Automatic Planning

Jörg Hoffmann

University of Innsbruck
DERI

Masterstudium Informatik Vertiefungsmodul (WM 9)

Local Search Topology

- Success of heuristic search depends on quality of heuristic function
- Measure quality in terms of topological properties of the "search space surface" (local minima etc.)
- Here: FF and related planners extremely efficient in many planning benchmarks
- *Can we say something about the quality of these planners' heuristic functions, in these benchmarks?*
- We focus exclusively on forward search in the state space

Automatic Planning 8. Local Search Topology

h^+ and h^{FF}

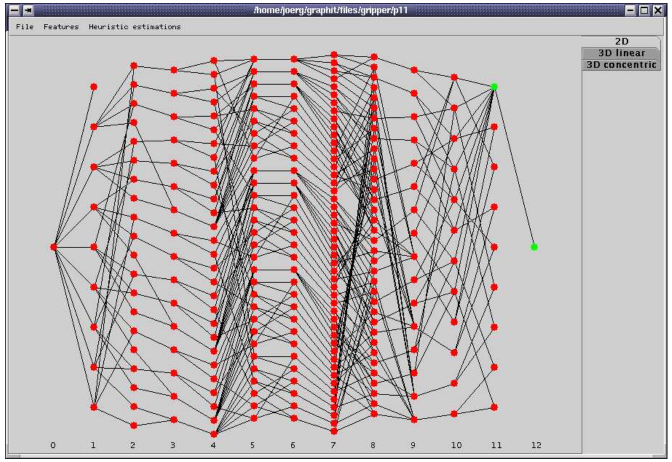
Definition
Let (P, A, I, G) be a STRIPS task, s a state in the state space. $h^+(s)$ is the length of an optimal plan for (P, A^+, s, G), or ∞ if there is no such plan.

- Computing h^+ is NP-hard
- h^{FF} approximates h^+ by a two-step (forward, backward) relaxed Graphplan process, computing some (not necessarily optimal) relaxed plan
- Related things are done in today's fastest planners
- No provable connection between h^+ and h^{FF} except for ∞ (see later)

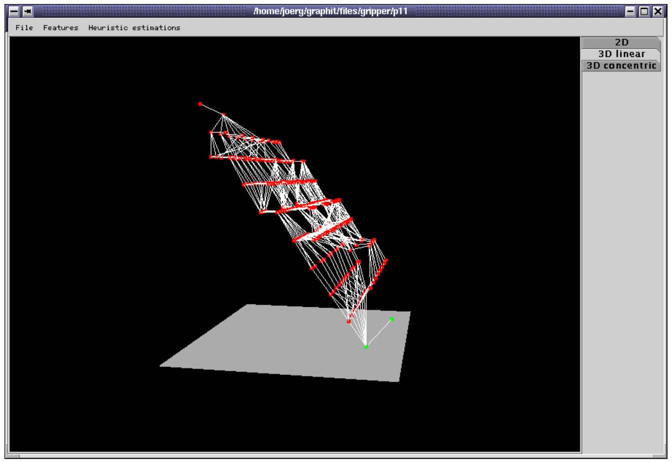
Automatic Planning
8. Local Search Topology

- A Reminder: h^+ and h^{FF}
- A Preview
- Local Minima & Co.
- 20 Planning Benchmark Domains
- Measuring Local Search Topology
- Proving Local Search Topology
- Sampling Local Search Topology

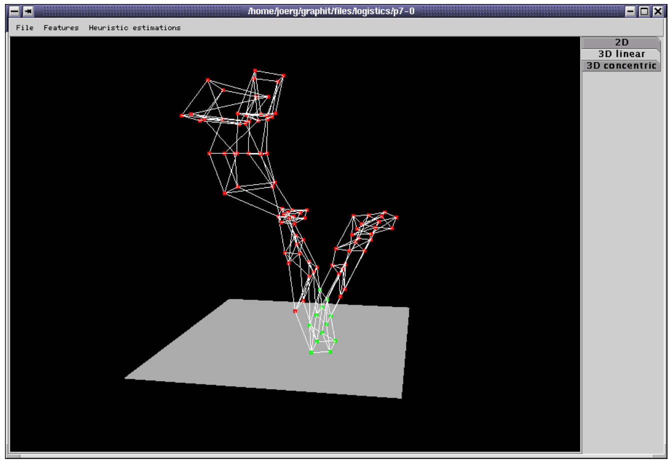
Visualized State Spaces



h^+ in Gripper

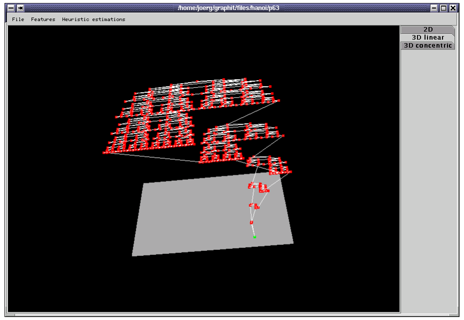


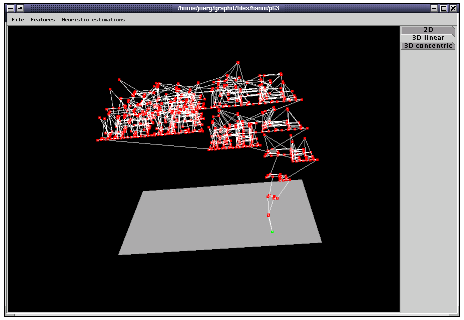
h^+ in Logistics



h^+ and h^{FF} in Towers of Hanoi

Instance with 6 discs:





Can you guess what the explanation is?

Automatic Planning
8. Local Search Topology

Local Search Topology

"Local Search Topology" [Frank et al,JAIR-1997] means topological phenomena of the state space graph annotated with heuristic values.

Here we'll investigate:

- Dead ends
- Local minima
- Benches

We consider *solvable* tasks only. Why???

State Space Graphs

We denote with S the set of reachable states:

$$S := \{s \mid s = I \vee \exists s' \in S, a \in A : result(s', \langle a \rangle) = s\}$$

and with E the set of edges (action applications) between them:

$$E := \{(s', s) \mid \exists a \in A : result(s', \langle a \rangle) = s\}$$

From what we know about I and G, what can we conclude about the graph (S, E)*?*

Annotated State Space Graphs

We annotate states $s \in S$ with a heuristic value $h(s)$:

Definition

Given a STRIPS task (P, A, I, G) and its state space graph (S, E), a heuristic function is a function $h : S \mapsto \mathbf{N_0} \cup \{\infty\}$ so that $h(s) = 0$ iff $G \subseteq s$.

That is, we assume that h is exact for goal states.

Proposition

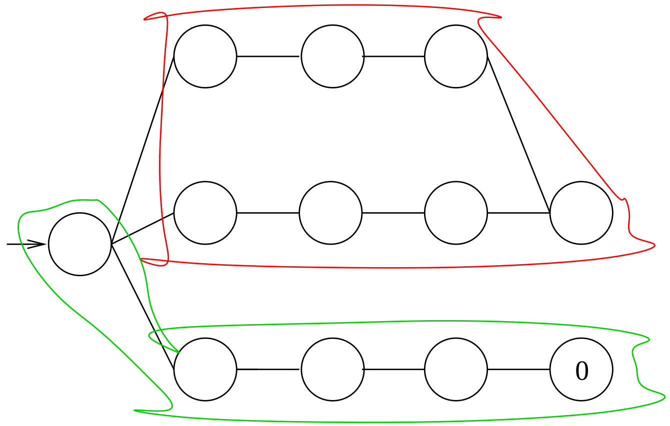
Given a STRIPS task (P, A, I, G) and its state space graph (S, E), for all $s \in S$ we have $h^+(s) = 0$ iff $h^{FF}(s) = 0$ iff $G \subseteq s$.

Proof.

Do you see why? □

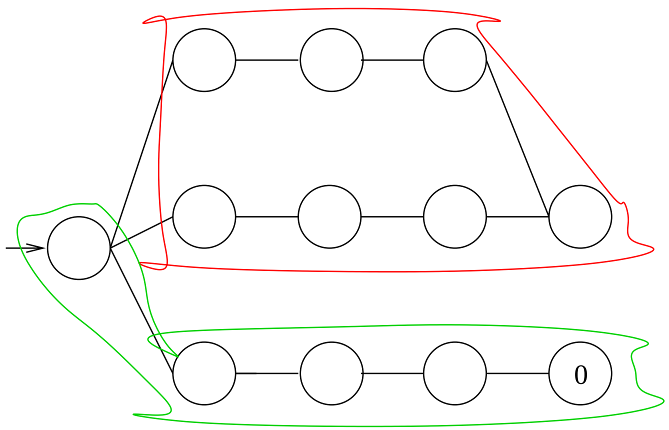
Dead Ends: Example

Dead ends are states from which no solution can be reached:



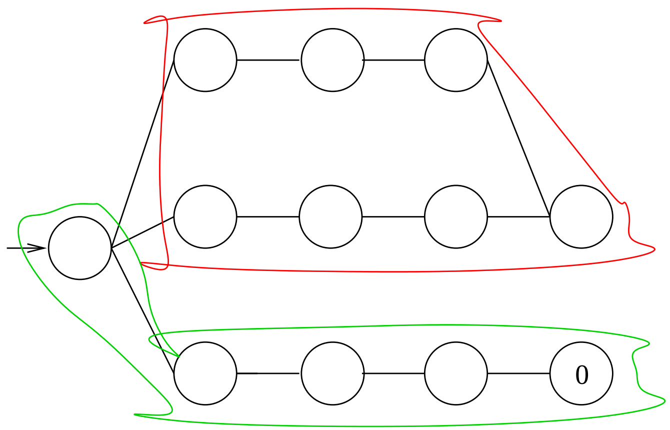
Local Minima: Example

Local minima are regions where all neighbours look worse:



Benches: Example

Benches are flat regions with better neighbours:



Dead Ends

Definition
Let $s \in S$ be a state. s is a dead end if $sd(s) = \infty$.

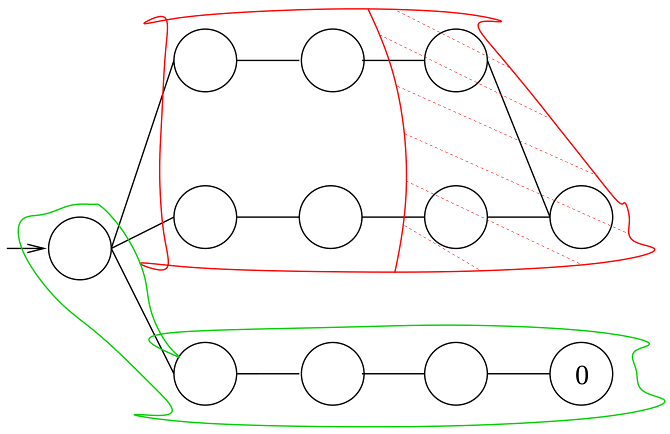
- Dead ends cannot arise if ???

Definition
Let $s \in S$ be a dead end state. s is recognized if $h(s) = \infty$.

- h returns ∞ to indicate that s is a dead end
- h is called completeness-preserving if $h(s) = \infty \Rightarrow sd(s) = \infty$

Dead Ends: Example

Recognized and unrecognized dead ends:



Dead Ends

Proposition
Let $s \in S$ be a state. If $h^+(s) = \infty$, then s is a dead end.

Proof.
Do you see why? □

The same is true for any h based on solving a relaxation.

Proposition
Let $s \in S$ be a state. $h^+(s) = \infty$ iff $h^{FF}(s) = \infty$.

Proof.
Do you see why? □

Dead Ends

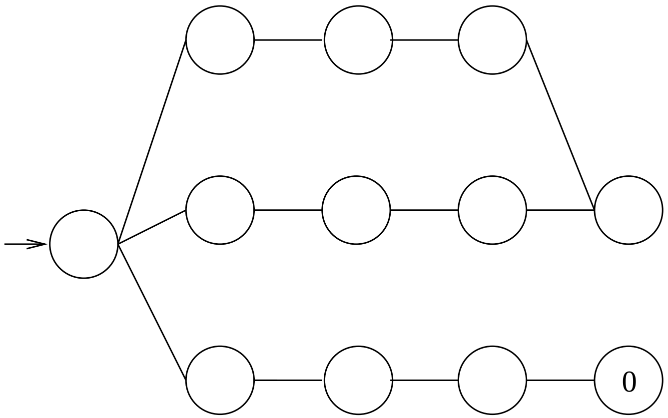
4 classes of behaviour with respect to dead ends:

Definition
A STRIPS task and its state space are called

1. *undirected, if $\forall (s, s') \in E : (s', s) \in E$,*
2. *harmless, if $\exists (s, s') \in E : (s', s) \notin E$, and $\forall s \in S : sd(s) < \infty$,*
3. *recognized, if $\exists s \in S : sd(s) = \infty$, and $\forall s \in S : sd(s) = \infty \Rightarrow h(s) = \infty$,*
4. *unrecognized, if $\exists s \in S : sd(s) = \infty \wedge h(s) < \infty$.*

Dead Ends: Example

Undirected/harmless/recognized/unrecognized:



Plateaus, I

A plateau is a region of states in S that all look the same:

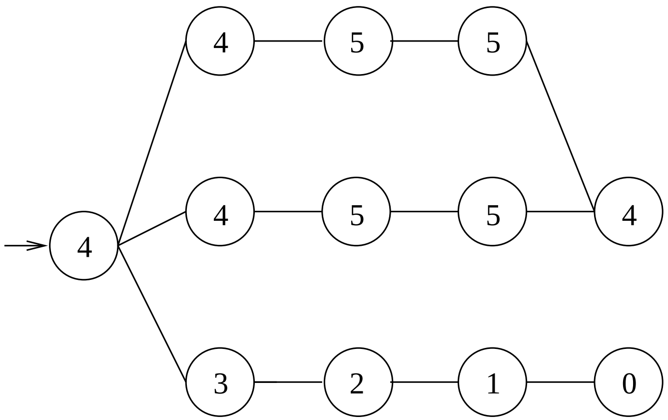
Definition

For $l \in \mathbf{N} \cup \{\infty\}$, a plateau of level l is a maximal subset P of S for which the induced subgraph in (S, E) is strongly connected, and $h(s) = l$ for each $s \in P$.

- Plateaus look the same both in terms of reachability, and in terms of heuristic value
- They form the smallest entity in the subsequent definitions and investigations

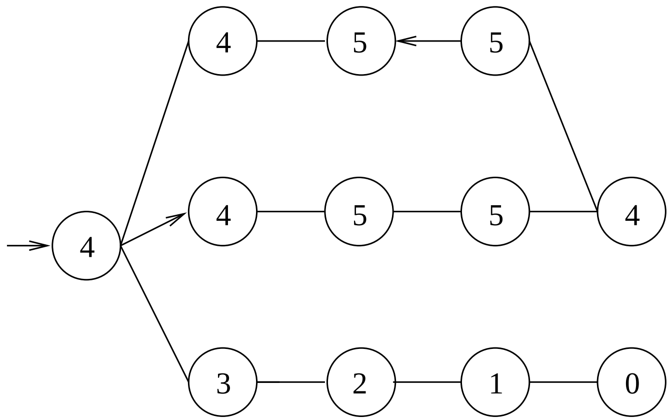
Plateaus: Example, I

What are the plateaus here?



Plateaus: Example, II

What are the plateaus here?



Plateaus, II

We distinguish between plateaus with the following defs:

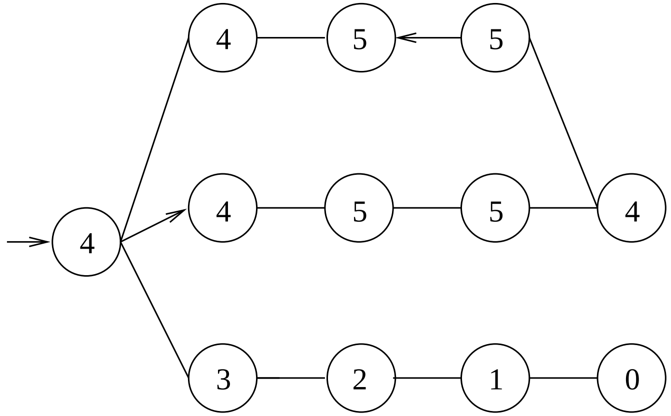
Definition

A path of connected states in S is flat *if all states on it have the same h value. For a plateau P of level l, an* exit *(of P) is a state s reachable from P on a (potentially empty) flat path, such that $h(s) = l$ and there exists a state s', $(s, s') \in E$, with $h(s') < h(s)$.*

- The standard definition [Frank et al,JAIR-1997] says that the exit lies *on* the plateau itself
- This is for undirected graphs; there, with the above definition any exit s of P indeed has $s \in P$ – *Exercise*
- In directed graphs, what matters is if we can escape the plateau without temporarily worsening the heuristic value

Plateaus: Example, III

Give me a plateau and its exit:



Local Minima & Co., I

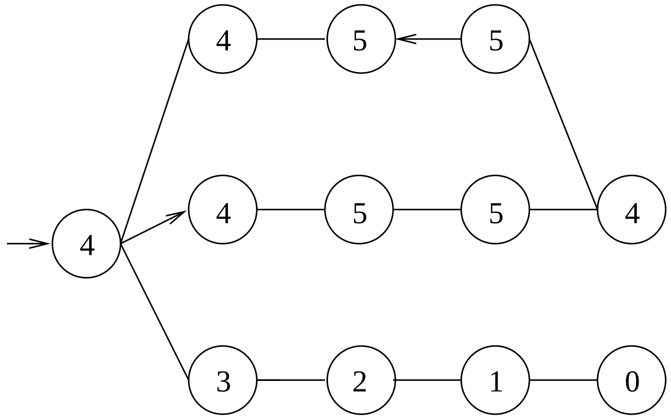
Definition

We distinguish 5 classes of plateaus:

1. *A* recognized dead end *is a plateau P of level ∞.*
2. *A* local minimum *is a plateau P of level $0 < l < \infty$ that has no exits.*
3. *A* bench *is a plateau P of level $0 < l < \infty$ that has at least one exit, and where at least one state on P is not an exit.*
4. *A* contour *is a plateau P of level $0 < l < \infty$ that consists entirely of exits.*
5. *A* global minimum *is a plateau P of level 0.*

Plateaus: Example, IV

What are the classes of all the plateaus?



A Concrete Example, I

Under the Manhattan Distance heuristic h^{MD}, what kind of plateau do we have here?

1	2	3
6	7	
4	8	9

A Concrete Example, II

Under the Manhattan Distance heuristic h^{MD}, what kind of plateau do we have here?

1	2	3
6	7	9
4	8	

There aren't any benches in the 8-puzzle under h^{MD} – *why?*

Local Minima & Co., II

For a heuristic or local search, the main difficulty is to *escape local minima and benches*

The difficulty of doing this can be assessed by, e.g.: ???

For benches the following will be relevant:

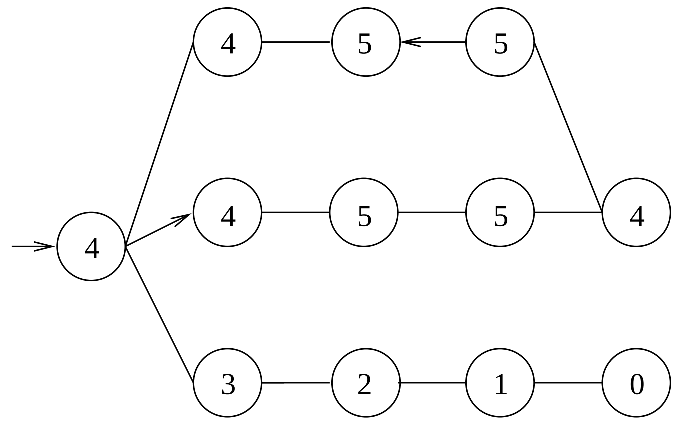
Definition

For a state $s \in S$ on a bench, the exit distance is the length of the shortest path to an exit. The maximal exit distance is the maximum over all s on benches, or 0 if there aren't any benches.

"How far do we need to go (in the worst case) to find an exit?"

Maximal Exit Distance: Example

Give me the exit distances of the states on benches:



Local Minima & Co., III

There are various easy-to-see implications. An important one that we will use:

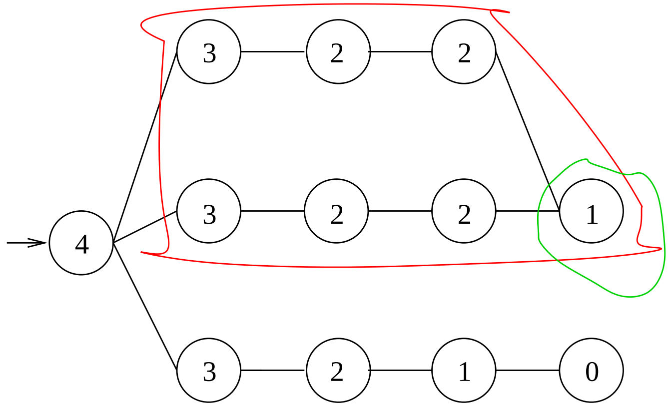
Proposition

If there exists a dead state $s \in S$ with $h(s) < \infty$, then there exists a local minimum $P \subseteq S$.

Proof.

Blackboard. □

Valleys



Automatic Planning
8. Local Search Topology

- A Reminder: h^+ and h^{FF}
- A Preview
- Local Minima & Co.
- 20 Planning Benchmark Domains
- Measuring Local Search Topology
- Proving Local Search Topology
- Sampling Local Search Topology

20 Benchmark Domains

I will show local search topology results for a large set of benchmark domains including all IPC-1 and IPC-2 examples.

A rough classification:

- Transportation domains: locations, vehicles, transportable objects. E.g. Logistics
- Construction domains: build a complex object out of its individual parts. E.g. Blocksworld, also Towers of Hanoi
- Others: . . .

(I also got theoretical results for the 10 IPC-3 and IPC-4 domains, but not the empirical results that we will consider.)

Transportation Domains

- Logistics, Gripper, Ferry: simple transportation, with different vehicles, sometimes with capacity restrictions
- Briefcaseworld: objects inside the briefcase are moved around via a conditional effect
- Driverlog: Logistics on a road map graph and with drivers
- Grid: key transportation on a grid map with locked positions of different shapes
- Miconic-STRIPS, Miconic-SIMPLE, Miconic-ADL: elevator control domains, passengers indicate where they are and wanna go; in "-ADL" with side constraints, VIPs must be served first etc.
- Zenotravel: Logistics with fuel usage, refueling operator
- Mystery, Mprime: Logistics on road map, fuel usage, no refueling. Mprime: transfer fuel between locations

Construction Domains

- Blocksworld-arm,Blocksworld-no-arm: sequential version, and parallel version moving blocks directly
- Depots: combines Blocksworld-arm with Logistics
- Freecell: the card game
- Hanoi: Towers of
- Assembly: a complex object must be assembled out of its parts, obeying ordering constraints etc.

Other Domains

- Simple-Tsp: TSP on a fully connected graph with uniform edge costs. Don't laugh – Graphplan actually dies on this. *Fully automatic* can be tough even if the domain happens to be simple. (Some guys have constructed the domain to show off the performance of their symmetry detection algorithm.)
- Movie: buy some snacks and rewind the tape
- Tireworld: replace n flat tires (jack up, remove wheel, . . .)
- Fridge: replace n broken compressors (unfasten screw, remove backplane, . . .)
- Schedule: process (paint etc.) objects on machines

Automatic Planning
8. Local Search Topology

- A Reminder: h^+ and h^{FF}
- A Preview
- Local Minima & Co.
- 20 Planning Benchmark Domains
- Measuring Local Search Topology
- Proving Local Search Topology
- Sampling Local Search Topology

Measuring Local Search Topology

Build the entire state spaces of example tasks, annotate them with the heuristic values, and measure topology parameters.

- Only possible for small instances
- Parameters: e.g. percentage of unrecognized dead ends, average size of valleys, ...

In what follows:

- 100 random instances of every domain (except special cases like Gripper)
- Summarize (average) data as function of “task size” as given by nr. of objects (trucks etc.)

Measuring Local Search Topology

Percentage of states with $h^+ < \infty$:

Domain	I_0	I_1	I_2	I_3	I_4
Assembly	100.0	99.8	99.5	99.0	99.9
Freecell	100.0			98.6	97.7
Miconic-ADL	100.0		88.5	90.9	84.7
Mprime	68.1	56.1	36.6	30.7	43.3
Mystery	52.7	36.3	28.8	21.4	14.1
Schedule	55.1	63.7	67.7	97.7	97.0

We call this the “relevant” state space. All others: 100%.

No differences here to h^{FF} since $h^{FF} = \infty \Leftrightarrow h^+ = \infty$.

Measuring Local Search Topology

Percentage, in relevant state space, of states with $sd = \infty, h^+ < \infty$ (unrecognized dead ends):

Domain	I_0	I_1	I_2	I_3	I_4
Freecell	0.0			0.4	2.8
Miconic-ADL	0.0		0.0	5.9	6.9
Mprime	13.3	37.9	55.5	41.5	58.8
Mystery	15.7	32.1	48.9	58.6	36.6

All other 16 domains: none.

Measuring Local Search Topology

Percentage, in relevant state space, of states located on valleys under h^+:

Domain	I_0	I_1	I_2	I_3	I_4
Blocksworld-arm	0.0	21.0		31.1	47.5
Freecell	0.0			0.4	2.8
Miconic-ADL	0.0		0.0	5.9	6.9
Mprime	13.3	38.3	56.2	42.2	65.9
Mystery	15.7	32.6	49.6	60.9	37.0
Schedule	24.8	32.6	33.1	24.3	19.9

What is the most significant observation here?

Measuring Local Search Topology

Percentage, in relevant state space, of states located on valleys under h^{FF}:

Domain	I_0	I_1	I_2	I_3	I_4
Assembly	0.0	0.0	0.0	0.0	0.2
Blocksworld-arm	0.0	21.0		31.1	47.5
Briefcaseworld	0.0		0.3		0.9
Freecell	0.0			0.9	2.8
Grid	2.6	4.2			3.1
Hanoi	0.0	0.0	0.0	33.2	77.7
Miconic-ADL	0.0		0.2	6.5	8.7
Miconic-SIMPLE	0.0	0.3	2.9	0.6	0.4
Mprime	13.3	38.4	56.1	42.2	65.9
Mystery	15.7	32.5	49.4	60.5	37.0
Schedule	24.2	32.2	32.3	24.3	18.8

"New" domains: Assembly, Briefcaseworld, Grid, Hanoi, Miconic-SIMPLE.

Measuring Local Search Topology

Maximal exit distance under h^+, excerpt:

Domain	I_0	I_1	I_2	I_3	I_4
Blocksworld-arm	1.0	3.0		3.5	4.9
Blocksworld-no-arm	0.0	1.0		1.5	2.4
Ferry	1.0	1.0	1.0	1.0	1.0
Fridge	2.5	4.5	4.0	6.0	7.7
Gripper	1.0	1.0	1.0	1.0	1.0
Hanoi	3.0	7.0	15.0	31.0	63.0
Logistics	1.0	1.0	1.0		1.0
Miconic-ADL	1.0		1.0	1.1	1.5
Miconic-SIMPLE	1.0	1.0	1.0	1.0	1.0
Miconic-STRIPS	1.0	1.0	1.0	1.0	1.0
Movie	1.0	1.0	1.0	1.0	1.0
Simple-Tsp	0.0	0.0	0.0	0.0	0.0
Tireworld	6.0				

In the other domains, inconclusive or slightly growing.

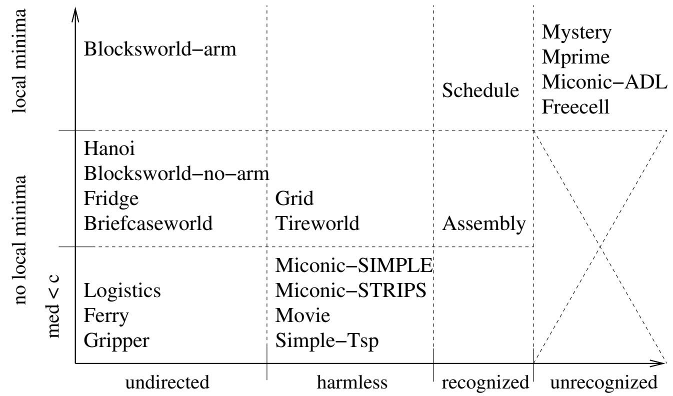
Measuring Local Search Topology

Maximal exit distance under h^{FF}, excerpt:

Domain	I_0	I_1	I_2	I_3	I_4
Blocksworld-arm	1.0	3.0		3.5	4.9
Blocksworld-no-arm	0.0	1.0		1.3	2.4
Ferry	1.0	1.0	1.0	1.0	1.0
Fridge	2.5	4.5	4.0	6.0	7.7
Gripper	1.0	1.0	1.0	1.0	1.0
Hanoi	3.0	9.0	23.0	23.0	23.0
Logistics	1.0	1.0	1.0		1.0
Miconic-ADL	1.0		1.0	1.1	1.1
Miconic-SIMPLE	1.0	1.0	1.0	1.0	1.0
Miconic-STRIPS	1.0	1.0	1.0	1.0	1.0
Movie	1.0	1.0	1.0	1.0	1.0
Simple-Tsp	0.0	0.0	0.0	0.0	0.0
Tireworld	6.0				

The Hypothesis: A Taxonomy of Planning Domains

h^+ behaves this way:



. . . and h^{FF} behaves similarly.

"Easy" Classes

Proposition

Given a solvable STRIPS task (P, A, I, G) with heuristic function h so that there are no local minima, and the maximal exit distance is d, enforced hill-climbing finds a solution after evaluating $O(|A|^{d+1} * h(I))$ states.

Proof.

Blackboard. □

Given a *domain* with no local minima, and constant $c \geq d$, enforced hill-climbing performs in the order of $(c+1) * h(I)$ – *polynomial* if $h(I)$ isn't exponential.

Does this prove that enforced hill-climbing is polynomial in the domains in the lowermost classes of the taxonomy?

Automatic Planning
8. Local Search Topology

Proving Local Search Topology

Prove, from the structural properties of a sub-class of instances, topological properties of a heuristic function.

- "Sub-class": in particular a planning domain
- In what follows, we verify the hypothesis for h^+
- Proofs *domain structure* $\Rightarrow$ *no (unrecognized) dead ends, no local minima, maximal exit distance bound*
- We only show some core lemmas, and a few examples
- Ideally, we would want criteria testable (quickly) by a machine – so we stick as closely to "syntax" as possible

Does S contain Dead Ends?

Definition

Let DEAD-END denote the following decision problem. Given a STRIPS task (P, A, I, G), does S contain a dead end?

Theorem

DEAD-END is **PSPACE**-complete.

Proof.

Membership: guess a state s and verify in PSPACE (PLANSAT for (P, A, I, s), PLANSAT for (P, A, s, G)) if it is in S and a deadend. Finished with NPSPACE = PSPACE = co-PSPACE. Hardness: Exercise. □

Does S contain Dead Ends?

We will derive two sufficient criteria. We need:

Definition
Let (P, A, I, G) be a STRIPS task. Two facts are inconsistent if there is no reachable state $s \in S$ that contains both of them. A set of facts F excludes a set of facts F' if each fact in F' is inconsistent with at least one fact in F.

"excludes" is *not* a symmetric definition: $\{at(A, L), at(T, L)\}$ excludes $\{in(A, T)\}$ but not vice versa!

Means: if F is true, then none of F' is true.

(There exist many good – typically fast and informative – approximation techniques for inconsistency: e.g., 2-planning graphs.)

Invertible Actions

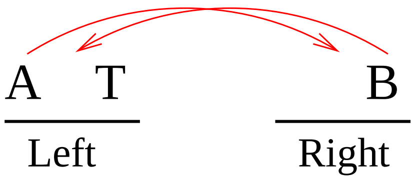
Definition
Let (P, A, I, G) be a STRIPS task. An action a is invertible if:

1. *there is an action $\overline{a} \in A$ such that*
   - 1.1 $pre(\overline{a}) \subseteq (pre(a) \cup add(a)) \setminus del(a)$
   - 1.2 $add(\overline{a}) = del(a)$
   - 1.3 $del(\overline{a}) = add(a)$
2. *$pre(a)$ excludes $add(a)$*
3. *$del(a) \subseteq pre(a)$.*

Lemma
Let (P, A, I, G) be a STRIPS task. If all actions $a \in A$ are invertible, then the state space to the task is undirected.

Proof.
Blackboard. □

Example Logistics



Blackboard.

At Least Invertible Actions

Definition
Let (P, A, I, G) be a STRIPS task. An action a is at least invertible, *if there is an action $\overline{a} \in A$ such that*

1. *$pre(\overline{a}) \subseteq (pre(a) \cup add(a)) \setminus del(a)$,*
2. *$add(\overline{a}) \supseteq del(a)$, and*
3. *$pre(a)$ excludes $del(\overline{a})$.*

"At least": $res(s, \langle a, \overline{a} \rangle) \supseteq s \dots$!

Example "Jog": $jog(x, y) = (\{at(x)\}, \{at(y), did\text{-}jog()\}, \{at(x)\})$

If a is invertible then a is also at least invertible: Blackboard.

Definition

Let (P, A, I, G) be a STRIPS task. An action a has irrelevant delete effects*, if*

$$del(a) \cap (G \cup \bigcup_{a \neq a' \in A} pre(a')) = \emptyset$$

An action has static add effects*, if*

$$add(a) \cap \bigcup_{a' \in A} del(a') = \emptyset$$

Example Tireworld: $inflate(wheel)$: $(\{NOTinflated(wheel)\}, \{inflated(wheel)\}, \{NOTinflated(wheel)\})$

Lemma

Let (P, A, I, G) be a solvable STRIPS task. If for all $a \in A$ either

1. *a is at least invertible, or*
2. *a has irrelevant delete effects and static add effects,*

then S contains no dead ends.

Proof.

Action sequence $a_1, \dots, a_n$ leading to state s, plan for initial state $p_1, \dots, p_m$. Construct plan for s as follows:

$M := \emptyset$
for $i := n \dots 1$ **do**
  if a_i is at least invertible by $\overline{a_i}$ **then**
    if $\overline{a_i} \notin M$ apply $\overline{a_i}$ **endif**
  else $M := M \cup \{a_i\}$
  endif
endfor
for $i := 1 \dots m$ **do**
  if $p_i \notin M$ **then** apply p_i **endif**
endfor

- All actions ($fasten(screw)$, $unfasten(screw)$, $\dots$) are invertible, except $inflate(wheel)$
- Say s is reached via $a_1, \dots, a_n$, and $p_1, \dots, p_m$ is a plan for the initial state
- "Undo $a_1, \dots, a_n$ except for the $inflate(wheel)$ actions; store the latter in M"
- "We are back to the initial state, except that some wheels – corresponding to M – are already inflated"
- "Execute $p_1, \dots, p_m$ except for inflating the wheels – M – that already are inflated"

Definition

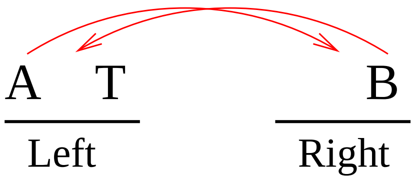
Let (P, A, I, G) be a STRIPS task. An action a is respected by the relaxation if, for any reachable state $s \in S$ such that a starts an optimal plan for (P, A, s, G), there is an optimal relaxed plan for (P, A, s, G) that also starts with a.

Definition

Let (P, A, I, G) be a STRIPS task. An action a is at least relaxed invertible, if there is an action $\overline{a} \in A$ such that $pre(\overline{a}) \subseteq (pre(a) \cup add(a)) \setminus del(a)$, and $add(\overline{a}) \supseteq del(a)$.

This is yet weaker than "invertible" and "at least invertible".

Example Logistics



Blackboard.

Does S contain Local Minima under h^+?

Lemma

Let (P, A, I, G) be a solvable STRIPS task whose state space does not contain unrecognized dead ends. If each action a

1. *is respected by the relaxation, and*
2. *has irrelevant delete effects or is at least relaxed invertible,*

then there are no local minima under evaluation with h^+.

Proof.

Blackboard.

Main line to remember: We can obtain a relaxed plan for $res(s, a)$ by either removing a from P^+ for s, or replacing it with $\overline{a}$ . . . ! □

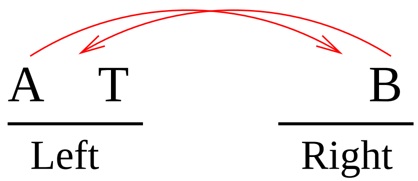
What is the Maximal Exit Distance under h^+?

There is (apparently) no nice “syntactic” criterion. Best I could find is this:

An action a has solution-irrelevant delete effects if, when a starts an optimal plan for s yielding state s', there is an optimal plan for s' that does not need the facts deleted by a.

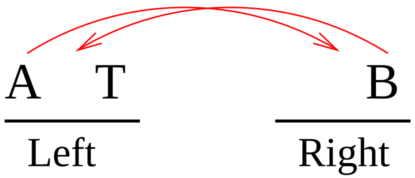
Say the prerequisites of the previous lemma hold. If, for any state s, there is an optimal plan for s that starts with at most d actions with non solution-irrelevant delete effects, then d is an upper bound on the maximal exit distance. *Do you see why?*

Example Logistics



Blackboard.

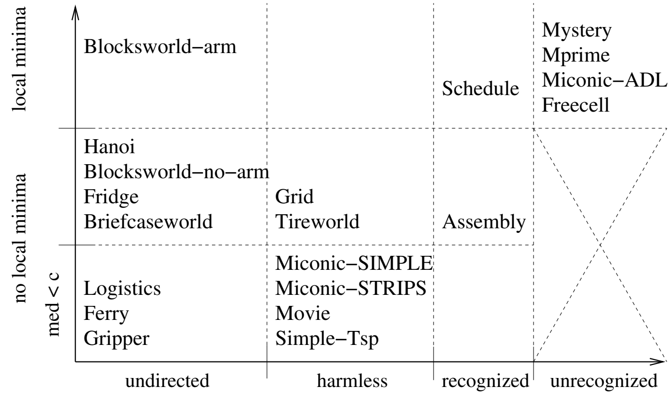
Example Logistics: So?



1. The state space is ???
2. There ??? local minima
3. The maximal exit distance is bounded by ???

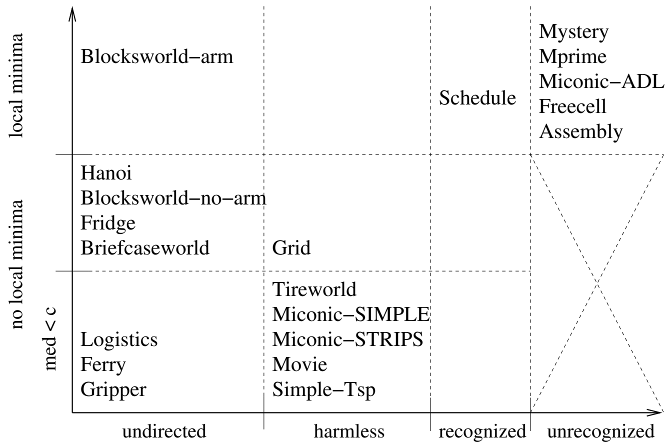
The Hypothesized Taxonomy

h^+ behaves this way:



The Proved Taxonomy

h^+ behaves this way:



"Example" Assembly

This is a little more complicated.

Lemma

Given an instance of the Assembly domain, and a state s, such that for all relevant leafs x in $NAG(s)$, $(available, x) \in s$. If

1. $(As, NA(s) \cup AC(s))$ *is cycle-free, and*
2. *for all* $(x, y) \in NA(s)$, *there is a sequence* $x = x_0, x_1, \dots, x_n$ *such that* $x_n \in pa(y)$, $(x_{i-1}, x_i) \in ao(y)$ *and* $(x_i, y) \in NA(s)$ *for* $1 \leq i \leq n$,

then s does not lie on a valley.

The good news: we don't consider this here.

Automatic Planning
8. Local Search Topology

- A Reminder: h^+ and h^{FF}
- A Preview
- Local Minima & Co.
- 20 Planning Benchmark Domains
- Measuring Local Search Topology
- Proving Local Search Topology
- Sampling Local Search Topology

Sampling Local Search Topology

Generate random states in large examples, measure topological properties of these states, do (simple) statistics.

- This is the only thing anybody previously did (in SAT [Frank et al, JAIR 1997])
- Only way to get empirical data about topology of large state spaces; applicable in any context

Here: verify *to what extent h^+ topology transfers to h^{FF}*

- For "no local minima" and "med $\leq$ c"! (dead end behavior provably identical)
- We plot topological data against domain parameters

Local Minima under h^{FF}

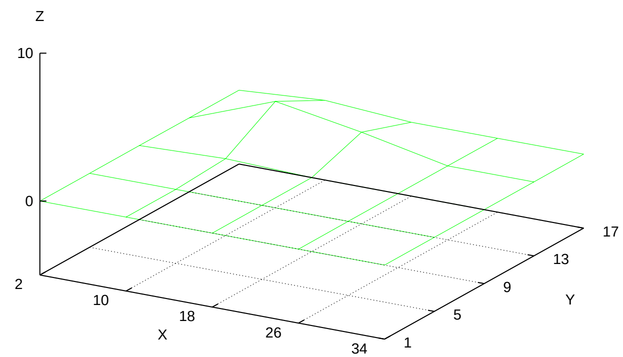
Average percentage of sample states on valleys:

Domain	I_0	I_1	I_2	I_3	I_4
Blocksworld-no-arm	0.0	0.0	0.0	0.1	0.0
Gripper	0.0	0.0	0.0	0.0	0.0
Hanoi	0.0	0.0	96.0	100.0	100.0
Movie	0.0	0.0	0.0	0.0	0.0
Simple-Tsp	0.0	0.0	0.0	0.0	0.0
Tireworld	0.0	0.0	0.0	0.0	0.0

(domains with 1 domain parameter)

Local Minima under h^{FF}

Average percentage of sample states on valleys:



Scaling floors x against passengers y in Miconic-SIMPLE.

Maximal exit distance under h^{FF}

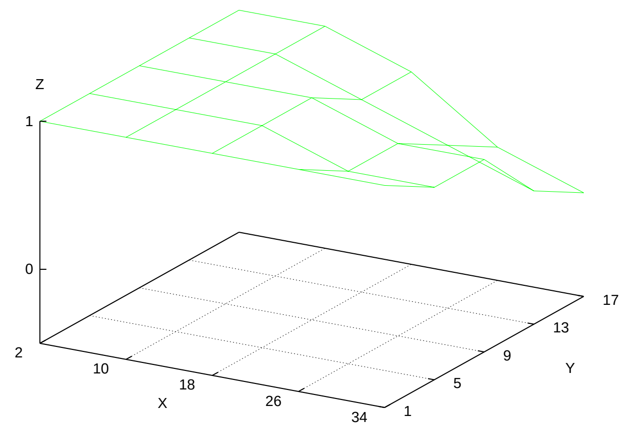
Average maximal exit distance of sample states:

Domain	I_0	I_1	I_2	I_3	I_4
Gripper	1.0	1.0	1.0	1.0	1.0
Movie	1.0	0.9	0.6	0.4	0.6
Simple-Tsp	0.0	0.0	0.0	0.0	0.0
Tireworld	6.0	6.0	6.0	6.0	2.0
Blocksworld-no-arm	0.3	1.8	2.8	3.8	3.7
Hanoi	6.0	23.0	12.0	2.0	2.0

(domains with 1 domain parameter)

Maximal exit distance under h^{FF}

Average maximal exit distance of sample states:



Scaling floors x against passengers y in Miconic-SIMPLE.

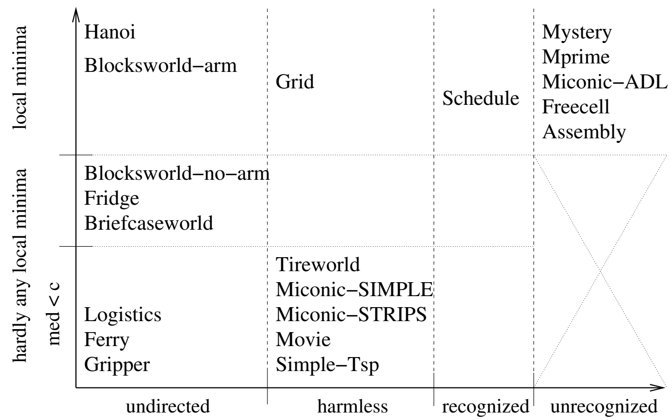
Sampling Local Search Topology

. . . this way, becomes pretty . . . when a domain has, say, more than 5 domain parameters.

Anyway:

The Sampled Taxonomy

h^{FF} behaves this way:



h^+ is an extremely informative heuristic in many Planning benchmarks.

In particular we can conclude that FF is “largely polynomial” in Gripper, Ferry, Logistics, Simple-Tsp, Movie, Miconic-STRIPS, Miconic-SIMPLE, and Tireworld.

(As well as Zenotravel, Satellite, Schedule, and Dining-Philosophers.)

Most important question: Does this phenomenon occur also in real domains? (Put in another way: “Is this good or bad news?”)

My intuition: “yes, but only in some domains, and not in such an extreme form”

- Jeremy Frank, Peter Cheeseman, and John Stutz, *When Gravity Fails: Local Search Topology*, JAIR, 1997.
- Jörg Hoffmann, *Local Search Topology in Planning Benchmarks: An Empirical Analysis*, IJCAI 2001.
- Jörg Hoffmann, *Local Search Topology in Planning Benchmarks: A Theoretical Analysis*, AIPS 2002.
- Jörg Hoffmann, *Utilizing Problem Structure in Planning: A Local Search Approach*, LNAI 2854, Springer-Verlag, 2003.
- Jörg Hoffmann, *Where “Ignoring Delete Lists Works: Local Search Topology in Planning Benchmarks”*, JAIR, 2005.