Everything You Always Wanted to Know About **Planning** (But Were Afraid to Ask)

Jörg Hoffmann

INRIA
Nancy, France

October 7, 2011


Woody Allen's
Joerg Hoffmann's
Was Sie schon immer
PLANEN
ueber Sex wissen wollten
(aber bisher nicht zu fragen wagten)

Everything You Always Wanted to Know About (Domain-Independent Classical) **Planning** (But Were Afraid to Ask)

Jörg Hoffmann

INRIA
Nancy, France

October 7, 2011

Agenda

1. Planning? What's that?
2. What is it good for?
3. Does it work?
4. Is it interesting to do research in?
5. And now, what?

Agenda

Planning? What's that?

"Selecting a goal-leading course of actions based on a high-level description of the world."

Planning? What's that?

"Selecting a goal-leading course of actions based on a high-level description of the world."

Details: [Ghallab *et al.* (2004)].

Planning? What's that?

"Selecting a goal-leading course of actions based on a high-level description of the world."

Details: [Ghallab *et al.* (2004)]. Here:

Definition (Planning Task)

A planning task is a 4-tuple $\langle X, A, s_0, s_* \rangle$:

- X: finite set of finite-domain state variables
- A: finite set of actions of form $\langle pre, eff \rangle$ (preconditions/effects; partial variable assignments)
- s_0: initial state (variable assignment)
- s_*: goal (partial variable assignment)

- Plan: action sequence transforming s_0 into state complying with s_*
- =compactly described transition system; **PSPACE**-complete

Example: FreeCell

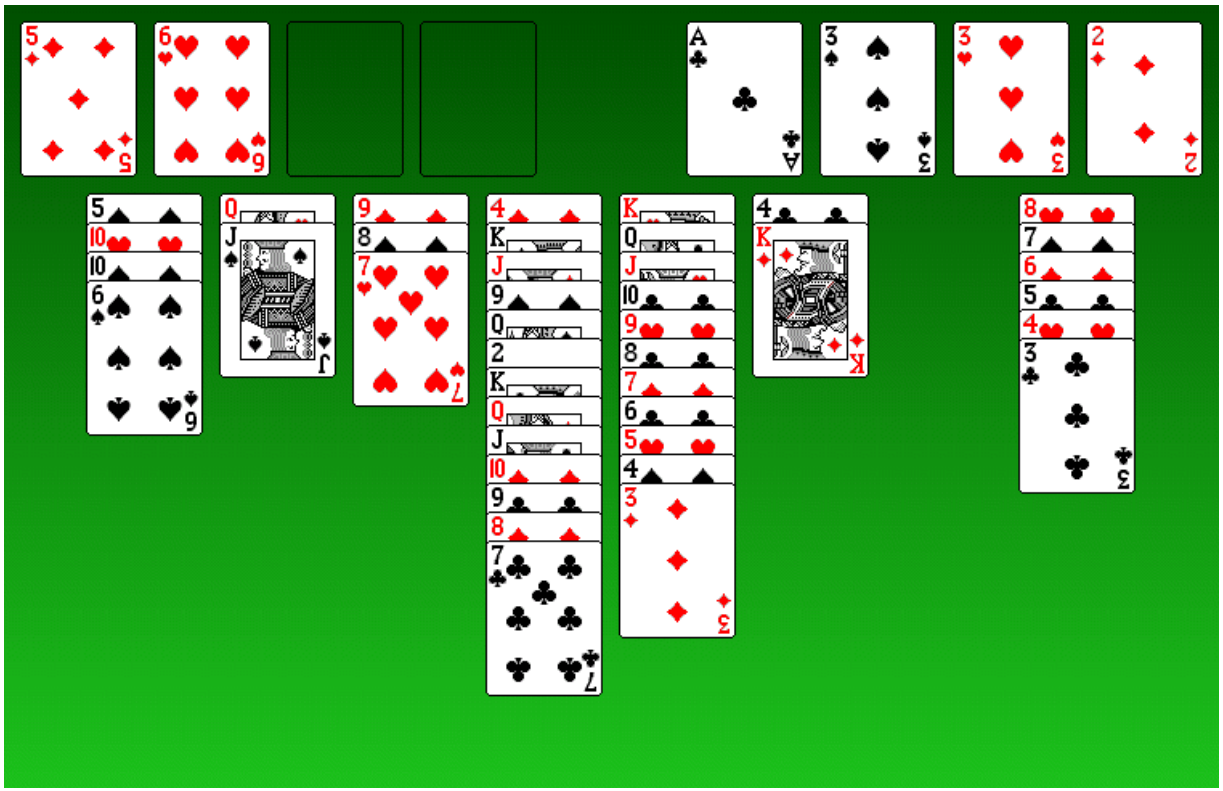
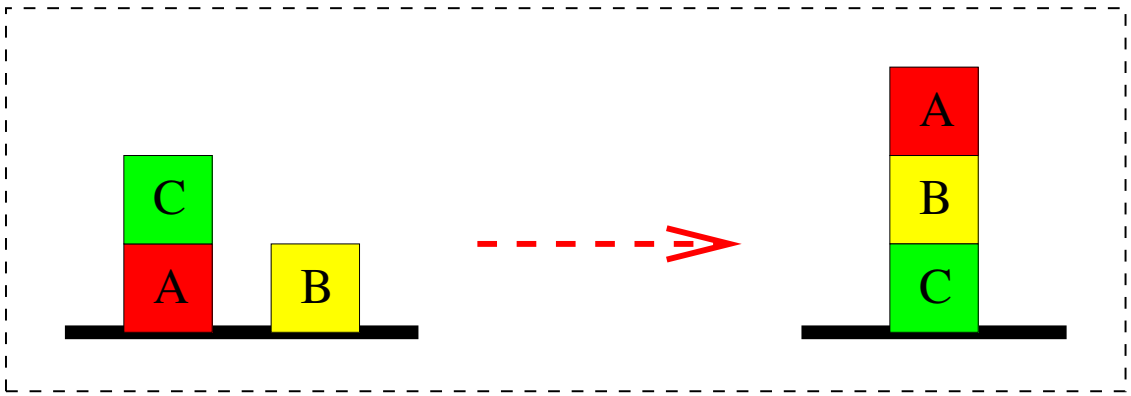


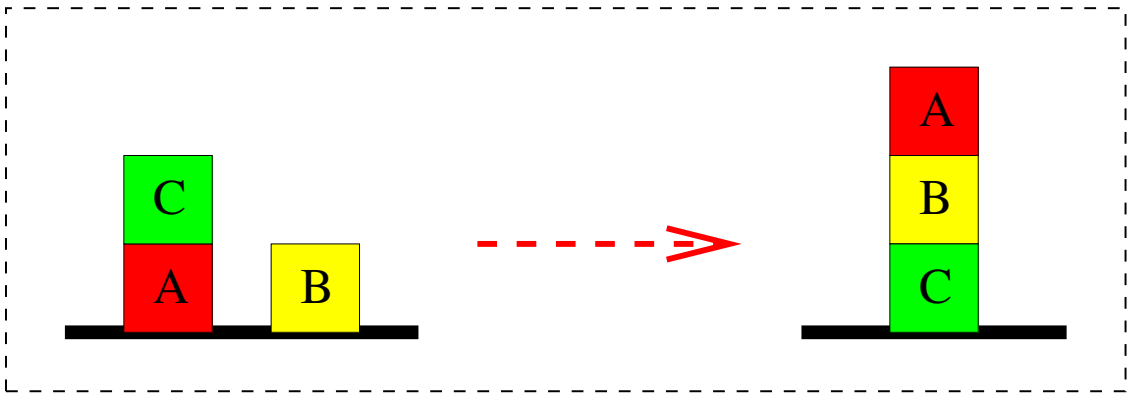
image credits: GNOME Project (GNU General Public License)

- Variables X: card positions
- Actions A: card moves
- Initial state s_0: start configuration
- Goal s_*: all cards "home"

This is planning (?)

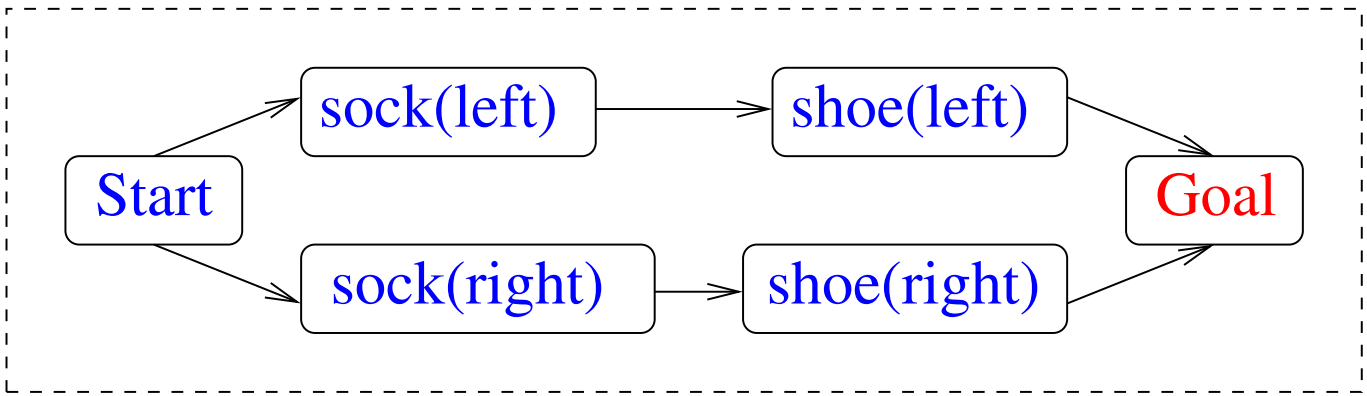


This is planning (?)

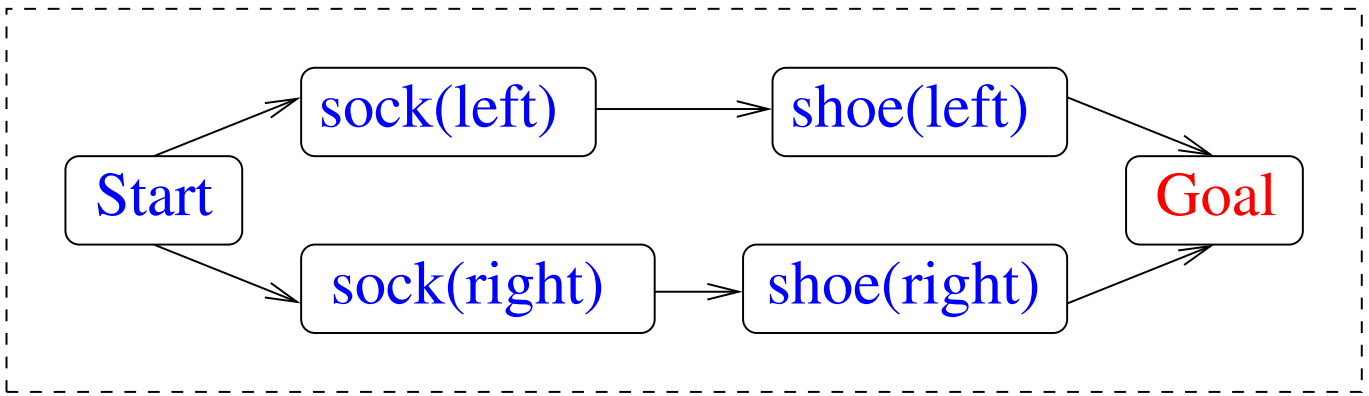


"The Sussman Anomaly"

This is planning (?)

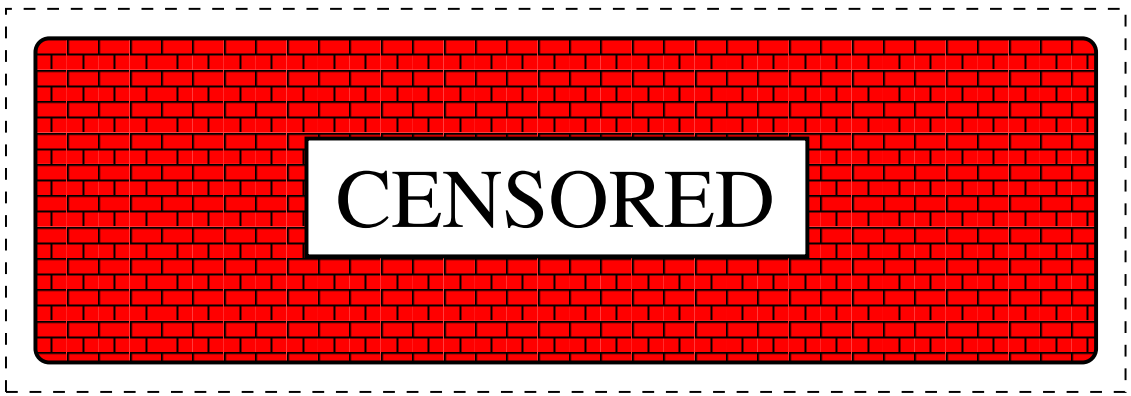


This is planning (?)



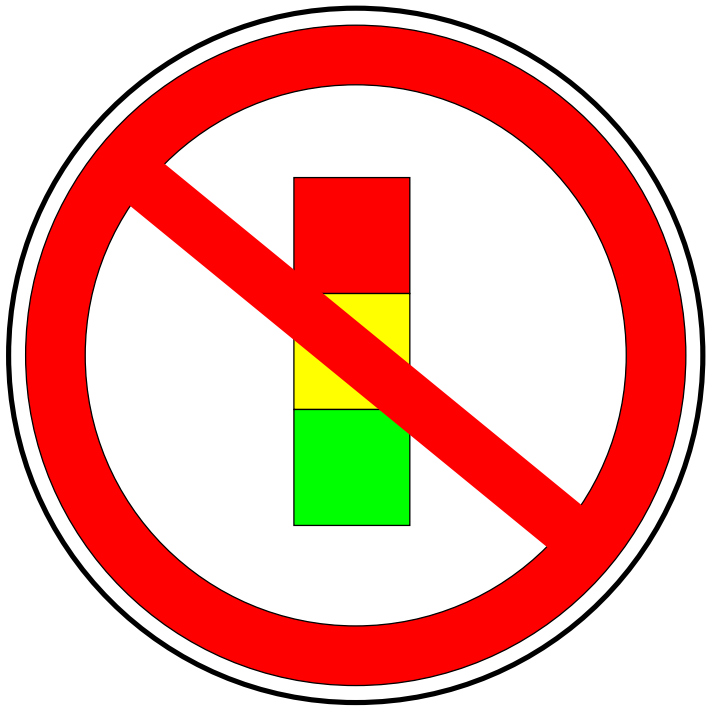
[Russell and Norvig (2003)]

This is planning (?)



"Bomb in the toilet"

Blocksworld verboten!



Agenda

What is it good for?

What are they (planning researchers) trying to do?

What is it good for?

What are they (planning researchers) trying to do?

Create *one* planning solver that will perform sufficiently well on *all* possible domains.

What is it good for?

What are they (planning researchers) trying to do?

Create *one* planning solver that will perform sufficiently well on *all* possible domains.

- **20th century:** (aka GOFAI)

  human intelligence $\Longrightarrow$ flexible problem solving

What is it good for?

What are they (planning researchers) trying to do?

Create *one* planning solver that will perform sufficiently well on *all* possible domains.

- **20th century:** (aka GOFAI)

  human intelligence $\Longrightarrow$ flexible problem solving

- **21st century:** (aka rock-bottom pragmatism)

  How can we earn money with this?

Selling point #1

Your problem is subject to frequent change

- Implement own solver $\Longrightarrow$ adapt the source code
- Use planning $\Longrightarrow$ adapt the planning *model*

Selling point #1

Your problem is subject to frequent change

- Implement own solver $\Longrightarrow$ adapt the source code
- Use planning $\Longrightarrow$ adapt the planning *model*

"Planning is a form of model-based software engineering."

- Classical app: space travel (cf. Apollo 13 . . .)

Selling point #1

Your problem is subject to frequent change

- Implement own solver $\Longrightarrow$ adapt the source code
- Use planning $\Longrightarrow$ adapt the planning *model*

"Planning is a form of model-based software engineering."

- Classical app: space travel (cf. Apollo 13 . . .)
- Controlling printers at Xerox [Ruml *et al.* (2011)]
- Composing business processes at SAP [Hoffmann *et al.* (2010)]

A conversation with Alexander Koller

Natural language sentence generation [Koller and Petrick (2011)]

- Planner input: grammar, intended meaning
- Planner output: sentence implementing meaning

A conversation with Alexander Koller

Natural language sentence generation [Koller and Petrick (2011)]

- Planner input: grammar, intended meaning
- Planner output: sentence implementing meaning

Jörg: *"Why don't you develop a domain-specific solver? That could be more effective."*

A conversation with Alexander Koller

Natural language sentence generation [Koller and Petrick (2011)]

- Planner input: grammar, intended meaning
- Planner output: sentence implementing meaning

Jörg: *"Why don't you develop a domain-specific solver? That could be more effective."*

Alexander: *"Well, that's not the research I'm interested in."*

A conversation with Alexander Koller

Natural language sentence generation [Koller and Petrick (2011)]

- Planner input: grammar, intended meaning
- Planner output: sentence implementing meaning

Jörg: *"Why don't you develop a domain-specific solver? That could be more effective."*

Alexander: *"Well, that's not the research I'm interested in."*

Jörg: *"...?"*

A conversation with Alexander Koller

Natural language sentence generation [Koller and Petrick (2011)]

- Planner input: grammar, intended meaning
- Planner output: sentence implementing meaning

Jörg: *"Why don't you develop a domain-specific solver? That could be more effective."*

Alexander: *"Well, that's not the research I'm interested in."*

Jörg: *"...?"*

Alexander: *"The planner works reasonably well, and I don't want to spend the time working out an alternative."*

A conversation with Core Security Technologies

Automatic penetration testing [Lucangeli *et al.* (2010)]

- Planner input: network configuration, target machine
- Planner output: hack (exploits) reaching target machine

A conversation with Core Security Technologies

Automatic penetration testing [Lucangeli *et al.* (2010)]

- Planner input: network configuration, target machine
- Planner output: hack (exploits) reaching target machine

Research department: *"Oh, but we could come up with something that works much better than the planner."*

A conversation with Core Security Technologies

Automatic penetration testing [Lucangeli *et al.* (2010)]

- Planner input: network configuration, target machine
- Planner output: hack (exploits) reaching target machine

Research department: *"Oh, but we could come up with something that works much better than the planner."*

Jörg: *"But . . ."*

A conversation with Core Security Technologies

Automatic penetration testing [Lucangeli *et al.* (2010)]

- Planner input: network configuration, target machine
- Planner output: hack (exploits) reaching target machine

Research department: *"Oh, but we could come up with something that works much better than the planner."*

Jörg: *"But . . . "*

Development department: *"Yeah, maybe, but you don't know when and my product deadline is next month!"*

A conversation with Core Security Technologies

Automatic penetration testing [Lucangeli *et al.* (2010)]

- Planner input: network configuration, target machine
- Planner output: hack (exploits) reaching target machine

Research department: *"Oh, but we could come up with something that works much better than the planner."*

Jörg: *"But . . . "*

Development department: *"Yeah, maybe, but you don't know when and my product deadline is next month!"*

Jörg: 

Selling point #2

Your problem is not trivial

- Implement own solver $\Longrightarrow$ costs time+money
- Write planning model $\Longrightarrow$ costs less time+money

Selling point #2

Your problem is not trivial

- Implement own solver $\Longrightarrow$ costs time+money
- Write planning model $\Longrightarrow$ costs less time+money

"Planning is a quick hack to get things up and running!"

- Rapid prototyping
- Prototype might end up being good enough ...

Agenda

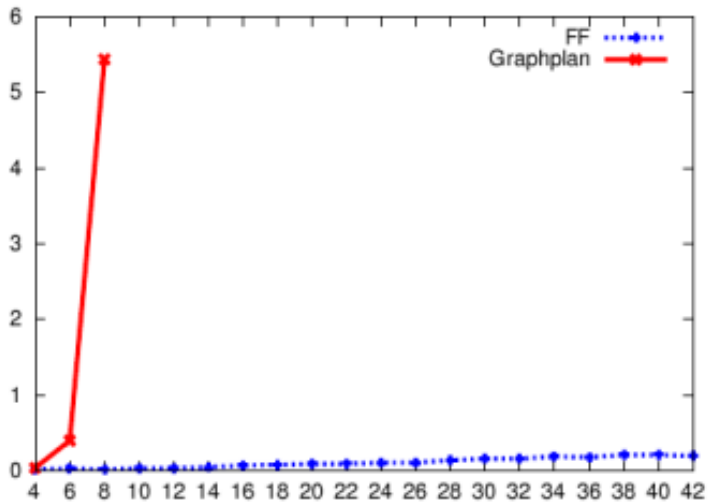
The Planning News (Year 2000)

The PLANNING Times

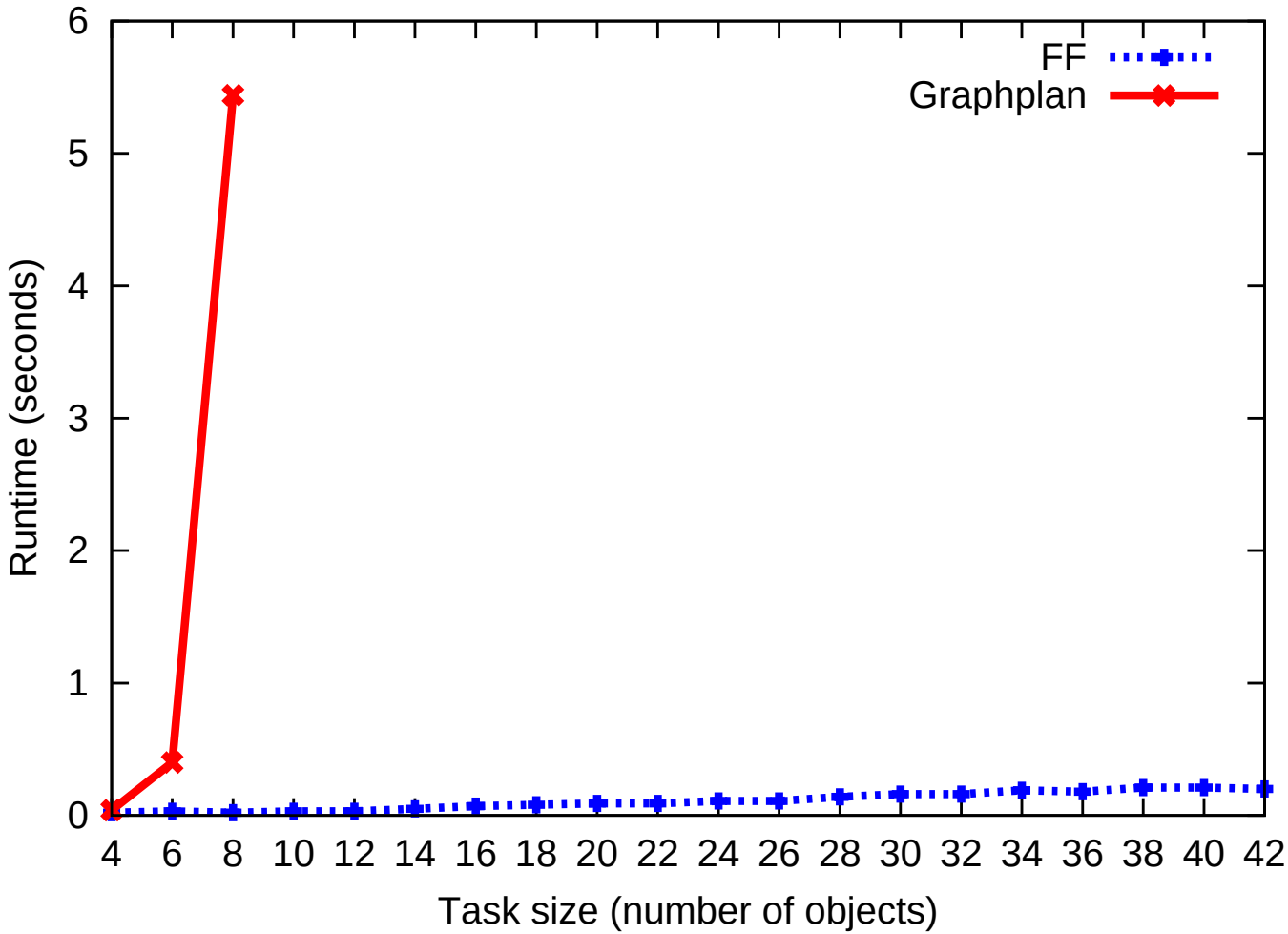
Edition April 2000

Planning, at last, WORKS!!!

Breckenridge, CO. Dramatic scenes yesterday at the 2nd International Planning Competition, when planners finally understood how to transport many balls from one room to another! Graphplan was left literally at the side of the road when FF got going. Joerg Hoffmann, its inventor, was unfortunately to busy celebrating to give us a coherent statement, but we're sure to cover you with red-hot news as matters develop further ...



The Planning News (Year 2000)



The IPC

IPC = International Planning Competition

- 1998, 2000, 2002, 2004, 2006, 2008, 2011
- PDDL [McDermott et al. (1998); Fox and Long (2003)]
- $\approx$ 40 domains, $\gg$ 1000 instances, 74 (!!) planners in 2011
- Optimal track vs. satisficing track
- Various others: uncertainty, learning, ...

`http://ipc.icaps-conference.org/`

Heuristic Forward Search

State Space

- Planning task $\langle X, A, s_0, s_* \rangle$
- State space: all states (assignments s to X) reachable from s_0
- Forward search: start at s_0, follow applicable A

Heuristic Function

- Function h : state space $\mapsto \mathbb{N}_0$
- Estimate distance to goal

Heuristic Search

- "Prefer to explore states with small h"
- A*, greedy best-first search, ...

The Relaxed Plan Heuristic

"What was once true remains true forever:"

Definition (Relaxed Plan Heuristic)

Let s be a state. A relaxed plan for s is a plan under the transition semantics $Result(s, a) = s \cup eff_a$. The minimal length of any relaxed plan for s is the relaxed plan heuristic $h^+(s)$.

The Relaxed Plan Heuristic

"What was once true remains true forever:"

Definition (Relaxed Plan Heuristic)

Let s be a state. A relaxed plan for s is a plan under the transition semantics $Result(s, a) = s \cup eff_a$. The minimal length of any relaxed plan for s is the relaxed plan heuristic $h^+(s)$.

- Footnote in [Bylander (1994)] ("boring sub-class")
- **PlanEx** easy, **PlanMin** hard

The Relaxed Plan Heuristic

"What was once true remains true forever:"

Definition (Relaxed Plan Heuristic)

Let s be a state. A relaxed plan for s is a plan under the transition semantics $Result(s,a) = s \cup eff_a$. The minimal length of any relaxed plan for s is the relaxed plan heuristic $h^+(s)$.

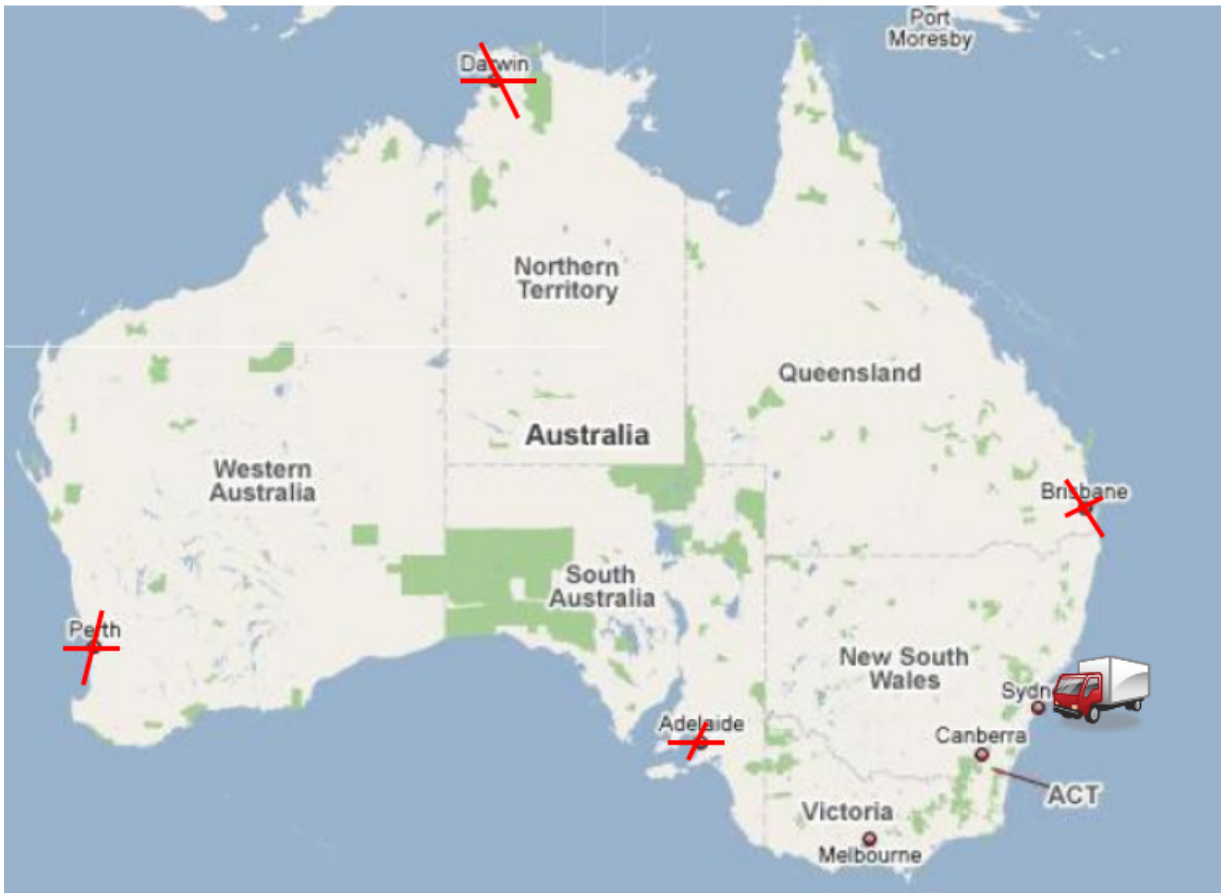
- Footnote in [Bylander (1994)] ("boring sub-class")
- **PlanEx** easy, **PlanMin** hard

- Upper-bound h^+ [McDermott (1996); Bonet *et al.* (1997)]
- Winners in ALL satisficing IPCs [Hoffmann and Nebel (2001) Gerevini *et al.* (2003); Richter and Westphal (2010)]

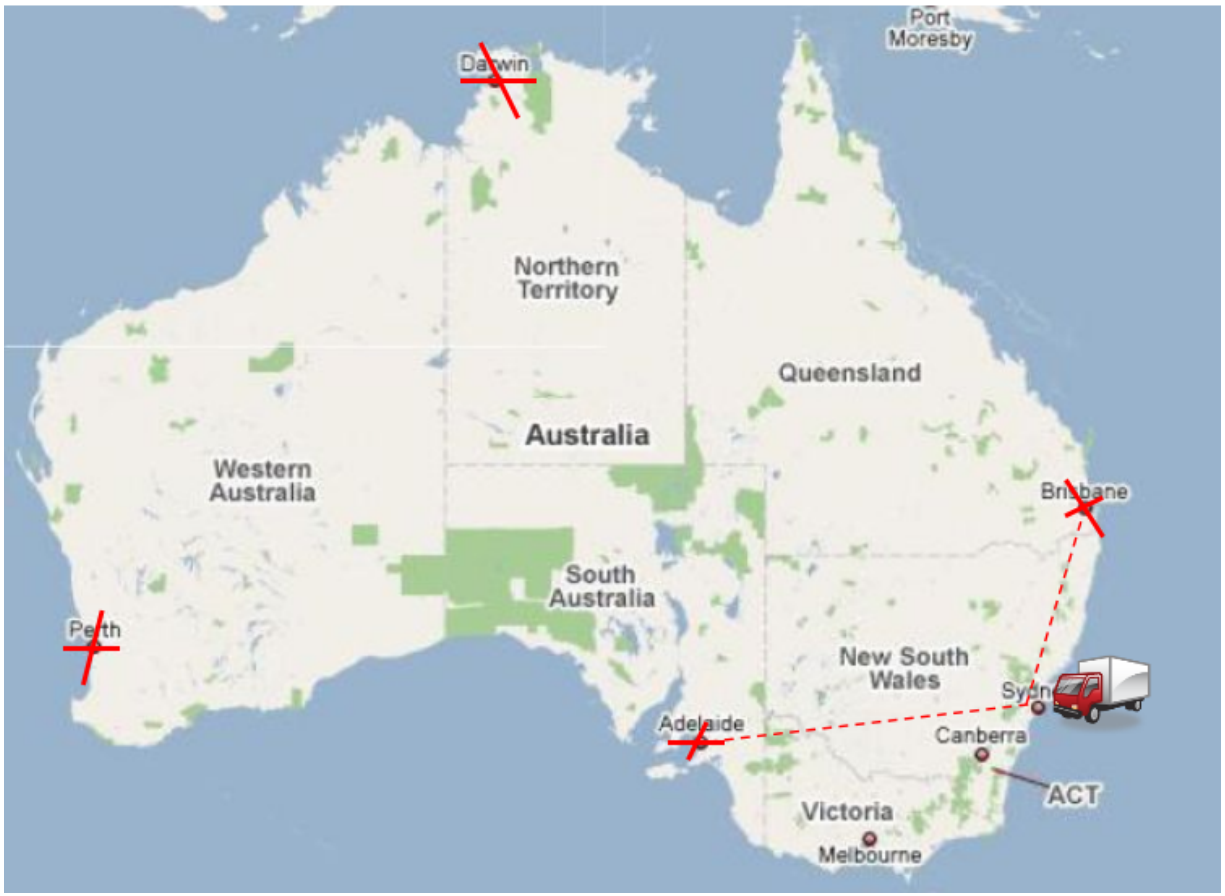
Relaxed Planning in TSP

Relaxed Planning in TSP

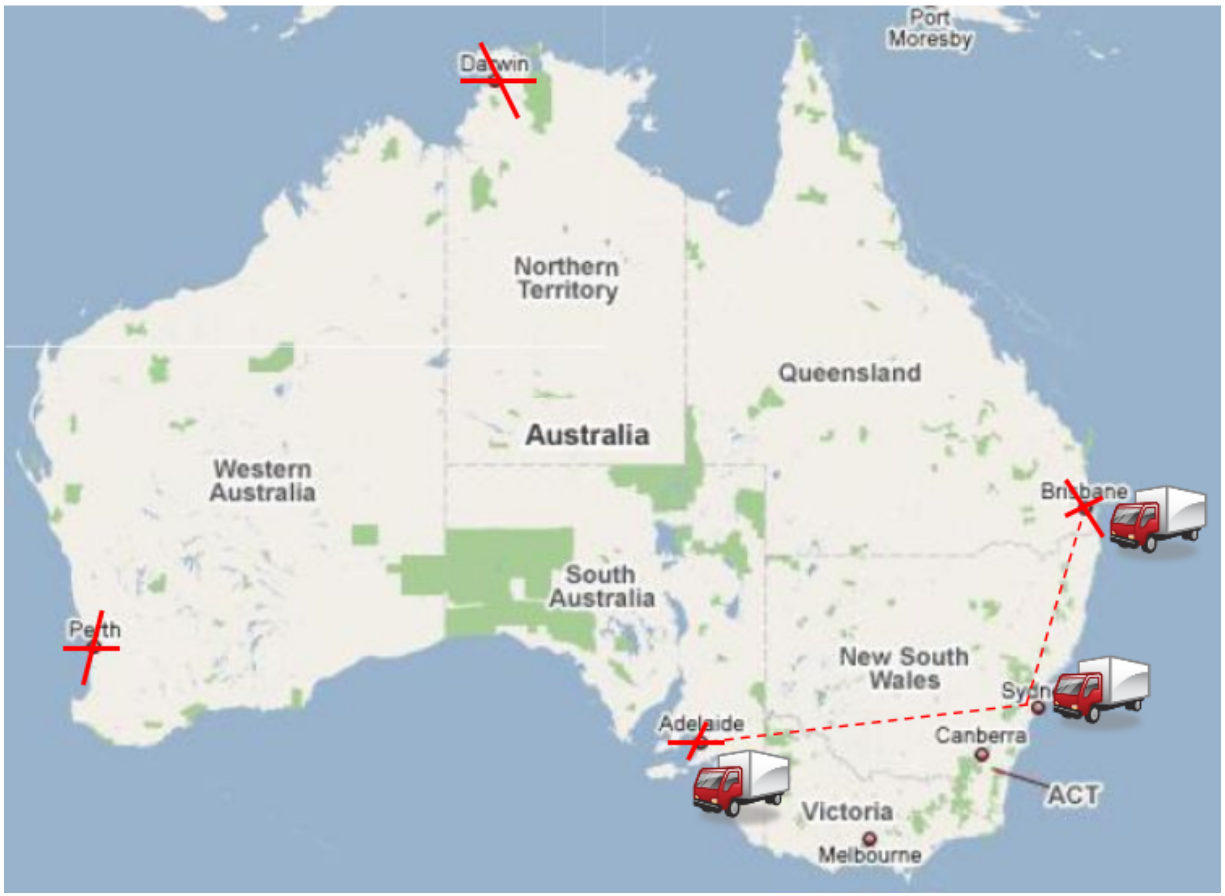
Relaxed Planning in TSP



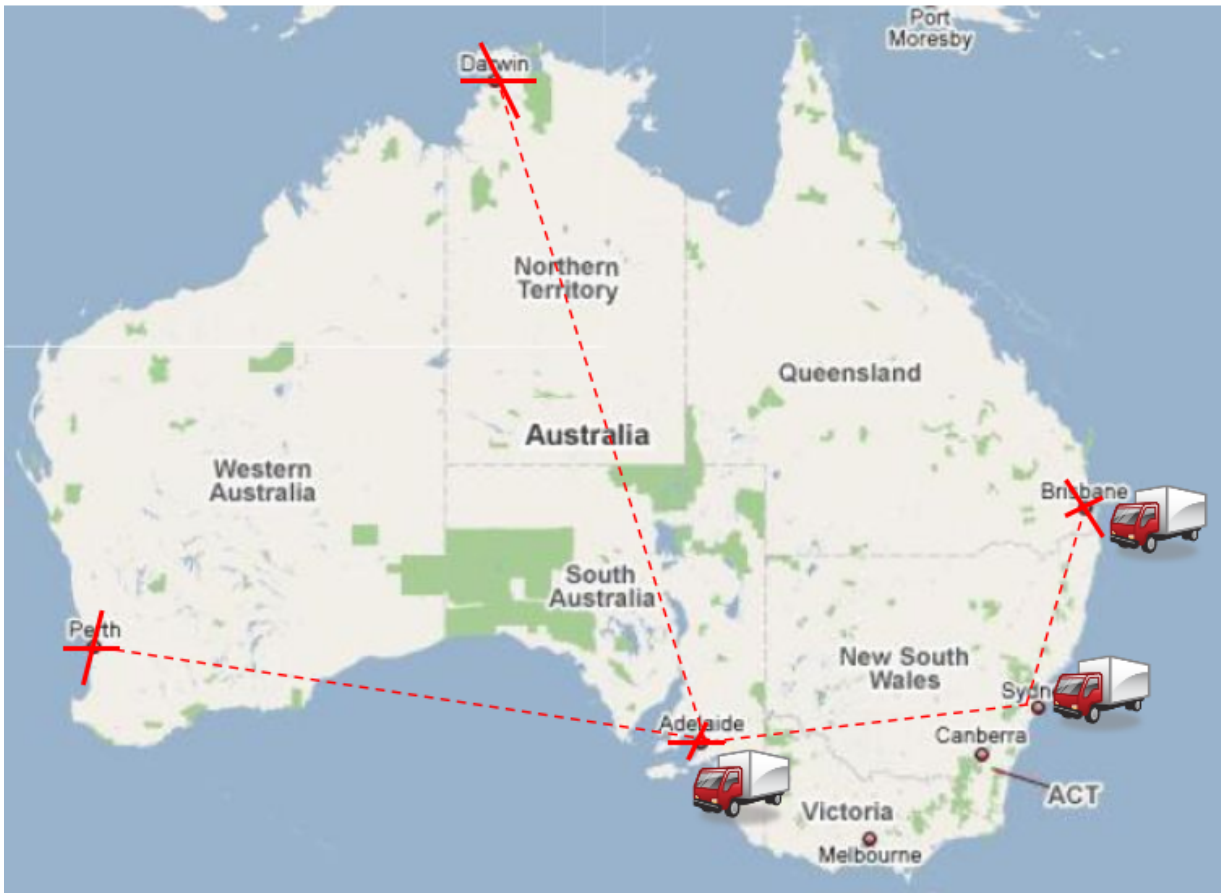
Relaxed Planning in TSP



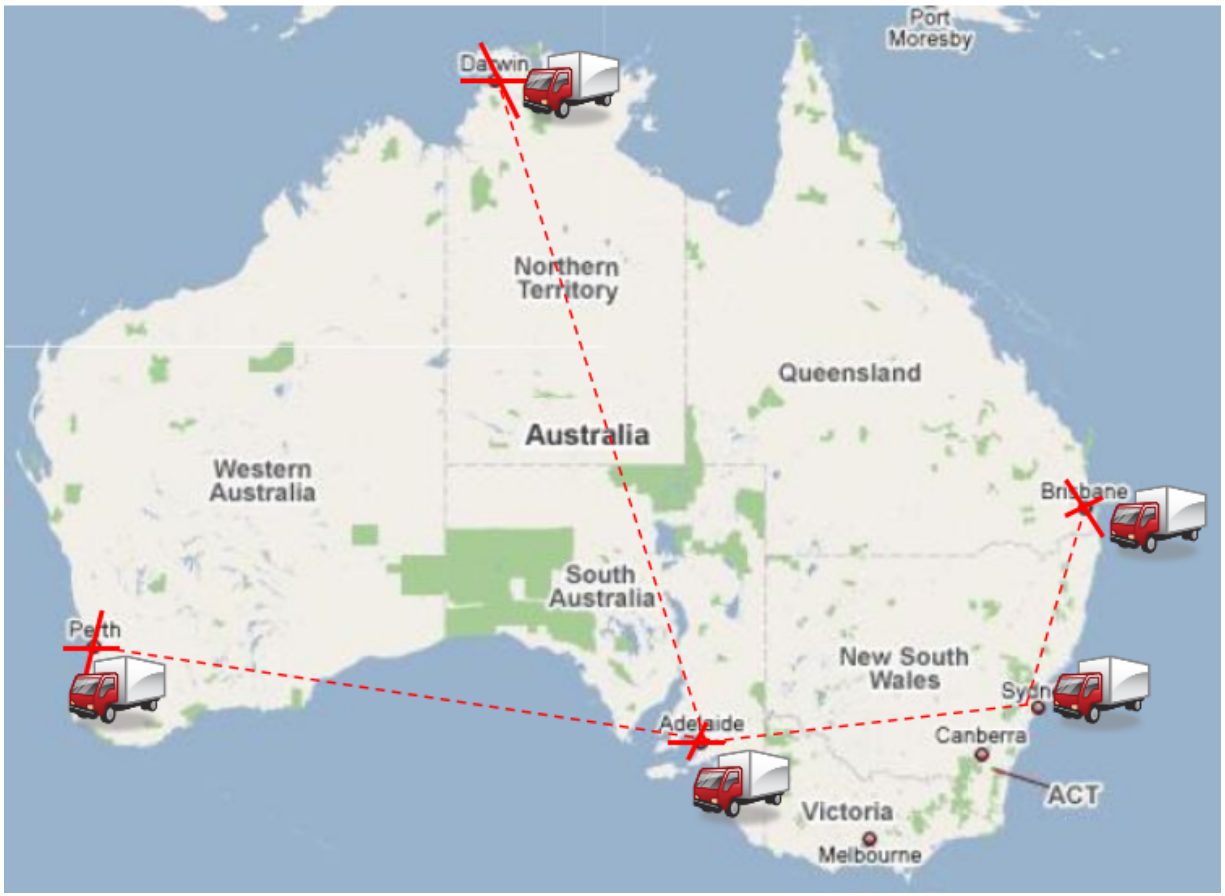
Relaxed Planning in TSP



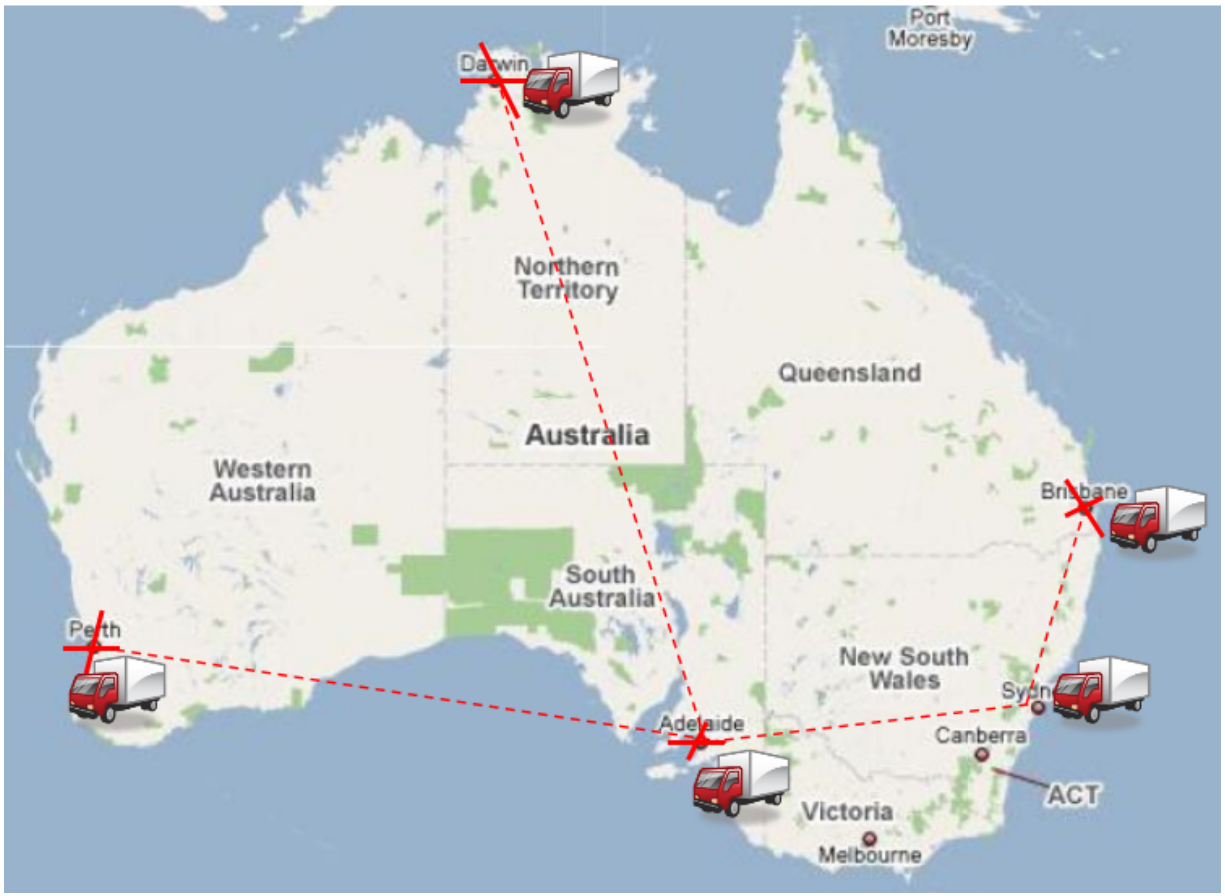
Relaxed Planning in TSP



Relaxed Planning in TSP

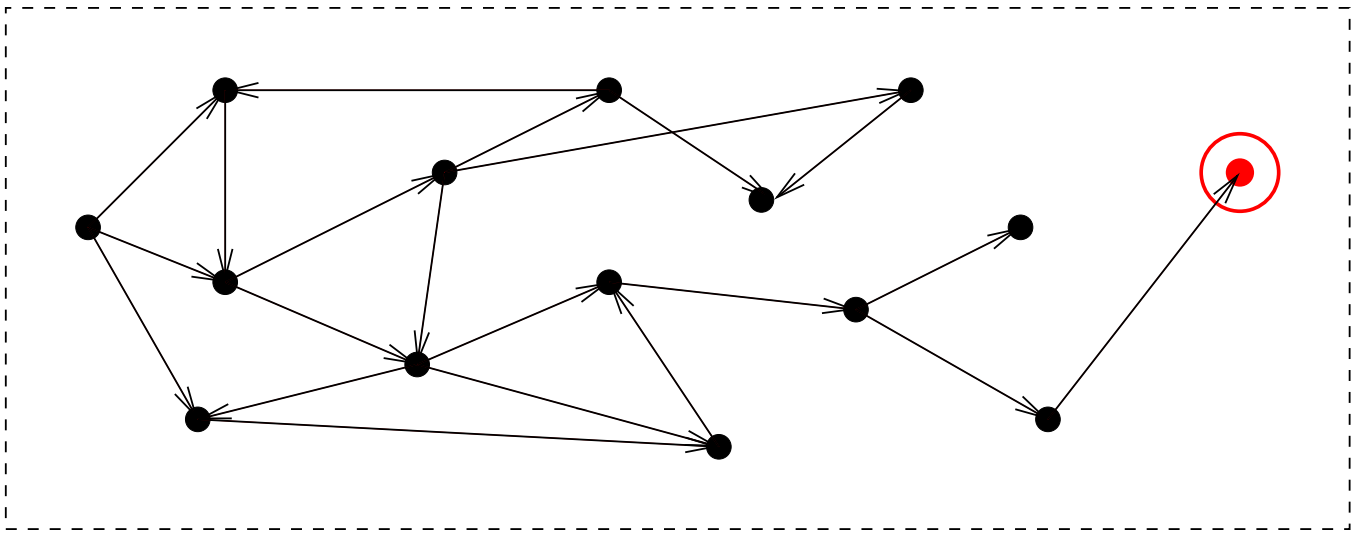


Relaxed Planning in TSP

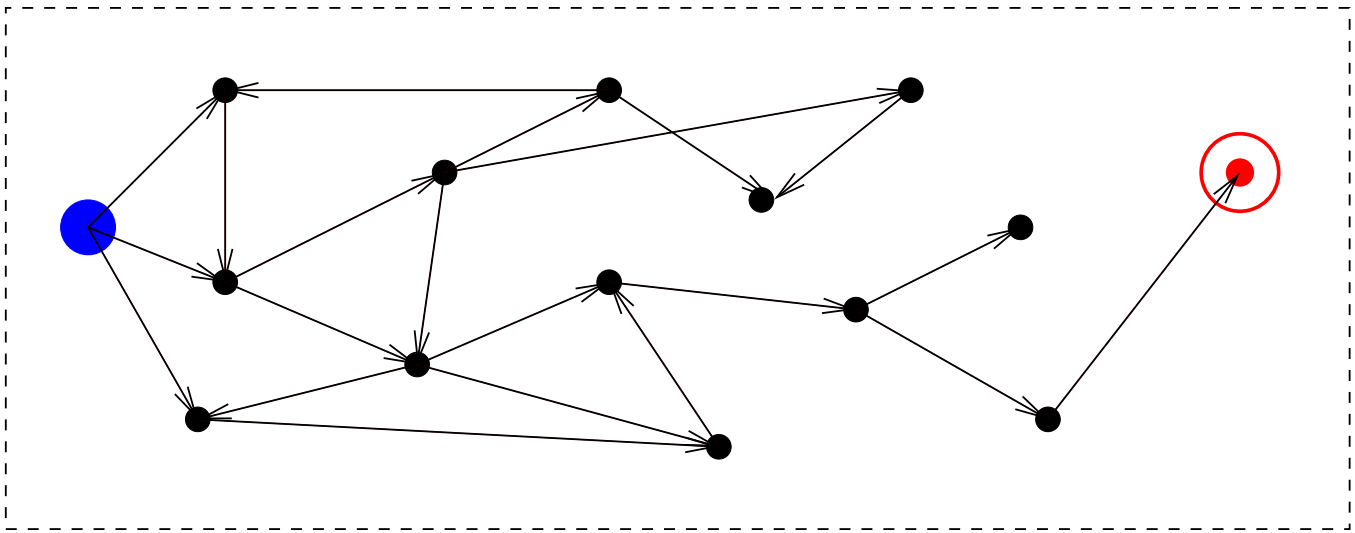


$h^+(\text{TSP}) = \text{minimum spanning tree}$

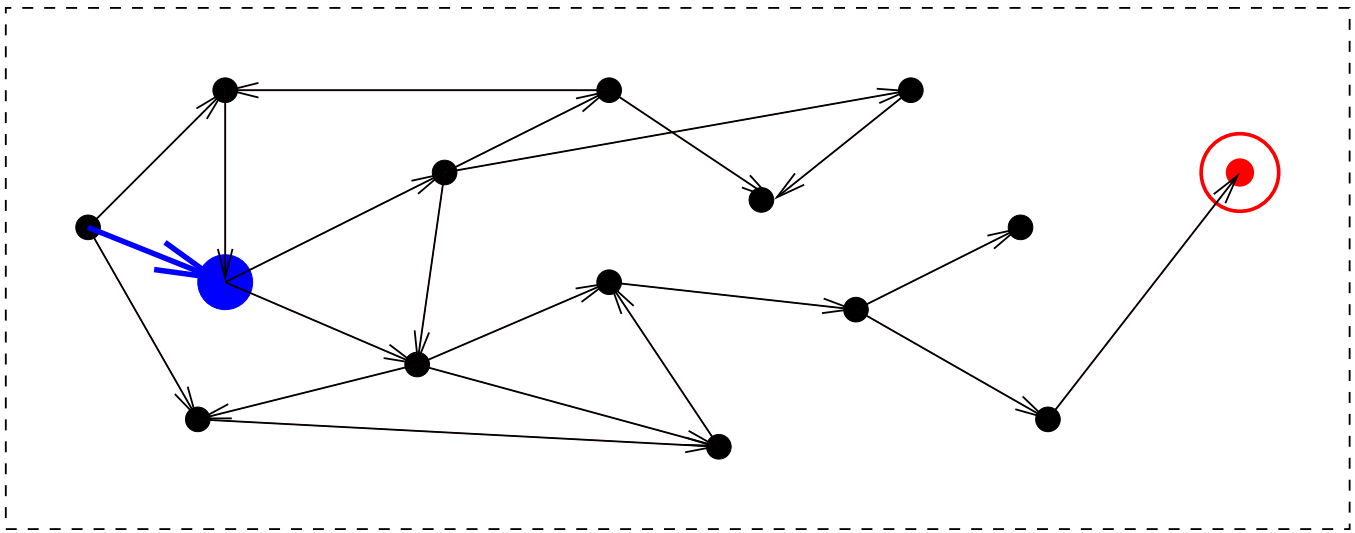
Relaxed Planning in Graphs



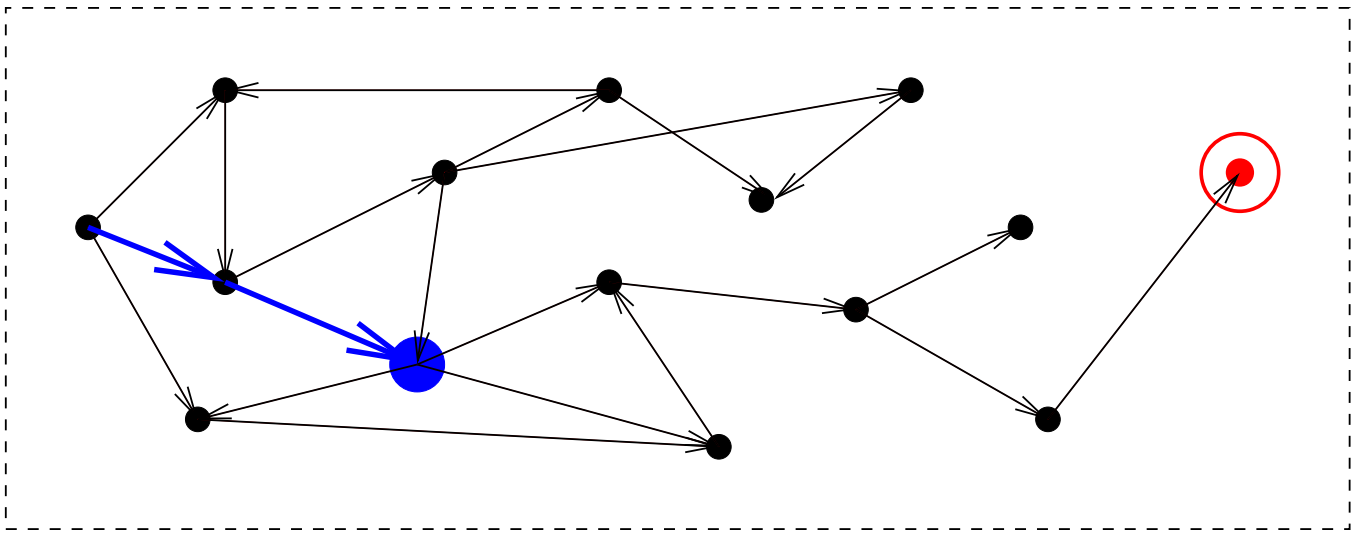
Relaxed Planning in Graphs



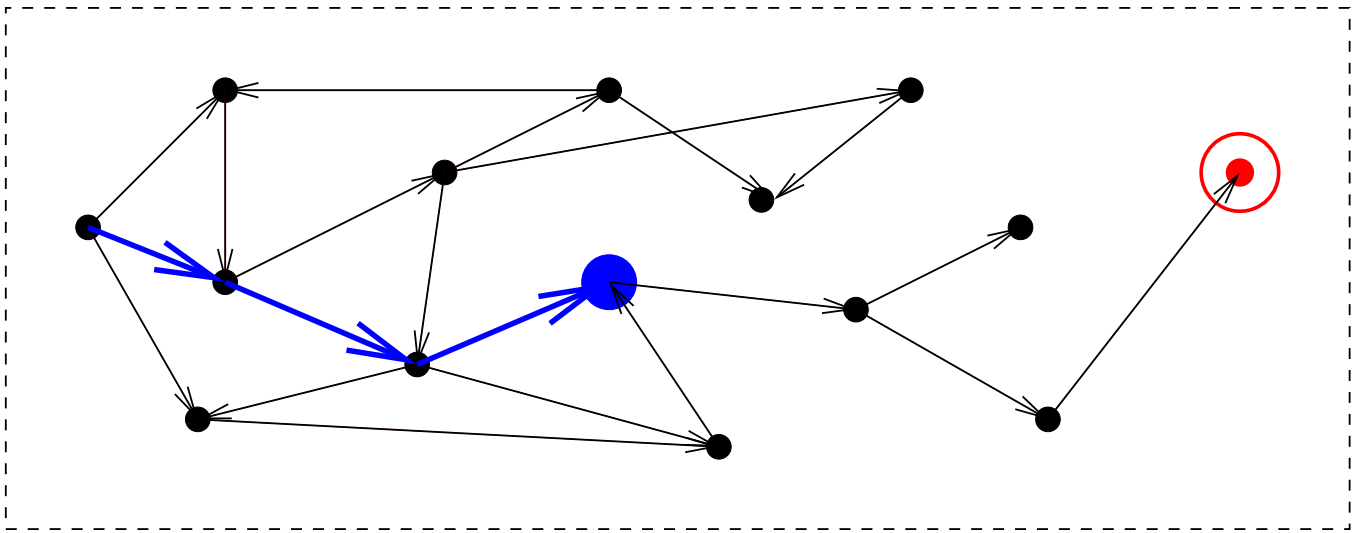
Relaxed Planning in Graphs



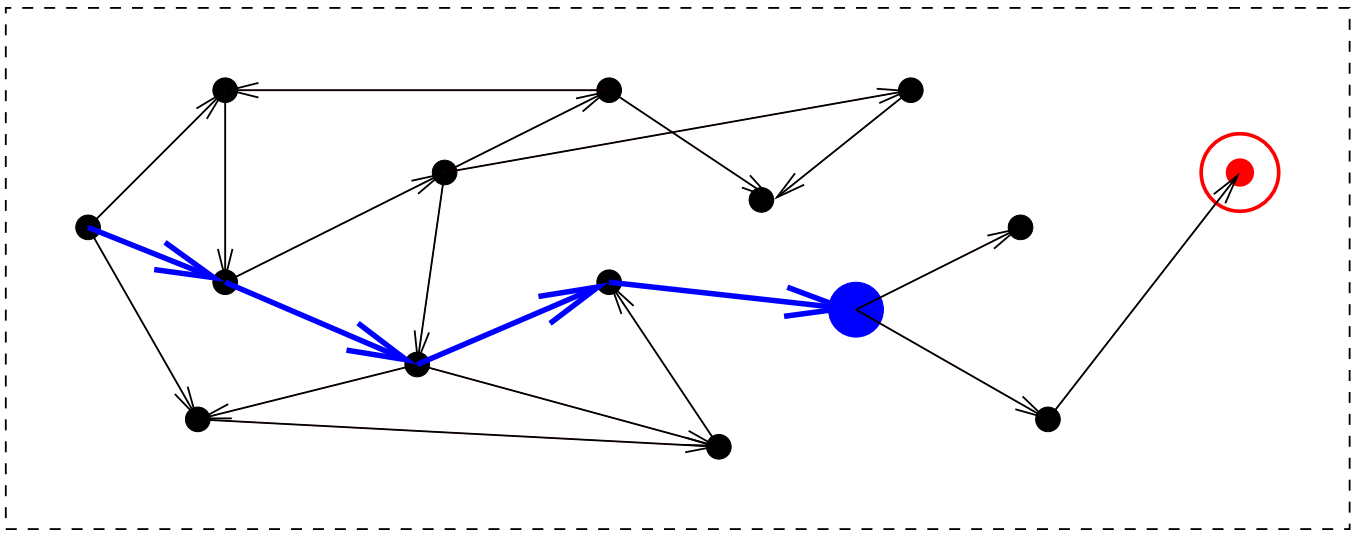
Relaxed Planning in Graphs



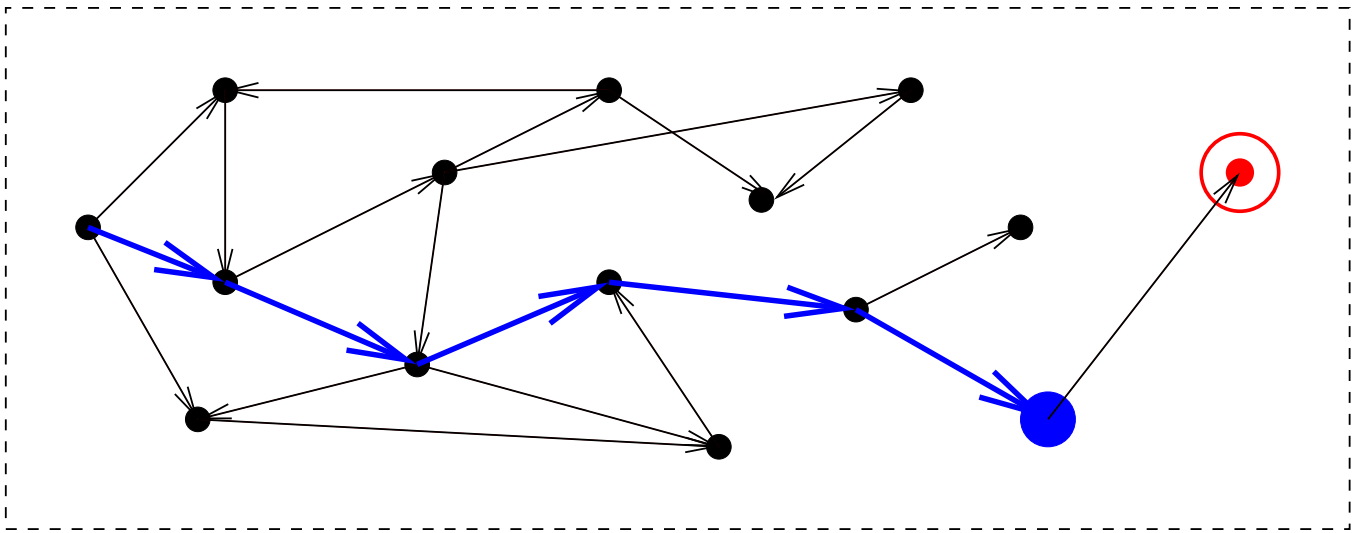
Relaxed Planning in Graphs



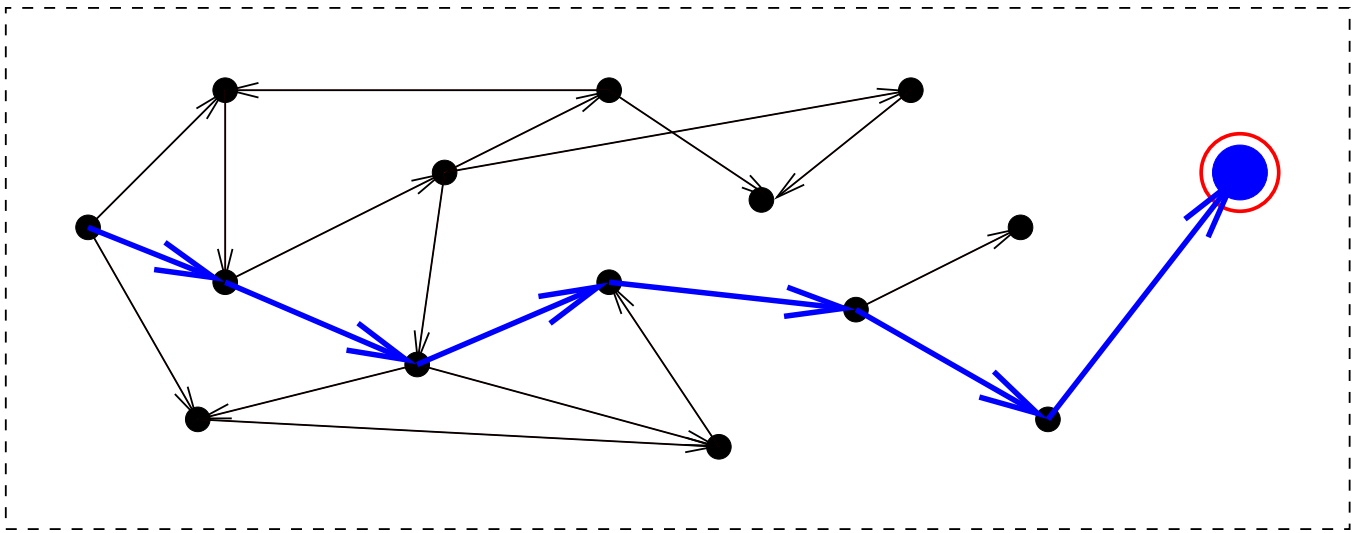
Relaxed Planning in Graphs



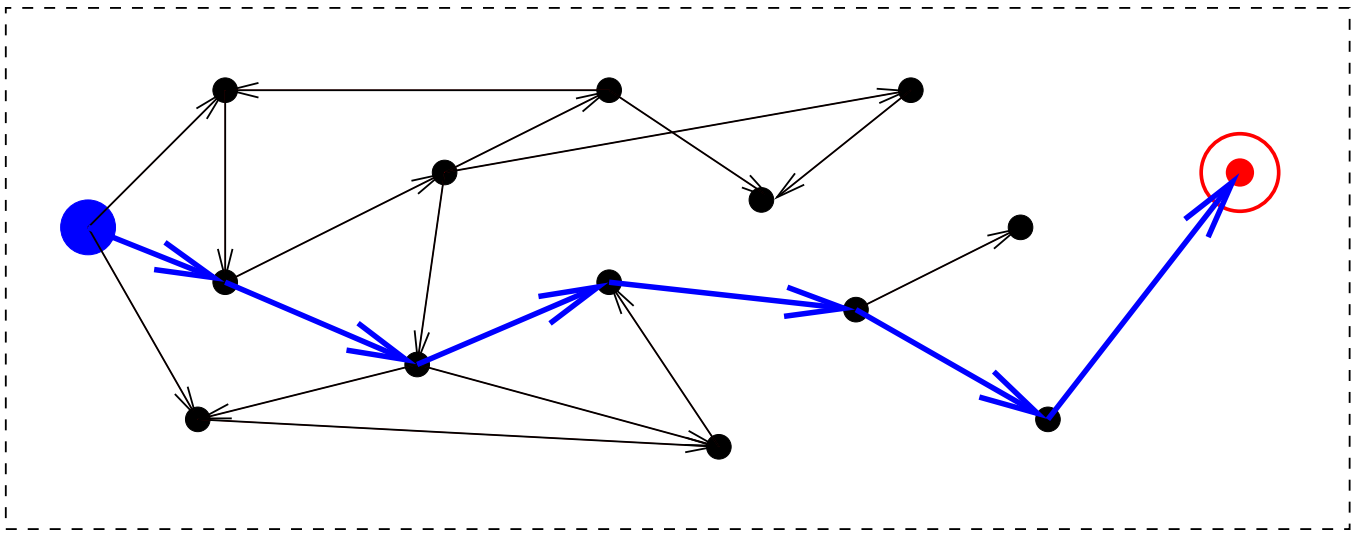
Relaxed Planning in Graphs



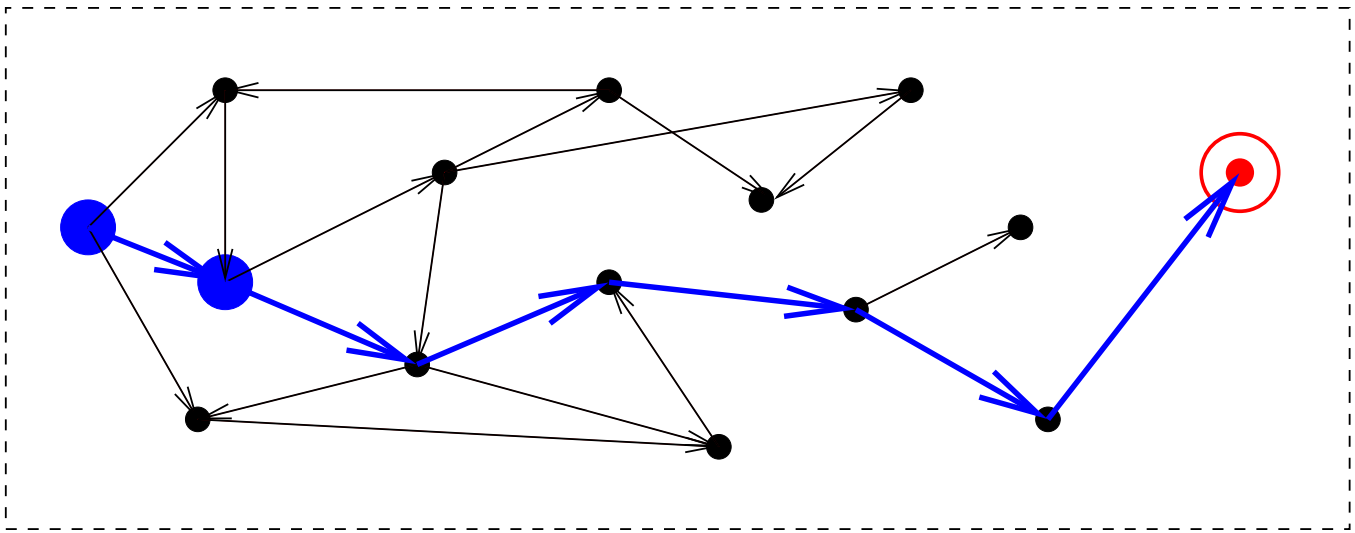
Relaxed Planning in Graphs



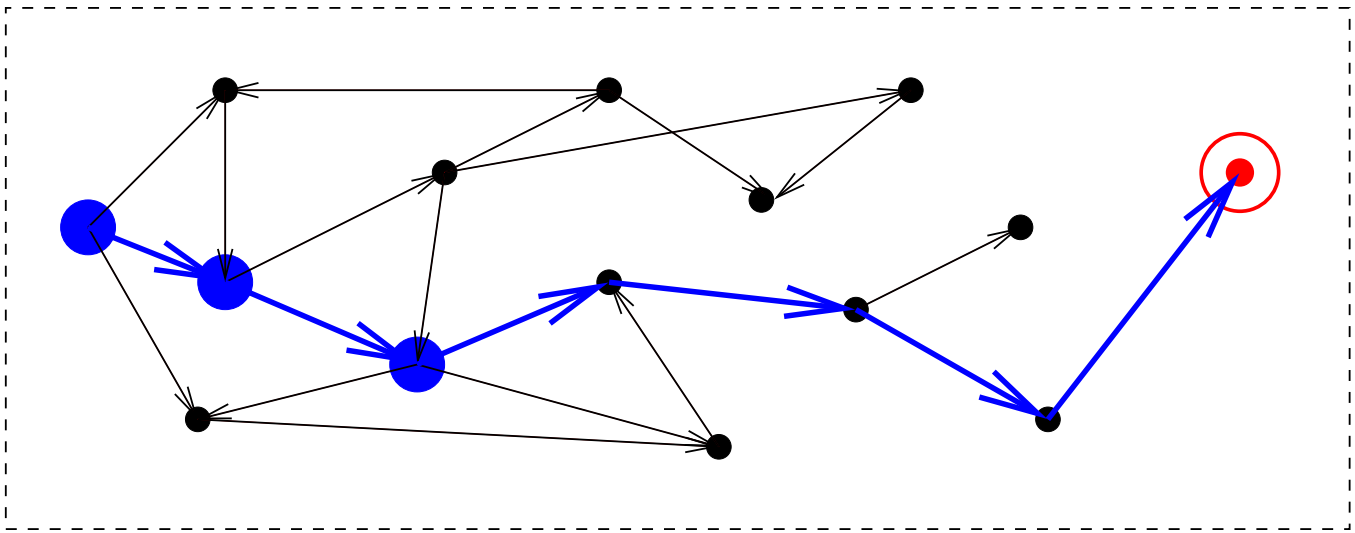
Relaxed Planning in Graphs



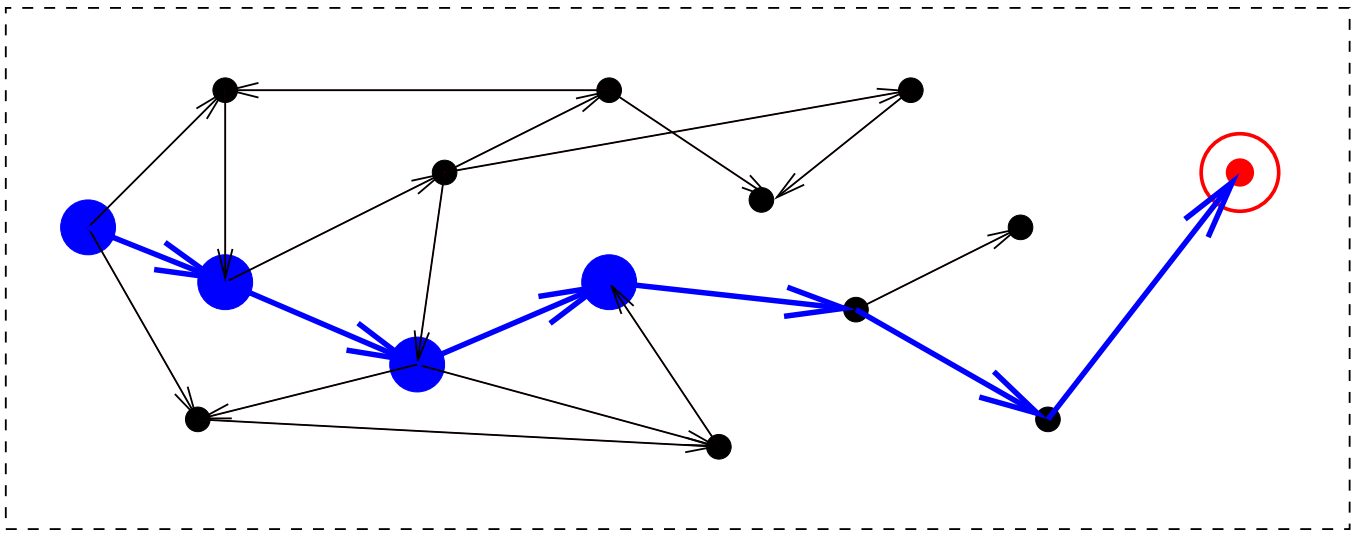
Relaxed Planning in Graphs



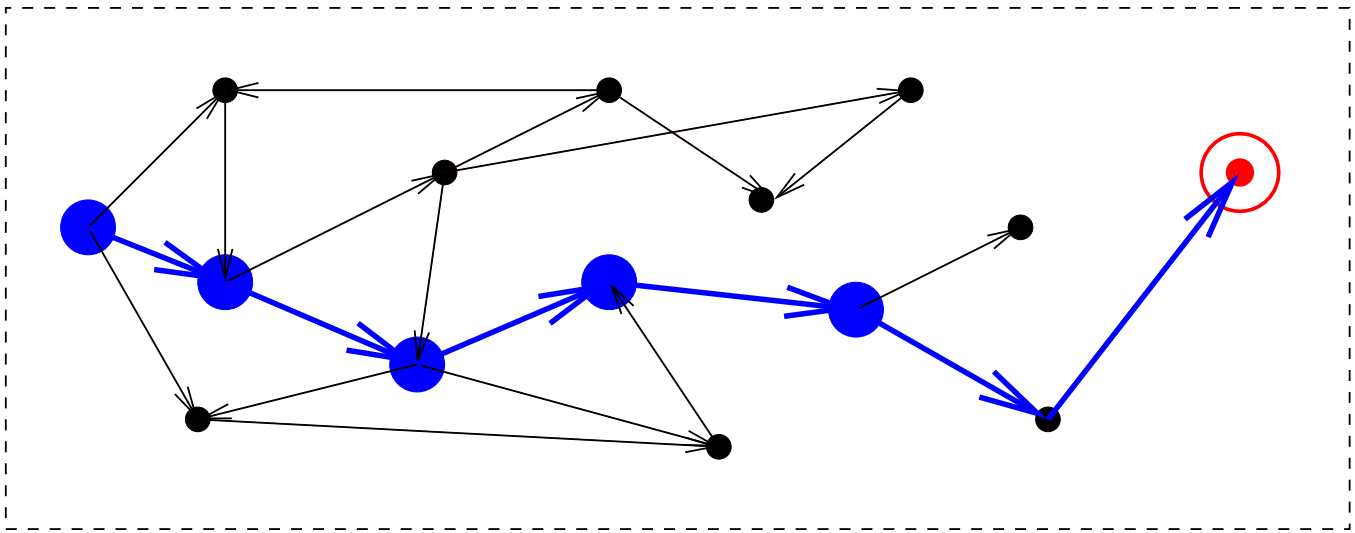
Relaxed Planning in Graphs



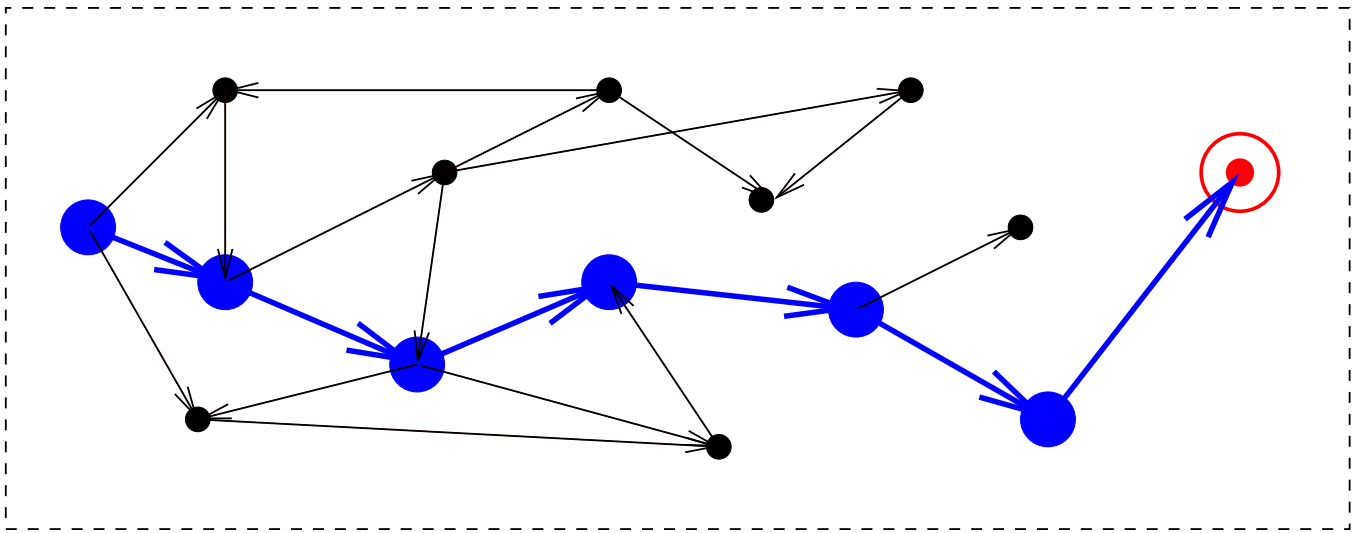
Relaxed Planning in Graphs



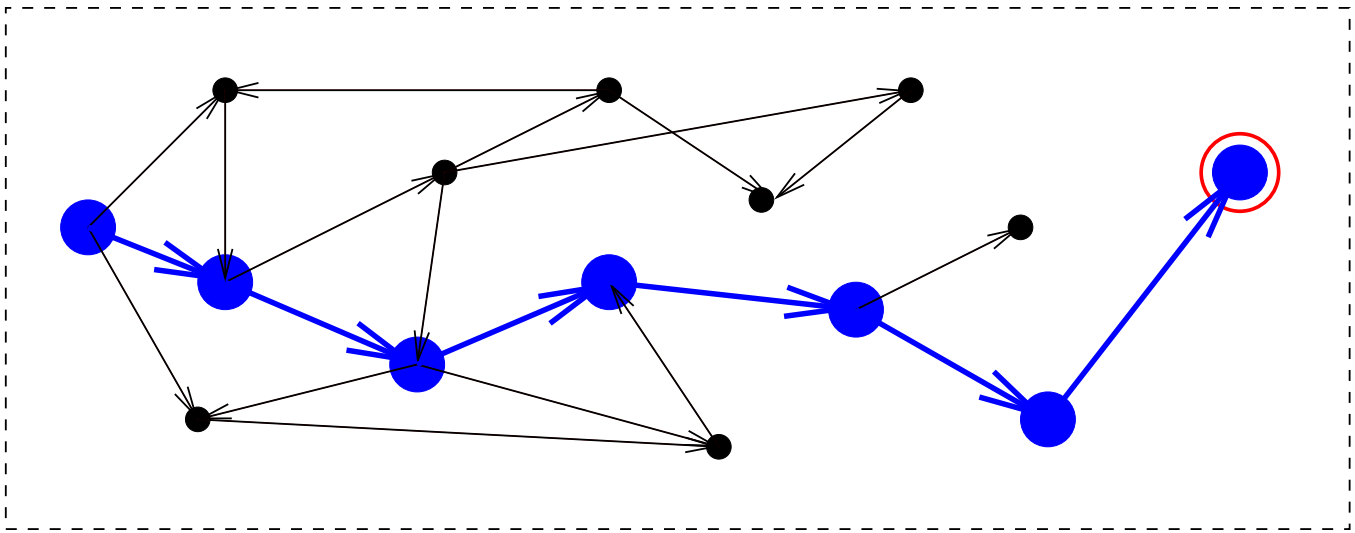
Relaxed Planning in Graphs



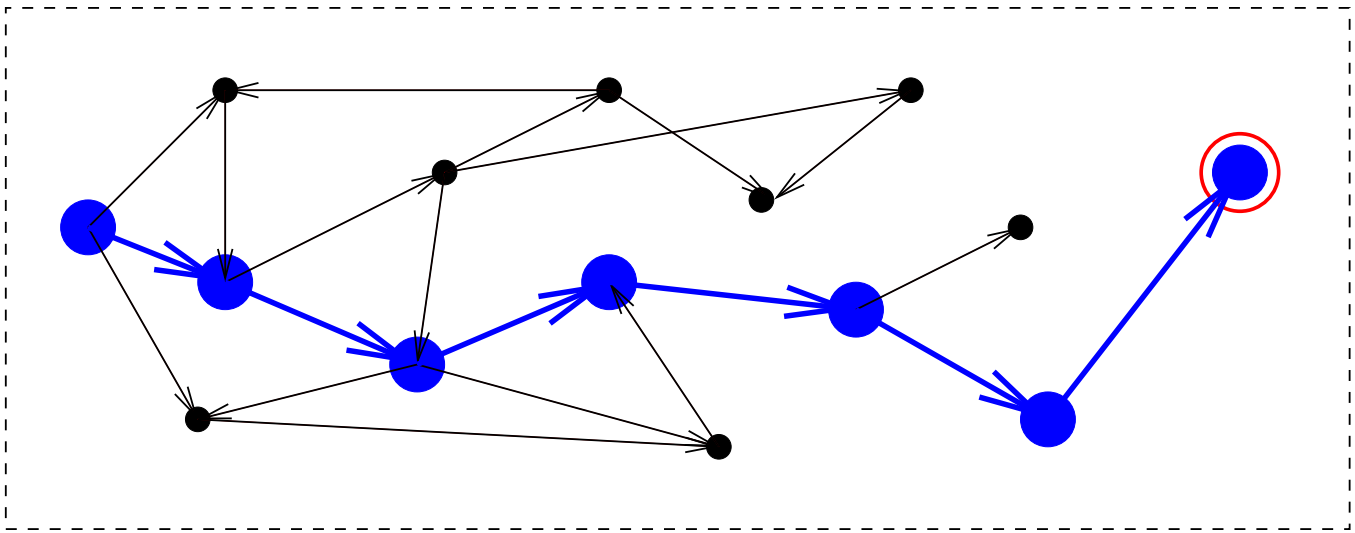
Relaxed Planning in Graphs



Relaxed Planning in Graphs



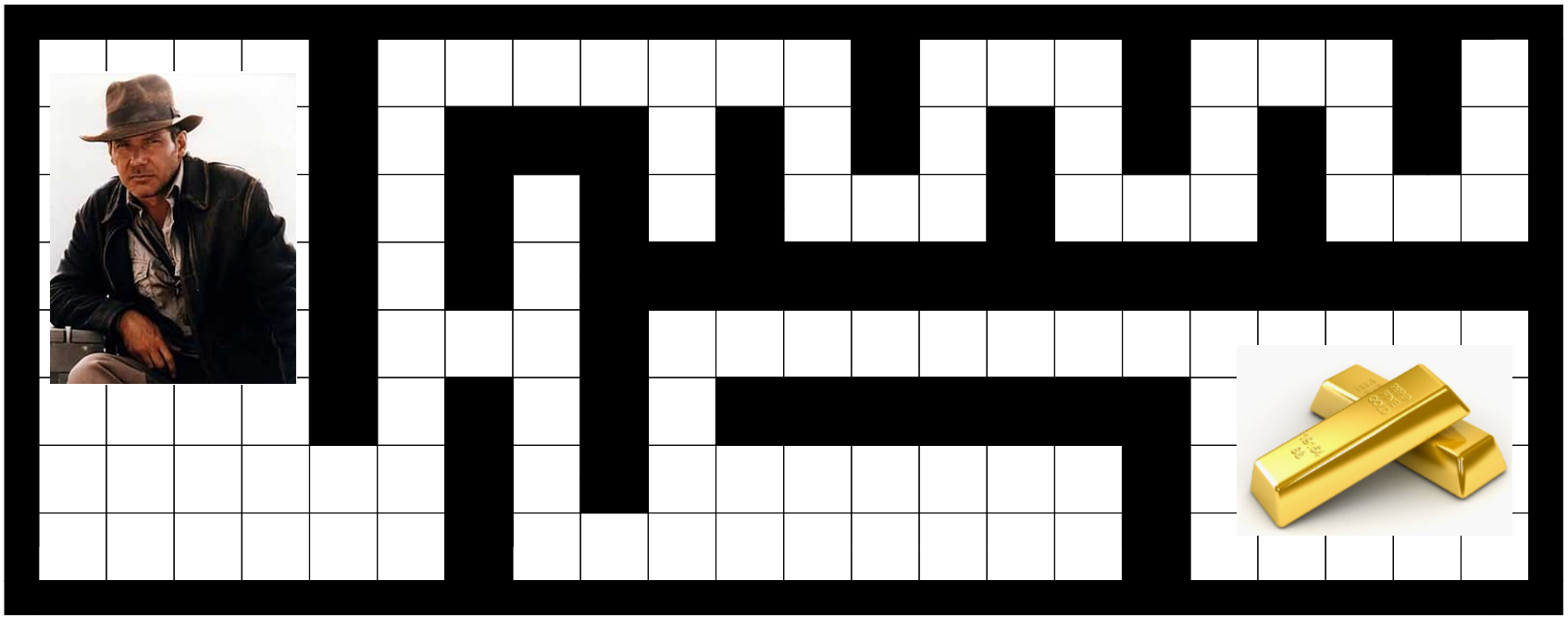
Relaxed Planning in Graphs



h^+(Graph-Distance) = real distance

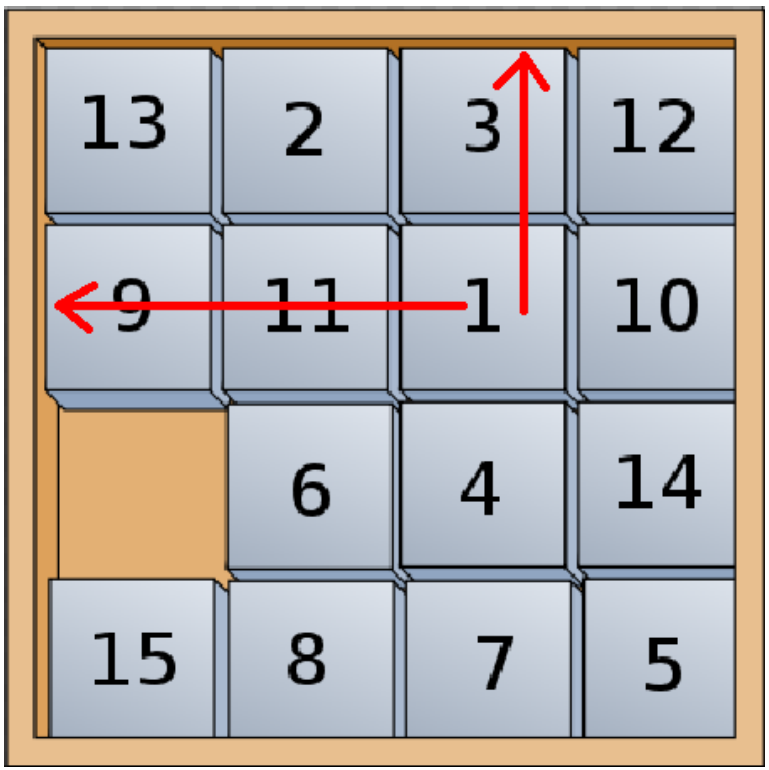
(shortest paths never "walk back")

Relaxed Path Planning



Relaxed Path Planning = Path Planning

Relaxed Planning in the 15-Puzzle



h^+(15-Puzzle) strictly dominates Manhattan distance

Relaxed Planning, Take-Home Message

Summary

- *"Act as if what was once true remains true forever"*
- General relaxation principle with interesting behavior (eg. TSP $\mapsto$ Minimum Spanning Tree)
- Great in (huge) set of planning competition benchmarks

Works also in applications!

- NL sentence generation [Koller and Hoffmann (2010)]
- Business process composition@SAP [Hoffmann *et al.* (2010)]
- Automatic penetration testing [Lucangeli *et al.* (2010)]

Relaxed Planning, Take-Home Message

Summary

- *"Act as if what was once true remains true forever"*
- General relaxation principle with interesting behavior (eg. TSP $\mapsto$ Minimum Spanning Tree)
- Great in (huge) set of planning competition benchmarks

Works also in applications!

- NL sentence generation [Koller and Hoffmann (2010)]
- Business process composition@SAP [Hoffmann *et al.* (2010)]
- Automatic penetration testing [Lucangeli *et al.* (2010)]

So is there any problem left?

Relaxed Planning, Take-Home Message

Summary

- *"Act as if what was once true remains true forever"*
- General relaxation principle with interesting behavior (eg. TSP $\mapsto$ Minimum Spanning Tree)
- Great in (huge) set of planning competition benchmarks

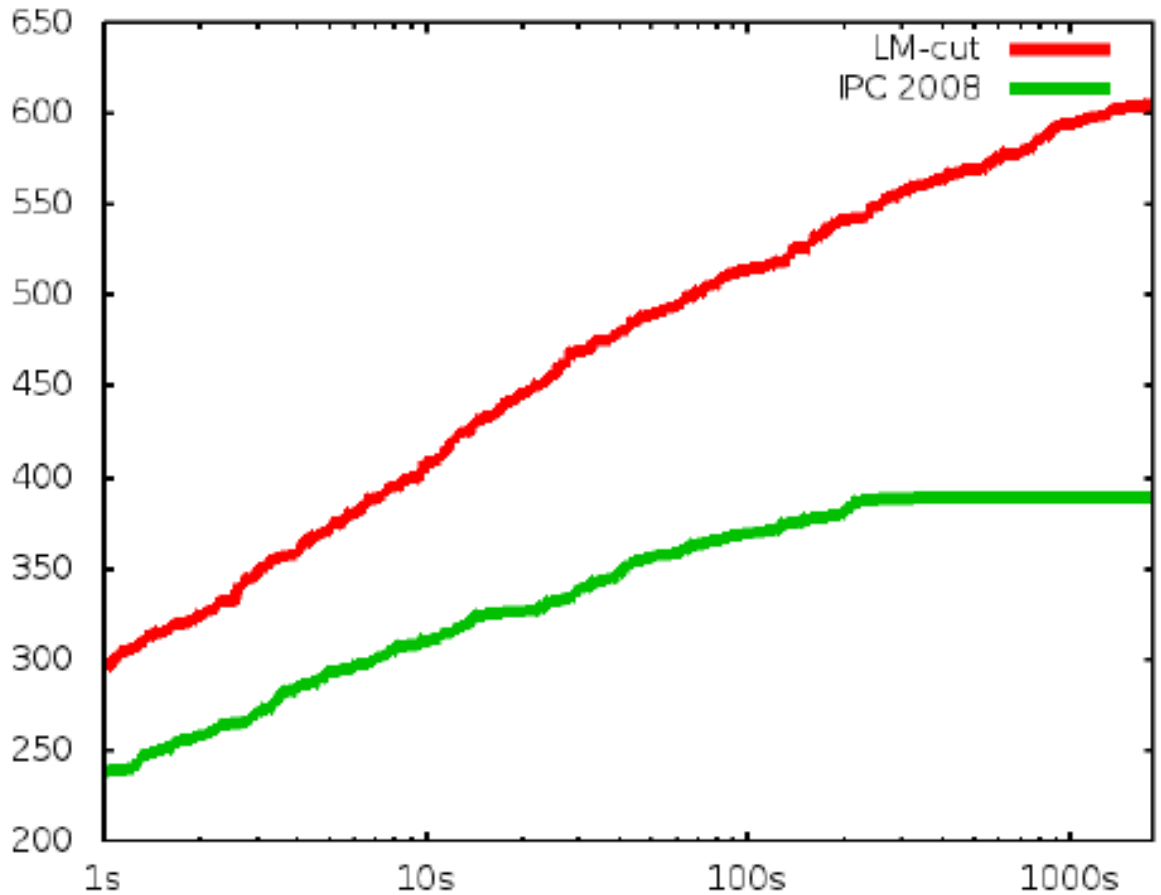
Works also in applications!

- NL sentence generation [Koller and Hoffmann (2010)]
- Business process composition@SAP [Hoffmann *et al.* (2010)]
- Automatic penetration testing [Lucangeli *et al.* (2010)]

So is there any problem left?

- *Optimal* planning
- h^+ doesn't work in *every* domain

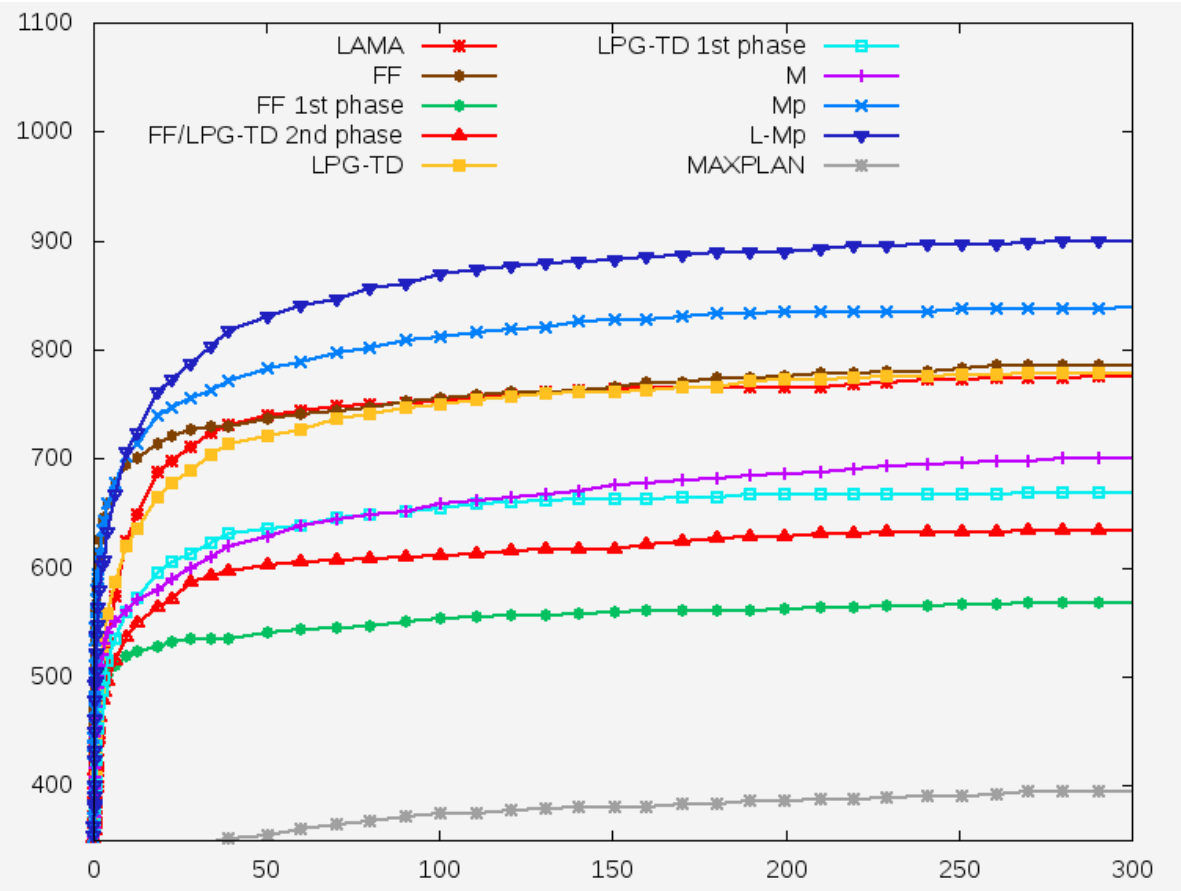
The Recent Boost in Optimal Planning



IPC 2008: best optimal planner

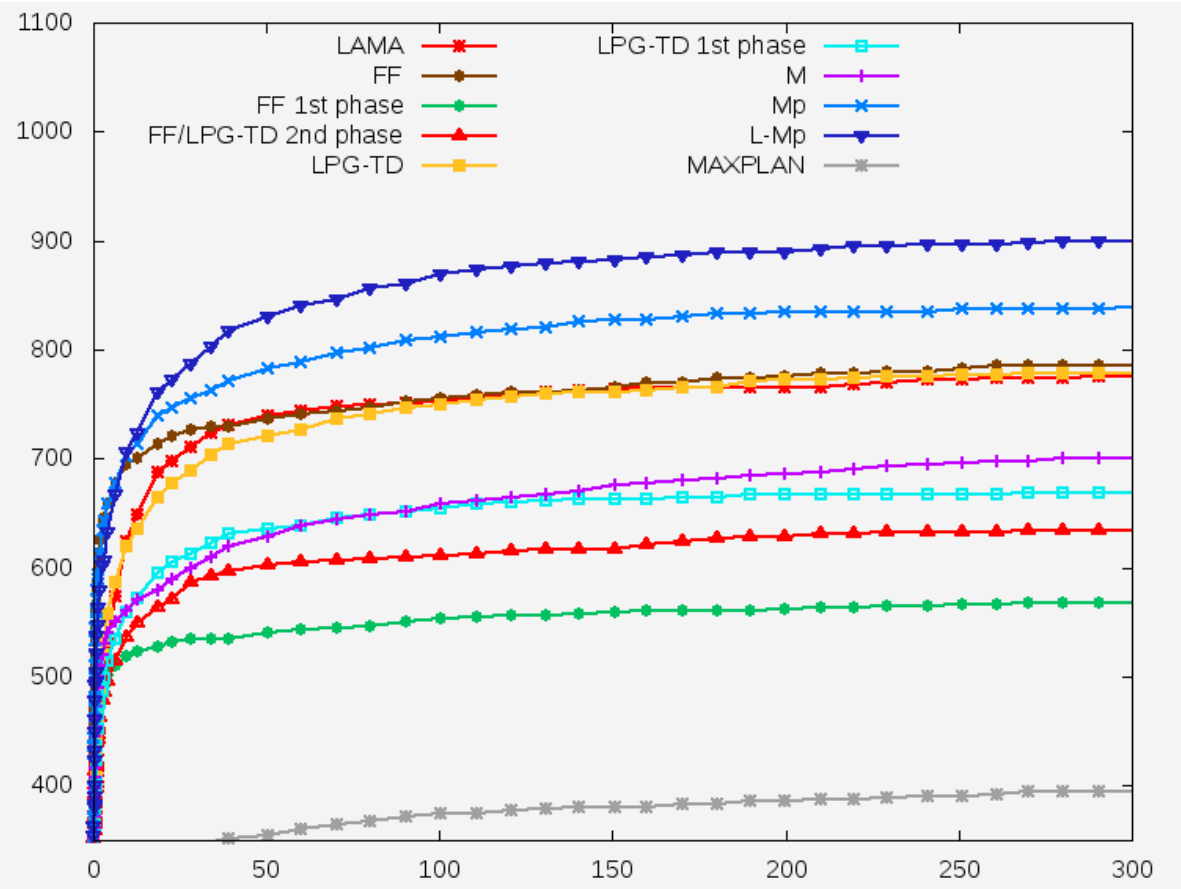
$h^{\text{LM-cut}}$: *lower*-bound h^+ [Helmert and Domshlak (2009)]

The Recent Boost in SAT-Based Planning



M, Mp: [Rintanen *et al.* (2006); Rintanen (2010)]; L-Mp:+LAMA

The Recent Boost in SAT-Based Planning



M, Mp: [Rintanen *et al.* (2006); Rintanen (2010)]; L-Mp:+LAMA

Wasn't very competitive at IPC'11, though ...

Agenda

Is it interesting to do research in?

Why, of course it is!

Is it interesting to do research in?

Why, of course it is!

(You didn't expect me to give you an honest answer to that, did you?)

Why is planning research interesting?

Planning vs. SAT

Why is planning research interesting?

Planning vs. SAT

- SAT: *"How to push around the bits in unit propagation?"*
- Planning: *"How to abstract a planning problem?"*

(abstraction $\equiv$ relaxation $\equiv$ heuristic)

Why is planning research interesting?

Planning vs. SAT

- SAT: *"How to push around the bits in unit propagation?"*
- Planning: *"How to abstract a planning problem?"*

(abstraction $\equiv$ relaxation $\equiv$ heuristic)

The life of a planning researcher?

while (not retired) **do**
    think up some new heuristic $h^{\text{foo-bar}}$
    run it on the benchmarks
endwhile

Why is planning research interesting?

Planning vs. SAT

- SAT: *"How to push around the bits in unit propagation?"*
- Planning: *"How to abstract a planning problem?"*

(abstraction $\equiv$ relaxation $\equiv$ heuristic)

The life of a planning researcher?

while (not retired) **do**
    think up some new heuristic $h^{\text{foo-bar}}$
    run it on the benchmarks
endwhile

Nope. *Understanding* heuristics = a natural science!

How do heuristics relate to each other?

Relaxed Plans

h^+

$h^{\max}$

Abstraction

pattern databases

merge-and-shrink

Critical Paths

h^1

h^2

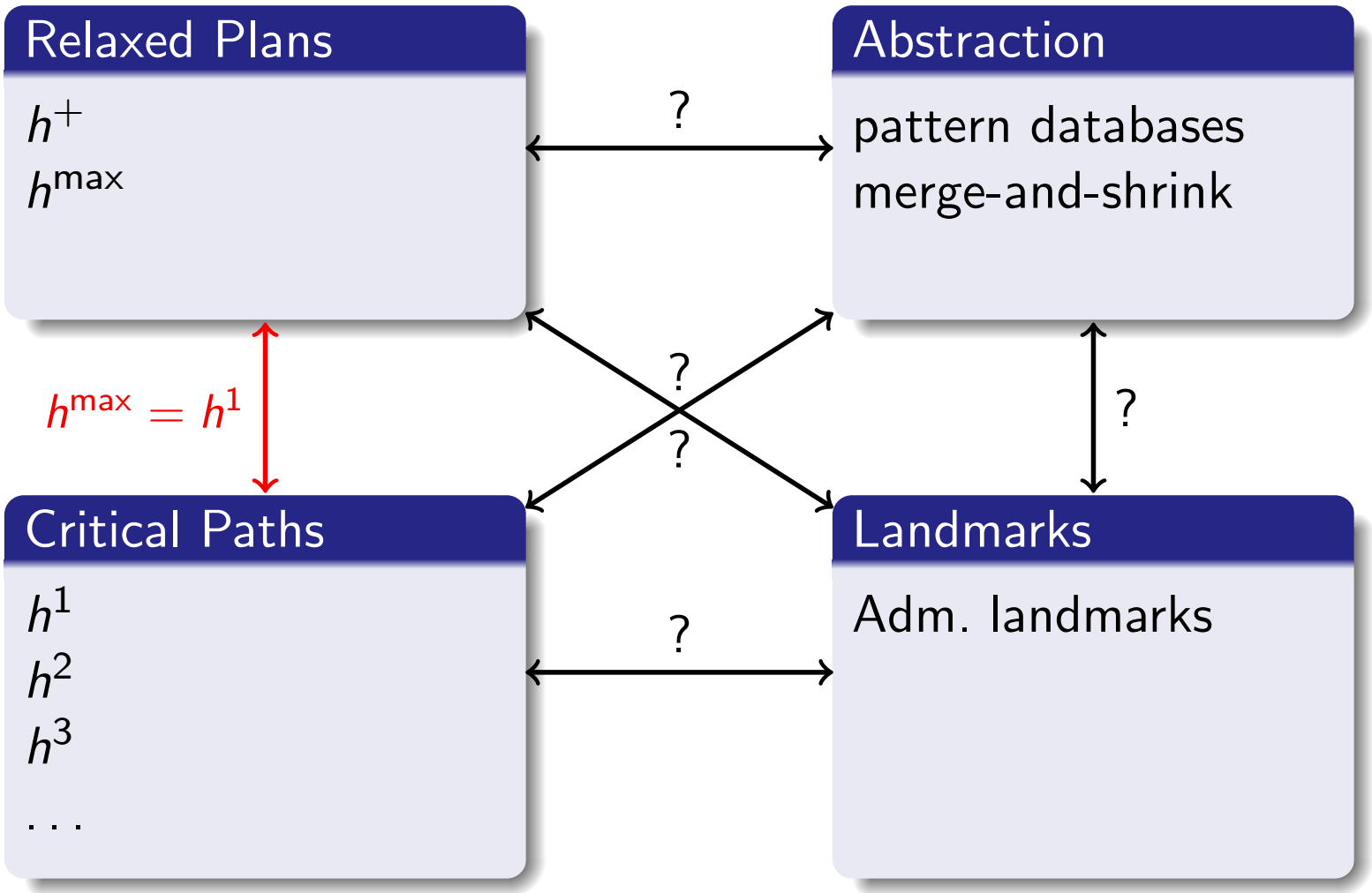
h^3

...

Landmarks

Adm. landmarks

How do heuristics relate to each other?



Compilability of Heuristics

[Helmert and Domshlak (2009)]:

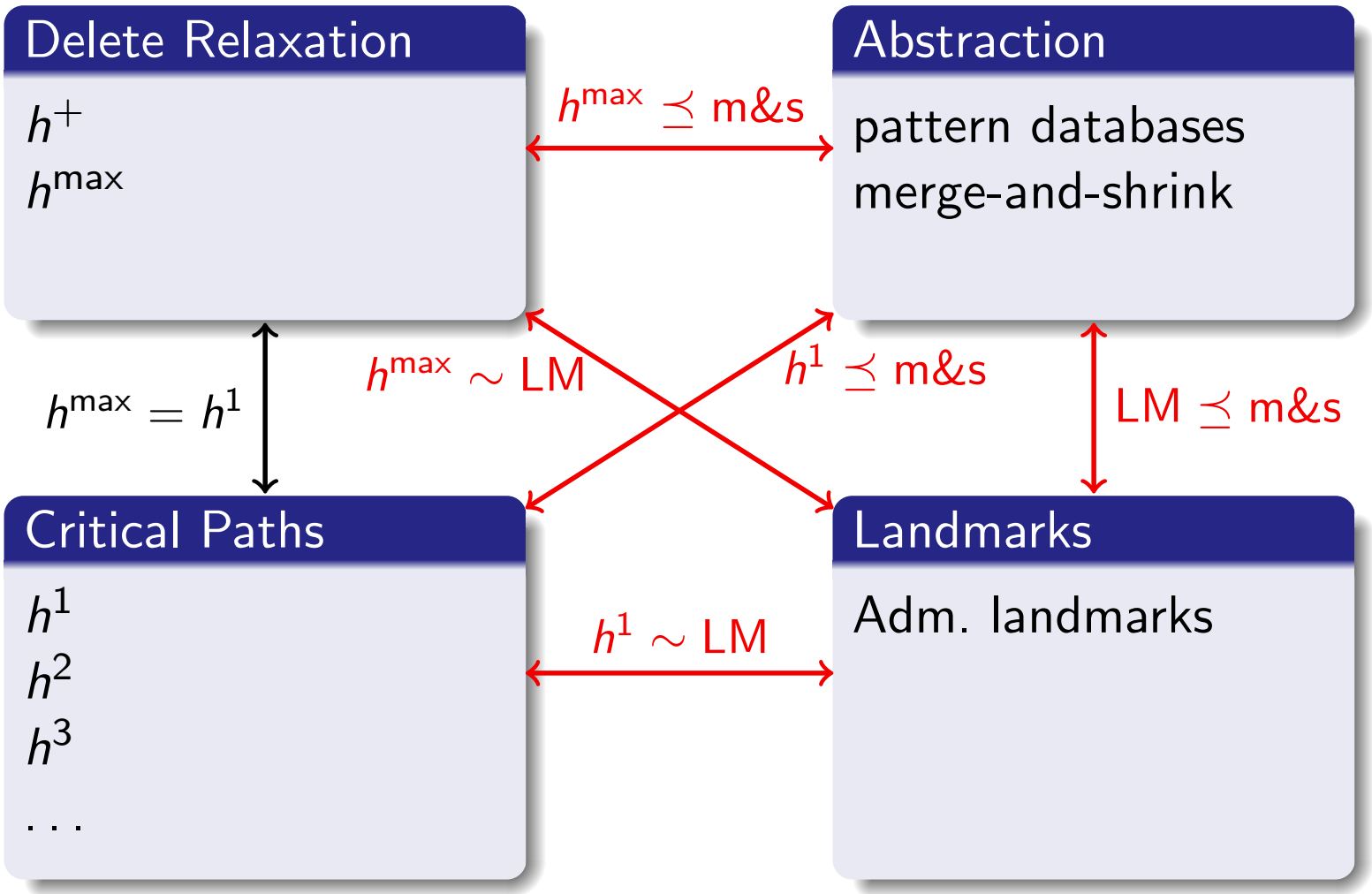
Comparing admissible heuristics!

Definition (Compilability)

Heuristic family $\mathcal{H}$ can be compiled into heuristic family $\mathcal{H}'$, $\mathcal{H} \preceq \mathcal{H}'$, if an algorithm **A** with the following properties exists:

- Input: heuristic $h \in \mathcal{H}$, state s
- Output: heuristic $h' \in \mathcal{H}'$ such that $h'(s) \geq h(s)$
- Runtime: polynomial

How do heuristics relate to each other?



So what?

It's nice to understand nature!

- Landmarks are incomparable to PDBs
- Landmarks are subsumed by merge-and-shrink
- Landmarks are equivalent to $h^{\max}$

So what?

It's nice to understand nature!

- Landmarks are incomparable to PDBs
- Landmarks are subsumed by merge-and-shrink
- Landmarks are equivalent to $h^{\max}$

It's nice to improve planners!

- LM $\preceq h^{\max}$ trivial
- $h^{\max} \preceq$ LM not trivial (and quite unexpected)
  $\Longrightarrow$ proof constructs LM so that heuristic will dominate $h^{\max}$

So what?

It's nice to understand nature!

- Landmarks are incomparable to PDBs
- Landmarks are subsumed by merge-and-shrink
- Landmarks are equivalent to $h^{\max}$

It's nice to improve planners!

- LM $\preceq h^{\max}$ trivial
- $h^{\max} \preceq$ LM not trivial (and quite unexpected)
  $\Longrightarrow$ proof constructs LM so that heuristic will dominate $h^{\max}$
- Implementation: LM-cut!!

Where does a given heuristic work well?

In which sub-class of planning does it deliver "good" estimates?

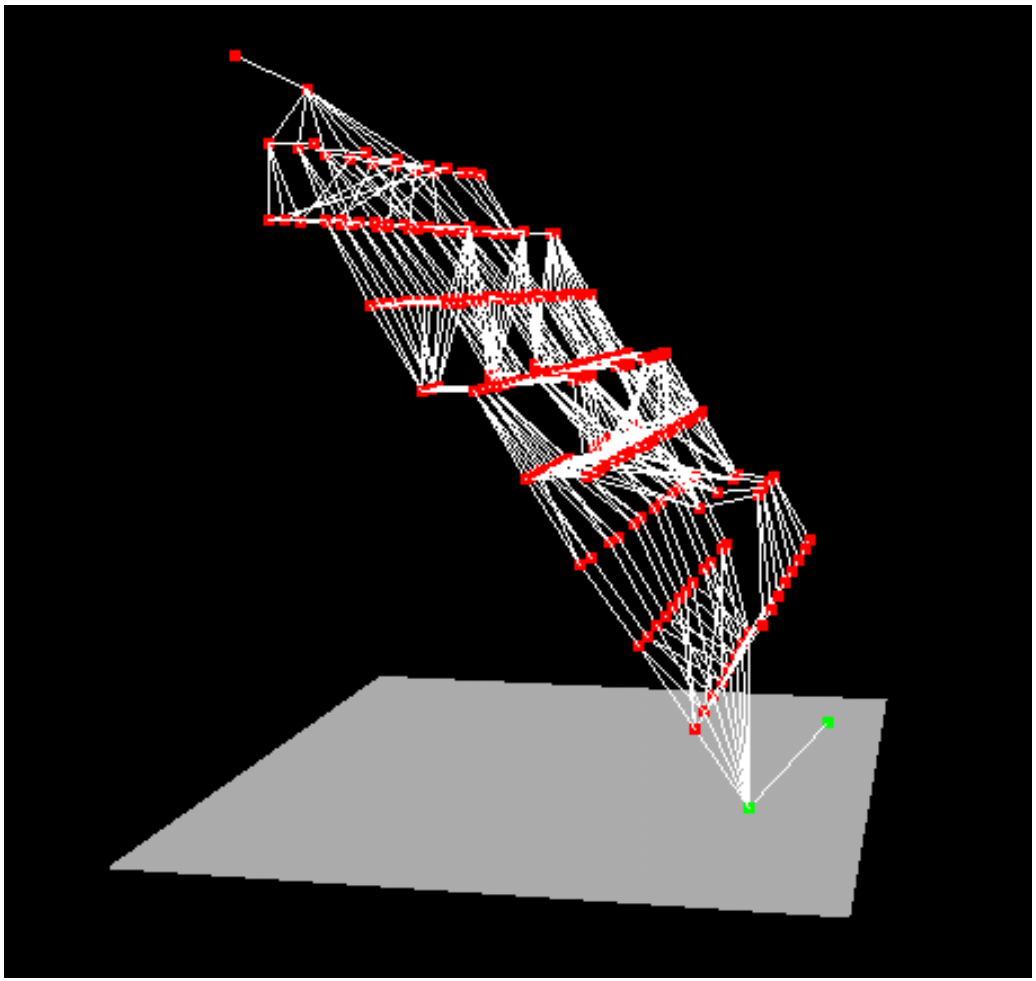
Where does a given heuristic work well?

In which sub-class of planning does it deliver "good" estimates?

Where does h^+ work well?

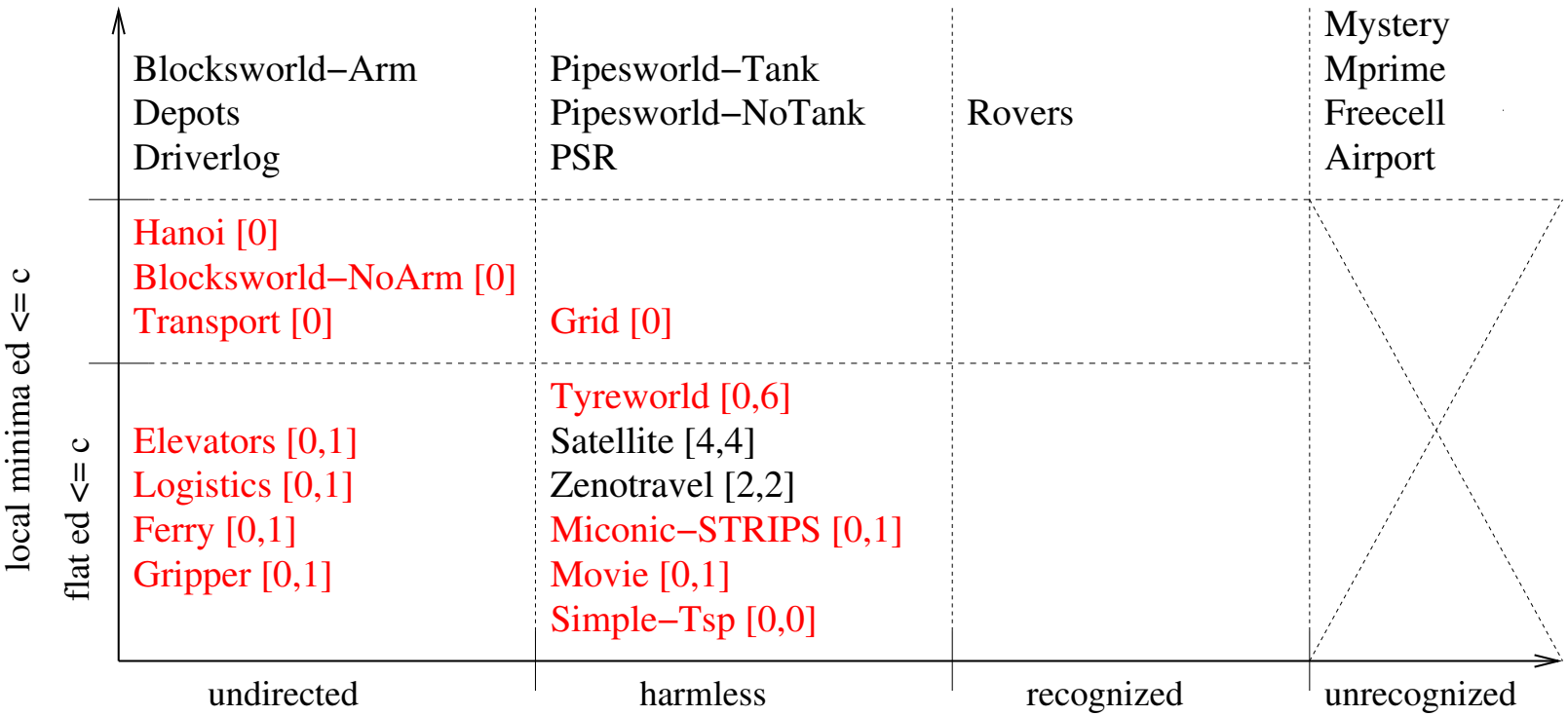
Could the computer determine automatically whether it does?

Where h^+ Works: "Gripper" Benchmark



Where h^+ Works [Hoffmann, AIPS'02, JAIR'05]

Results proved per-domain by hand:



Red == no local minima. Can we recognize this automatically?

Can we recognize this automatically?

Yes we can! [Hoffmann (2011)]

Basic result:

If the "causal graph" is acyclic and all actions are "invertible", then there are no local minima under h^+.

- "Causal graph", "invertible": syntax of input model
- Static analysis in split seconds

Can we recognize this automatically?

Yes we can! [Hoffmann (2011)]

Basic result:

If the "causal graph" is acyclic and all actions are "invertible", then there are no local minima under h^+.

- "Causal graph", "invertible": syntax of input model
- Static analysis in split seconds

$\Rightarrow$ **Analyzing search topology without running any search!**

- Can also derive bound on lookahead required to find state with strictly smaller h^+ value

The TorchLight Tool

Global Analysis

- Sufficient criterion for "no local minima at all"
- What about domains *with* local minima?

The TorchLight Tool

Global Analysis

- Sufficient criterion for "no local minima at all"
- What about domains *with* local minima?

Local Analysis

- Sufficient criterion for "state s is no local minimum"
- Randomly sample states s
- Success rate = percentage of s where criterion applies

The TorchLight Tool

Global Analysis

- Sufficient criterion for "no local minima at all"
- What about domains *with* local minima?

Local Analysis

- Sufficient criterion for "state s is no local minimum"
- Randomly sample states s
- Success rate = percentage of s where criterion applies

Diagnosis

- Reasons for analysis failure

  $\Longrightarrow$ action effects (potentially) causing local minima!

Hoffmann vs. TorchLight

Zenotravel	Hanoi [0]
Satellite	Airport [0]
Rovers	Blocksworld–Arm [30]
PSR	Mystery [39]
Pipesworld–Tank	Pipesworld–Tank [40]
Pipesworld–NoTank	Mprime [49]
Mystery	PSR [50]
Mprime	Freecell [56]
Freecell	Blocksworld–NoArm [57]
Driverlog	Pipesworld–NoTank [76]
Depots	Grid [80]
Blocksworld–Arm	Depots [81]
Airport	Zenotravel [95]
Tyreworld	Tyreworld [100]
Transport	Transport [100]
Simple–Tsp	Simple–Tsp [100]
Movie	Satellite [100]
Miconic–STRIPS	Rovers [100]
Logistics	Movie [100]
Hanoi	Miconic–STRIPS [100]
Gripper	Logistics [100]
Grid	Gripper [100]
Ferry	Ferry [100]
Elevators	Elevators [100]
Blocksworld–NoArm	Driverlog [100]

- Success rate: average per-domain from single sample state per-instance

Hoffmann vs. TorchLight

Zenotravel	Hanoi [0]
Satellite	Airport [0]
Rovers	Blocksworld–Arm [30]
PSR	Mystery [39]
Pipesworld–Tank	Pipesworld–Tank [40]
Pipesworld–NoTank	Mprime [49]
Mystery	PSR [50]
Mprime	Freecell [56]
Freecell	Blocksworld–NoArm [57]
Driverlog	Pipesworld–NoTank [76]
Depots	Grid [80]
Blocksworld–Arm	Depots [81]
Airport	Zenotravel [95]
Tyreworld	Tyreworld [100]
Transport	Transport [100]
Simple–Tsp	Simple–Tsp [100]
Movie	Satellite [100]
Miconic–STRIPS	Rovers [100]
Logistics	Movie [100]
Hanoi	Miconic–STRIPS [100]
Gripper	Logistics [100]
Grid	Gripper [100]
Ferry	Ferry [100]
Elevators	Elevators [100]
Blocksworld–NoArm	Driverlog [100]

- Not all domains are "fully recognized" . . .
- . . . mostly because Hoffmann is too optimistic

Hoffmann vs. TorchLight

Zenotravel	Hanoi [0]
Satellite	Airport [0]
Rovers	Blocksworld–Arm [30]
PSR	Mystery [39]
Pipesworld–Tank	Pipesworld–Tank [40]
Pipesworld–NoTank	Mprime [49]
Mystery	PSR [50]
Mprime	Freecell [56]
Freecell	Blocksworld–NoArm [57]
Driverlog	Pipesworld–NoTank [76]
Depots	Grid [80]
Blocksworld–Arm	Depots [81]
Airport	Zenotravel [95]
Tyreworld	Tyreworld [100]
Transport	Transport [100]
Simple–Tsp	Simple–Tsp [100]
Movie	Satellite [100]
Miconic–STRIPS	Rovers [100]
Logistics	Movie [100]
Hanoi	Miconic–STRIPS [100]
Gripper	Logistics [100]
Grid	Gripper [100]
Ferry	Ferry [100]
Elevators	Elevators [100]
Blocksworld–NoArm	Driverlog [100]

- Some new domains are "fully recognized" . . .
- . . . mostly because Hoffmann is too pessimistic

Hoffmann vs. TorchLight

Zenotravel	Hanoi [0]
Satellite	Airport [0]
Rovers	Blocksworld–Arm [30]
PSR	Mystery [39]
Pipesworld–Tank	Pipesworld–Tank [40]
Pipesworld–NoTank	Mprime [49]
Mystery	PSR [50]
Mprime	Freecell [56]
Freecell	Blocksworld–NoArm [57]
Driverlog	Pipesworld–NoTank [76]
Depots	Grid [80]
Blocksworld–Arm	Depots [81]
Airport	Zenotravel [95]
Tyreworld	Tyreworld [100]
Transport	Transport [100]
Simple–Tsp	Simple–Tsp [100]
Movie	Satellite [100]
Miconic–STRIPS	Rovers [100]
Logistics	Movie [100]
Hanoi	Miconic–STRIPS [100]
Gripper	Logistics [100]
Grid	Gripper [100]
Ferry	Ferry [100]
Elevators	Elevators [100]
Blocksworld–NoArm	Driverlog [100]

- Success rates are more than a "yes/no" answer!

Agenda

Quo Vadis, Planning?

+ Many effective heuristics and thus planners

Quo Vadis, Planning?

+ Many effective heuristics and thus planners
+ Understanding grows, can even predict performance

Quo Vadis, Planning?

+ Many effective heuristics and thus planners
+ Understanding grows, can even predict performance
+ Great for rapid prototyping

Quo Vadis, Planning?

+ Many effective heuristics and thus planners
+ Understanding grows, can even predict performance
+ Great for rapid prototyping
– No-one knows our technology and no-one can write PDDL

Quo Vadis, Planning?

+ Many effective heuristics and thus planners
+ Understanding grows, can even predict performance
+ Great for rapid prototyping
– No-one knows our technology and no-one can write PDDL

Make planning accessible . . .

- Modeling environments [Vaquero *et al.* (2007); Simpson *et al.* (2007)]
- Model learning techniques [Cresswell *et al.* (2009); Castillo *et al.* (2010)]
- Domain analysis (e.g. TorchLight in current work)

. . . and get it out there!

Your Questions, Answered

- Planning? What's that?

  *"**One** solver for **all** problems that can be cast as finding a path in a large transition system."*

- What is it good for?

  "Cost-effective software engineering (fast, cheap, flexible)."

- Does it work? *"Yes!"*

- Is it interesting to do research in? *"Yes!"*

- And now, what?

  `http://www.loria.fr/~hoffmanj/PlanningForDummies.zip`

Last Slide

Thanks for listening.

Any more questions?

References

Blai Bonet, Gábor Loerincs, and Héctor Geffner. A robust and fast action selection mechanism for planning. In *Proceedings of the 14th National Conference of the American Association for Artificial Intelligence*, 1997.

Tom Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1–2):165–204, 1994.

Luis A. Castillo, Lluvia Morales, Arturo González-Ferrer, Juan Fernández-Olivares, Daniel Borrajo, and Eva Onaindia. Automatic generation of temporal planning domains for e-learning problems. *Journal of Scheduling*, 13(4):347–362, 2010.

Stephen Cresswell, Thomas Leo McCluskey, and Margaret Mary West. Acquisition of object-centred domain models from planning examples. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling*, 2009.

Maria Fox and Derek Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20:61–124, 2003.

Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. Planning through stochastic local search and temporal action graphs. *Journal of Artificial Intelligence Research*, 20:239–290, 2003.

References

Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling*, 2009.

Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

Jörg Hoffmann, Ingo Weber, and Frank Michael Kraft. SAP speaks PDDL. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

Jörg Hoffmann. Analyzing search topology without running any search: On the connection between causal graphs and h^+. *Journal of Artificial Intelligence Research*, 41:155–229, 2011.

Alexander Koller and Joerg Hoffmann. Waking up a sleeping rabbit: On natural-language sentence generation with FF. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling*, 2010.

References

Alexander Koller and Ronald Petrick. Experiences with planning for natural language generation. *Computational Intelligence*, 27(1):23–40, 2011.

Jorge Lucangeli, Carlos Sarraute, and Gerardo Richarte. Attack planning in the real world. In *Proceedings of the 2nd Workshop on Intelligent Security*, 2010.

McDermott et al. *The PDDL Planning Domain Definition Language*. The AIPS-98 Planning Competition Comitee, 1998.

Drew McDermott. A heuristic estimator for means-ends analysis in planning. In *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems*, 1996.

Silvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010.

Jussi Rintanen, Keijo Heljanko, and Ilkka Niemelä. Planning as satisfiability: parallel plans and algorithms for plan search. *Artificial Intelligence*, 170(12-13):1031–1080, 2006.

References

Jussi Rintanen. Heuristics for planning with SAT. In *Proceeedings of the 16th International Conference on Principles and Practice of Constraint Programming*, 2010.

Wheeler Ruml, Minh Binh Do, Rong Zhou, and Markus P. J. Fromherz. On-line planning and scheduling: An application to controlling modular printers. *Journal of Artificial Intelligence Research*, 40:415–468, 2011.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice-Hall, Englewood Cliffs, NJ, 2003.

Ron M. Simpson, Diane E. Kitchin, and T. L. McCluskey. Planning domain definition using GIPO. *Knowledge Engineering Review*, 22(2):117–134, 2007.

Tiago Stegun Vaquero, Victor Romero, Flavio Tonidandel, and José Reinaldo Silva. itSIMPLE 2.0: an integrated tool for designing planning domains. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling*, 2007.