
Movie					+	+	+	+				
Mprime	+	+	+				+	+	+	+	+	+
Mystery					+		+	+	+			
Schedule			+	+		−	+	+	+	+	+	+
Tireworld	+		+	+		−	+	+		+	+	+
Tsp	+	+	+	+	+	+	+	+	+	+	+	+

Factorial Experiments $C \in O_1 \times \cdots \times O_n$
Example 2

- [Röger&Helmert, ICAPS'10]
- How to combine heuristic estimators?
  - $O_1 = 2^{\{h^{FF}, h^{CG}, h^{cea}\}}$
  - $O_2 = \{max, sum, tie\text{-}break, pareto, alternation, alternation\text{-}TB\}$
- $4 * 6 + 3 * 1 = 27$ combinations ...
- (Granted, large n more headache than large $|O_i|$)

How to combine heuristic estimators?

	Coverage	Quality	Speed	Guidance
h^{cea}	74.62	68.67	65.27	65.65
h^{FF}	73.85	70.55	66.81	64.07
h^{CG}	72.66	65.36	64.16	60.43
h^{cea}, h^{FF}				
Maximum	72.69	67.26	62.15	64.02
Sum	73.75	68.42	63.75	*65.67
Tie-breaking	72.44	67.14	62.90	64.67
Pareto	*76.20	*70.71	66.32	*68.90
Alternation	***77.95**	***73.70**	***67.84**	***70.14**
Alternation-TB	*75.42	70.21	66.23	*68.48
h^{FF}, h^{CG}				
Maximum	*74.76	68.76	65.29	*65.08
Sum	*75.01	67.99	65.41	*65.35
Tie-breaking	72.59	66.13	64.66	*64.41
Pareto	*74.93	67.84	65.87	*66.19
Alternation	***78.73**	***73.28**	***69.22**	***69.28**
Alternation-TB	*74.75	67.45	66.06	*66.18
h^{cea}, h^{CG}				
Maximum	74.06	67.95	63.63	65.51
Sum	*74.76	67.70	64.12	*65.67
Tie-breaking	73.78	67.41	63.36	64.99
Pareto	74.52	67.70	64.48	*66.52
Alternation	***75.20**	***69.18**	64.42	*66.39
Alternation-TB	74.58	67.79	**64.59**	***66.59**
h^{cea}, h^{FF}, h^{CG}				
Maximum	72.21	66.54	61.13	63.71
Sum	73.47	67.52	62.98	65.24
Tie-breaking	72.49	66.95	61.90	64.34
Pareto	*76.29	70.16	66.01	*69.18
Alternation	***79.80**	***74.62**	***68.56**	***71.91**
Alternation-TB	*76.05	70.15	65.83	*69.16

Cross-domain summary:

- Coverage score: 100 solved, 0 else
- Quality score: like IPC'08, i.e., $100 * q^*/q$
- Speed score: interpolate logarithmically between 1 sec and time-out 1800 sec
- Guidance score: interpolate logarithmically between 100 and 1000000 expansions

How to combine heuristic estimators?

Domain	h^{cea}	h^{FF}	h^{CG}	Max.	Sum	Tie-br.	Pareto
Airport	–3	+7	+18	+2/–1	+3/–2	+7	+6
Assembly	+20	+15	+24	+20	+20	+20	+14
Depot	+2	–1	+3/–1	+2	+3/–1		–2
Driverlog	+1	+1	+1	+2	+1	+2	+2
FreeCell	+1/–1	+3/–2	+10/–1	+4/–1	+3/–1	+5	–1
Grid	+1	+1	+1	+1		+1	
Logistics-1998	–4	+4	–4			–1	
Miconic-FullADL	–1	+4/–1	+2	–1	+1/–1	–1	+1
MPrime		+8	–1	+6			–1
Mystery	+1	+3/–1		+1			–1
Openstacks	+5		+4	+5	+5	+5	
OpticalTelegraphs			+3				+2
Pathways	+5	+7	+4	+5	+6	+5	+5
Pipesw.-NoTankage	+13	+7	+15/–1	+14	+12	+12	+9/–1
Pipesw.-Tankage	+4/–1	+4/–3	+7/–3	+4	+4/–1	+5/–2	+2/–2
PSR-Large	–2	+1/–1	–2	+1	+3	+3	+2
PSR-Middle					+1	+1	+1
Rovers	+7	+5	+7	+7	+8	+8	+7
Satellite	–3	+1	–9				
Schedule	+9	+3/–12	+9	+9	+9	+9	+9
Storage	+2	–1	+1	+2		+2	
TPP	+3/–4	+3	+1	+3	+3	+5	+2/–4
Trucks		+2	+4/–1		+2		+1/–1
Total	+74/–19	+79/–22	+114/–23	+88/–3	+84/–6	+90/–4	+63/–13

Per-domain zoom-in:

Coverage differences when switching to Alternation ("+" new solved, "–" now unsolved).

Outline

- Evaluation? What's that?
- On the IPC and other bizarre rituals
- Do your homework!
- Buried beneath tons of data
- **The black art of counting black sheep**
- Understanding the world

LPG vs. FF in “Mystery”

task	LPG	FF
prob-01	0.01	0.00
prob-02	0.22	0.00
prob-03	0.04	0.00
prob-04	–	–
prob-05	–	–
prob-06	86.33	–
prob-08	–	–
prob-09	0.08	0.01
prob-10	14.41	–
prob-11	0.01	0.00
prob-12	–	–
prob-13	–	–
prob-14	990.78	1.72
prob-15	1.39	0.04
prob-16	–	–
prob-17	1.29	0.03
prob-19	0.38	0.73
prob-20	0.27	0.02
prob-21	–	–
prob-22	–	–
prob-23	–	–
prob-24	–	–
prob-25	0.00	0.00
prob-26	0.06	0.04
prob-27	12.05	0.00
prob-28	0.00	0.00
prob-29	0.05	0.00
prob-30	0.95	0.01

Counting Black Sheep

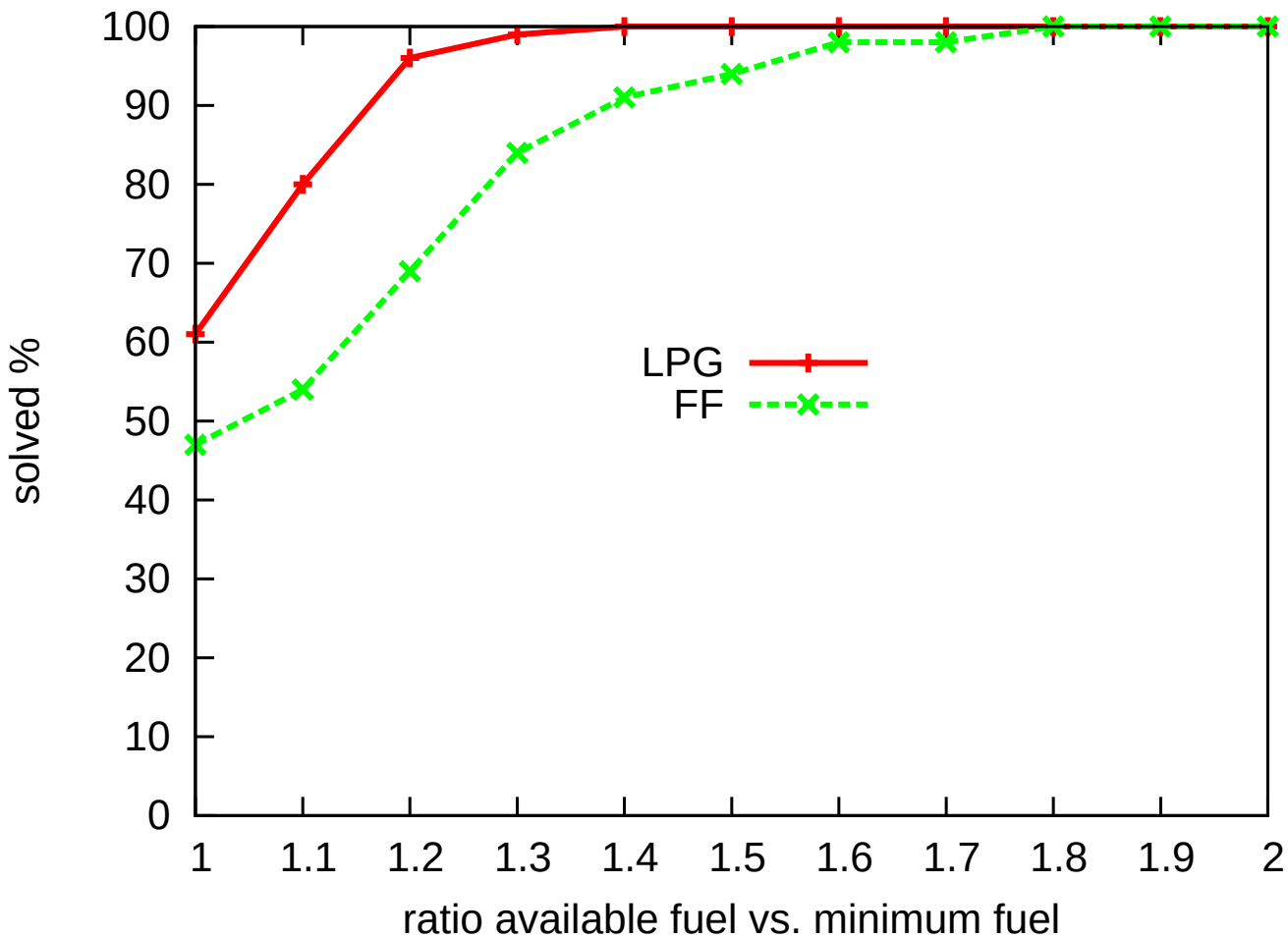
An astronomer, a physicist and a mathematician are on a train in Scotland. The astronomer looks out of the window, sees a black sheep standing in a field, and remarks:

“ How odd. Scottish sheep are black.”

“ No, no, no!” says the physicist. “ Only some Scottish sheep are black.”

The mathematician rolls his eyes at his companions’ muddled thinking and says, “ In Scotland, there is at least one sheep, at least one side of which is black.”

LPG vs. FF in “NoMystery”

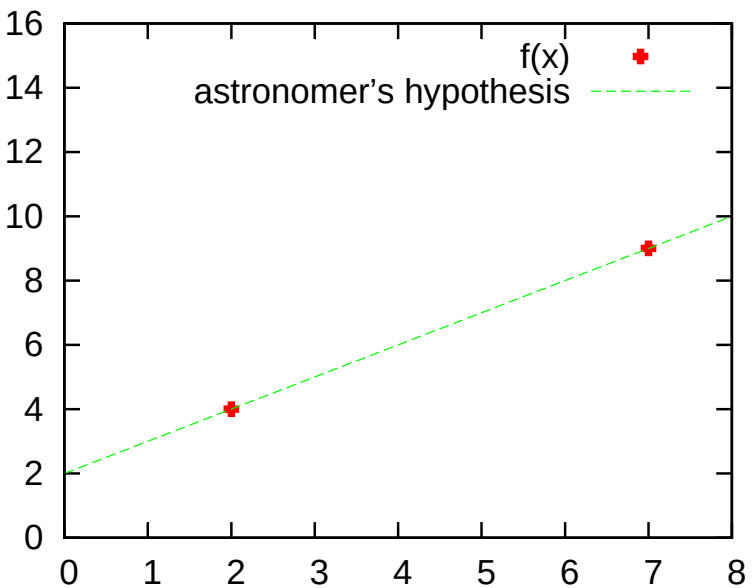


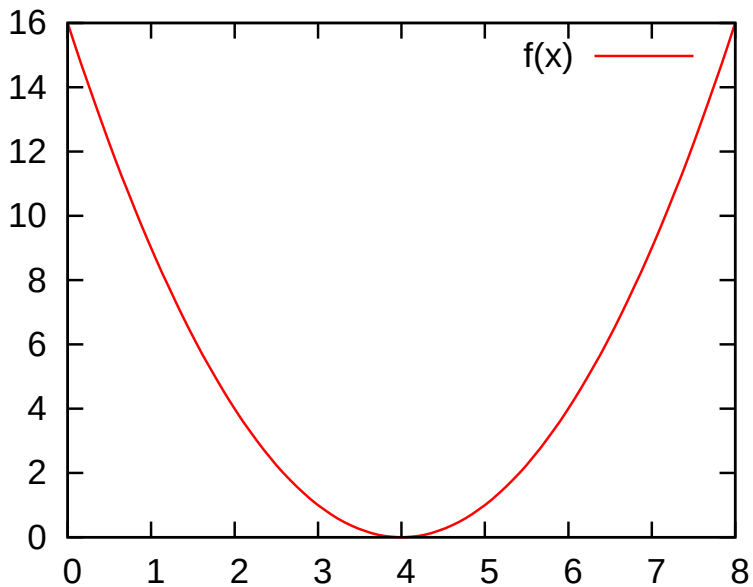
The “Performance Function”

- Performance is a function of algorithm and planning problem:

$$f(A, P)$$

- Running a test $\Longrightarrow$ one point of that function
- Experiments: “What is the form of $f(A, P)$?”





The “Performance Function”, ctd.

Why is it difficult to determine “the form of $f(A, P)$”?

(1) Form a priori completely unknown (unlike $f(x) = ax^2 + bx + c$)

(2) “A” is highly complex/structured

(3) “P” is highly complex/structured

(2,3) $\Longrightarrow$ want to know “what kind of” algorithm/task:

$$p(\mathcal{F}_1^A(A), \ldots, \mathcal{F}_n^A(A), \mathcal{F}_1^P(P), \ldots, \mathcal{F}_m^P(P))$$

- $\mathcal{F}^A/\mathcal{F}^P$: algorithm/problem features
- What features? All relevant ones, ideally
- Which are those? It’s a kind of magic . . .

The Performance Function in NoMystery

What did we do better in NoMystery?

$$p(\ \mathcal{F}_1^A(A) \in \{\mathrm{FF}, \mathrm{LPG}\},\\ \mathcal{F}_1^P(P) = \text{size, roadmap, etc.},\\ \mathcal{F}_2^P(P) = \text{avail vs. min fuel ratio}\ \ )$$

- We changed exactly one problem feature – $\mathcal{F}_2^P(P)$
- In Mystery, unsystematically changed everything
- Same for IPC! No notion of "problem features", no good for counting sheep!

"There exists a sheep with a black side" vs.
"The more gene X has property Y, the blacker is the sheep"

Changing a single algorithm feature $\mathcal{F}_i^A$ at a time

== ablation studies!

	F				E				H			
domain	−−	−E	H−	HE	−−	−F	H−	HF	−−	−F	E−	EF
Assembly	+	+	+	+		−	+	+	+	+	+	+
Blocksworld-3ops	+		+	−	+	−	+	+	+	+	+	+
Blocksworld-4ops		+			−	−	−			−	+	+
Briefcaseworld	+	−			−	−			−	−	−	−
Bulldozer	+	+	−		−	−	−		−	−	−	−
Freecell	+	+	+	+	+	+	+	+	+	+	+	+
Fridge	−				−	−	−	−	+	+	+	+
Grid	+		+	+	+		+	+	+	+	+	+
Gripper	+	+	+	+	−	+	+	+	+	+	+	+
Hanoi		+			+	+	+	+				
Logistics	−	−	+			−	+	+	+	+	+	+
Miconic-ADL		+		+	+	+	+	+		+	+	+
Miconic-SIMPLE	+	+	+		+	+	+	+	+	+	+	+
Miconic-STRIPS	+	+	+	+		+	+	+	+	+	+	+
Movie					+	+	+	+				
Mprime	+	+	+				+	+	+	+	+	+
Mystery					+		+	+	+			
Schedule			+	+		−	+	+	+	+	+	+
Tireworld	+		+	+		−	+	+		+	+	+
Tsp	+	+	+	+	+	+	+	+	+	+	+	+

Changing a single *problem* feature $\mathcal{F}_i^P$ at a time

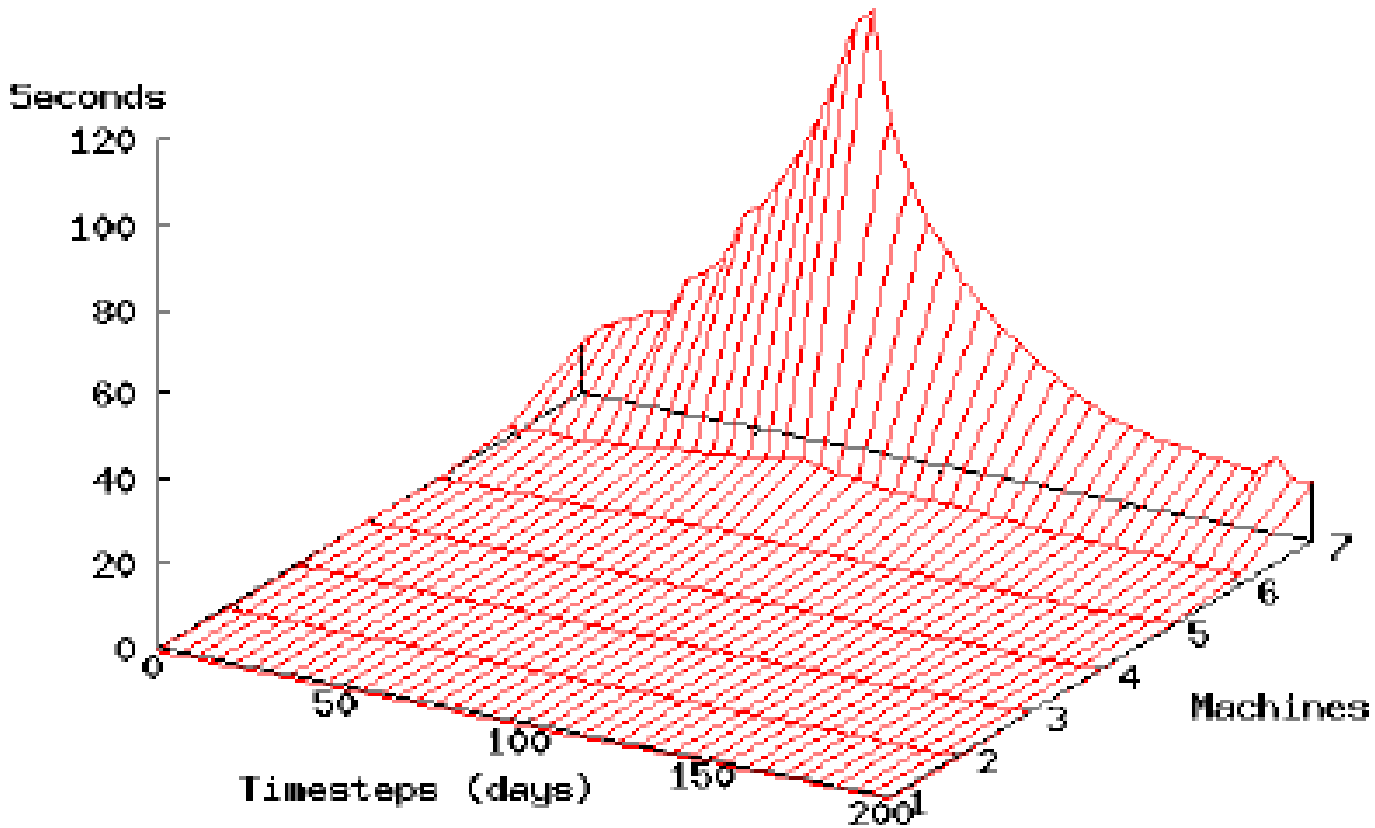
What are useful problem features?

- A simple one: the *domain*
- Presenting results per-domain ≡ vary only $\mathcal{F}_1^P \in \{\text{domains}\}$

- More simple ones: instance size parameters
- Scaling size param ≡ vary only $\mathcal{F}_i^P = \text{number-of-trucks}$ etc.

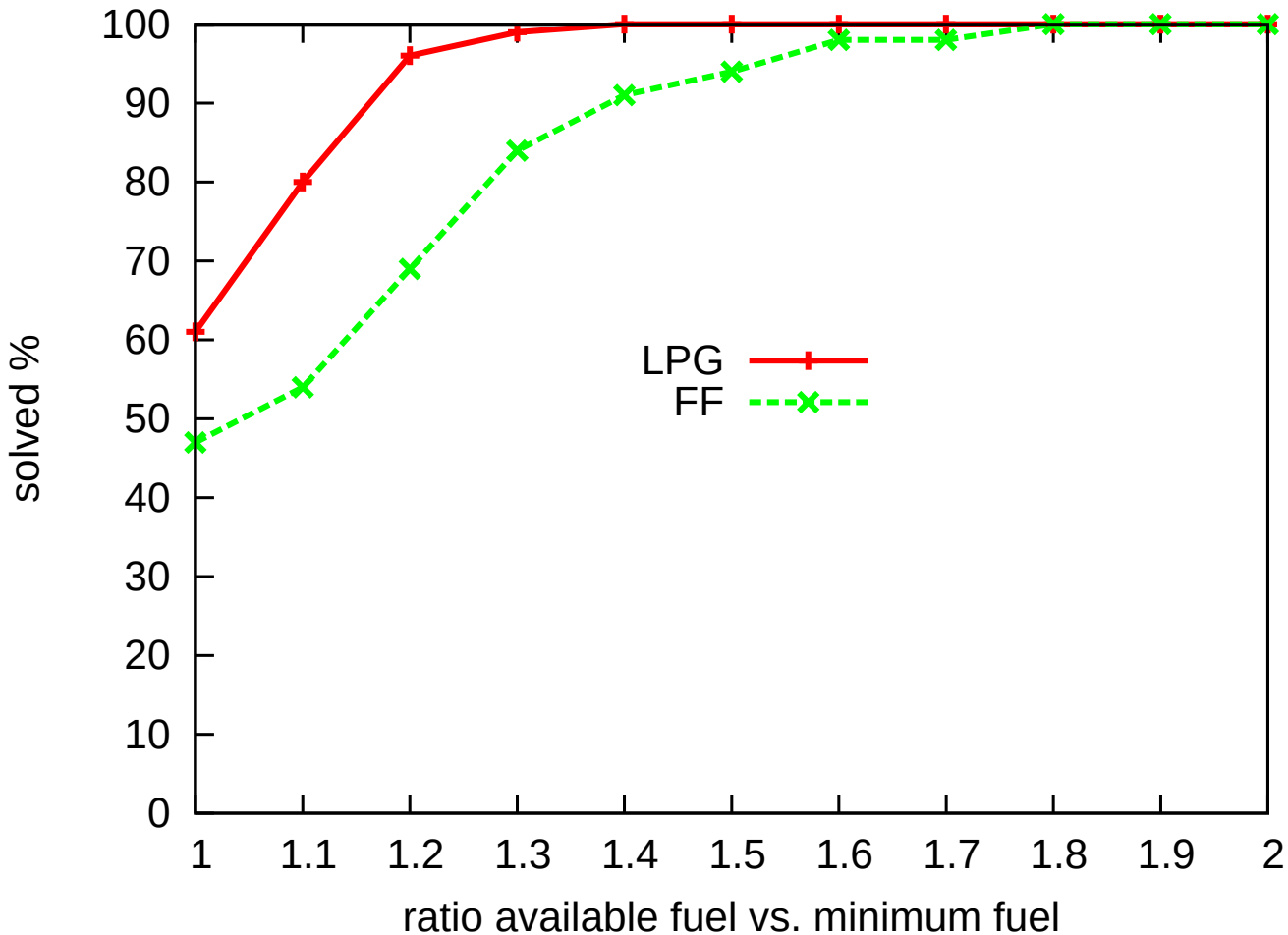
- More subtle $\mathcal{F}_i^P$ relevant to algorithms: an art form!
- Work hard, keep your eyes open, use your intuition, ...
- ... copy from others 

$\mathcal{F}_i^P$ = amount of uncertainty in model



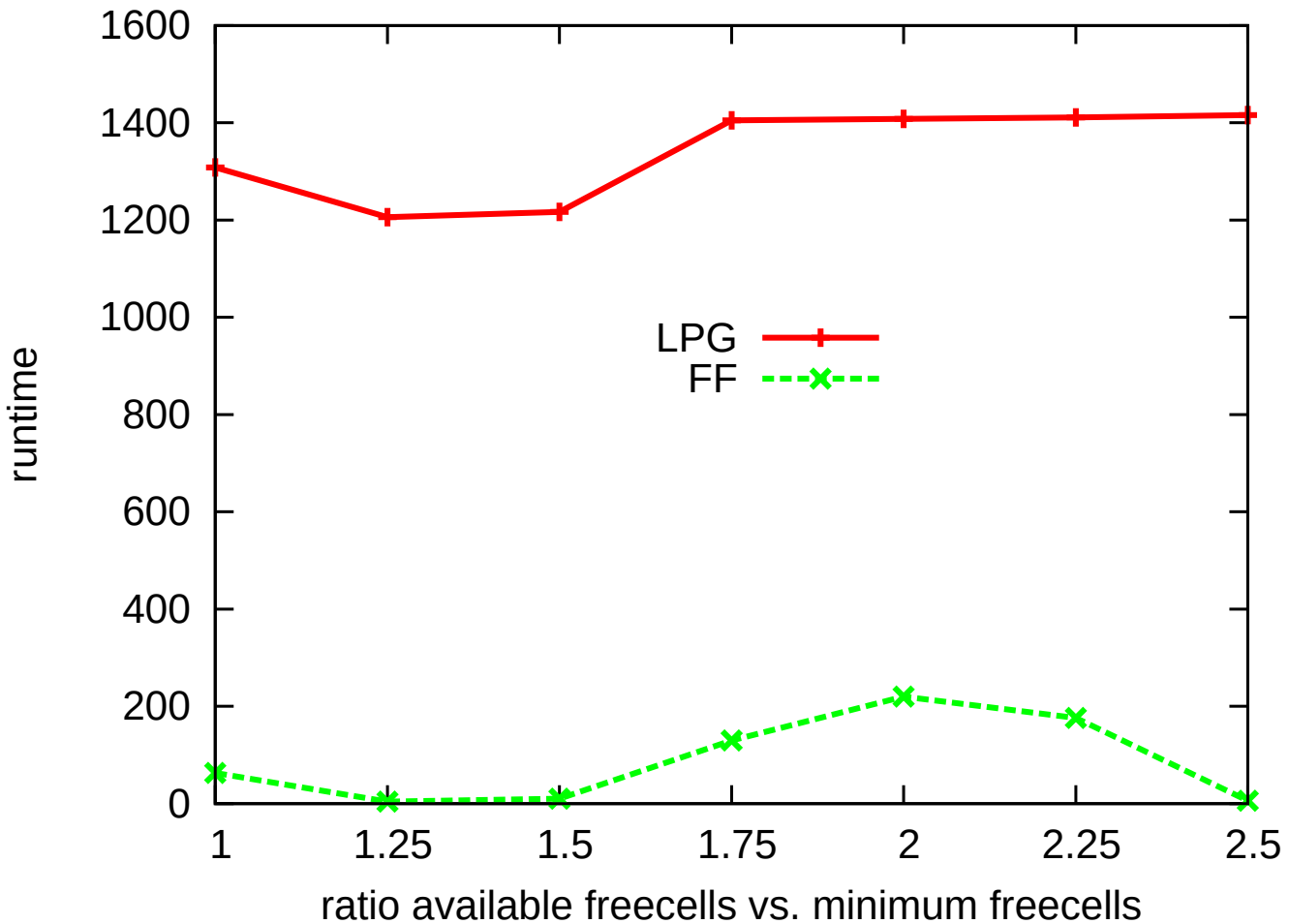
[Sarraute&Buffet&Hoffmann, SecArt'11]

$\mathcal{F}_i^P$ = ratio available fuel vs. minimum fuel



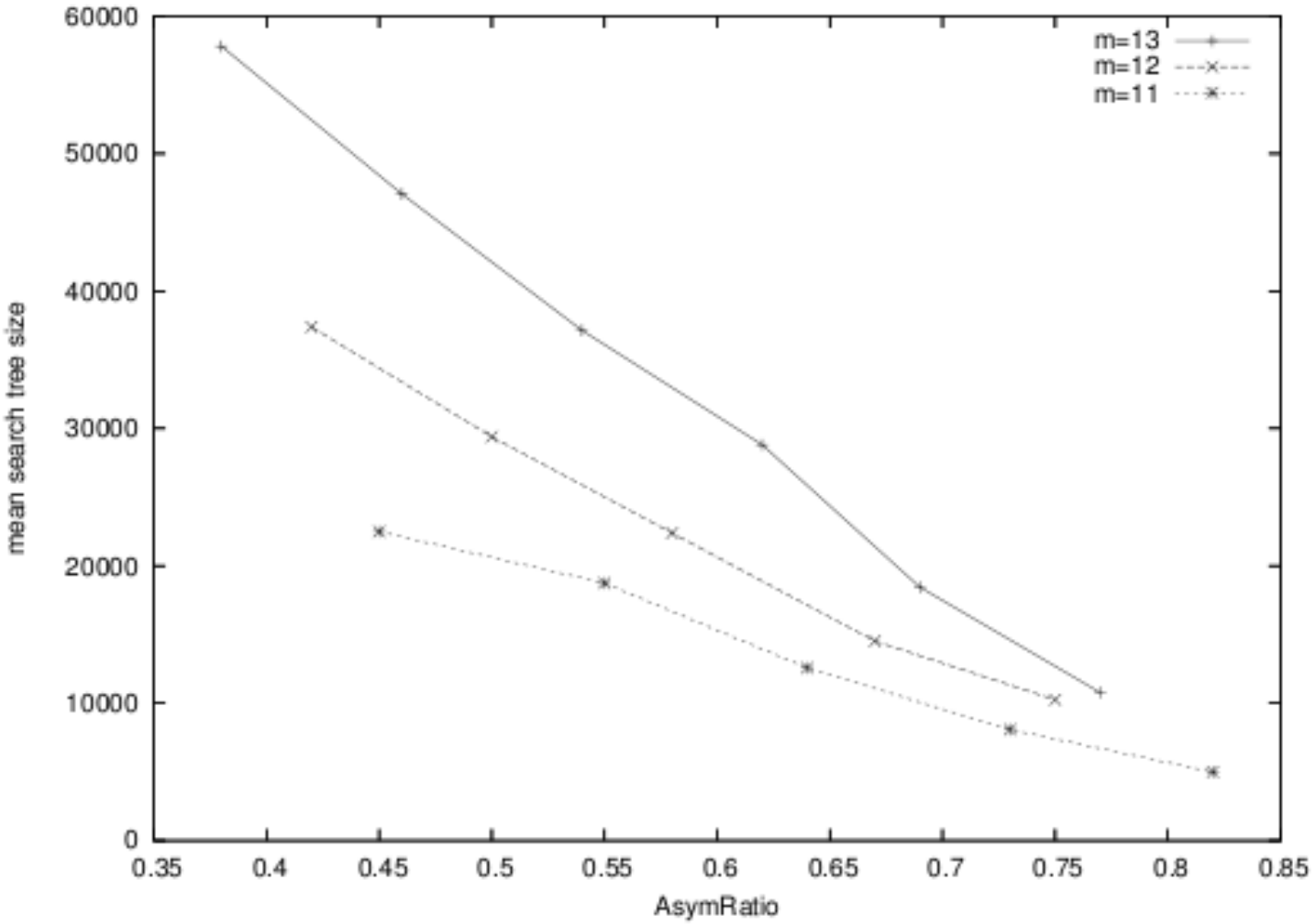
[Hoffmann&Kautz&Gomes&Selman, IJCAI'07]

$\mathcal{F}_i^P$ = ratio available freecells vs. minimum freecells



[Hoffmann, never to be published]

$\mathcal{F}_i^P$ = "AsymRatio" $\frac{max_{g \in G} cost(g)}{cost(\bigwedge_{g \in G} g)}$



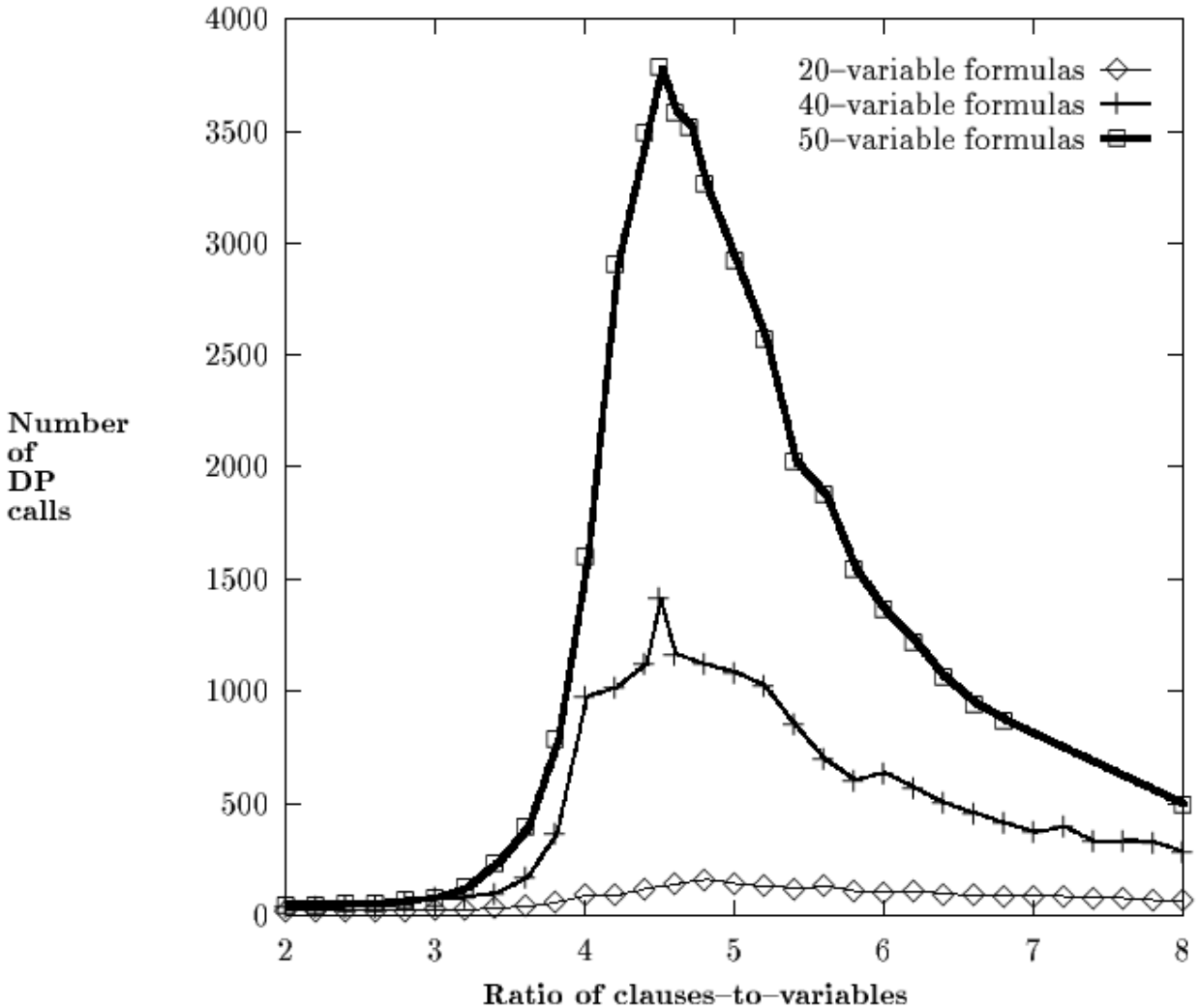
[Hoffmann&Gomes&Selman, LMCS'07]

$\mathcal{F}_i^P$ = "Conformant Width"

	Domain-Parameter	# Unknown Fluents	Width
1	Safe-n combinations	n	1
2	UTS-n locs	n	1
3	Ring-n rooms	$4n$	1
4	Bomb-in-the-toilet-n bombs	n	1
5	Comm-n signals	n	1
6	Square-Center-$n \times n$ grid	$2n$	1
7	Cube-Center-$n \times n \times n$ cube	$3n$	1
8	Grid-n shapes of n keys	$n \times m$	1
9	Logistics n pack m locs	$n \times m$	1
10	Coins-n coins m locs	$n \times m$	1
11	Block-Tower-n Blocks	$n \times (n-1) + 3n + 1$	$n \times (n-1) + 3n + 1$
12	Sortnet-n bits	n	n
13	Adder n pairs of bits	$2n$	$2n$
14	Look-and-Grab m objs from $n \times n$ locs	$n \times n \times m$	m
15	1-dispose m objs from $n \times n$ locs	$n \times n \times m$	m

[Palacios&Geffner, JAIR'09]

$\mathcal{F}_i^P$ = Constrainedness



[Mitchell&Selman&Levesque, AAAI'92]

Outline

- Evaluation? What's that?
- On the IPC and other bizarre rituals
- Do your homework!
- Buried beneath tons of data
- The black art of counting black sheep
- **Understanding the world**

Empirical CS == Natural Science

"In Lincolnshire, summer 1666, an apple fell straight to the ground."

"Everywhere, always, apples fall straight to the ground."

"It's because of gravity!"

Your Empirical CS == Natural Science

Domain	Lorg	Linv	Aorg	Ainv	INorg	INinv	IR	BestOf
Airport	0.706	1.360	0.716	1.187	0.743	1.028	0.879	2.182
Assembly	0.835	0.835	1.082	0.850	1.082	0.851	0.835	1.268
Blocks	0.996	1.141	0.997	1.154	0.996	1.049	1.127	1.374
Depots	0.882	2.048	1.499	1.194	0.787	1.633	1.042	2.624
Driverlog	1.197	1.197	1.100	1.100	1.245	1.114	1.090	1.269
FreeCell	0.946	1.007	1.004	1.000	0.948	0.985	0.928	1.455
Grid	0.994	0.987	1.025	2.521	1.513	1.212	2.189	2.647
Log	0.802	0.858	0.805	0.858	0.800	0.812	0.858	0.890
Mic-ADL	0.999	0.996	1.061	1.045	1.000	1.003	1.082	1.186
Mic-Sim	1.209	1.198	1.602	3.554	1.018	1.126	3.554	3.554
Mic-STR	2.591	2.843	1.451	2.843	1.083	1.118	2.843	2.846
MPrime	0.818	0.818	0.813	0.818	0.795	0.795	0.818	0.818
Mystery	0.871	0.871	0.860	0.871	0.826	0.826	0.871	0.871
Openst	1.028	1.029	1.036	1.030	1.017	1.030	1.057	1.063
Optical-T	0.596	0.596	0.662	0.434	0.688	0.600	0.358	0.688
Pathways	1.042	1.042	1.042	1.000	1.042	0.999	1.044	1.060
Philos.	0.722	0.722	0.054	0.133	0.671	0.146	0.041	0.759
Pipe-NoT	0.856	0.647	0.737	0.960	0.853	0.734	0.745	1.807
Pipe-T	0.752	0.846	0.723	0.839	0.899	0.688	0.789	1.770
PSR-L	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.001
PSR-M	1.008	1.000	1.000	1.000	1.000	1.000	1.000	1.010
PSR-S	1.007	0.994	1.012	1.003	1.016	1.005	1.020	1.043
Rovers	1.000	0.992	0.878	0.991	0.994	0.979	1.004	1.216
Satellite	1.028	1.024	0.920	1.146	1.036	1.011	1.225	1.272
Schedule	0.956	1.052	1.942	1.044	0.993	1.035	1.026	2.262
Storage	0.882	0.699	0.929	0.785	0.909	0.781	0.958	1.025
TPP	2.742	2.815	2.814	2.519	2.702	2.707	2.905	3.381
Trucks	1.160	1.161	0.785	2.526	0.674	1.686	1.564	3.634
Zenotravel	0.848	0.848	0.848	0.848	0.848	0.848	0.848	0.848

Observation:

"In instance $\alpha\beta\gamma$ of domain XYZ, my planner is faster than version ABC of planner foo-bar."

Generalization/Formalization:

"If the instance has property *X* then algorithms of type *Y* have property *Z*."

Explanation:

"It's because of search space property ϕ!"

How Good is Almost Perfect?

[Helmert&Röger, AAAI'08]:

Definition

Let $\mathcal{T}$ be a planning task, and let $c \in \mathbb{N}$. Define the heuristic function $h^* - c$ as $(h^* - c)(s) := \max(0, h^*(s) - c)$. Define $N^c(\mathcal{T})$ as the number of states s where $g(s) + (h^* - c)(s) < h^*(\mathcal{T})$.

$N^c(\mathcal{T})$: number of states that *must* be expanded by A* with almost-perfect heuristic $h^* - c$.

Theorem

In Gripper, $N^1(\mathcal{T}_n)$ grows exponentially with the number of balls. In Miconic-Simple, there exist scaling families of tasks $\mathcal{T}_n$ where $N^4(\mathcal{T}_n)$ grows exponentially with n. In Blocksworld, there exist scaling families of tasks $\mathcal{T}_n$ where $N^1(\mathcal{T}_n)$ grows exponentially with n.

How Good is Almost Perfect?

Observation:

- A* doesn't scale in the IPC instances of trivial domains like Gripper, with any of the known admissible heuristics

Generalization/Formalization:

- The search space of A* must necessarily grow exponentially in these domains, even with almost perfect heuristics
- (In contrast to known tractability results for almost perfect heuristics)

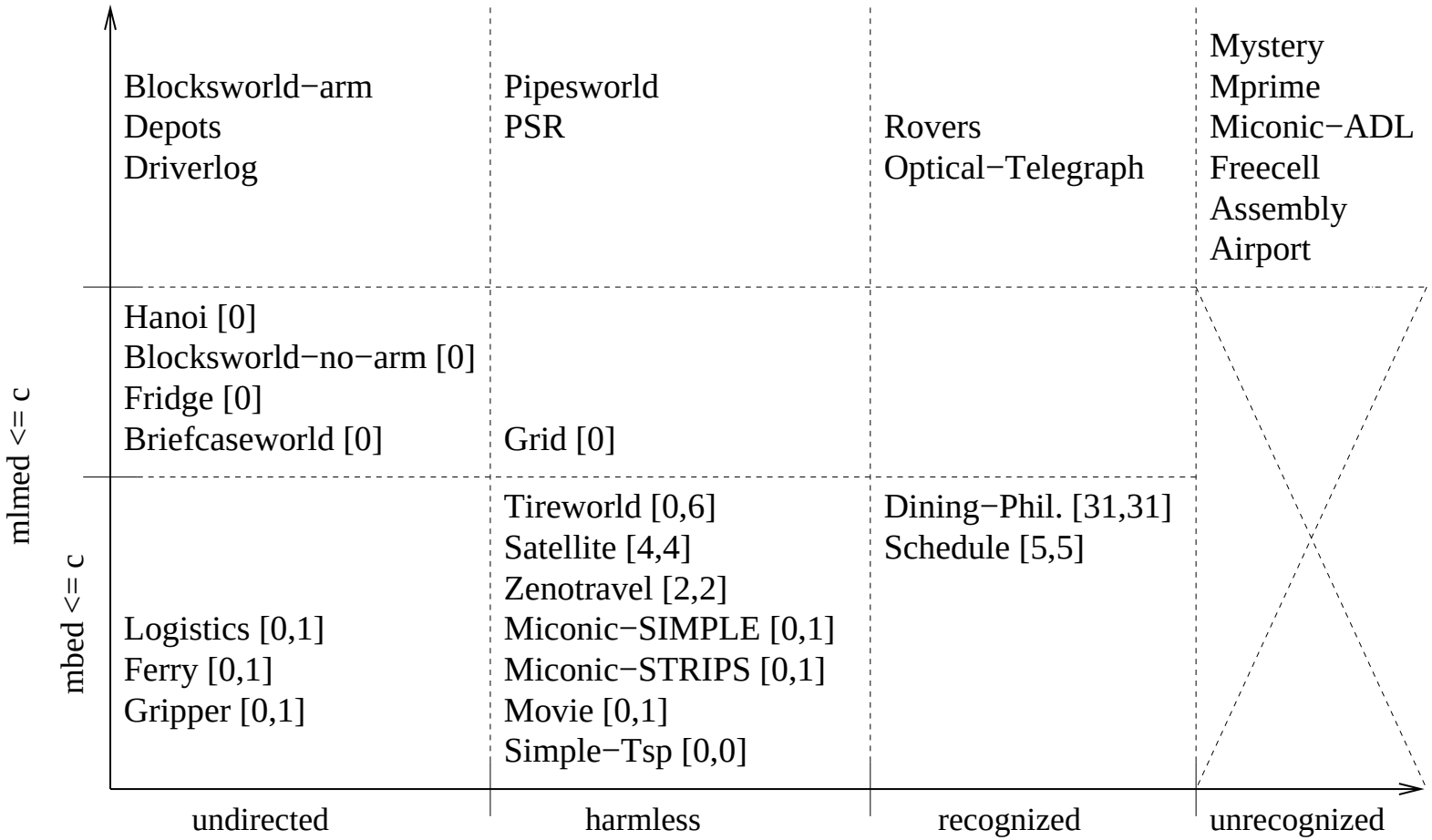
Explanation:

- Goal state can be reached in many different ways (transpositions)
- (Main proof argument)

Best Paper Award at AAAI'08

Where Ignoring Delete Lists Works

[Hoffmann, AIPS'02, JAIR'05]:



h^+ "exit distance" from states on local minima/benches

Where Ignoring Delete Lists Works

Observation:

- Relaxed plan heuristics seem to work well in some domains, but not in others

Generalization/Formalization:

- Taxonomy of domain categories sharing topological properties of idealized heuristic h^+

Explanation:

- Connections between "optimal actions" in real and relaxed versions of respective domains
- (Main proof argument)

2002 Award for Best European Dissertation in AI

Final Punchline

It's about understanding the world

not about "my apple flies faster than yours"

p.s. Are we solving the right problem here?

Natural Language Generation: [Koller&Hoffmann, ICAPS'10]

- Performance: Ok based on trivial modification of FF
- *Why planning? PDDL cheaper to write than code*
- Main issue: PDDL modeling (understand planner reaction)

Attack Path Generation: (with Core Security Technologies)

- Performance: Ok based on easy modification of FF
- *Why planning? PDDL cheaper to write than code*
- Main issue: PDDL modeling (understand planner reaction)

Creating business processes at SAP: [Hoffmann et al, AAAI'10]

- Performance: Ok based on easy adaptation of FF
- Why planning? Flexibility required
- Main issue: "PDDL" modeling (5 years, 200 people, special GUI, design patterns, naming conventions, governance process, review meetings, council supervision, educational training)

References

- "How Not To Do It":
  `http://www.cse.unsw.edu.au/~tw/hownotto.pdf`
- IJCAI'01 tutorial on empirical methods in AI:
  `http://www.cse.unsw.edu.au/~tw/ijcai2001.ppt`
- Toby Walsh's web page on empirical methods in CS and AI:
  `http://www.cse.unsw.edu.au/~tw/empirical.html`
- P. Cohen, "Empirical Methods for AI", MIT Press, 1995.
- C. Domshlak, J. Hoffmann, and A. Sabharwal, *Friends or Foes? On Planning as Satisfiability and Abstract CNF Encodings*, Journal of Artificial Intelligence Research 36: 415-469, 2009.
- C. Gomes, C. Fernandez, B. Selman, and C. Bessiere, *Statistical Regimes Across Constrainedness Regions*, Constraints 10(4): 317-337, 2005.
- C. Gomes, B. Selman, and N. Crato, *Heavy-Tailed Distributions in Combinatorial Search*, Principles and Practice of Constraint Programming, 3rd International Conference (CP'97).

References

- M. Helmert, P. Haslum, and J. Hoffmann, *Flexible Abstraction Heuristics for Optimal Sequential Planning*, Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07).
- M. Helmert, Gabriele Röger, *How Good is Almost Perfect?*, Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08).
- J. Hoffmann, *Local Search Topology in Planning Benchmarks: A Theoretical Analysis*, Proceedings of the 6th International Conference on Artificial Intelligence Planning and Scheduling (AIPS'02).
- J. Hoffmann, *Where Ignoring Delete Lists Works: Local Search Topology in Planning Benchmarks*, Journal of Artificial Intelligence Research 24: 685–758, 2005.
- J. Hoffmann, *Where Ignoring Delete Lists Works, Part II: Causal Graphs*, Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS'11).

References

- J. Hoffmann, C. Gomes, and B. Selman, *Structure and Problem Hardness: Goal Asymmetry and DPLL Proofs in SAT-based Planning*, Logical Methods in Computer Science 3 (1-6), 2007.
- J. Hoffmann, H. Kautz, C. Gomes, and B. Selman, *SAT Encodings of State-Space Reachability Problems in Numeric Domains*, Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07).
- J. Hoffmann and B. Nebel, *The FF Planning System: Fast Plan Generation Through Heuristic Search*, Journal of Artificial Intelligence Research 14: 253–302, 2001.
- J. Hoffmann, I. Weber, and F. Kraft, *SAP Speaks PDDL*, Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10).
- M. Katz and C. Domshlak, *Optimal Additive Composition of Abstraction-based Admissible Heuristics*, Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS'08).

References

- A. Koller and J. Hoffmann, *Waking Up a Sleeping Rabbit: On Natural-Language Sentence Generation with FF*, Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS'10).
- A. Koller and R. Petrick, *Experiences with planning for natural language generation*, Computational Intelligence 27(1): 23-40, 2011.
- D. Mitchell, B. Selman, and H. Levesque, *Hard and Easy Distributions of SAT Problems*, Proceedings of the 10th National Conference of the American Association for Artificial Intelligence (AAAI'92).
- R. Nissim, J. Hoffmann, and M. Helmert, *Computing Perfect Heuristics in Polynomial Time: On Bisimulation and Merge-and-Shrink Abstraction in Optimal Planning*, Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11).

References

- H. Palacios and H. Geffner, *Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width*, Journal of Artificial Intelligence Research 35: 623-675, 2009.
- G. Röger and M. Helmert, *The More, the Merrier: Combining Heuristic Estimators for Satisficing Planning*, Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS'10).
- W. Ruml, M. Do, R. Zhou, and M. Fromherz, *On-line Planning and Scheduling: An Application to Controlling Modular Printers*, Journal of Artificial Intelligence Research 40: 415-468, 2011.
- C. Sarraute, O. Buffet, and J. Hoffmann, *Penetration Testing == POMDP Solving?* Proceedings of the 3rd Workshop on Intelligent Security (SecArt'11), at IJCAI'11.